

Geschäftsprozessmodellierung und Reengineering mittels prozess- und objektorientierter Methoden

Diplomarbeit

Zur Erlangung des Grades eines Diplom-Volkswirts
an der Wirtschaftswissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

vorgelegt von

Veit Köppen

(Matrikel-Nr. 155489)

Prüfer: Prof. Dr. Hans Gernert

Berlin, 15.Juli 2003

Inhaltsverzeichnis

1	Abbildungsverzeichnis.....	4
2	Tabellenverzeichnis	6
3	Abkürzungsverzeichnis.....	7
4	Einleitung.....	8
5	Geschäftsprozessmodellierung	11
5.1	ARIS	13
5.1.1	Die einzelnen Modelle des Fachkonzeptes in ARIS.....	14
5.2	Petri-Netze	16
5.3	UML.....	17
5.3.1	Die Sichtweisen der UML	18
6	Transformation der statischen Sicht	20
6.1	Die statische Sicht der ARIS – Welt.....	20
6.1.1	Das Organigramm.....	20
6.1.2	Das Entitäten – Relationen – Modell.....	24
6.1.3	Die Leistungssicht.....	26
6.1.4	Darstellung der Sichten in der Datenrepräsentation	26
6.2	Die statische Sicht der UML – das Klassendiagramm	29
6.2.1	Organigrammdarstellung in der UML	34
6.3	Umwandlung von prozess- zu objektorientierter Datenrepräsentation.....	36
6.3.1	Klassendiagramm und eERM	36
6.3.2	Klassendiagramm und Organigramm	39
6.4	Ergebnis	41
7	Transformation der dynamischen Sicht	42
7.1	Dynamisches Verhalten im ARIS – Haus.....	42
7.1.1	Die Funktionssicht	43
7.1.2	Die Steuerungssicht	45
7.2	Dynamisches Verhalten in der UML.....	50
7.2.1	Das Anwendungsfalldiagramm.....	50
7.2.2	Das Zustandsdiagramm.....	52
7.2.3	Das Aktivitätsdiagramm	54
7.3	Petri – Netze.....	56

7.4	Transformation des Anwendungsfalldiagramms	61
7.5	Transformation Funktionsebene	64
7.5.1	Transformation von Aktivitätsdiagramm und Funktionsbaum.....	64
7.5.2	Transformation von Zustandsdiagramm und Funktionsbaum	68
7.5.3	Fazit Transformation der Funktionsebene	70
7.6	Transformation von eEPK und Aktivitätsdiagramm	71
7.6.1	Transformation eEPK und Petri – Netz	71
7.6.2	Transformation Aktivitätsdiagramm und Petri – Netz	78
7.6.3	Resultat der Transformation	82
7.7	Transformation eEPK und Zustandsdiagramm.....	85
7.7.1	Transformation Zustandsdiagramm und Petri – Netz.....	86
7.7.2	Resultat der Transformation	89
8	Zusammenführung beider Sichten	92
9	Zusammenfassung und Ausblick	94
9.1	Die UML 2.0.....	94
9.2	Die ARIS - Weiterentwicklungen.....	95
9.3	Andere Ansätze.....	95
9.4	Zusammenfassung	96
10	Anhang.....	99
10.1	Anwendungsfalldiagramm und eEPK	99
10.2	Petri – Netz – Transformation	101
11	Literaturverzeichnis	105

1 Abbildungsverzeichnis

ABBILDUNG 1: THE BUSINESS PROCESS ANALYSIS / MODELLING MAGIC QUADRANT	9
ABBILDUNG 2: ARIS - KONZEPT	14
ABBILDUNG 3: KANTEN ZWISCHEN UNTERSCHIEDLICHEN SYMBOLAUSPRÄGUNGEN	22
ABBILDUNG 4: ABSTRAKTIONSEBENE 1	24
ABBILDUNG 5: ABSTRAKTIONSEBENE 2	24
ABBILDUNG 6: ORGANIGRAMM DER STUFE 1 ALS EERM	28
ABBILDUNG 7: ORGANIGRAMM DER STUFE 2 ALS EERM	28
ABBILDUNG 8: UML - KLASSENDARSTELLUNG	29
ABBILDUNG 9: UML - GENERALISIERUNG	30
ABBILDUNG 10: UML - MULTIPLIZITÄT	30
ABBILDUNG 11: UML - ROLLENAUSPRÄGUNGEN	30
ABBILDUNG 12: ASSOZIATIONSKLASSE	31
ABBILDUNG 13: AUFGELÖSTE ASSOZIATIONSKLASSE	32
ABBILDUNG 14: MEHRGLIEDRIGE ASSOZIATION	33
ABBILDUNG 15: AUFGELÖSTE MEHRGLIEDRIGE ASSOZIATION	33
ABBILDUNG 16: AGGREGATION UND KOMPOSITION	33
ABBILDUNG 17: UML – ORGANIGRAMM	35
ABBILDUNG 18: PERSONEN IM KLASSENDIAGRAMM	36
ABBILDUNG 19: TRANSFORMATION ASSOZIIERTER KLASSEN	37
ABBILDUNG 20: ORGANIGRAMM IN KLASSENNOTATION	41
ABBILDUNG 21: OBJEKTORIENTIERTER FUNKTIONSBAUM	44
ABBILDUNG 22: VERRICHTUNGSORIENTIERTER FUNKTIONSBAUM	44
ABBILDUNG 23: PROZESSORIENTIERTER FUNKTIONSBAUM	44
ABBILDUNG 24: EREIGNIS	46
ABBILDUNG 25: UND - VERKNÜPFUNG	46
ABBILDUNG 26: XOR - VERKNÜPFUNG	46
ABBILDUNG 27: OR – VERKNÜPFUNG	46
ABBILDUNG 28: OR ZU XOR – UND – DARSTELLUNG	47
ABBILDUNG 29: XORUND DARSTELLUNG	47
ABBILDUNG 30: TRANSFORMATION EINES OR - VERKNÜPFUNG MIT DREI EINGÄNGEN	48
ABBILDUNG 31: ANWENDUNGSFALLDIAGRAMM	51
ABBILDUNG 32: OBJEKTE IM ZUSTANDSDIAGRAMM NACH TOGETHER	53
ABBILDUNG 33: PASSIVE KOMPONENTE	56
ABBILDUNG 34: MARKIERTE PASSIVE KOMPONENTE	56
ABBILDUNG 35: AKTIVE KOMPONENTE	56
ABBILDUNG 36: FLUß	56

ABBILDUNG 37: EXKLUSIVE ODER ZUSAMMENFÜHRUNG IM PETRI - NETZ	57
ABBILDUNG 38: BEDINGUNGS- EREIGNIS - NETZ: DIE 4 JAHRESZEITEN	59
ABBILDUNG 39: BE – NETZ 2 KONSUMENTEN UND 1 PRODUZENT	60
ABBILDUNG 40: ST - NETZ 2 KONSUMENTEN UND 1 PRODUZENT	60
ABBILDUNG 41: PROZESSAUSWAHLMATRIX	62
ABBILDUNG 42: AKTIVITÄTSDIAGRAMM (OBJEKTORIENTIERT)	65
ABBILDUNG 43: AKTIVITÄTSDIAGRAMM MIT MEHREREN INTERNEN AKTIONEN (PROZESSORIENTIERT)	66
ABBILDUNG 44: AKTIVITÄTSDIAGRAMM (PROZESSORIENTIERT)	67
ABBILDUNG 45: ZUSTANDSDIAGRAMM (OBJEKTORIENTIERT)	69
ABBILDUNG 46: ZUSTANDSDIAGRAMM (PROZESSORIENTIERT)	70
ABBILDUNG 47: EEPK ANTRAG AUSFÜLLEN	75
ABBILDUNG 48: STELLEN - TRANSITIONS – NETZ ANTRAG AUSFÜLLEN	76
ABBILDUNG 49: STELLEN - TRANSITIONS - NETZ ANTRAG AUSFÜLLEN (ÜBERFÜHRBAR)	77
ABBILDUNG 50: EPK MIT VERKNÜPFUNG	78
ABBILDUNG 51: STELLEN - TRANSITIONS - NETZ MIT VERKNÜPFUNG	78
ABBILDUNG 52: AKTIVITÄTSDIAGRAMM MIT KONNEKTOREN	80
ABBILDUNG 53: PETRI NETZ FÜR AKTIVITÄTSDIAGRAMM MIT KONNEKTOREN	80
ABBILDUNG 54: AKTIVITÄTSDIAGRAMM MIT OBJEKTEN UND VERANTWORTLICHKEITSBEREICHEN	81
ABBILDUNG 55: PETRI - NETZ FÜR AKTIVITÄTEN MIT OBJEKTEN UND VERANTWORTLICHKEITEN	81
ABBILDUNG 56: UND – ÜBERTRAGUNG	84
ABBILDUNG 57: ZUSTANDSDIAGRAMM MIT KONNEKTOREN	87
ABBILDUNG 58: PETRI - NETZ FÜR ZUSTANDSDIAGRAMM MIT KONNEKTOREN	87
ABBILDUNG 59: ZUSTANDSDIAGRAMM MIT OBJEKTEN	88
ABBILDUNG 60: PETRI - NETZ FÜR ZUSTANDSDIAGRAMM MIT OBJEKTEN	88
ABBILDUNG 61: ODER - VERFEINERUNG	90
ABBILDUNG 62: FUNKTION MIT DATENELEMENTEN	93
ABBILDUNG 63: KLASSENDIAGRAMM MIT OPERATIONEN	93
ABBILDUNG 64: PETRI - NETZ DER EPK MIT FUNKTIONEN ALS AKTIVER KOMPONENTE	103
ABBILDUNG 65: PETRI - NETZ DER EEPK MIT FUNKTIONEN ALS AKTIVER KOMPONENTE	103

2 Tabellenverzeichnis

TABELLE 1: ÜBERSICHT DER ARIS - MODELLE DES FACHKONZEPTES	16
TABELLE 2: SYMBOLTYPEN DES ORGANIGRAMMS	21
TABELLE 3: EERM-ELEMENTE	26
TABELLE 4: UML TRANSFORMATIONSELEMENTE	34
TABELLE 5: ÜBERFÜHRUNG EERM UND KLASSENDIAGRAMM	39
TABELLE 6: ELEMENTE DER PROZESSAUSWAHLMATRIX	50
TABELLE 7: ELEMENTE DES ZUSTANDSDIAGRAMM	53
TABELLE 8: ELEMENTE DES AKTIVITÄTSDIAGRAMMS	55
TABELLE 9: PROZESSE IN PETRI - NETZEN	58
TABELLE 10: WICHTIGE PETRI - NETZE	61
TABELLE 11: TRANSFORMATION PROZESSAUSWAHLMATRIX UND ANWENDUNGSFALLDIAGRAMM	63
TABELLE 12: TRANSFORMATION FUNKTIONSBAUM UND AKTIVITÄTSDIAGRAMM	68
TABELLE 13: TRANSFORMATION FUNKTIONSBAUM UND ZUSTANDSDIAGRAMM	70
TABELLE 14: TRANSFORMATION EEPK UND PETRI – NETZ	74
TABELLE 15: TRANSFORMATION AKTIVITÄTSDIAGRAMM UND PETRI - NETZ	82
TABELLE 16: TRANSFORMATION EEPK UND AKTIVITÄTSDIAGRAMM	85
TABELLE 17: TRANSFORMATION ZUSTANDSDIAGRAMM UND PETRI - NETZ	89
TABELLE 18: TRANSFORMATION EEPK UND ZUSTANDSDIAGRAMM	91
TABELLE 19: ÜBERFÜHRUNG EEPK UND ANWENDUNGSFALLDIAGRAMM	100
TABELLE 20: AUFTEILUNGS- UND ZUSAMMENFÜHRUNGSUMWANDLUNGEN	102
TABELLE 21: ÜBERFÜHRUNG DER PETRI - NETZE	104

3 Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
BPR	Business Process Reengineering
bzw.	beziehungsweise
Case	Computer Aided Software Engineering
EDV	elektronische Datenverarbeitung
EPK(n)	Ereignisgesteuerte Prozesskette(n)
eEPK(n)	erweiterte Ereignisgesteuerte Prozesskette(n)
ERM	Entitäten Relationen Modell
eERM	erweitertes Entitäten Relationen Modell
et al.	et alii
Hrsg.	Herausgeber
IT	Informationstechnologie
OMG	Object Management Group
OMT	Object Modeling Technique
OOSE	Object Oriented Software Engineering
S.	Seite(n)
SAP	Software Anwendungen Programme ???
UML	Unified Modelling Language
z.B.	zum Beispiel

4 Einleitung

Aufgrund der ständigen Entwicklung neuer Unternehmensvoraussetzungen muss für die Wettbewerbsfähigkeit eines Unternehmens die Flexibilität hoch sein. Anpassungen an Veränderungen müssen rasch erfolgen können. Durch den Einsatz von Informationstechnologie können Bearbeitungszeiten und somit Kosten dann reduziert werden, wenn die vorhandene Software effizient die Abläufe des Unternehmens unterstützt. Aus diesem Grund stehen angepasste Komponentensysteme oder Individualsoftware häufig im Entscheidungsbereich bei Änderungen auf wettbewerbspolitische Maßnahmen.

Durch diese Arbeit soll ein Beitrag zur Kostenreduktion von Individualsoftware für Geschäftsprozesse, die nicht Zwingenderweise durch Workflows abgebildet werden, geleistet werden. Da ein hoher Anteil einzelner Projekte durch die Kommunikation von Anforderungen und Möglichkeiten erfolgt, können Kosten in dieser Phase durch Modelle eingespart werden, die sowohl auf betriebswirtschaftlicher wie auch informationstechnischer Seite verstanden werden. Ziel ist dabei nicht das Wissen beider Seiten in ein Modell zu integrieren, sondern eine Überführung der jeweiligen Methoden zu erreichen. Dabei wird nicht nur die Entwicklung von Software als einseitiger Schritt verstanden, sondern auch die Verwendung bereits vorhandener informationstechnischer Software genutzt. Aus diesem Grund steht das Business Process Reengineering im Vordergrund. Eine Überführung kann in dieser Arbeit nur dann als ausreichend angesehen werden, wenn auch der Umkehrschluss nachvollziehbar bleibt.

In Fachabteilungen werden die Wissensstrukturen beispielsweise durch prozessorientierte Modellmethoden abgebildet. Diese Arbeit konzentriert sich dabei auf die Architektur integrierter Informationssysteme (ARIS), da die IDS Scheer AG mit fast 39000¹ verkauften Lizenzen Marktführer im Bereich der Business Process Analyse ist und die führende Rolle der Business Process Management Modellierungswerkzeuge innehat.² Abbildung 1 zeigt die Positionierung der IDS Scheer AG auf diesem Toolmarkt, wobei die Ausführbarkeit und Vollständigkeit der verwendeten Modelle Entscheidungskriterium sind.

¹ Stand 31.03.2003 aus: Quartalsbericht IDS Scheer AG vom 31.03.2003.

² Vgl. Bullinger (2001), S. 148.

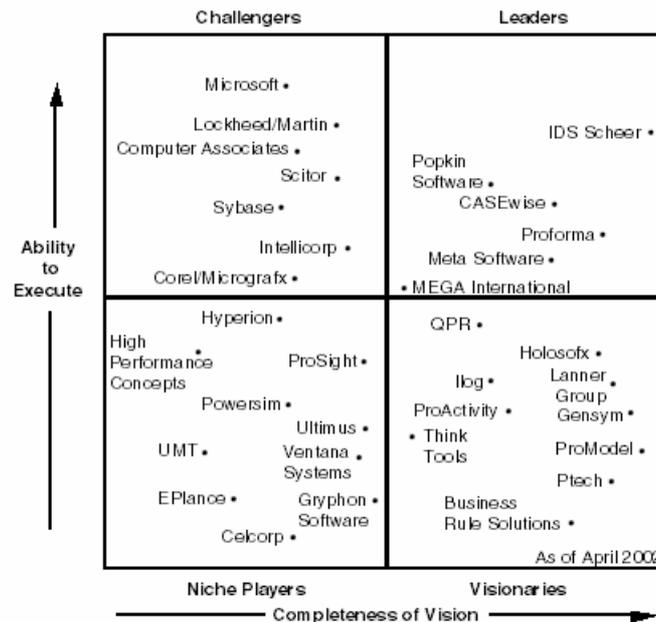


Abbildung 1: The Business Process Analysis / Modelling Magic Quadrant³

Seit dem Beginn der objektorientierten Modellierung durch beispielsweise OMT und OOSE hat sich diese im Bereich der Softwareentwicklung durchsetzen können. Die Unified Modeling Language (UML) stellt eine Vielzahl von Modellen zur Verfügung, die geeignet sind, die Softwareentwicklung zu unterstützen. Sie ist durch die ständige Entwicklung der OMG an die sich verändernden Bedingungen angepasst und wird durch eine Vielzahl von Entwicklern genutzt. Weiterhin gibt es Modellierungstools, wie z.B. TogetherSoft und Rational Rose, die eine Generierung von Codegerüsten aufgrund der UML – Modelle ermöglichen. Dadurch wird wiederum eine Kostenreduktion erreicht, denn auf der einen Seite kann Zeit gespart werden, die durch die Implementierung dieser Codegerüste verwendet wird und auf der anderen Seite ist dadurch eine Dokumentation des Codes gewährleistet, der somit leichter verständlich und übersichtlicher für andere beteiligte Entwickler ist. Aus diesem Grund werden die objektorientierten Modelle der UML in dieser Arbeit betrachtet.

Da es sich bei der UML und ARIS um teilweise semiformale Beschreibungsmethoden handelt, wird für die Überführung die Petri – Netz – Methode herangezogen. Diese soll für die Angleichung und Interpretation einzelner Modellelemente genutzt werden. Als Resultat soll dann eine unmittelbare Überführung im Vordergrund stehen. Vorteil des Zwischenschrittes ist aber das gleichzeitige Erhalten einer dritten Modelldarstellung, die wiederum Klarheiten zwischen den Beteiligten über die beschriebenen Abläufe schaffen kann.

³ Quelle: Gartner Research, Gartner (2003).

In einem ersten Abschnitt werden die wichtigsten Begriffsdefinitionen und die verwendeten Methoden näher beschrieben und die für diese Arbeit wichtigen Modelle ermittelt.

Darauf folgt die Überführung der statischen Informationen, die in den beiden Beschreibungs-darstellungen gesondert auftritt.

Im größten Abschnitt dieser Arbeit beschäftigt sich mit der Überführung der dynamischen Elemente und Beschreibungen. Hierbei wird schrittweise unter Zuhilfenahme der Petri – Netz – Methode eine Überführung überprüft. Am Ende soll die direkte Transformation beider Beschreibungssprachen stehen.

Auch die Zusammenführung von Dynamik und Statik wird unterschiedlich abgebildet und soll dargestellt werden.

Anschließend sollen weitere Ansätze die sich mit der Problematik befassen kurz dargestellt werden. Da es eine Vielzahl von solchen Bestrebungen gibt, wird hier nur ein kurzer Überblick über einige Ansätze gegeben.

Da am besten Modellierungssprachen anhand von Beispielen deutlich gemacht werden können, finden sich in dieser Arbeit solche Beispiele wieder. Dabei handelt es sich im Wesentlichen um einfache Abstraktionen der Realität, die in den betreffenden Methoden dargestellt werden.

5 Geschäftsprozessmodellierung

„Ein Geschäftsprozess ist eine zusammengehörige Abfolge von Unternehmensverrichtungen zum Zweck einer Leistungserstellung. Ausgang und Ergebnis des Geschäftsprozesses ist eine Leistung, die von einem internen oder externen „Kunden“ angefordert und abgenommen wird.“⁴

Die Modellierung von Geschäftsprozessen stellt sich aufgrund ihrer vielfältigen Komponenten kompliziert dar. So ist es schwierig Daten, Funktionen und Funktionsträger und ihre räumlichen, zeitlichen und logischen Zusammenhänge modellhaft wiederzugeben.⁵ Die Geschäftsprozessmodellierung ist jedoch ein wesentlicher Faktor für die Analyse und Optimierung von Unternehmen. Denn durch die Nutzung informationstechnischer Ressourcen können Effizienzsteigerungen erzielt werden, die zu einer Besserstellung des Unternehmens im Wettbewerb führen und somit dessen Fortbestand sichern.

Modelle werden zur Erklärung von Sachverhalten benutzt. Dabei abstrahieren sie von der Wirklichkeit, das heißt eine Vereinfachung der Realität wird durch das Weglassen von Informationen, die für die Beschreibung bzw. Betrachtung des Sachverhaltes uninteressant sind, erreicht.

Für die Modellierung bieten sich unterschiedliche Modellierungsmethoden an. Dabei ist zusätzlich der Modellierungszweck für die Modelle wichtig. Die wichtigste Zielsetzung von Geschäftsprozessmodellen ist die Prozessanalyse.

Geschäftsprozessmodelle wenden sich an eine Vielzahl von Nutzern, wie zum Beispiel Führungskräfte, prozessausübende Mitarbeiter oder auch Informatiker, die Informationssysteme für das Unternehmen erstellen oder anpassen sollen. Diese unterschiedlichen Zielgruppen besitzen ungleiche Abstraktionsebenen. Somit kann nicht das gleiche Modell genutzt werden, da die Abstraktion des Modells von entscheidender Bedeutung für das Verständnis der beteiligten Nutzer ist. Durch Aus- und Einblenden der jeweils relevanten Informationen kann aus einem Grundmodell die jeweils benötigte Sicht gewählt werden. Dabei ist es die Aufgabe des Modellierers zu entscheiden, welche Informationen für welchen Nutzer sichtbar sein sollen. Diese Aufgabe ist jedoch bei komplexen Prozessen nicht trivial bzw. nur mit enormen Auf-

⁴ Siehe Scheer (1998), S.3.

⁵ Vgl. Chen (1994), S.1.

wand zu realisieren. Weiterhin verfügt der Modellierer meist nicht über das Wissen aller zukünftigen Modellnutzer, so dass wichtige Informationen bereits im Grundmodell nicht beachtet werden.

Zusätzlich ist ein Unternehmen ständig dem Prozess der Wettbewerbsanpassung unterlegen. Damit es auf die sich verändernden Strukturen der Umwelt reagieren kann, muss sich auch die Unternehmung verändern können und mit ihr ihre Prozesse. Dadurch veralten die erstellten Modelle relativ schnell.

Eine Lösung des Problems der Veränderung von Unternehmen stellt das Business Process Reengineering (BPR) bzw. das Business Reengineering dar. Dabei handelt es sich eigentlich um zwei Aktivitäten. Zum einen muss der bisherige Verlauf der Geschäftstätigkeit analysiert werden um danach das neue Unternehmen zu entwickeln. Business Reengineering setzt sich also aus dem Reverse Business Engineering (das bisherige Unternehmen verstehen) und dem Forward Business Engineering (Entwicklung des neuen Unternehmens) zusammen.⁶

Neben den Informationen werden durch die unterschiedlichen Anwender auch verschiedene Methoden verstanden. So existieren z.B. prozess-, verrichtungs- und objektorientierte, sowie aktionsbedingte Modellierungsmethoden.

Für die heutige Entwicklung eines Unternehmens ist die Einbeziehung von Computertechnologie von besonders hoher Bedeutung. Damit Veränderungen im Unternehmen auch in die Software schnell implementiert werden können, ist eine schnelle und richtige Kommunikation zwischen Fachabteilung und EDV – Kräften besonders wichtig. Eine einfache und schnell verständliche Kommunikation wird dabei durch das Einsetzen von grafischen Modellen möglich. Hierbei werden jedoch unterschiedliche Informationen von den differenzierten Teilnehmern genutzt und vermittelt, die zudem über unterschiedliche Darstellungsweisen verfügen.

Damit ein möglichst gleicher Informationsstand auf beiden beteiligten Seiten erreicht werden kann, soll in dieser Arbeit versucht werden, eine eindeutige Transformation von betriebswirtschaftlicher Fachsicht zur Systementwicklungssicht zu entwickeln. Eineindeutig aus dem Zusammenhang des Reengineerings, da sowohl die Hinrichtung der Transformation, wie auch die Rückrichtung von Bedeutung sind. Die betriebswirtschaftliche Fachsicht wird durch das Konzept der Architektur integrierter Informationssysteme modelliert. Die Systementwicklungssicht wird dagegen auf der Grundlage von objektorientierten Programmiersprachen und ihrer grafischen Modellierung durch die Unified Modeling Language repräsentiert.

⁶ Vgl. Jacobson (1994), S. 77 ff.

Für die Transformation beider Ebenen wird in dieser Arbeit teilweise auf die theoretischen Konstrukte der Petri – Netze zurückgegriffen, um die formale Transformation zu gewährleisten. Zwar könnte die Geschäftsprozessmodellierung auch in ihrer Gesamtheit durch diese Netze erfolgen, jedoch bleibt dabei die Frage offen, ob dieses mathematische Konstrukt auf den agierenden Seiten jeweils als gleich verstanden wird. Deshalb sollen sie hier nur als Transformationsmittel dienen.

5.1 ARIS

Die „Architektur integrierter Informationssysteme“ [ARIS] wurde 1991 von August-Wilhelm Scheer zur vollständigen Beschreibung von Informationssystemen entwickelt. Sie generiert Modelle von betrieblichen Geschäftsvorfällen. Ausgangspunkt stellt dabei der Geschäftsprozess in natürlicher Sprache dar, der sowohl mit formalen wie auch halbformalen Beschreibungsmöglichkeiten in ein Modell gefasst wird.

Das von der IDS Scheer AG entwickelte, auf dem ARIS-Konzept basierende Softwaresystem ARIS-Toolset ist zum internationalen Marktführer von Business Process Engineering-Tools geworden.⁷

Das ARIS - Konzept besteht aus den drei Ebenen Fachkonzept, Datenverarbeitungskonzept und Implementierung. Dabei stellt das Fachkonzept den Ausgangspunkt für die Implementierung dar, wobei eine lockere Kopplung die langfristige Gültigkeit der Modelle sicherstellt. Diese Arbeit konzentriert sich aus diesem Grund auf das Fachkonzept.

Im ARIS – Haus wird zusätzlich eine Sichteneinteilung vorgenommen.⁸ Dabei wird die Leistungssicht in die anderen Modelle integriert, sie verfügt über keine eigenen Modelle. Die vier Sichten, die eigene Modelle bereitstellen, sind die Organisations-, die Funktions-, sowie die Steuerungs- und die Datensicht. In der Organisationssicht werden Organisationsstrukturen der modellierten Unternehmung abgebildet. Die Funktionssicht beinhaltet alle relevanten und in eine Hierarchie gebrachten Funktionen der Geschäftsprozessmodellierung. In der Datensicht werden alle benötigten Daten einheitlich modelliert, sowohl Kontroll- als auch Nutzdaten. Die

⁷ Vgl. dazu Scheer (1998), S. V.

⁸ Vgl. Scheer (1998), S. 33.

Steuerungssicht verbindet die anderen drei Sichten, erst hierbei werden die eigentlichen Prozesse beschrieben.

Abbildung 2 zeigt die gesamte Struktur des ARIS – Konzeptes, die durch das .ARIS Toolset unterstützt wird.

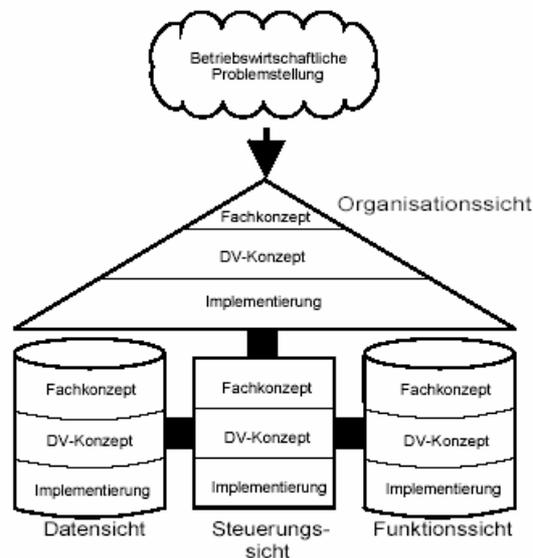


Abbildung 2: ARIS - Konzept⁹

5.1.1 Die einzelnen Modelle des Fachkonzeptes in ARIS

Für die einzelnen Sichten des Fachkonzeptes gibt es unterschiedliche Modelle. Die wichtigsten sollen hier beschrieben und eingeordnet werden.

Das Organigramm bildet die Struktur der Aufgabenträger mit ihren Verknüpfungen ab und ist somit ein Modell der Organisationssicht. Der Schichtkalender spiegelt die Verfügbarkeit von Ressourcen wider. Für die weitere Betrachtung ist dieser nicht relevant, da er als eine Verknüpfung mit dem Organigramm oder der eEPK dargestellt werden kann und seine Informationen in den beiden anderen Diagrammen als Attribute eingepflegt werden können.

Das wichtigste Modell der Funktionssicht ist der Funktionsbaum. Hier werden die Vorgänge, Prozesse, Teilfunktionen und Elementarfunktionen hierarchisch visualisiert. Dabei kann sowohl eine objekt-, prozess- oder verrichtungsorientierte Beschreibung erfolgen. Das Y-Diagramm beschreibt Aufgaben auf hoch aggregierter Ebene und kann als Funktionsbaum

⁹ siehe Scheer (1997), S.17.

uminterpretiert werden. Eine Funktion kann zusätzlich durch ein Zieldiagramm näher beschrieben werden. Dieses Modell entspricht nur dem Veränderungskalkül der Unternehmung. Die Informationen können in den Funktionsattributen hinterlegt werden.

Das von Chen 1976 entwickelte Entität – Relationen – Modell¹⁰ [ERM] ist eine der weitverbreitetsten Entwurfsmethoden für semantische Datenmodelle und wird durch ARIS genutzt. Alle Daten der Unternehmung können hierbei in einem ERM modelliert werden. Hierzu zählen neben den Objekten des Unternehmensumfeldes auch die Objekte die Ereignisse widerspiegeln. Zusätzlich werden auch die Beziehungen unter den einzelnen Objekten abgebildet. Da das ursprüngliche ERM – Konzept verändert wurde, wird es im Weiteren als erweitertes ERM [eERM] bezeichnet. Das Materialflussdiagramm stellt die Verbindung zwischen Funktionsinput- bzw. –outputmaterialien dar. Es soll aufgrund der alleinigen unterstützenden Funktionsbeschreibung hier keine weitere Rolle spielen. Das Data Warehouse Strukturdiagramm bildet die statische Beschreibung eines Data Warehouses ab und soll aufgrund der starken Konvergenz mit dem eERM in dieser Arbeit nicht weiter betrachtet werden. Das Berechtigungshierarchiediagramm unterstützt die Organisationsmodellierung durch Berechtigungsdaten. Seine Daten sind alle im ERM abgebildet. Es ist daher ebenfalls nicht Gegenstand dieser Arbeit.

Das wichtigste Modell der Steuerungssicht ist die erweiterte Ereignisgesteuerte Prozesskette [eEPK]. Sie verbindet die Organisation, Daten und die Funktionen zu einem Prozess. Daneben stellt das Funktionszuordnungsdiagramm für einzelne Funktionen deren Datenverarbeitung bzw. deren Organisationsverantwortlichen dar. Da diese Modellierung auch in der eEPK erfolgen kann, wird dieses Modell nicht weiter betrachtet. Informationsflussdiagramme stellen die Verknüpfung von Funktionen durch Informationsdaten dar. Auch dies kann in die eEPK integriert werden. Analog verhält sich das Ereignisdiagramm. Im Vorgangskettendiagramm werden dieselben Sachverhalte abgebildet, aus diesem Grund wird sich im weiteren auf die eEPK konzentriert.

Tabelle 1 zeigt alle allgemeinen Modelle des Fachkonzeptes. Spezielle Modelle werden hierbei nicht weiter betrachtet, da sie nur Erweiterungen der allgemeinen Modelle darstellen.

¹⁰ siehe hierzu Chen (1976).

Sicht	Modell
Organisation	Organigramm, Schichtkalender
Funktion	Funktionsbaum, Y-Diagramm, Zieldiagramm
Daten	eERM, Materialdiagramm, Data Warehouse Strukturdiagramm, Berechtigungshierarchiediagramm
Steuerung	eEPK, Funktionszuordnungsdiagramm, Informationsflussdiagramme, Ereignisdiagramm, Vorgangskettendiagramm

Tabelle 1: Übersicht der ARIS - Modelle des Fachkonzeptes¹¹

5.2 Petri-Netze

Die Petri-Netze haben ihren Ursprung in der Dissertation von C.A. Petri im Jahre 1962¹².

Petri-Netze sind ein mächtiges Darstellungsmittel für semantische Prozessmodellierung durch Validierbarkeit und Verifizierbarkeit der Ergebnisse sowohl in statischer wie auch dynamischer Sichtweise. Die grafische Syntax von Petri - Netzen ist genau definiert und hat eine Semantik, sowohl auf informationstechnischer (endlicher Zustandsautomat), wie auch auf mathematischer Ebene. Darüber hinaus haben sich diese Systeme in der Praxis entschieden bewährt.¹³

Bei Petri – Netzen wird unterschieden in passive und aktive Komponenten, sowie in den Fluss von Informationen und Gegenständen. Dieser Fluss kann nur in abwechselnder Folge, das heißt Fluss zwischen passiver und aktiver sowie zwischen aktiver und passiver Komponente erfolgen. Somit können Komponenten des gleichen Typs nicht direkt miteinander verbunden sein. In Petri – Netzen werden aktive und passive Komponenten gleichrangig behandelt.

Die Informationen und Gegenstände werden durch Marken im Petri – Netz gekennzeichnet.

¹¹ Vgl. IDS Scheer AG (2000), Kap. 4.

¹² Vgl. Baumgarten (1990), S. 14 oder Reisig (1985) S.III.

¹³ Vgl. Reisig (1985a), IX.

5.3 UML

Die Unified Modeling Language [UML] ist die erste standardisierte Modellierungssprache in der Softwareentwicklung. Sie setzt sich aus einer Vielzahl von Modellierungssprachen zusammen. Seit Ende 1997 gilt die UML als anerkannte Modellierungssprache durch die Object Management Group [OMG], die eine Weiterentwicklung ständig vorantreibt. Dabei wird die Weiterentwicklung durch eine Vielzahl unterstützender Unternehmen gefördert. Seit September 2001 gilt die Version 1.4, die die Grundlage der in dieser Arbeit betrachteten Diagramme darstellt.

Die UML ist eine Sprache zur Spezifikation, Visualisierung, Erstellung und Dokumentation von Nutzgegenständen der Softwaresysteme, sowie für Geschäftsmodelle und andere Nicht-Softwaresysteme.¹⁴ Sie ist somit keine Methode, die die Vorgehensweise der Softwareentwicklung bestimmt, sondern eine Notation, die unterstützende Leistungen bezwecken soll.

Die UML verfolgt folgende Ziele¹⁵:

- Versorgung der Nutzer mit einer benutzerfreundlichen, ausdrucksstarken Visualisierungssprache zur Entwicklung und zum Austausch aussagekräftiger Modelle,
- Ausstattungen zur Erweiterung und Spezialisierung der Basiskonzepte,
- Spezifikationsunterstützung, die unabhängig von Programmiersprachen und Entwicklungsprozessen ist,
- Bereitstellen einer formalen Basis zum Verstehen der Modellierungssprache,
- Förderung des Wachstums des Object – Tool – Marktes,
- Unterstützung hochgradiger Entwicklungskonzepte, wie Komponenten, Kollaborationen, Framework und Pattern,
- Integration der besten Gewohnheiten.

¹⁴ Vgl. OMG (2001), S.1-1.

¹⁵ Vgl. OMG (2001), S.1-4.

5.3.1 Die Sichtweisen der UML

Zur Differenzierung besitzt die UML unterschiedliche Sichtweisen, die durch verschiedene Diagrammtypen die Blickwinkel der einzelnen Modelle fokussieren. Der Ausgangspunkt von Modellen ist entscheidend für die Problemherangehensweise und –lösung. Die Unterteilung führt durch diesen Sachverhalt zu den folgenden Ergebnissen¹⁶ :

- Jedes komplexe System wird am besten durch eine möglichst geringe Anzahl von nahezu unabhängigen Sichtweisen angenähert, wobei ein alleiniger Betrachtungswinkel nicht ausreichend ist.
- Jedes Modell kann einen anderen Grad von Genauigkeit ausdrücken.
- Die besten Modelle sind mit der Realität verbunden.

Für die einzelnen Sichtweisen werden zugehörige grafische Diagramme benutzt. Die Sichtweisen können unterteilt werden in Benutzersicht, statische Sicht, Verhaltenssicht, sowie Interaktions-, Gliederungs- und Implementierungssicht.¹⁷

Für die Beschreibung eines Systems werden hierbei nicht alle Sichtweisen benutzt, sondern nur eine Auswahl. Für die Geschäftsprozessmodellierung spielt so zum Beispiel die Implementierungssicht, die durch Komponenten- und Verteilungsdiagramme dargestellt wird, nur eine sehr geringe Rolle.

Die Benutzersicht wird durch Anwendungsfalldiagramme¹⁸ modelliert. Dabei kann ein Anwendungsfall aus einer Menge von Aktivitäten bestehen, er muss jedoch von einem Akteur ausgelöst werden. Ein Anwendungsfall ist eine komplette, unteilbare Beschreibung.¹⁹

Klassendiagramme²⁰ stellen die wichtigste Diagrammart der UML dar. Sie beschreiben die statische Sicht der Softwareentwicklung. Klassen sind die Konstrukte bzw. Vorlagen der im

¹⁶ Vgl. OMG (2001), S.1-2.

¹⁷ Vgl. Rumbaugh (1999), S.23 f.

¹⁸ im Englischen: Use – Case diagram.

¹⁹ Vgl. Oestereich (2001), S.174 f.

²⁰ im Englischen Class diagram.

Informationssystem gebildeten Elemente. Sie beschreiben neben den Eigenschaften der Objekte auch die Funktionen der Objekte, die diese Daten verändern. Eine Interaktion zwischen den Klassen wird im Klassendiagramm durch Assoziationen abgebildet.

Paketdiagramme²¹ visualisieren die Gliederungssicht der UML. Sie fassen dabei zusammengehörige Klassen zu Paketen zusammen. Die Interaktion zwischen den Paketen wird dabei ebenfalls dargestellt. Da die Informationen eines Paketdiagramms wiederum in den Klassendiagrammen abgebildet werden können, wird im Weiteren auf diese Diagrammtypen nicht weiter eingegangen.

Die Verhaltenssicht wird mit Hilfe von Aktivitäts- und Zustandsdiagrammen aufgezeigt. Aktivitätsdiagramme²² beschreiben hierbei den prozeduralen Ablauf mit Hilfe von Aktivitäten. Zustandsdiagramme²³ bilden den Ablauf der Zustände und somit ihre Veränderungen ab.

Interaktionsdiagramme werden zum Teil der Verhaltenssicht unterstellt. Sequenzdiagramme²⁴ zeigen den zeitlichen Ablauf der beschriebenen Umgebung auf. Kollaborationsdiagramme²⁵ verdeutlichen die Interaktionen zwischen beteiligten Objekten.

²¹ Im Englischen Package diagram.

²² Im Englischen Activity diagram.

²³ Im Englischen State diagram oder State machine.

²⁴ Im Englischen Sequence diagram.

²⁵ Im Englischen Collaboration diagram.

6 Transformation der statischen Sicht

Die statische Sicht beinhaltet bei der Modellierung die unveränderlichen Informationen. Das können Informationsträger und Verknüpfungen zwischen diesen sein. Hierbei ist jedoch von der Veränderung der Daten zu abstrahieren. Welche genaue Ausprägung das Informationsobjekt hat, kann durch die statische Sicht nicht wiedergespiegelt werden, nur der Aufbau der Informationsträger.

Diese Datenobjekte werden in den verschiedenen Beschreibungssprachen unterschiedlich modelliert. Der folgende Abschnitt beschäftigt sich mit der Transformation der beiden Modellierungssprachen. Hierbei soll gemäß dem Prinzip des Business Reengineering sowohl die Überführung von der UML zur ARIS Darstellung, wie auch die Rückführung betrachtet werden.

6.1 Die statische Sicht der ARIS – Welt

Zur statischen Sicht des ARIS – Fachkonzeptes gehören sowohl die Organisations- als auch die Datensicht. Weiterhin sind die Elemente der Leistungssicht dazuzuzählen.

Da alle Informationsträger in der Datensicht gespeichert und auch die Beziehungen zwischen ihnen im eERM durch Relationen abgebildet werden können, wird in einem ersten Schritt eines umfassenden eERM erstellt. Hierbei werden die Informationen der Leistungs- und der Organisationssicht vollständig integriert.

6.1.1 Das Organigramm

Im Organigramm wird die Struktur der Unternehmung visualisiert. Dabei entsprechen die unterschiedlichen Symbole verschiedenen Aufgabenträgern. Je nach Strukturierungskriterium können die gleichen Symbole unterschiedliche Bedeutung haben. Des Weiteren kann eine Typisierung der Organisationseinheiten erfolgen. Als zusätzliches Beschreibungselement tritt der Standort auf, der neben der Zusatzinformation für die Organisationseinheit auch hierar-

chisch dargestellt werden kann²⁶. Tabelle 2 zeigt die verwendeten Symbole und ihren zugehörigen Typen an.

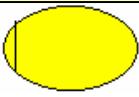
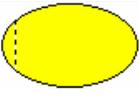
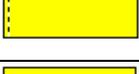
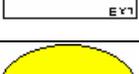
Symbol	Typ
	Organisationseinheit
	Organisationseinheitstyp
	Organisationseinheitstyp
	Stelle
	Personentyp
	Person
	Person
	Standort
	Gruppe

Tabelle 2: Symboltypen des Organigramms

Die Verknüpfung der einzelnen Objekte wird über Kanten erreicht, die wiederum unterschiedliche Bedeutung haben können. So ist neben der Kantenrichtung auch der Kantentyp für die Modellierung der Organisationssicht relevant. Es gibt im Wesentlichen drei unterschiedlichen Kantentypen für Organisationseinheiten. So kann unterschieden werden in fachlich vorgesetzt, disziplinarisch vorgesetzt und ist bildend für.

Daneben spezifizieren die Verbindungen zwischen unterschiedlichen Symbolen den Zugehörigkeitstyp. Abbildung 3 zeigt sowohl Kanten zwischen Organisationseinheiten, als auch Kanten, die den Typ einer solchen Einheit spezifizieren. Die Attribute der Kanten werden hierbei mit ausgedrückt, um die Bedeutung der Kante aufzuzeigen. Für die Beziehung von Organisationseinheit und Organisationseinheitstyp ist nur die Kante „ist vom Typ“ möglich. Auch die Kante „befindet sich an“ kann nur zwischen Standort und Organisationseinheit auftreten. Für die Beziehung zwischen Organisationseinheiten und der Beschreibung einer Gruppe existiert

²⁶ Vgl. IDS Scheer AG (2000), S.4-73 f.

die Kante „ist zugeordnet“. Die anderen Kanten werden je nach Beschreibung durch die Realität, das heißt durch die Beschreibung der Fachabteilung ermittelt.

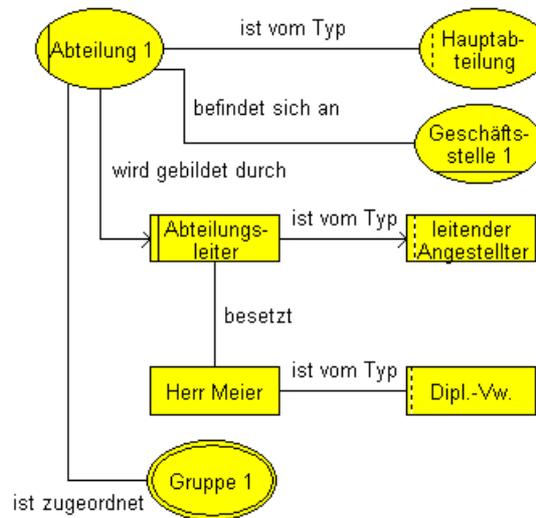


Abbildung 3: Kanten zwischen unterschiedlichen Symbolausprägungen

Um eine Transformation der Organisationssicht in die UML – Darstellung leichter zu erreichen, wird das Modell zuerst verändert dargestellt. Hiernach erfolgt die Umwandlung in das eERM – Modell. Somit kann die schrittweise Durchführung nachvollzogen werden. Eine direkte Transformation des Organigramms in das eERM ist zwar auch möglich, jedoch müssen bei diesem Schritt beide Teilschritte (also die Abstraktion und die Überführung) gemacht werden.²⁷ Bei der Abstraktion erhält man dann ein vereinfachtes grafisches Modell, das alle Informationen enthält, jedoch einfacher in die eERM – Notation überführbar ist, da weniger Elemente verwendet werden. Auch die direkte Transformation des abstrahierten Organigramms in ein UML – Klassendiagramm ist dadurch leichter möglich.

Eine Reduzierung der eingesetzten Symbole kann durch die Informationsverschiebung auf Attributsebene erreicht werden. So sind die Informationen nicht mehr unmittelbar grafisch dargestellt, aber immer noch vorhanden. Dabei kann ein Attribut für die Typzugehörigkeit verwendet werden. Dies wird durch die Abstraktionsebene 1 verdeutlicht. Hier bestehen alle Elemente aus den gleichen Organisationstypen. Der eigentliche Informationswert für die unterschiedlichen Symbole wurde in die Attributsebene verlagert, wobei die Attribute in Abbildung 4 mit dargestellt werden. Das bedeutet, die Attribute und Bezeichnungen, sowie Kanteninformationen bleiben erhalten, jedoch die Symbolgrafiken sind von einem einheitlichen Typ. Ist aufgrund der Reduktion der Elemente die Kantenvielfalt nicht mehr gegeben, so

²⁷ Siehe hierzu Kapitel 6.3.2.

müssen nicht vorhandene Kantenbeschreibungen in einen Attributwert der jeweiligen Beziehung übergehen. Es lassen sich also alle nicht unmittelbaren Informationen in den jeweiligen Attributen unterbringen. Hierbei sollte auf eine feste Attributnutzung geachtet werden, damit alle Informationen wieder richtig rückinterpretiert werden können. So könnten die Informationen über den grafischen Typ des Elementes an erster Stelle, d.h. im ersten freien Attribut, abgelegt werden. Kantenbeschreibungen sollten so im ersten Attribut der jeweiligen Kante hinterlegt werden.

Mit einer weiteren Abstraktion kann das Organigramm durch Reduktion der Elemente vereinfacht werden. So können Typzugehörigkeitsbeziehungen in die einzelnen Organisationsobjekte dann mittels Attributen geführt werden, wenn keine zusätzliche Spezifikation dieser Beschreibungselemente vorliegt. Analog kann mit den Standortelementen verfahren werden, wenn die alleinige Information des Standortes vorliegt und dieser nicht weiter modelliert ist. Für die Ausprägung der Typzugehörigkeit und den Standort kommen somit zwei Attribute für die Beschreibung in Frage. Die Information über die Typzugehörigkeit kann im zweiten freien Attribut und die des Standortes im dritten gespeichert werden. Abbildung 5 stellt die zweite Abstraktionsebene dar. Hierbei ist zu beachten, dass die Organisationseinheitstypen keiner genaueren Spezifizierung des Modells entsprechen, sondern ihr Informationsgehalt lediglich durch den Namen gegeben ist. Auch der Standort trägt nur durch seine Bezeichnung zur Erklärung bei. Ebenso verhält es sich mit der Information der Gruppe, die auch nur als Information in einem Attribut gespeichert werden kann. Zur Verdeutlichung wurden die Attribute wiederum an die einzelnen Objekte geschrieben.

Das Ergebnis beider Filterungen ist ein Modell, das nur noch aus einem Element, mit verschiedenen Attributen, und den untereinander bestehenden Verknüpfungen besteht.

Abbildung 4 und Abbildung 5 zeigen beide Ergebnisse, die durch die Ausgangsgrafik aus Abbildung 3 erreicht wurden. Die Abstraktion 2 kann jedoch nur unter den oben genannten Bedingungen sinnvoll verwendet werden. Dabei handelt es sich bei der zweiten Stufe um eine Verringerung der Elemente für die spätere Transformation. Sie ist lediglich ein Optimierungsprinzip zur Verringerung des Darstellungsaufwandes und nicht zwingend notwendig, da die Optimierung eventuell besser auf Systemebene angewendet wird.

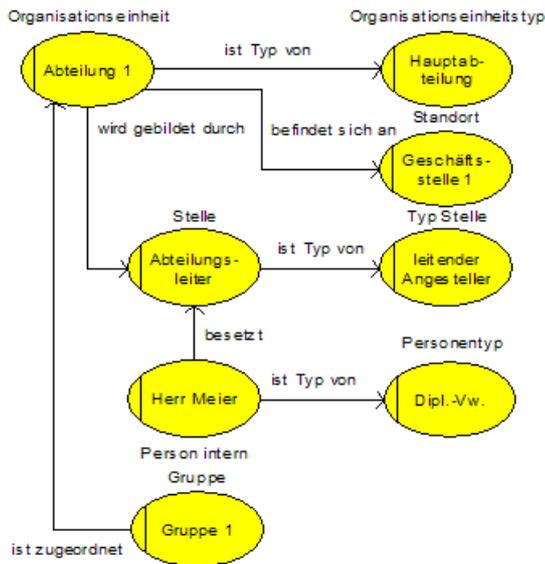


Abbildung 4: Abstraktionsebene 1

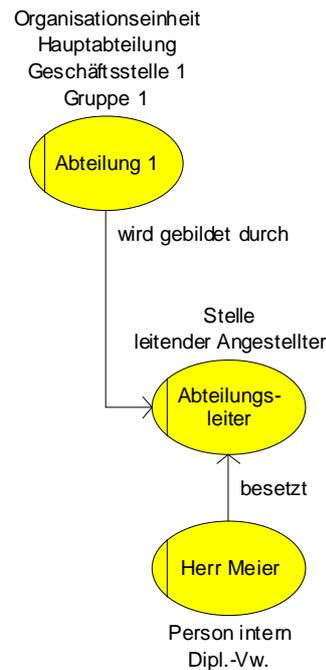


Abbildung 5: Abstraktionsebene 2

Auch der Rückschluss von beiden Abstraktionsebenen hin zum ursprünglichen Modell der Abbildung 3 ist gegeben. Die grafische Information des Ursprungsmodells ist durch das erste Attribut der jeweiligen Elemente vorhanden. Somit kann die Abstraktionsebene 1 direkt wieder in das Ausgangsmodell überführt werden. Für den Fall, dass sich auch die Abstraktionsebene 2 modellieren lässt, sind alle Informationen zur Rückführung gegeben. Das zweite Attribut repräsentiert die Beziehung zur typisierten Organisationseinheit, deren Inhalt dem Wert des Attributes entspricht. Da die Kantenbeziehung eindeutig ist, muss kein zusätzliches Attribut verwendet werden. Auch die Information über Standorte kann somit rücküberführt werden.

6.1.2 Das Entitäten – Relationen – Modell

Im ERM werden Daten und ihre Verknüpfungen dargestellt. Es stellt eine statische Sichtweise dar. Bei den modellierten Daten wird zwischen Kontroll- und Flussdaten unterschieden. In der Datensicht werden alle statischen Informationen der Geschäftsprozesse abgelegt. Somit sind auch alle Ereignisse, Organisationseinheiten, Dokumente usw. in dieser Sicht vorhanden. Im weiteren wird auf das durch das ARIS – Toolset unterstützte erweiterte ERM zurückgegriffen, da es sich hierbei um eine Erweiterung des von Chen 1976 entwickelten Konzeptes handelt, ohne dieses zu verändern.

Hauptbestandteil des eERM sind die Entitäten, die durch Relationen verknüpft sind. Eine Entität ist definiert als reales oder abstraktes Ding, das von Interesse für den betrachteten Ausschnitt ist. Relationen stellen die Konnektoren von Entitäten dar.²⁸

Beide besitzen als Ausprägungen Attribute, die die eigentlichen Informationsträger sind. Somit kann eine Entität als eine logische Zusammenfassung von Einzelinformationen gesehen werden. Je nach Abstraktionstiefe können Entitäten abstrakte bis detaillierte Güte besitzen. Um die Übersichtlichkeit der Modelle zu gewähren, können Attribute auch zu Attributgruppen zusammengefasst werden. Da es sich hierbei um eine bloße Aggregation von Informationen handelt, die an anderer Stelle wieder aufgelöst werden muss, zum Beispiel in einem anderen Diagramm, wird im folgenden die Gruppierung nicht betrachtet, sondern als aufgelöste Einzel – Attribute angesehen werden. Zusätzlich ist für die eindeutige Identifizierung die Verwendung von Schlüsselattributen notwendig.

Eine Erweiterung des ERM wird durch die Generalisierung geschaffen. Hierdurch können spezielle Objekte einem allgemeineren Objekt zugeordnet werden.

Der uminterpretierte Beziehungstyp stellt sowohl eine Beziehung wie auch eine Entität dar. Er kann jedoch aufgelöst werden durch entsprechende Beziehungen und Entitäten. Somit wird gleichzeitig das Modell verfeinert, da die Abstraktion, die durch den uminterpretierten Beziehungstypen geschaffen wurde, aufgelöst wird. Im weiteren wird davon ausgegangen, dass der uminterpretierte Beziehungstyp aufgelöst wurde und aus diesem Grund wird er nicht weiter betrachtet.

„Ein Datencluster beschreibt eine logische Sicht auf eine Ansammlung von Entity- und Beziehungstypen eines Datenmodells, die zur Beschreibung eines komplexen Objektes benötigt wird.“²⁹ Ein Datencluster dient somit im Wesentlichen der Übersichtlichkeit der Modelle. Er steht für ein hinterlegtes eERM und soll aus Vereinfachungsgründen durch das ihm hinterlegte eERM ersetzt werden. Der Datencluster könnte aber auch durch ein eigenes Klassendiagramm überführt werden. Auf dies wird aus Gründen der Übersichtlichkeit verzichtet.

Die vorhandenen Elemente des eERM, die für die Transformation von Bedeutung sind, werden in Tabelle 3 zusammengefasst.

²⁸ Vgl. Scheer (1997), S.31.

²⁹ Siehe IDS(2000), S. 4-35.

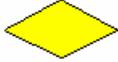
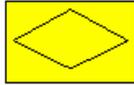
Symbol	Bedeutung
	Entität
	Relation
	Uminterpretierte Relation
	Generalisierung
	Attribut
	Schlüsselattribut

Tabelle 3: eERM-Elemente

6.1.3 Die Leistungssicht

In der Leistungssicht werden Input- und Outputdaten erfasst. Der Leistungssicht selbst stehen keine eigenen Modelle zur Verfügung. Sie wird jedoch in der Steuerungssicht integriert. Dort werden so zum Beispiel ??? in der eEPK zur Leistungsbeschreibung eines Prozesses dargestellt. Bei den Leistungsmerkmalen kann die Transformation in das eERM durch einzelne Entitäten realisiert werden, denn jedes Leistungselement kann in der Datenrepräsentation als ??? mit seinen verschiedenen Attributen gesehen werden.

6.1.4 Darstellung der Sichten in der Datenrepräsentation

Um den Übergang der beiden Modellierungssprachen zu vereinfachen, ist eine Transformation in der ARIS – Fachwelt möglich. Hierzu können die Organisations- und Leistungssicht im eERM zusammengeführt werden. Bei dieser Transformation dürfen jedoch die Daten der einzelnen Sicht nicht eliminiert werden, sondern müssen eins zu eins übertragen werden. Somit

ist dann auch die Rücktransformation, d.h. die Erstellung von Organigrammen und Leistungselementen aus einem eERM möglich, die für das Reverse Engineering notwendig ist.

Das Organigramm in seiner oben beschriebenen vereinfachten Darstellungsweise Abstraktionsebene 1 oder 2 besitzt nunmehr nur noch ein Ausprägungssymbol. Dieses kann dem Entitätstyp zugeordnet werden. Die beschreibenden Attribute der Organisationseinheit werden als Attribute der Entität übernommen. Die Kanten des Organigramms können entweder auf Relationen oder als Generalisierung abgebildet werden, wobei sich der Name der Relation bzw. der Generalisierung aus dem Kantentyp ergibt. Die Generalisierung wird für die Kanten verwendet, die Verfeinerungen darstellen. So werden Beziehungen zwischen Elementen die auf gleicher Stufe existieren, wie z.B. die Kanten „ist Typ von“ oder „besetzt“ durch die Generalisierung abgebildet. Ebenfalls durch die Generalisierung interpretierbar sind Kanten, bei denen eine detailliertere Darstellung der gleichen Objekte abgebildet wird. Durch Relationen werden hingegen Kanten ausgedrückt, die Weisungsbefugnisse (auf gleicher oder unterschiedlicher Hierarchisierungsebene) ausdrücken, Informationen über Standorte liefern und Beziehungen zwischen unterschiedlichen Elementen darstellen. Die weiteren zugehörigen Werthinterlegungen spiegeln sich in den Attributen der Relation wieder.

Für die eineindeutige Transformation, d.h. um darstellen zu können welches Element der zusammengefassten Datensicht zu welcher ARIS - Sicht gehört, muss jeder Entität und jeder Verbindung zwischen den Entitäten ein zusätzliches Attribut hinzugefügt werden, das die Information der jeweiligen Sicht beinhaltet. So kann durch diese Dokumentation die Rückübertragung in die jeweilige Sicht erfolgen.

Abbildung ??? zeigt eine solche Transformation von der Leistungssicht in die Datensicht.

Um die Organisationssicht in der Datensicht zu beschreiben wird das Modell der Organisationssicht in die Abstraktionsebene 1 oder wenn möglich 2 überführt. Hiernach werden die Organisationseinheiten auf Entitäten abgebildet. Die Attribute werden mit übernommen. Die Kanten werden durch Relationen oder Generalisierungen ersetzt. Damit eine Rücktransformation möglich ist, müssen alle so gebildeten Elemente durch ein Attribut (Sicht) gekennzeichnet werden. Abbildung 6 zeigt das Ergebnis dieser Transformation am Beispiel des Modells von Abbildung 4. Kann die Abstraktionsebene 2 durchgeführt werden, so vereinfacht sich die Darstellung zur Abbildung 7.

Eine weitere Überführung von Organigrammen wurde bereits im obigen Abschnitt angesprochen. Hierbei wird der Zwischenschritt der Abstraktion nicht durchgeführt. Alle Elemente des Organigramms werden dabei als Entität abgebildet, die zusätzlich die Information, welches

grafische Element sie in der Organisationssicht besitzen in einem Attribut (Typ) übernehmen. Als weiteres, jedoch nicht zwingend notwendiges Element erhalten sie ein Attribut (Sicht), das angibt, dass es sich dabei um Elemente der Organisationssicht handelt. Die Kanten des Organigramms werden in das eERM als Relationen zwischen zwei Organisationselementen übertragen, so dass die Notation Entität – Relation – Entität hergestellt wird. Zur Rücküberführung werden auch an den Beziehungen zwischen Entitäten Attribute hinterlegt, die sowohl die Kantenbeschreibung wie die Zugehörigkeit zur Organisationssicht als Information tragen. Das Attribut der Sichtzugehörigkeit kann zwar entfallen, da Organisationseinheiten nur in der Organisationssicht untereinander in der so gebildeten Form zusammenhängen. Es empfiehlt sich jedoch zum einen als Kontrolle und zum anderen für die bessere Verständlichkeit das Attribut einzupflegen.

Das Ergebnis dieser Transformation entspricht dem Ergebnis bei Verwendung der Abstraktion 1. Da die Komplexität aber größer ist, sollte die Überführung mittels der Abstraktion 1 gewählt werden. Der Zusammenhang zwischen den einzelnen Ausgangsmodellen ist dadurch leichter zu erkennen.

Automatischer Schritt !!!

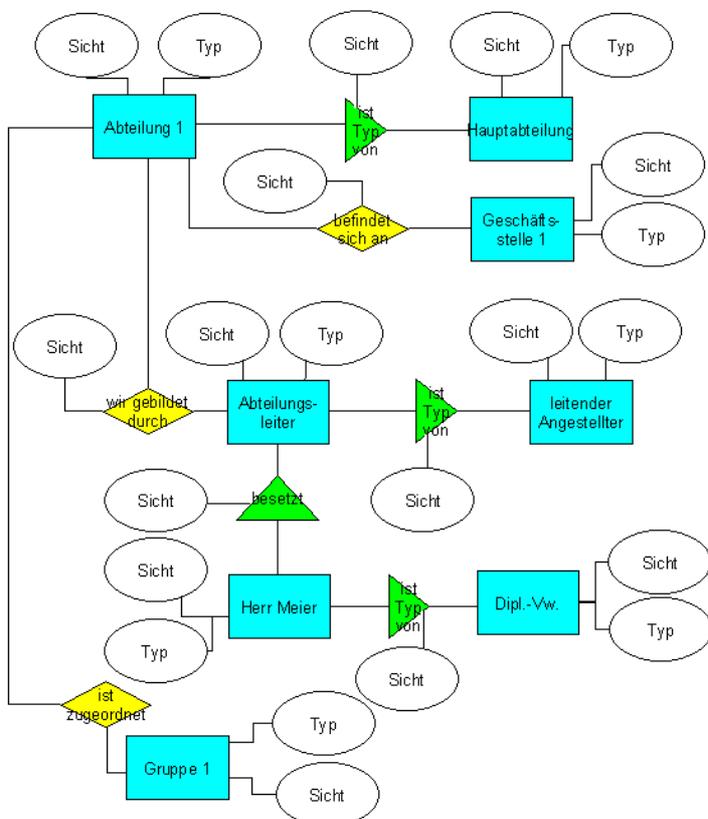


Abbildung 6: Organigramm der Stufe 1 als eERM

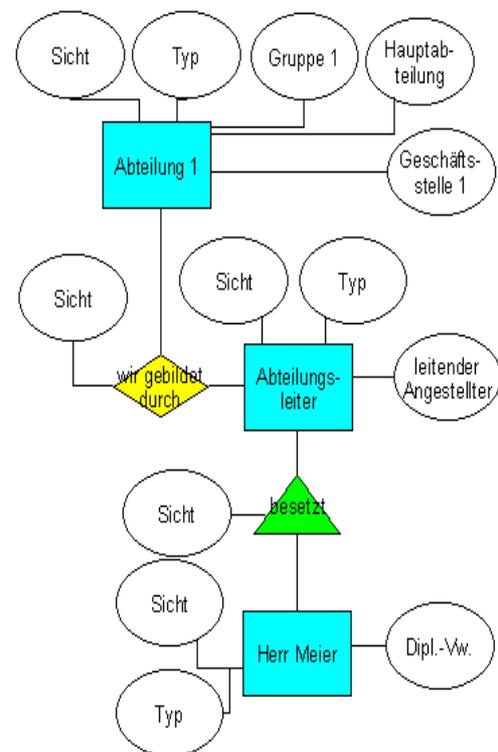


Abbildung 7: Organigramm der Stufe 2 als eERM

Da sowohl bei allen Entitäten, wie auch bei allen Relationen die Information enthalten ist, dass diese Elemente zur Organisationssicht gehören, können aus der Vielzahl der Datenelemente diejenigen der Organisationssicht entnommen werden. Bei der Transformation vom eERM zum Organigramm werden Entitäten zu Organisationseinheiten, während die Relationen zu den Kanten werden. Dabei müssen auch alle Attribute mit übernommen werden. Somit erhält man dann ein abstraktes Organigramm, das sich wiederum in die ursprüngliche Form durch Übersetzung der jeweiligen Attribute in grafische Notation bringen lässt.

6.2 Die statische Sicht der UML – das Klassendiagramm

In der statischen Sicht werden die festen Bestandteile eines Systems dargestellt. Dabei stellt das Klassendiagramm die grafische Darstellung bei der objektorientierten Modellierung dar.

Ein Klassendiagramm besteht aus den einzelnen Klassen und aus der Darstellung der Beziehungen zwischen diesen. Eine Klasse ist eine Definition von Operationen, Attributen und der Semantik für eine Menge von Objekten. Alle Objekte dieser Klasse folgen dieser Definition.³⁰

Beziehungen zwischen den Klassen können unterteilt werden in Generalisierung bzw. Spezialisierung, sowie Assoziation, Aggregation und Komposition.

Klassen werden durch Rechtecke repräsentiert, die durch einen eindeutigen Namen zwischen den Klassen differenzieren. Als zusätzliche Bestandteile können im darunter durch einen Strich abgeteilten Bereich Attribute definiert werden. Im ebenfalls fakultativen dritten Bereich des Rechteckes befinden sich dann die einzelnen Operationen der Klasse. Abbildung 8 zeigt die möglichen Ausprägungen.

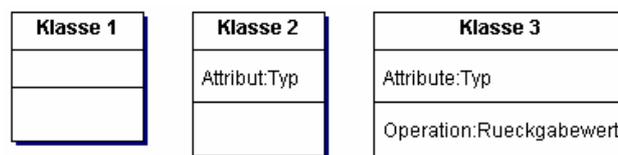


Abbildung 8: UML - Klassendarstellung

Die Generalisierung bzw. Spezialisierung gibt an, dass die untergeordnete Klasse alle Eigenschaften der Superklasse erhält. Nur zusätzliche Erweiterungen werden in der Subklasse aufgelistet. Abbildung 9 repräsentiert die Generalisierung.

³⁰ Vgl. Oestereich (2001), S.186.

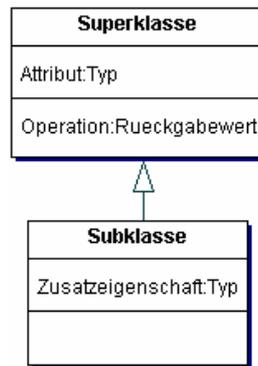


Abbildung 9: UML - Generalisierung

Beziehungen zwischen den Klassen werden als Assoziation mit Hilfe eines Striches dargestellt. Informationen über die Kardinalität der Klassenbeziehung werden in der Multiplizität abgebildet. Die Multiplizität ist der Bereich von erlaubten Kardinalitäten.³¹ Dabei ist die Möglichkeit der Einstufung sehr groß, da sowohl Wertebereiche, wie beliebige Zahlen, aber auch die unbeschränkte, durch einen Stern gekennzeichnet, und die optionale Beziehung, verdeutlicht durch die Null, abgebildet werden können. Abbildung 10 zeigt mögliche Beispiele.



Abbildung 10: UML - Multiplizität

Des Weiteren können der Beziehungsname oder die Rollen der Klassen als nähere Information für das Modell genutzt werden. Aussagekräftiger sind dabei die Rollenausprägungen, da dadurch die Richtung der Beziehung mit in die Information einfließt. Abbildung 11 veranschaulicht dies an einem Beispiel. Sowohl der Mann, wie auch die Frau können miteinander verheiratet sein. Dabei übernehmen sie die jeweilige Rolle. Die Frau ist mit ihrem Ehemann verheiratet und vis-a-vis. Die alleinige Assoziation „verheiratet“ kann diese Information nicht liefern.



Abbildung 11: UML - Rollenausprägungen

³¹ Vgl. Oestereich (2001), S.228.

Bei Beziehungen, die nicht bidirektional sind, sollte dies dann auch verdeutlicht werden. Durch gerichtete Assoziationen ist es möglich, die Beziehungsrichtung mit anzugeben. Die Verwendung der Rollen macht dies jedoch auch möglich, da nur noch eine Rolle dargestellt wird. Ein weiteres wichtiges Merkmal der UML sind attributierte Assoziationen. Hier werden die Assoziationen zwischen zwei Klassen durch zusätzliche Informationsträger, die entweder einzelne Attribute oder gesamte Klassen sein können, näher spezifiziert. Da die Attribute jedoch keiner der beteiligten Klassen direkt zugeschrieben werden können, müssen sie in einer Klasse enthalten sein, die sich direkt auf die Beziehung bezieht. In Abbildung 12 ist die Beziehung zwischen dem Unternehmen und einem Mitarbeiter durch die Klasse „Arbeitsverhaeltnis“ näher bestimmt.

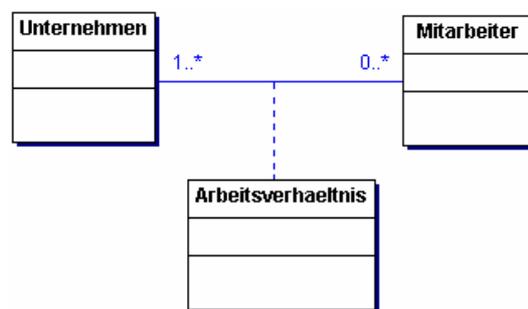


Abbildung 12: Assoziationsklasse

Da im Feindesign solche Beziehungen aufgebrochen werden, sollen sie an dieser Stelle auch schon aufgelöst werden. Die Assoziationsklasse wandelt sich dann in eine eigenständige Klasse, die mit den verbundenen Klassen Beziehungen eingeht. Die Beziehung zwischen den Ursprungsklassen wird dabei aufgelöst. Auch die Multiplizitäten der attribuierten Beziehung werden mit übernommen. Abbildung 13 stellt die gleiche Aussage wie Abbildung 12 dar. Aufgrund der Verringerung der zu transformierenden Elemente wird im Weiteren von aufgelösten Assoziationsklassen ausgegangen. Für die eventuelle Rücktransformation hin zur attribuierten Assoziation sollte die entsprechende Klasse mit einem eindeutigen Stereotyp (zum Beispiel Assoziationsklasse) gekennzeichnet werden.

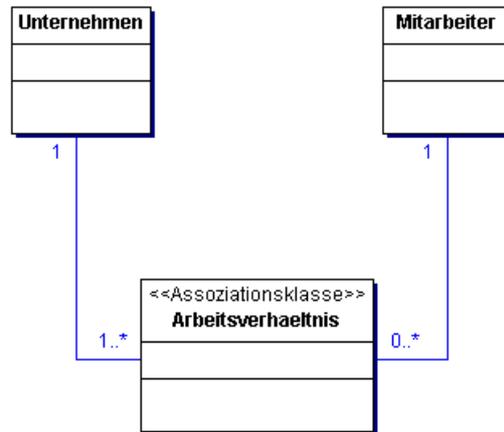


Abbildung 13: aufgelöste Assoziationsklasse³²

Um Beziehungen zu modellieren, an denen mehr als zwei Klassen partizipieren, stellt die UML mit der mehrgliedrigen Assoziation (Abbildung 14) ein Element zur Verfügung, das dies ermöglicht. Hierzu muss jedoch erwähnt werden, dass diese Beziehung in den meisten Fällen nicht so modelliert wird, sondern bereits aufgelöst dargestellt wird.³³ Hierzu wird die Raute durch eine Klasse ersetzt. Nun fungiert die so zwischengeschaltete Klasse als Vermittler zwischen den anderen beteiligten Klassen. Abbildung 15 zeigt die identische Aussage zur mehrgliedrigen Assoziation der Abbildung 14. In der weiteren Betrachtung werden deshalb keine mehrgliedrigen Beziehungen betrachtet, sondern falls sie in der Modellierung auftreten, werden sie durch ihre gleichwertige Klassendarstellung aufgelöst. Um eine eventuelle Rücktransformation der mehrgliedrigen Beziehung zu ermöglichen, bzw. die Verständlichkeit des Modells zu erhöhen, sollte die Information, dass es sich um eine aus einer mehrgliedrigen Beziehung gebildeten Klasse handelt, zum Beispiel durch einen entsprechenden Stereotyp dargestellt werden.

³² Siehe Oestereich (2001), S.233

³³ Together unterstützt die mehrgliedrige Assoziation nicht.

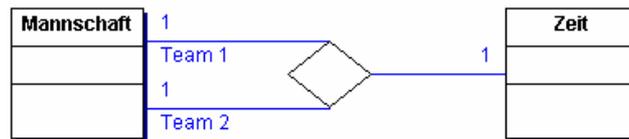


Abbildung 14: mehrgliedrige Assoziation

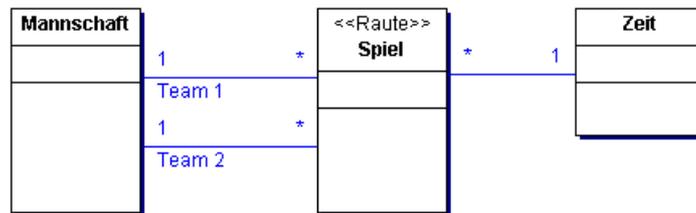


Abbildung 15: aufgelöste mehrgliedrige Assoziation

Des Weiteren gibt es die Möglichkeit eine Ganzes-Teile-Hierarchie mittels der Beziehung Aggregation abzubilden. Dabei wird die Zusammensetzung eines Objektes aus einer Anzahl von Einzelementen beschrieben.³⁴ Die Komposition stellt eine strengere Aggregation dar, bei der die beteiligten Teile vom zugehörigen Objekt existenzabhängig sind.³⁵ Abbildung 16 verdeutlicht die grafische Darstellung beider Beziehungen.

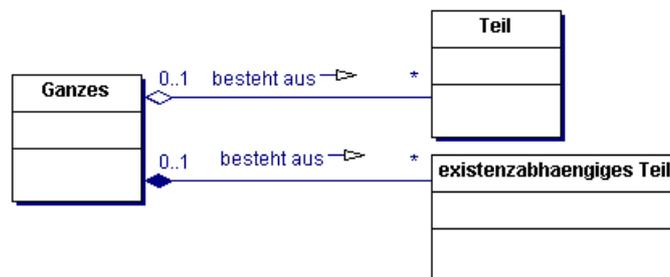


Abbildung 16: Aggregation und Komposition

Tabelle 4 zeigt die für die Transformation von der statischen UML - Sicht zur ARIS - Datensicht wesentlichen Elemente auf.

³⁴ Vgl. Oesterreich (2001), S. 243.

³⁵ Vgl. Oesterreich (2001), S. 245.

UML - Darstellung	Beschreibung
	Klasse
	Attribut
	Generalisierung
	Assoziation
	Aggregation
	Komposition
	Rolle
	Multiplizität
	Stereotyp

Tabelle 4: UML Transformationselemente

6.2.1 Organigrammdarstellung in der UML

Um die notwendigen Akteure eines Anwendungssystems zu ermitteln, bzw. die Aufbauorganisation abzubilden, können Klassendiagramme verwendet werden. Für jedes Element der zu beschreibenden Organisation kann eine eigene Klasse gebildet werden. Gemeinsamkeiten der einzelnen Einheiten werden für notwendige Gruppierungen in einer gemeinsamen Oberklasse zusammengeführt. Auch die Verwendung von Stereotypen für die Unterscheidung von Organisationseinheiten bietet sich an. So können beispielsweise Abteilungen als abstrakte Klassen dargestellt werden, da sich kein Objekt dieser im Anwendungssystem bilden lassen wird, aber die Mitarbeiter einer Abteilung als Objekte zur Laufzeit vorhanden sind, die ähnliche Eigenschaften, wie Bearbeitungsprofile und -berechtigungen aufweisen. Diese können in der übergeordneten Organisationseinheit Abteilung beschrieben werden.

Neben der Generalisierung treten auch andere Kantenbeziehungen auf. So werden in Organigrammen häufig disziplinarische Weisungsbefugnisse aufgezeigt. Aber auch funktionale oder fachliche Weisungsbefugnisse, sowie Beziehungen zur Darstellung von zugehörigen Elementen sind Abbildungen eines Organigramms. Hierbei werden unterschiedliche Beziehungen der

einzelnen Elemente durch stereotypisierte Kanten unterschieden. Abbildung 17 verdeutlicht mögliche Ausprägungen für eine Organigrammdarstellung im UML – Klassendiagramm.

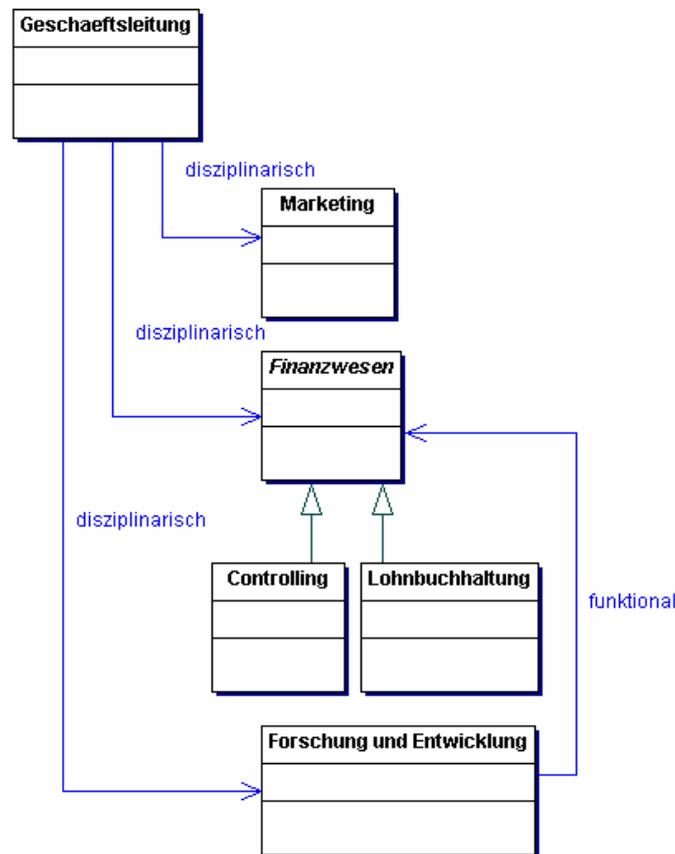


Abbildung 17: UML – Organigramm

In Abbildung 17 ist die Organisationseinheit „Geschaeftsleitung“ allen anderen Organisationseinheiten disziplinarisch vorgesetzt. Die Abteilung Finanzwesen wird durch die ihr untergeordneten Abteilungen Controlling und Lohnbuchhaltung gebildet. Da sie eine Zusammenfassung der Gemeinsamkeiten dieser beiden Abteilungen darstellt und von ihr keine Objekte gebildet werden sollen, wird sie als abstrakte Klasse modelliert. Die funktionale Abhängigkeit durch die Forschungs- und Entwicklungsabteilung wird zusätzlich abgebildet.

Problematisch stellt sich die Modellierung von Personen im Klassendiagramm dar. Diese sind gewöhnlich Objekte zur Laufzeit und müssten aus diesem Grund im Klassendiagramm von einer Superklasse gebildet werden, die das Gerüst für die Mitarbeiter darstellt. Da sie aber daneben auch durch die Stelle beschrieben werden, müssen sie die Eigenschaften, die in der Stellenklasse beschrieben sind, ebenfalls aufweisen. Im Allgemeinen existieren in den jeweiligen zu modellierenden Unternehmensbereichen unterschiedliche Stellen, die durch identische Personen besetzt werden können. So kann ein Mitarbeiter während seiner Zeit im Unternehmen seine Stelle wechseln. Er ist zwar weiterhin die Person, hat aber dann

andere Aufgaben (Operationen) durchzuführen. Aus diesem Grund ist es ratsam zwischen den Objekten zur Laufzeit (natürliche Person) und den Personen, die eine Stelle besetzen, zu differenzieren. Letztere werden dann im Klassendiagramm als eine Subklasse von der Stellenklasse und eine Subklasse der Mitarbeitersuperklasse gebildet. Abbildung 18 verdeutlicht die Zusammenführung beider Klassen. Die Bildung der „Stellenmitarbeiterklasse“ ist notwendig, da Objekte stets von nur einer Klasse gebildet werden können. Das Objekt kann dann direkt von dieser Klasse gebildet werden.

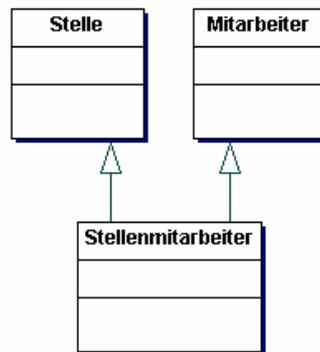


Abbildung 18: Personen im Klassendiagramm

Stereotypen !!!

6.3 Umwandlung von prozess- zu objektorientierter Datenrepräsentation

6.3.1 Klassendiagramm und eERM

Das Klassendiagramm der UML besitzt eine starke Ähnlichkeit mit dem eERM. Für die Transformation werden die Informationen der Datenänderung, also der Operationen einer Klasse, nicht benötigt. Diese werden daher bei der Transformation nicht beachtet. Für die Umwandlung von der UML hin zum ARIS - Modell stellt das somit kein Problem dar. Damit jedoch eine eindeutige Abbildung ermöglicht wird, müssen alle Informationen beider Sichten in die jeweilige andere Sicht mit übernommen werden. Das Entitäten – Relationen – Modell kann alle seine Informationen im UML – Klassendiagramm hinterlegen. Da das Klassendiagramm zusätzliche Informationen zum Beispiel über die Operationen und vertiefende Be-

schreibungen für die Art der Verknüpfung von Klassen darstellen kann, ist eine Verschiebung dieser Informationen in die ERM - Struktur notwendig. Dadurch wird das eERM zusätzlich erweitert. Operationen können wiederum als Entitäten aufgefasst werden, wobei die benötigten Daten in Form von Attributen hinterlegt sind. Der eigentliche Verarbeitungsprozess, also die Darstellung der Operationen wird weder im eERM noch im Klassendiagramm veranschaulicht. Aus diesem Grund ist eine Transformation auf dieser Ebene möglich. Die zusätzlichen Beziehungen können in den Relationen dann wiederum durch Attribute übertragen werden. Hingegen können die Daten, sowohl im eERM, als auch im Klassendiagramm integriert werden. Auch die Information über Kardinalitäten sind in beiden Modellen vorhanden. Hier muss jedoch bei der Überführung vom UML – Klassendiagramm beachtet werden, dass die Information mächtiger ist, da hier ein Wertebereich von Kardinalitäten angegeben wird.³⁶ Diese Information geht bei der bloßen Überführung in das eERM verloren. Aus diesem Grund müssen die Wertebereichsgrenzen als zusätzliche Attribute in die jeweilige Relation mit übernommen werden. So erhält jede Relation zwei Attribute pro beteiligter Entität. Zur Herausstellung welche Attribute zusammengehören wird die Attributsgruppierung verwendet. Die Verbindung zwischen den einzelnen Informationsträgern wird zwar in den Modellen unterschiedlich gehandhabt, aber ist bei beiden ausgeprägt. So finden sich die Informationen sowohl im Klassendiagramm wie auch im eERM wieder. Abbildung 19 zeigt die Transformation zweier assoziierter Klassen.

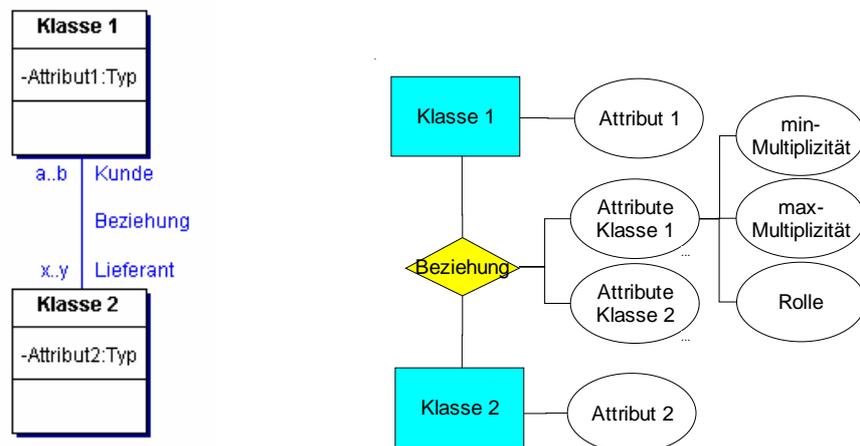


Abbildung 19: Transformation assoziierter Klassen

In der Erweiterung des ursprünglichen Modells von Chen können neben den Obergrenzen auch Untergrenzen (min – max – Notation) und somit ein konkreter Wertebereich für die

³⁶ Vgl. Oesterreich (2001), S.228.

Kardinalitäten angegeben werden.³⁷ Dies wird im allgemeinen aber nur für die vier wesentlichen Fälle (0,1), (0,n), (1,1) und (1,n) abgebildet.³⁸ Wird die Modellierung nach der min – max - Notation im eERM durchgeführt, so ist die Transformation der Multiplizitäten einfacher als in der oben beschriebenen Weise, da eine Eins-zu-Eins-Abbildung direkt möglich ist. Da die Generalisierung und Spezialisierung Abstraktionsprinzipien zur hierarchischen Visualisierung der Semantik eines UML - Klassenmodells dienen, sind sie geeignet, die Generalisierung des ARIS – eERM – Modells darzustellen. Hierbei handelt es sich um eine eindeutige Überführung, da beide Darstellungsformen den gleichen Sachverhalt widerspiegeln. Komposition und Aggregation sind besondere Beziehungen, die aus Gründen der Eindeutigkeit durch ein Identifizierungsattribut in der eERM – Modellierung abgebildet werden müssen. Hierzu wird die entsprechende Relation mit einem Attribut (Komposition oder Aggregation) versehen. Um die Rollen, die bei den Beziehungen im Klassendiagramm modelliert sind, nicht bei der Transformation zu verlieren, müssen auch sie als Attribut der Relation implementiert werden, wobei darauf zu achten ist, dass für jede Rolle ein eigenständiges Attribut verwendet wird.

Schlüsselattribute stellen besondere Attribute des eERM dar. Für die Übernahme der Informationen in das Klassendiagramm müssen sie mit einem Stereotyp (z.B. Schlüsselattribut) gekennzeichnet werden. Sind Informationen über die Stereotypen einer Klasse vorhanden, muss diese als ein Attribut der dazugehörigen Entität übernommen werden. Somit können die Elemente sowohl vom Klassendiagramm in das eERM überführt werden, als auch umgekehrt. Die einzelnen Elemente, die sich gegenüberstehen werden in Tabelle 5 aufgeführt.

³⁷ Vgl. IDS (2001), S.4-38.

³⁸ Vgl. IDS (2001), S.4-39.

eERM - Element	Klassendiagramm - Element

Tabelle 5: Überführung eERM und Klassendiagramm

6.3.2 Klassendiagramm und Organigramm

Die Modellierung des Organigramms kann auch ohne den Zwischenschritt der Transformation in das eERM in ein Klassendiagramm transformiert werden. Analog verhält es sich mit der direkten Überführung vom Klassendiagramm in das Organigramm.

Wichtig hierbei ist, dass alle Klassen als Elemente einer Organisationsabbildung identifiziert werden können. Dies kann durch den Einsatz von Stereotypen problemlos erfolgen, wenn diese eindeutig verwendet werden.

Damit eine Unterscheidung von Datenklassen und Organisationsklassen erfolgen kann und somit eine Transformation automatisiert bzw. einfacher möglich ist, sollten Klassen der Organisation auch einen entsprechenden Stereotyp erhalten. Somit ist dann analog zur Transformation vom ARIS – Organigramm zum eERM auch im UML – Klassendiagramm eine Transformation in beide Richtungen möglich, obwohl hier keine eigentliche Transformation vorgenommen wird, da es sich weiterhin um dasselbe Modell handelt. Für die Transformation in die eERM – Sicht bietet sich jedoch diese Vereinfachung an, da somit nicht zusätzlich recherchiert werden muss, welche Klassen einfache Daten widerspiegeln und bei welchen Klassen es sich um weiter zu transformierende Organisationseinheiten handelt.

Im Organigramm gibt es eine Vielzahl von Organisationseinheitssymbolen und zugehörige nähere Erläuterungen. Für die Transformation dieser Elemente in das Klassendiagramm sind Klassen mit unterschiedlichen Stereotypen notwendig, um eine Rücktransformation des erstellten Klassendiagramms zum Ursprungsorganigramm zu gewährleisten. Die Beschreibung durch Organisationseinheitstypen kann entweder durch eine einfache Assoziation oder aber durch eine Spezialisierung erfolgen. Dies ist abhängig vom jeweiligen Modell. Sinnvoll ist hierbei meiner Ansicht nach die in Kapitel 6.1.4 erläuterte Verwendung von Generalisierung und einfacher Beziehung. Bei der vorgeschlagenen Transformation der Kanten kann eine Darstellung von Organisationsteilen sowohl im Organigramm als auch im Klassendiagramm erfolgen und jeweils in das andere Modell überführt werden. Alle Klassen bilden dann im Organigramm die jeweilige Darstellung. Um die gleiche Mächtigkeit in beiden Darstellungsweisen zu gewährleisten, sollten alle Symbole des Organigramms durch einen entsprechenden Stereotyp im Klassendiagramm dargestellt werden. So können Organisationseinheiten den Stereotyp <<Organisationseinheit>>, Standorte <<Standort>> etc. im Klassendiagramm zugewiesen bekommen. Generalisierung und Assoziation werden im Organigramm als jeweilige Kante dargestellt. Auch hier muss für die Rücktransformation eine eindeutige Abbildung vorliegen. Da Verbindungen zwischen Elementen der Organisationsabbildung auch jeweils im Organigramm vorhanden sind, ist eine Stereotypisierung der Verbindungen zwischen den Klassenelementen zwar redundant, jedoch als Kontrollmittel hilfreich. Eine Transformation der Abbildung 3 in das entsprechende UML Klassendiagramm stellt Abbildung 20 dar. Auch die Überführung von dieser Abbildung hin zum Organigramm ist durch die verwendeten Elemente ohne weiteres möglich.

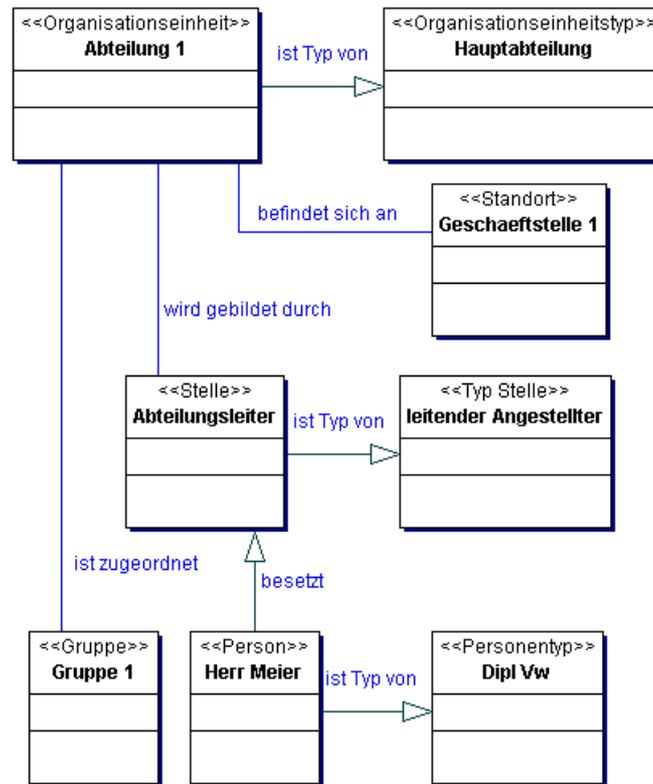


Abbildung 20: Organigramm in Klassennotation

6.4 Ergebnis

Für die Transformation der statischen Sichtweisen beider Modellierungsarten ist zusammenfassend zu sagen, dass eine ausreichende Transformation auf beiden Seiten möglich ist. Jedoch sollte dabei nicht jedes Element der jeweiligen Sicht transformiert werden, da einige Aussagen in der anderen Modellierungswelt unverständlich sind. Beim Forward wie auch beim Reverse Engineering ist es aber von wesentlicher Bedeutung alle Informationen in die jeweilige Betrachtungssicht zu übernehmen, damit keine doppelte Erarbeitung erfolgen muss. So sollten Informationen, die für nur eine Sicht von Relevanz sind zwar mit übernommen werden, aber zur weiteren Bearbeitung dann ausgeschlossen werden. Dies könnte durch geschützte Bereiche oder Filteransichten erfolgen.

Praktische Erfahrungen haben weiterhin ergeben, dass die Datenmodellierung von Unternehmen nur sehr selten in ARIS erfolgt. Dieser Schwachpunkt ergibt sich zum einen aus der ungenügenden Darstellungsmacht und zum anderen aus der nicht Weiterverwendbarkeit der erstellten Modelle.

7 Transformation der dynamischen Sicht

Um Verhalten abzubilden, werden unterschiedliche Methoden angewendet. Für die Geschäftsprozessmodellierung hat sich die prozessorientierte Methode bis zum heutigen Tag durchgesetzt. Die OMG, die die objektorientierte Betrachtungsweise der UML maßgeblich gestaltet, hat erst in den letzten Jahren verstärkt versucht, sich in diesem Bereich zu etablieren. Beide Formen werden von unterschiedlichen Anwendern genutzt. So wird vor allem in Fachabteilungen die ARIS – Methodik angewendet, da sie zum einen leicht zu verstehen ist und zum anderen direkt für dieses Umfeld konzipiert wurde. Mit der Entwicklung einer objektorientierten Beschreibungssprache ist ein Element für die Entwicklung von Anwendungssystemen geschaffen worden, das sich vor allem an IT – Fachkräfte richtet. Da viele Anwendungen aus dem Prozess der Automatisierung entstehen, versucht die OMG auch Geschäftsprozesse durch die UML zu gestalten. Auch mit der UML ist die Geschäftsprozessmodellierung möglich.³⁹ Durch ihre Verwendung kann die Implementierung einfacher erfolgen, da die Ausgangsmodelle schon durch die Entwickler verstanden werden.

Neben diesen beiden Konzepten haben sich auch graphenorientierte Methoden zur Darstellung von dynamischen Verhalten entwickelt. Hier ist besonders die Petri – Netz - Methode zu nennen, die einen formalen Charakter hat und sich dadurch für die Analyse von Geschäftsprozessen eignet. Durch die Transformation in ein Petri – Netz kann sichergestellt werden, dass die Modelle der unterschiedlichen Modellmethoden auch übertragbar sind und gleiche Aussagen widerspiegeln.

7.1 *Dynamisches Verhalten im ARIS – Haus*

Neben der statischen Modellierung können auch die datenverändernden Bestandteile eines Unternehmens dargestellt werden. Hierbei werden die Funktionssicht und die Zusammenführung aller Sichten, die Steuerungssicht, genutzt.

³⁹ Vgl. OMG (2002).

7.1.1 Die Funktionssicht

Um den Zusammenhang zwischen Input und Output darzustellen werden in der Regel Funktionen benutzt. „Eine Funktion ist eine fachliche Aufgabe bzw. Tätigkeit an einem Objekt zur Unterstützung eines oder mehrerer Unternehmensziele.“⁴⁰

Zur Abstraktion können Funktionen in unterschiedlichen Verdichtungsstufen eingesetzt werden. Dabei kann der Geschäftsprozess selbst als die oberste Verdichtungsstufe angesehen werden. Durch Verfeinerungen kann dieser Prozess aufgegliedert werden. Die unterste Stufe bilden die Elementarfunktionen, die keine betriebswirtschaftlich sinnvolle Zerlegung mehr ermöglichen.⁴¹ Die Gruppierung der einzelnen Funktionen kann dabei unterschiedlich erfolgen. So haben Brombacher/Bungert⁴² eine Einteilung in objektorientiert, prozessorientiert oder verrichtungsorientiert vorgenommen. Für diese Form der Unterteilung von Funktionen ist das Modell des Funktionsbaums im ARIS – Haus in die Sicht des Fachkonzeptes integriert. Abbildung 21 bis Abbildung 23 zeigen die unterschiedlichen Gruppierungen der Einteilung auf. Während bei der objektorientierten Betrachtung Funktionen des gleichen Objektes in einem Funktionsbaum gegliedert werden, ist die Verrichtung gleicher Sachverhalte das wesentliche Kriterium bei der verrichtungsorientierten Modellierungsweise. Während bei diesen beiden Gestaltungsmöglichkeiten die Reihenfolge der Funktionen keine Bedeutung hat, ist sie beim prozessorientierten Funktionsbaum von großer Bedeutung, da sie gleichzeitig den Ablauf des Prozesses angibt.

⁴⁰ Siehe IDS Scheer AG (2000), S. 4-1 und 10-787.

⁴¹ Vgl. IDS Scheer AG (2000), S. 4-2.

⁴² Siehe IDS Scheer AG (2000), S. 4-3.

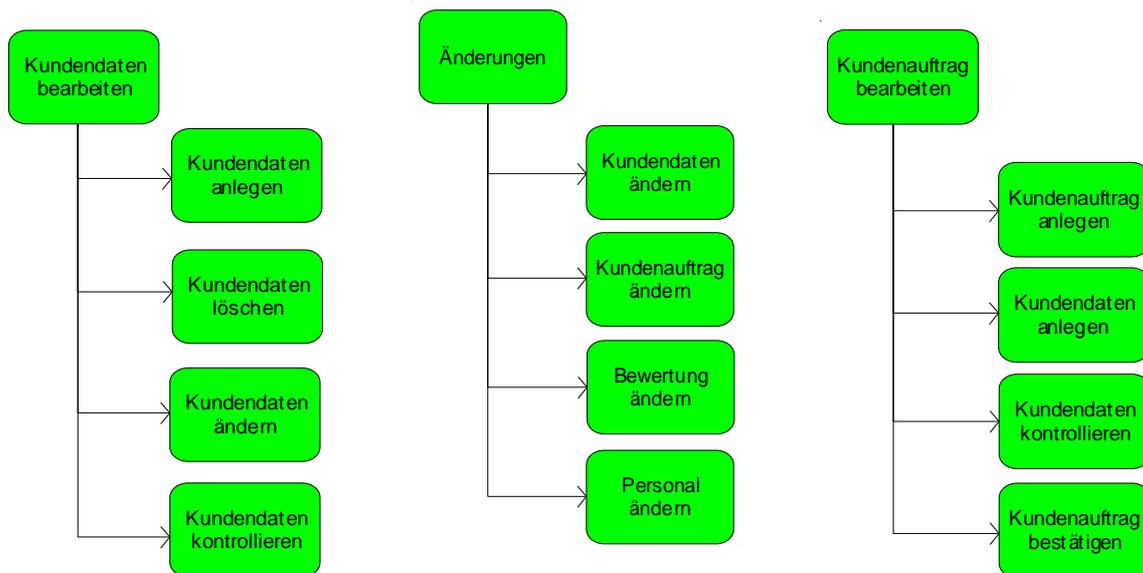


Abbildung 21: objektorientierter Funktionsbaum

Abbildung 22: verrichtungsorientierter Funktionsbaum

Abbildung 23: prozessorientierter Funktionsbaum

Für die Transformation in die objektorientierte Sichtweise bietet sich an dieser Stelle vor allem die Gruppierung nach Bearbeitung eines Objektes an, da dies dann direkt dem betreffenden Objekt zugeschrieben werden kann. Zum Beispiel durch die Integration der Operationen, die das Pendant von Funktionen bilden, in das zugehörige Beschreibungsmodell des Objektes, der Klasse. Aber auch die anderen Gruppierungen lassen sich in der UML - Darstellung abbilden.

Daneben gibt es weitere Diagrammtypen der Funktionssicht, die hier kurz erläutert werden sollen, für die weitere Betrachtung aber keine Rolle spielen.

Das Y-CIM-Modell stellt nur einen Ordnungsrahmen aller Funktionen eines Produktionsbetriebes dar, bei dem die Funktionen in betriebswirtschaftlich-planerische und technische Funktionen unterteilt werden. Somit werden die Aufgaben innerhalb eines Modells geordnet, bei dem die Aufteilung von durchführender und kontrollierender bzw. zielsetzender Funktion im Vordergrund steht. Die betrachteten Aufgaben treten dabei als grobe Abstraktion, das heißt als Tätigkeiten eines großen Bereiches, auf. Die einzelnen Funktionen können mittels Funktionsbaum oder eEPK hierarchisiert sein. Aus diesem Grund wird das Y - Diagramm im weiteren Verlauf nicht weiter betrachtet.

Das SAP – Applikationendiagramm ist ebenfalls ein Modell der Funktionssicht im Fachkonzept. Es bietet die Verknüpfung zu einem SAP – R/3 – Referenzmodell.

Zieldiagramme dienen zur Veranschaulichung von Zielen der Unternehmung. Diese können definiert und hierarchisiert werden. Da Zieldiagramme schon vor der eigentlichen Geschäfts-

prozessmodellierung erstellt werden sollten, damit ersichtlich wird, welche Ziele durch sie erreicht werden können, sind sie für die weitere Betrachtung nicht von Bedeutung.

Für zeitlich-logische Verhaltensweisen von Geschäftsprozessen werden Ereignisgesteuerte Prozessketten genutzt. Diese sind Bestandteil des Fachkonzeptes der Steuerungssicht.

7.1.2 Die Steuerungssicht

Die Steuerungssicht ist der zentrale Bestandteil des ARIS – Hauses. Hier werden die jeweils voneinander unabhängigen anderen Sichten zusammengeführt. So werden die Elemente der Organisation, der Datensicht und die einzelnen Funktionen in den zeitlich-logischen Kontext gebracht. Dabei stehen unterschiedliche Modelle für unterschiedliche Sichtenverknüpfung zur Verfügung.

Das wichtigste Modell ist die erweiterte Ereignisgesteuerte Prozesskette (eEPK). Sie kann sowohl die anderen drei Sichten, wie auch nur zwei von ihnen verbinden. Dadurch ist sie Kernelement der folgenden Betrachtungen.

Für die Verbindung von Organisations- und Funktionssicht stellt ARIS weiterhin das Funktions-/org. Ebenendiagramm zur Verfügung. Es kann damit ein grober Überblick über die Zusammenhänge von Organisationsebenen mit grob-granularen Funktionen gegeben werden. Da es sich jedoch nicht für die Entwicklung von Modellen eignet, die für die Softwareentwicklung hilfreich sind, wird es in dieser Arbeit nicht weiter betrachtet.

Das Vorgangskettendiagramm stellt den gleichen Sachverhalt wie die eEPK dar, wobei eine spaltenweise Unterteilung der Sichten erfolgt. Da es sich nur um eine andere Darstellungsweise zur eEPK handelt, ist eine weitere Betrachtung nicht notwendig.

Funktionen, die direkt an der Wertschöpfung des Unternehmens beteiligt sind, können in einer Wertschöpfungskette im zugehörigen Diagramm dargestellt werden. Hierbei handelt es sich um eine rein betriebswirtschaftliche Sicht, die für die Entwicklung von Software nicht genutzt werden kann. Das Wertschöpfungskettendiagramm gibt einen groben Überblick auf die wichtigsten Funktionen des Unternehmens, die durch hinterlegte Modelle weiter beschrieben sind.

Für die Modellierung von eEPKs steht die Menge der Elemente der anderen Sichten zur Verfügung. Die für die Transformation wesentlichen wurden im oberen Abschnitt beschrieben. Zusätzlich werden Elemente der Steuerung in die Modellierung integriert. Dabei handelt es sich zum einen um Ereignisse, die durch das Eintreten eines betriebswirtschaftlich relevanten

Zustandes eines Informationsobjektes, der den nachfolgenden Ablauf des Geschäftsprozesses steuert oder beeinflusst, definiert sind. Sie sind Auslöser oder Ergebnis von Funktionen. Ein Ereignis ist nur auf einen Zeitpunkt bezogen, das heißt es stellt kein zeitverbrauchendes Geschehen dar.⁴³



Abbildung 24: Ereignis

Daneben werden für eventuelle Entscheidungsverläufe oder zeitgleiches Geschehen Verknüpfungen verwendet. Dabei haben sich mit der Entwicklung von ARIS eine Reihe von vereinfachenden Verknüpfungen ergeben, die jedoch alle mit Hilfe der ursprünglichen Einfach – Verknüpfungen darstellbar sind. Die UND – Verknüpfung (Abbildung 25) soll parallele Schritte ermöglichen oder Nebenläufigkeiten wieder zusammenführen. Die XOR – Verknüpfung (Abbildung 26) stellt das Exklusive Oder dar, hierbei kann eine Aufteilung entweder in die eine oder in die andere Richtung erfolgen. Analog verhält sich auch die Entscheidungszusammenführung. Das Inklusive Oder wird durch die OR – Verknüpfung (Abbildung 27) repräsentiert. An dieser Stelle ist keine eindeutige Aussage darüber möglich, in welchen Zweig die Aktivität läuft. Es kann sowohl einer, mehrere oder alle Zweige aktiviert sein bzw. werden. Dies ist für die Modellierung in anderen Ausdrucksformen meist nicht darstellbar. Aus diesem Grund wird das Inklusive Oder äquivalent umgeformt in Ausdrucksweisen des Exklusiven Oder, sowie des Und. Somit kann dann die Transformation der Verknüpfungen in die anderen Modellierungsarten erfolgen. Da es sich bei der Umformung um eine Äquivalenzaussage handelt, ist auch die Rücktransformation zum Inklusiven Oder möglich.



Abbildung 25: UND - Verknüpfung



Abbildung 26: XOR - Verknüpfung



Abbildung 27: OR – Verknüpfung

Die Umformung eines Inklusiven Oder in eine äquivalente logische Aussage mit zwei Eingängen und einem Ausgang wird in Abbildung 28 aufgezeigt. Dabei können die folgenden drei Fälle auftreten: Erstens liegt nur der linke Eingang vor; Zweitens, nur der rechte Eingang ist aktiviert und drittens beide Eingangskanten sind belegt. Alle drei Fälle können nur exklusiv auftreten. Somit kann die Einteilung der auftretenden Möglichkeiten durch ein Exklusives

⁴³ Vgl IDS Scheer AG (2000); S. 4-93.

Oder differenziert werden. Da die beiden ersten Fälle jeweils nur aus einer Kante bestehen, können sie durch das Exklusive Oder direkt mit der Ausgangskante verbunden werden. Der dritte Fall hingegen verbindet beide Eingänge zuerst durch ein Und, das hiernach durch das Exklusive Oder zur Ausgangskante geführt wird.

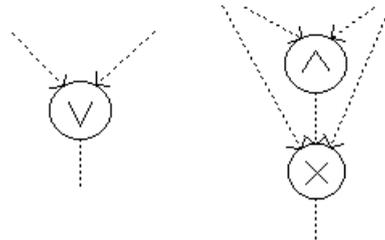


Abbildung 28: OR zu XOR – UND – Darstellung

Da bei Ereignissen und Funktionen jeweils nur eine Ausgangskante vorhanden sein kann, müssen die zusätzlich entstandenen Kanten durch eine UND-Verknüpfung verbunden werden. Somit erhält man dann Abbildung 29⁴⁴. Aus Gründen der Übersichtlichkeit werden im Folgenden die beiden verbindenden UND-Verknüpfungen nicht betrachtet, da in Petri – Netzen aktive und passive Komponenten mehr als einen Ein- bzw. Ausgang besitzen können. Auch bei den UML Modellen ist es möglich, mehrere Eingangs- oder Ausgangstransitionen zu modellieren. Bei ausführlicher Übertragung im Hinblick auch auf die Rücktransformation sollte die Verwendung der zusätzlichen UND - Verknüpfungen garantiert werden, da somit die Regeln der EPK eingehalten werden.

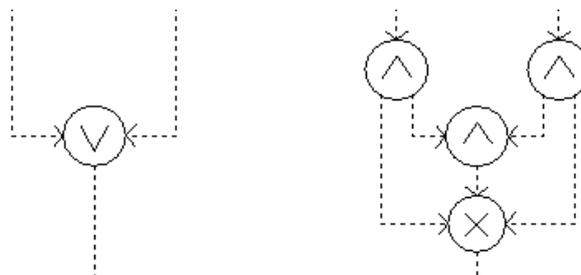


Abbildung 29: XORUND Darstellung

Für die Umformung von Inklusiven Oder mit mehr als zwei Eingängen wird ein zusätzlicher Zwischenschritt benötigt, der eine Abbildung in Inklusives Oder mit jeweils nur zwei Eingängen erzeugt. Diese können dann wiederum mit den obigen Umformungsschritten in eine äquivalente Abbildung ohne Inklusives Oder überführt werden. Abbildung 30 zeigt eine solche Umformung anhand eines Inklusiven Oder mit drei Eingängen. Hierbei treten neben den drei

⁴⁴ Vgl. hierzu Rittgens (2000), S. 11.

Einzel-Eingängen auch noch die Kombination zweier oder die Möglichkeit, dass alle drei Eingangskanten aktiviert sind. Dies lässt sich durch zwei Inklusive Oder mit jeweils zwei Eingängen abbilden, wobei jedes Oder sowohl zwei Einzelaktivitäten berücksichtigt, wie auch ihre Kombination. Der Fall, dass alle Eingänge aktiviert sind, wird durch die Verknüpfung beider Oder abgedeckt.

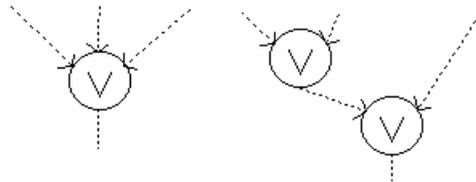


Abbildung 30: Transformation eines OR - Verknüpfung mit drei Eingängen

Die äquivalente logische Umformung eines OR mit weiteren Eingängen kann sukzessiv in eine Darstellung von OR – Verknüpfungen mit jeweils nur zwei Eingängen erfolgen. Für n Eingänge werden somit $n - 1$ inklusive OR benötigt. Diese müssen wiederum durch jeweils ein XOR und ein UND ersetzt werden. Das führt zu einer Erhöhung der verwendeten Verknüpfungen auf $2 \cdot (n - 1)$ Elemente. Die Aussage bleibt dabei die gleiche. Wie in der Darstellungsweise des Inklusiven Oder sind in der transformierten Abbildung $2^n - 1$ Fälle modelliert. Dadurch wird eine Transformation in die anderen Modellierungsarten gewährleistet. Der damit verbundene Mehraufwand kann auch automatisiert verwaltet werden, so dass ein erhöhter Arbeitsaufwand durch den Menschen bei der Modellierung nicht auftritt.

Die Darstellung von einem Eingang und mehreren Ausgängen erfolgt analog. Eine Verwendung von Verknüpfungen mit gleichzeitig mehreren Ein- und Ausgängen ist in der ARIS – Modellierung nicht vorgesehen und auch nicht sinnvoll.

Ein wichtiger Grundsatz bei der Modellierung von EPK_n ist die Einhaltung der Reihenfolge von Ereignissen und Funktionen. Jede Funktion muss mindestens ein Vorereignis besitzen und als Ergebnis mindestens ein Ereignis liefern. Ereignisse können nur mit Funktionen verknüpft werden. Durch die Anpassung der Software an die Praxis kann diese strikte Form der Modellierung bei Trivialereignissen aufgebrochen werden. So dürfen dann zwei Funktionen direkt miteinander verknüpft werden. Da es sich zum einen jedoch um die gleiche Aussage handelt und die Trivialereignisse problemlos integriert werden können und zum anderen nur bei der strikten Befolgung des Modellierungsgrundsatzes eine Transformation in ein Petri – Netz möglich ist, werden in dieser Arbeit eEPK_n nach strenger Modellierungsmethode betrachtet.

Ist in einer eEPK nur die zeitlich-logische Verhaltensweise von Funktionen modelliert, so spricht man von schlanker EPK. Bei der eEPK können alle Elemente der anderen Sichten integriert werden.

Im Vorgangskettendiagramm werden die gleichen Sachverhalte abgebildet.⁴⁵ Hier wird lediglich eine Darstellung in Spaltenweise benutzt. Aus diesem Grund ist die eEPK für komplexe Modelle besser geeignet. Alle Aussagen zur eEPK können auch auf das Vorgangskettendiagramm bezogen werden.

Das Wertschöpfungskettendiagramm dient zur Darstellung von Funktionen, die eine Wertschöpfung im Unternehmen abbilden.⁴⁶ Dabei handelt es sich um eine abstrakte Darstellung der Funktionen, die sowohl hierarchisch, als auch prozessorientiert von einander abhängig sind. Die Informationen werden in einer eEPK verfeinert, so dass auf dieses Diagramm im Folgenden nicht weiter eingegangen, sondern von der Verfeinerung durch die eEPK ausgegangen wird.

Für die Verbindung von Funktions- und Organisationssicht bietet sich das Funktions- / Organisationsebenenendiagramm an. Jedoch wird es im ARIS Toolset anders benutzt, als im zugehörigen Methodenhandbuch. Aufgrund der unterschiedlichen und dadurch missverständlichen Symbolverwendung, soll das Funktions- / Organisationsebenenendiagramm in dieser Arbeit nicht weiter betrachtet werden.

Das Funktionszuordnungsdiagramm stellt die Verbindung der unterschiedlichen Sichten mit einer einzelnen Funktion her. Da stets nur eine Funktion pro Diagramm verwendet werden kann, ist dieser Diagrammtyp für die Transformation nicht hilfreich.

Ein weiterer Diagrammtyp, der die Funktionssicht mit der Organisationssicht verbindet ist die Prozessauswahlmatrix. Hauptprozesse werden durch unterschiedliche Szenarien beschrieben, die wiederum einzelne Prozesse aufweisen. Diesen Elementen der Funktionssicht können Organisationseinheiten zugeordnet werden. In Tabelle 6 werden die Elemente der Prozessauswahlmatrix aufgeführt.

⁴⁵ Vgl. IDS Scheer (2000), S. 4-111.

⁴⁶ Vgl. IDS Scheer (2000), S. 4-114.

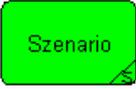
Symbol	Bedeutung
	Hauptprozess
	Szenario
	Prozess
	Organisationseinheit
	Organisationseinheit

Tabelle 6: Elemente der Prozessauswahlmatrix

7.2 Dynamisches Verhalten in der UML

Für die Darstellung von Verhalten gibt es in der UML eine Vielzahl von Modellelementen. Für die zeitlich-logische Ablauffolge bieten sich in der UML Verhaltensdiagramme an.

7.2.1 Das Anwendungsfalldiagramm

Anwendungsfalldiagramme sind Ausgangspunkt von Projektplanungen. Sie sollen auch den Überblick über die notwendigen Punkte des betrachteten Systems gewähren. Da das Anwendungsfalldiagramm eine übersichtliche Skizze liefern soll, kommt es mit einer geringen Menge von Elementen aus. Abbildung 31 beinhaltet alle wesentlichen Elemente eines Anwendungsfalldiagramms. Akteure können dabei sowohl reale Personen wie auch außerhalb des Systems stehende Anwendungssysteme sein. Ein Anwendungsfall beschreibt komplett, unteilbar die Aktivitäten eines Systems aus der Sicht von Akteuren, wobei mindestens ein Ak-

teur den Anwendungsfall initiieren muss.⁴⁷ Eine Verbindung mit dem Klassendiagramm erfolgt beispielsweise über die Akteure, da sie eine Klasse mit dem Stereotyp „Akteur“ darstellen.⁴⁸ Die include Beziehung verdeutlicht, dass ein Anwendungsfall Teil eines anderen Anwendungsfalles ist. Hingegen zeigt die exclude Beziehung eine Erweiterung eines bestehenden Anwendungsfalles an. Um das Verhalten und die Semantik zu vererben, kann die Generalisation verwendet werden. Die Systemgrenze stellt die Abgrenzung der Anwendungsfälle von anderen Systemen dar.

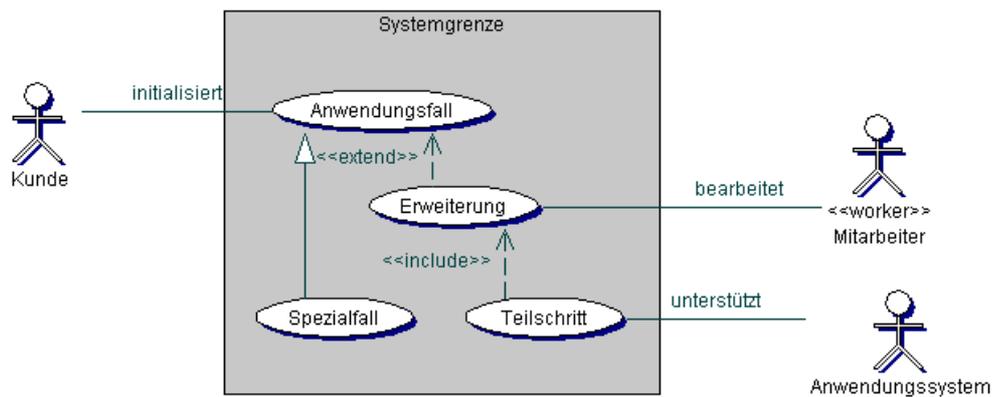


Abbildung 31: Anwendungsfalldiagramm

⁴⁷ Vgl. Oesterreich (2001), S. 174.

⁴⁸ Vgl. Oesterreich (2001), S. 178.

7.2.2 Das Zustandsdiagramm

Die Basis der Zustandsdiagramme sind Statecharts von David Harel aus dem Jahr 1987.⁴⁹ Zustandsdiagramme veranschaulichen eine hypothetische Maschine, die sich zu jedem Zeitpunkt in einer Menge von endlichen Zuständen befindet.⁵⁰ Ein Zustand ist definiert als die Zusammenfassung einer Menge von möglichen Attributwerten eines Objektes.⁵¹ Daneben existieren Start- und Endzustände, die den Anfang bzw. das Ende des Modells anzeigen. Der Wechsel der Zustände wird mittels Transitionen hergestellt. Eine Transition kann dabei auch wieder auf denselben Zustand führen. Ereignisse lösen Zustandsübergänge aus.⁵² Eine unterstützende Beschreibung der Zustände durch beteiligte Objekte ist nicht direkt möglich. Case Tools ermöglichen zwar die Einbeziehung von Objekten, eine Kopplung an die jeweiligen Zustände ist nicht möglich. Dies wird auch ersichtlich aus der Beschreibung des Zustandsdiagramms, da es nur für eine Entität⁵³ die jeweils möglichen Zustände repräsentiert.⁵⁴ Für die Beschreibung von Geschäftsprozessen ist es erforderlich, eine Vielzahl von Objekten in den abbildenden Zustandsautomaten zu integrieren. Somit sind Zustandsdiagramme, wie sie in der UML angeboten werden, für die Geschäftsprozessmodellierung nicht einsetzbar. Sie können nur die Informationen von schlanken EPKn widerspiegeln.

Bei der Verwendung von Case - Tools kann diese Einschränkung behoben werden, wobei dann aufgrund der unterschiedlichen Behandlungen in den einzelnen Werkzeugen keine oder nur eine bedingte Überführung in andere Case - Tools möglich ist. Als Beispiel für die Integration von Objektinformationen stellt sich anhand von Together eine mögliche Lösung wie folgt dar. Dabei sollen die einzelnen Objekte, die an unterschiedlichen Zuständen im System beteiligt sind, verbunden werden. Hierfür bietet sich die Verwendung von Notizen an. Diese dürfen sowohl mit Zuständen wie auch Objekten verlinkt sein. Da dieser Verbindungstyp kei-

⁴⁹ Vgl. Fowler (2000), S. 107.

⁵⁰ Vgl. Oesterreich (2001), S. 265.

⁵¹ Vgl. Oesterreich (2001), S. 266.

⁵² Vgl. Oesterreich (2001), S. 269.

⁵³ Entitäten können im Zusammenhang der UML Instanzen von Klassen, Anwendungsfälle, Akteuren, Operationen und Methoden sein.

⁵⁴ Vgl. OMG (2001), S. 3-137.

ne Richtung aufweist, aber beispielsweise zwischen In- und Output unterschieden werden muss, ist es möglich, diese Information in der Notiz zu hinterlegen. Abbildung 32 verdeutlicht den so ermöglichten Objektfluss.

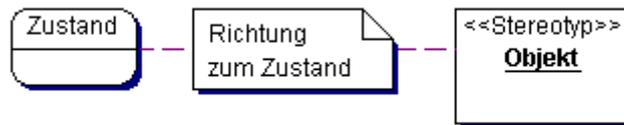


Abbildung 32: Objekte im Zustandsdiagramm nach Together

Tabelle 7 zeigt die Elemente des Zustandsdiagramms, wobei die Objekte eine Ausnahme bilden.

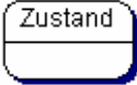
Symbol	Bezeichnung
	Zustand
	Transition
	Startzustand
	Endzustand
	Synchronisation
	Objekt
	Notiz mit Verbindungslinie

Tabelle 7: Elemente des Zustandsdiagramm

7.2.3 Das Aktivitätsdiagramm

Aktivitätsdiagramme sind spezielle Zustandsdiagramme, bei denen der Zustand mit einer internen Aktion als Aktivität dargestellt wird.⁵⁵ Sie kombinieren Ideen aus Techniken des Ereignisdiagramms⁵⁶, der Modellierungstechnik SDL, von Workflow-Modellierung und Petri-Netzen.⁵⁷ Jede Aktivität kann nur einmal verwendet werden, da sie durch ihre Eingangs- und Ausgangstransitionen eindeutig beschrieben ist.

Die Transitionen stellen den Übergang zwischen den einzelnen Aktivitäten dar. Sie werden ausgelöst durch Ereignisse, die die Beendigung des vorhergehenden Aktionszustandes, das Erreichen eines Objektzustandes, das Auftreten eines Signals oder die Erfüllung von Bedingungen anzeigen.⁵⁸

Entscheidungen können sowohl durch eine spezielle (Entscheidungs-)Aktivität, genannt Raute, dargestellt werden⁵⁹, als auch durch mehrere Ausgangstransitionen aus einer Aktivität. Beide Darstellungsformen sind äquivalent. Aufgrund der Erleichterung der Transformation wird deshalb stets von der Modellierung durch die Entscheidungsraute ausgegangen. Die Aufteilung bzw. Zusammenführung von Nebenläufigkeiten wird mittels der Synchronisation veranschaulicht. Hierbei ist zu beachten, dass aufgeteilte Nebenläufigkeiten unbedingt wieder zusammengeführt werden müssen. Jedes Aktivitätsdiagramm beginnt mit einem einzigen Startpunkt und endet in mindestens einem Endpunkt. Verantwortlichkeiten über die einzelnen Aktivitäten können durch Bereiche⁶⁰ gekennzeichnet werden.

Wenn der Objektzustand durch eine Aktivität verändert wird, kann dies durch das Hinzufügen der einzelnen Objekte an die Aktivität erfolgen. Aber auch zur Unterstützung können im Aktivitätsdiagramm Objekte mit den jeweiligen Aktivitäten verknüpft sein. Der Objektfluss wird mittels gestrichelter Linie dargestellt, wobei er den Steuerungsfluss ersetzt, wenn er eine re-

⁵⁵ Vgl. Oesterreich (2001), S. 252.

⁵⁶ Nach Jim Odell.

⁵⁷ Vgl. Fowler (2000), S. 115.

⁵⁸ Vgl. OMG (2001), S. 2-176.

⁵⁹ Vgl. Oesterreich (2001), S. 253.

⁶⁰ Im Englischen: Swim Lane.

dundante Beschränkung liefert.⁶¹ Da der Objektfluss nur unterstützend verwendet wird, ist diese Beschränkung nicht zutreffend und der Steuerungsfluss bleibt erhalten.

Tabelle 8 zeigt die Elemente des Aktivitätsdiagramms.

Symbol	Bezeichnung
	Aktivität
	Transition
	Startpunkt
	Endpunkt
	Synchronisation
	Entscheidung
	Objekt
	Verantwortlichkeitsbereich

Tabelle 8: Elemente des Aktivitätsdiagramms

Zusätzlich werden Sequenz- und Kollaborationsdiagramme zu den Verhaltensdiagrammen gezählt. Sie stellen dabei das Verhalten als Interaktion dar. Sequenzdiagramme gehen dabei auf den zeitlichen Verlauf eines Systems ein, wohingegen Kollaborationsdiagramme die Verbindungen der einzelnen modellierten Gegenstände abbilden.

Für die Geschäftsprozessmodellierung sind beide Diagrammartentypen von geringerer Bedeutung, da sie nur eine beschränkte Betrachtungsweise auf die Unternehmensprozesse gewähren.

⁶¹ Vgl. OMG (2001), S. 3-164.

7.3 Petri – Netze

Petri – Netze verbinden Formalität und Anschaulichkeit für die Gestaltung von Geschäftsprozessmodellen.⁶²

Petri – Netze sind gerichtete Graphen, die aus passiven und aktiven Komponenten bestehen. Diese werden gleichberechtigt behandelt. Für die Simulation der erstellten Modelle kann mittels Markierungen die aktuelle Position im Netz festgehalten werden. Diese Markierungen können nur in den passiven Komponenten auftreten. Die aktiven Komponenten verarbeiten die Marken. Zusätzlich gibt es den Fluss von Informationen, der zwischen den Komponenten erfolgt, wobei Komponenten des gleichen Typs nicht miteinander verbunden werden dürfen. Aktive Komponenten mit einem Eingangsinformationsfluss löschen Informationen. Aktive Komponenten mit einem Ausgangsinformationsfluss erzeugen Informationen.



Abbildung 33: passive
Komponente



Abbildung 34: markierte
passive Komponente



Abbildung 35: aktive
Komponente



Abbildung 36:
Fluß

Eine Verwendung von Kombinationen unterschiedlicher Elemente ermöglicht die Veränderung der Marken im Netz. Verzweigt eine aktive Komponente den Informationsfluss, so entsteht eine UND Nebenläufigkeit, da die Information nach dem Schalten an mehr als einer passiven Komponente vorhanden ist. Die Zusammenführung paralleler Aktivitäten erfolgt ebenfalls durch die aktive Komponente, die durch gleichzeitiges Vorschalten mehrerer passiver Komponenten erst schalten kann, wenn alle Informationsträger belegt sind. Verzweigt eine passive Komponente, so handelt es sich um eine ODER Nebenläufigkeit, da die nachfolgenden aktiven Komponenten um die Marken konkurrieren. Hierbei handelt es sich um die exklusive Oder Nebenläufigkeit. Bei der Zusammenführung durch eine passive Komponente handelt es sich um eine inklusive ODER Zusammenführung, denn die Weiterleitung der Information kann durch beide aktiven Komponenten erfolgen. Die Zusammenführung entspricht dann einer exklusiven ODER Zusammenführung, wenn entweder sichergestellt werden kann, dass nur ein Zweig der Zusammenführung aktiviert ist oder die aktiven Komponenten, die zusammengeführt werden sollen in Konkurrenz stehen. Abbildung 37 zeigt, dass durch Hinzufügen einer markierten passiven Komponente sichergestellt werden kann, dass nur ein

⁶² Vgl. Rittgens (2000), S. 3.

Zweig aktiviert war, da beide aktiven Komponenten nun um die hinzugefügte passive Komponente konkurrieren. Kann sichergestellt werden, dass nur ein Zweig aktiviert wird, z.B. durch die ODER Nebenläufigkeit so ist die Konkurrenzkomponente nicht notwendig. Um die Modelle einfach zu halten, wird im Folgenden davon ausgegangen, dass die ODER Zusammenführung sich wie eine exklusive ODER Zusammenführung verhält, also stets nur einer der zusammengeführten Zweige aktiviert ist.

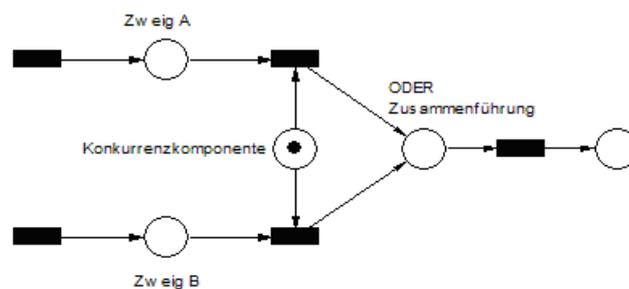


Abbildung 37: exklusive ODER Zusammenführung im Petri - Netz

Zusätzlich lassen sich die Komponenten detaillieren und das Netz durch Hinzufügen weiterer Elemente erweitern.⁶³ Bei dieser Verfeinerung werden einzelne Elemente durch neue Teilnetze ersetzt, die einen detaillierteren Überblick verschaffen. Im Umkehrschluss lassen sich aber auch Teile des Modells abstrahieren. Ergänzungen können durch das Hinzufügen neuer Teilstrukturen gemacht werden. Tabelle 9 fasst die wichtigsten Prozesse von Petri – Netzen zusammen.

⁶³ Vgl Reisig (1985), S. 67.

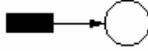
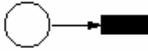
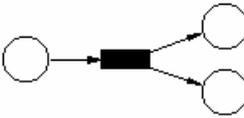
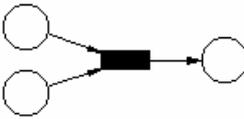
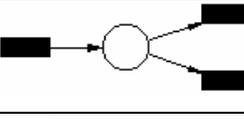
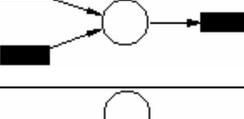
Teilnetz	Bedeutung
	Erzeugung von Marken
	Löschen von Marken
	Und - Nebenläufigkeit
	Und - Synchronisation
	Oder – Alternative
	Oder – Entscheidung
	Verfeinerung

Tabelle 9: Prozesse in Petri - Netzen

Es existieren eine Vielzahl von Petri – Netz – Typen, die sich hauptsächlich durch ihre unterschiedlichen Gestaltungsmöglichkeiten im Bezug auf die Verwendung von Marken unterscheiden lassen.

Bedingungs – Ereignis – Netze stellen das einfachste Netzmodell dar. Je Komponente und Übergang ist nur die Verwendung einer Marke zulässig. Diese Marken sind somit vom Objekttyp boolean. Die Marken werden dabei nicht unterschieden, sondern sind identisch. Die aktiven Komponenten werden als Ereignisse interpretiert, die passiven Komponenten als Bedingungen bezeichnet.⁶⁴

„Ein Ereignis eines Netzes aus Bedingungen und Ereignissen kann eintreten, wenn alle seine Vorbedingungen erfüllt und alle seine Nachbedingungen unerfüllt sind. Solche Ereignisse heißen aktiviert. Ist ein Ereignis aktiviert und tritt es ein, so werden dadurch seine Vorbedingungen unerfüllt und seine Nachbedingungen erfüllt.“⁶⁵

⁶⁴ Vgl. Balzert (2001), S. 347.

⁶⁵ Siehe Reisig (1985a), S. 18.

Abbildung 38 stellt den Sachverhalt der 4 Jahreszeiten dar, wobei die Jahreszeiten die Informationsträger sind und der Wechsel zwischen ihnen durch die zugehörigen Startereignisse der nächsten Jahreszeit symbolisiert sind. Die Markierung zeigt an, welche Jahreszeit vorherrscht.

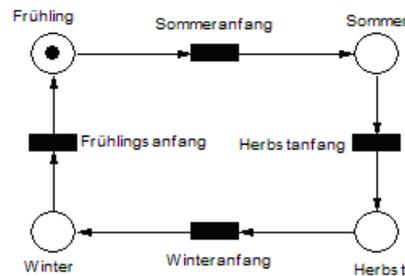


Abbildung 38: Bedingungs- Ereignis - Netz: Die 4 Jahreszeiten⁶⁶

Stellen – Transitions – Netze erweitern das Bedingungs – Ereignis – Netz durch die Aggregation gleicher markierter Stellen. Hier ist die Verwendung mehrerer Marken pro Komponente möglich. Zusätzlich können die einzelnen passiven Komponenten mit einer Kapazität belegt werden. Auch die Übergänge können eine Anzahl von Marken verlangen. Die Marken sind weiterhin identisch. Somit sind die Objekte nun vom Datentyp integer. Die aktiven Komponenten werden als Transition bezeichnet, die passiven Komponenten als Stelle. „Formal sind die Netze aus Bedingungen und Ereignissen gleich denjenigen Netzen aus Stellen und Transitionen, die für jede Stelle die Kapazität 1 und für jeden Pfeil das Gewicht 1 haben.“⁶⁷ Abbildung 39 zeigt ein Bedingungs- und Ereignisnetz von einem Produzenten und zwei Verbrauchern. Die Konsumenten konkurrieren um das Produkt des Produzenten an der Pufferstelle. Der Produzent kann stets nur ein Produkt produzieren (BE – Netz). Der gleiche Sachverhalt kann in Abbildung 40 durch ein Stellen – Transitions – Netz abgebildet werden. Dabei können die Konsumenten zusammen dargestellt werden. Die Information, welcher Konsument das Produkt entnimmt, ginge jedoch verloren. Ein Vorteil besteht in der Verringerung der Komplexität des Modells.

⁶⁶ Nach Reisig (1985), S. 3.

⁶⁷ Siehe Reisig (1985a), S. 30.

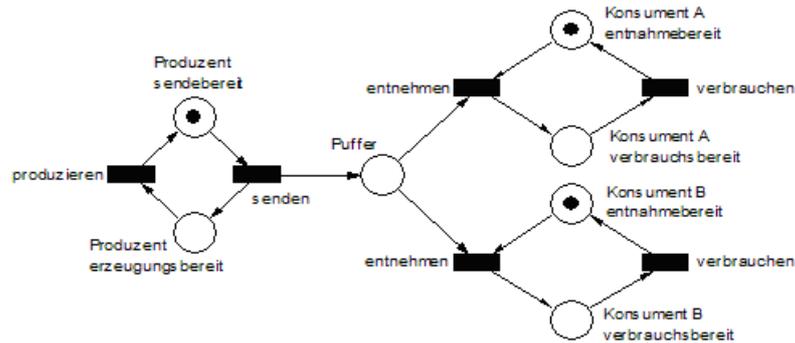


Abbildung 39: BE – Netz 2 Konsumenten und 1 Produzent

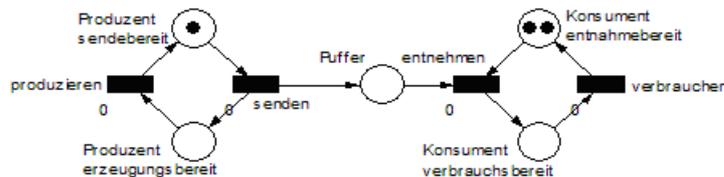


Abbildung 40: ST - Netz 2 Konsumenten und 1 Produzent

Im Kanal – Instanzen – Netz wird mit umgangssprachlichen Mitteln ein Modell konstruiert. Dadurch können jedoch keine präzisen Schaltregeln gefunden werden.⁶⁸ Die passive Komponente stellt die Ressourcen des Modells dar und wird als Kanal bezeichnet. Die Instanz ist die Bezeichnung der aktiven Komponente und verrichtet Aufgaben an den modellierten Sachverhalten. Aufgrund der Auslegungsmöglichkeiten dieser Netze sind sie für einen ersten Grobentwurf geeignet, für die weitere Spezifikation jedoch unbrauchbar. Aus diesem Grund wird in dieser Arbeit nicht weiter auf diesen Netztyp eingegangen.

Netze mit individuellen Marken⁶⁹ unterscheiden die im Netz vorhandenen Marken. Eine Veränderung dieser kann in den aktiven Komponenten erfolgen. Die Kapazitätsangaben werden aus dem Stellen – Transitions – Netz übernommen. Dieses Netz stellt eine Erweiterung des Stellen – Transitions – Netzes dar. Da die Marken unterschieden werden können, ist es möglich, dass die detaillierten Informationen des Bedingungs - Ereignis – Netzes wieder hinzugefügt werden.

Durch Erhöhung der Komponenten der jeweiligen Netze ist eine gleichwertige Darstellung im Bedingungs – Ereignis – Netz möglich. Damit ist jedoch auch eine Erhöhung der Komplexität des Modells verbunden. Es bleibt jeweils abzuwägen, welches Modell herangezogen wird. Für die Geschäftsprozessmodellierung ist nach Ansicht des Autors die Verwendung von Stel-

⁶⁸ Vgl. Reisig (1985a), S. 63.

⁶⁹ Diese Netze werden in der Literatur auch als Prädikat – Transitions – Netz bezeichnet.

len – Transitions – Netzen ausreichend. Bei einfachen Modellen kann auf das Bedingungs – Ereignis – Netz zurückgegriffen werden.

Tabelle 10 fasst die gängigsten Modelle zusammen. Dabei sind die aktiven und passiven Komponenten sowie die maximal zulässige Anzahl der Objekte bzw. die Art der Objekte pro Komponente aufgliedert.

Bezeichnung	Passive Komponente	Aktive Komponente	Elemente
Bedingungs – Ereignis – Netz	Bedingungen	Ereignisse	1, identisch
Stellen – Transitions – Netz	Stellen	Transitionen	Kapazitätsabhängig, identisch
Kanal – Instanzen - Netz	Kanäle	Instanzen	-
Individuelle Marken	Stellen	Transitionen	Objekte, individuell

Tabelle 10: Wichtige Petri - Netze

7.4 Transformation des Anwendungsfalldiagramms

Da bei einem Anwendungsfalldiagramm kein Zusammenhang zwischen den Anwendungsfällen modelliert wird, sondern nur die Beziehungen zwischen den Akteuren, die mit dem System, zum Beispiel als Kunde oder Sachbearbeiter, in Verbindung stehen, kann keine Überführung in die eEPK erfolgen. Bei den Anwendungsfällen handelt es sich um eine sehr abstrakte bzw. zusammengefasste Beschreibung von Vorgängen. In der ARIS – Modellierung wird für diese Form die Prozessauswahlmatrix verwendet, da auch die Akteure in diesem Diagramm, durch Organisationseinheiten, mit einbezogen werden können und eine einfache Strukturierung erfolgen kann. Für die Anwendungsfälle werden Szenarien verwendet. Bei den Akteuren kann zwischen Personen und Anwendungssystemen durch die Verwendung unterschiedlicher Symbole differenziert werden. Die Spezialisierung kann durch eine Kante zwischen zwei Szenarien hergestellt werden, wobei die Information, dass es sich hierbei um eine Spezialisierung handelt mit in den Attributen (z.B. der Kantenrolle) hinterlegt sein muss. Analog verhält sich die Extend – Beziehung. Für Teilschritte bieten sich Prozesse an, hierbei kann aber zusätzlich die Include – Beziehung in der zugehörigen Kante hinterlegt werden.

Die Informationen über die Beziehungen zwischen den Anwendungsfällen und den Akteuren muss in den Kanten zwischen den Organisationseinheiten und Prozessen, bzw. Szenarien ebenfalls vorhanden sein. Auch hierbei bietet sich das Attribut Kantenrolle an. Die System-

grenze kann durch einen Hauptprozess beschrieben werden, da er ein Anwendungssystem darstellt.⁷⁰

Somit sind die wesentlichen Informationen des Anwendungsfalles auch in der Prozessauswahlmatrix beschreibbar. Abbildung 41 stellt das Anwendungsfalldiagramm der Abbildung 31 dar.

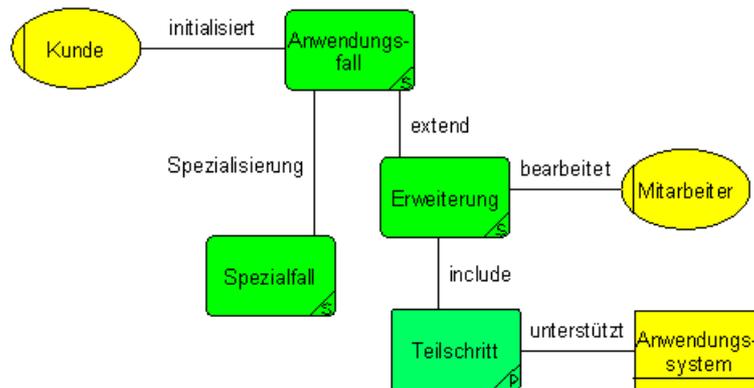


Abbildung 41: Prozessauswahlmatrix

Für die Transformation von Prozessauswahlmatrix und Anwendungsfalldiagramm ergibt sich somit die folgende Transformationstabelle.

⁷⁰ Vgl. IDS Scheer (2000), S. 4-120.

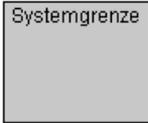
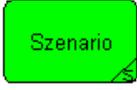
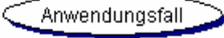
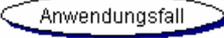
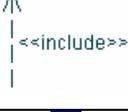
Prozessauswahlmatrix	Anwendungsfalldiagramm
	
	
	
	
	
in den Kanten hinterlegt	
in den Kanten hinterlegt	
in den Kanten hinterlegt	

Tabelle 11: Transformation Prozessauswahlmatrix und Anwendungsfalldiagramm

Für die Transformation von einer Prozessauswahlmatrix zum Anwendungsfalldiagramm müssen die oben genannten Beschreibungen eingepflegt sein. Es darf nur ein Hauptprozess verwendet werden. Da dies eine sehr starke Einschränkung ist, wird sich das Anwendungsfalldiagramm hauptsächlich aus anderen Diagrammtypen, wie der eEPK ableiten lassen⁷¹, wobei jedoch eine Rücktransformation auf Grund der ungenügenden Information nicht möglich ist. Die Abbildung kann jedoch in der Prozessauswahlmatrix erfolgen. Anwendungsfalldiagramme dienen der Abgrenzung von Projekten, die im Wesentlichen einen Überblick verschaffen sollen. Da bei vorliegender eEPK bereits detailliertere Informationen vorliegen, ist eine Erstellung von Anwendungsfalldiagrammen nicht sinnvoll. Es kann dann direkt zu einem anderen Diagrammtyp der UML eine Verbindung hergestellt werden.

⁷¹ Siehe Anhang A.

7.5 Transformation Funktionsebene

7.5.1 Transformation von Aktivitätsdiagramm und Funktionsbaum

Für die Transformation des Funktionsbaums bietet sich das Aktivitätsdiagramm an, da sich in beiden Modellen ähnliche Sachverhalte widerspiegeln. Ist der Funktionsbaum objektorientiert dargestellt, kann das entsprechende Aktivitätsdiagramm für das jeweilige Objekt eine Verhaltensbeschreibung liefern. Anderenfalls kann das Diagramm als eine Abbildung der jeweiligen Tätigkeiten auf mehrere Objekte bezogen werden. Ein wesentliches Merkmal von Funktionsbäumen ist ihre Hierarchisierung durch Hinterlegungen mit anderen Modelltypen. Dabei können Funktionen wiederholt genutzt werden, was eine direkte Abbildung in das Aktivitätsdiagramm erschwert. Um eine Transformation zu ermöglichen, werden Funktionen, die einmalig auftreten, in eine Aktivität mit gleichem Namen abgebildet. Dies soll aufzeigen, dass die interne Aktion der Funktion entspricht. Um Redundanz von Informationen zu vermeiden, wird die interne Aktion, die bei Eintritt der Aktivität ausgeführt wird, nicht extra mit angezeigt. Für die Überführung einer Funktion, die mehrfach verwendet wird, ist die zugehörige Namenskonvention der Aktivität zu beachten. Hierbei könnte jeder Aktivitätsname aus dem Funktionsnamen und einer eindeutigen, nicht relevanten Nummer bestehen. Die interne Aktion hingegen, die die eigentliche Abbildung der Funktion darstellt, trägt den Namen der zugehörigen Funktion. Dadurch ist auch eine Rückführung vom Aktivitätsdiagramm zum Funktionsbaum möglich, wobei jede Aktivität, wenn sie keine interne Aktion besitzt, mit einer Funktion übersetzt wird. Jede Aktivität, die eine interne Aktion hat, kann durch eine Funktion dargestellt werden, die den Namen der Aktion erhält.

Funktionsbäume stellen zudem eine Hierarchisierung dar, die nach unterschiedlichen Kriterien erfolgen kann. Dabei kann jede Funktion einer Hierarchieebene als gleichwertig angesehen werden. Um eine Überführung in das Aktivitätsdiagramm zu ermöglichen, muss die Struktur ersichtlich bleiben. Die Aufteilungsregeln des Aktivitätsdiagramms ermöglichen hierbei die Unterscheidung zwischen den einzelnen Ebenen. Alle Aktivitäten innerhalb einer Aufteilungsregel gehören zur gleichen Hierarchieebene. Jede Aufteilungsregel muss bei der Abbildung wieder zusammengeführt werden, so dass eine eindeutige Zugehörigkeit gewährleistet ist. Die Wahl der Aufteilungsregel ergibt sich aus der jeweiligen Logik. Nicht alle Un-

terfunktionen⁷² müssen notwendigerweise gleichzeitig abgearbeitet werden, wenn die Oberfunktion⁷³ erreicht ist. Aber auch die Bearbeitung mehrerer Unterfunktionen kann vorgesehen sein. Bei der Betrachtung von Funktionsbäumen nach Brombacher/Bungert⁷⁴ ergibt sich die Aufteilungsregel bei objekt- und verrichtungsorientierten Funktionsbäumen als Entscheidung. Hier werden nur einzelne Unterfunktionen bearbeitet. Damit die Darstellung im Aktivitätsdiagramm nicht zu restriktiv ist, sollte vor Beendigung der jeweiligen Oberfunktion eine weitere Möglichkeit gegeben werden, eine Funktion der unteren Ebene aufzurufen. Dies kann durch eine Transition modelliert werden, die die Entscheidungszusammenführung mit der Entscheidungsaufteilung verbindet. Abbildung 42 gibt den gleichen Sachverhalt wie Abbildung 21 wider. Dabei handelt es sich um einen objektorientierten Funktionsbaum, so dass die Entscheidungsraute verwendet wird, die zusätzlich eine Eingangstransition von der Zusammenführung erhält. Somit können bei Auslösung der Funktion „Kundendaten bearbeiten“ auch mehrere Unterfunktionen durchgeführt werden. Um eine Abbildung in den Funktionsbaum zu erhalten, kann beispielsweise die Raute je nach Gliederungsart⁷⁵ bezeichnet werden.

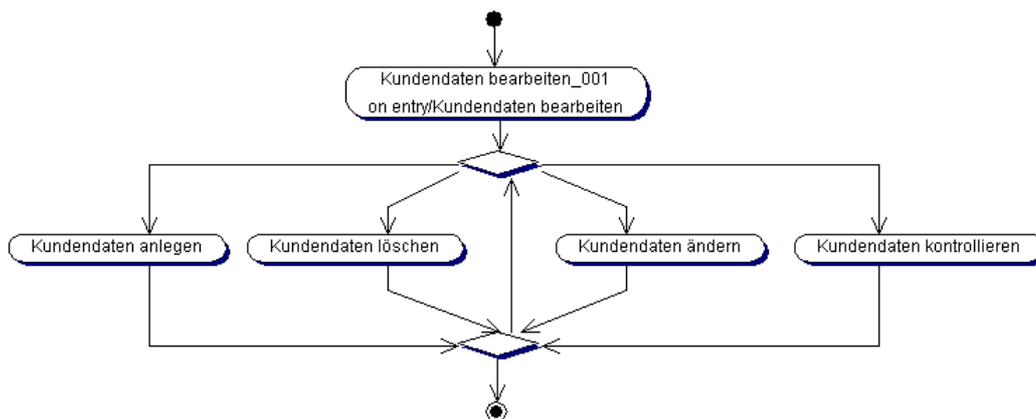


Abbildung 42: Aktivitätsdiagramm (objektorientiert)

Problematisch ist die Hierarchisierung, da die Oberfunktion durch ihre zugehörigen Unterfunktionen eine nähere Beschreibung erhält. Im Aktivitätsdiagramm ist hingegen eine Schachtelung nicht vorgesehen und die Oberfunktion bereits abgeschlossen. Durch den vorgeschriebenen Ablauf kann die näher zu beschreibende Aktivität dann jedoch als „leere“ Aktivität

⁷² Eine Unterfunktion ist Teil der untergeordneten Hierarchieebene.

⁷³ Eine Oberfunktion fasst die ihr hierarchisch untergeordneten Funktionen zusammen und stellt damit eine Abstraktion dieser dar.

⁷⁴ Vgl. IDS (2000), S. 4-3.

⁷⁵ Für diesen Funktionsbaum „objektorientiert“.

angesehen werden, die durch die ihr folgenden Aktivitäten erläutert wird. Somit stellt das Aktivitätsdiagramm in dieser Weise eine Verfeinerung dar. Hierbei ist die Wiederverwendung der Aktivitäten zwar durch die Namensgebung eingeschränkt, jedoch die Abbildung möglich, da die innere Aktion der Funktion entspricht. Somit handelt es sich um eine verfeinerte Beschreibung der inneren Aktion der jeweiligen höher geordneten Aktivität.

Bei prozessorientierter Darstellung wird hingegen eine Hierarchisierung der Unterfunktionen verwendet, da alle untergeordneten Funktionen unbedingt in der Reihenfolge durchgeführt werden müssen, damit die Oberfunktion abgeschlossen werden kann. Somit bieten sich zwei unterschiedliche Darstellungsweisen für den prozessorientierten Funktionsbaum an. Zum einem können alle Unterfunktionen innerhalb der Oberfunktion als interne Aktionen abgebildet werden. Hierbei darf der modellierte Funktionsbaum jeweils nur eine Ebene von der Oberfunktion besitzen. Zusätzlich ist die Reihenfolge der internen Aktionen relevant. Die erste Funktion kann hierbei durch die Bezeichnung „on entry“ erfolgen, alle weiteren nachfolgenden Aktionen werden dann mit „do“ in Verbindung gebracht. Die letzte Aktion kann „on exit“ erfolgen, dies ist jedoch eventuell verwirrend und nicht zwingend notwendig.



Abbildung 43: Aktivitätsdiagramm mit mehreren internen Aktionen (prozessorientiert)

Eine weitere Möglichkeit zur Darstellung von prozessorientierten Funktionsbäumen ist die Verwendung von Einzelaktivitäten für jede Funktion, wobei die vorhergehende Aktivität abgeschlossen sein muss, bevor die nächste durchgeführt werden kann. Da eine Hierarchisierung von Aktivitäten nicht möglich ist, kann eine Nebenläufigkeit zwischen der Oberaktivität und den Unteraktivitäten diese Hierarchisierung ausdrücken, wobei die Oberaktivität dann eine als interne Aktion rein wartende Funktion ausübt, die bei Beendigung der letzten Unteraktivität abgeschlossen wird. Durch die Nebenläufigkeit wird die zeitliche Gleichwertigkeit von Oberaktivität und deren Unteraktivitäten gewährleistet. Abbildung 44 stellt den Sachverhalt von Abbildung 23 dar. Hierbei kann der Funktionsbaum auch mehrere Ebenen aufweisen, die im Aktivitätsdiagramm durch verschachtelte Nebenläufigkeiten auftreten. Für die Rücküberführung ist die Unterscheidung von Ober- und Unterfunktion dadurch gegeben, dass einer Oberfunktion mehrere Unterfunktionen zugeordnet werden. Für den Fall, dass die Oberfunktion nur

durch eine Unterfunktion dargestellt wird, ist eine grafische Anordnung im Aktivitätsdiagramm ausreichend, um die Hierarchisierung abzubilden.⁷⁶ Damit die Oberaktivität erst bei Abschluss der letzten untergeordneten Funktion erfolgt, kann eine Visualisierung der Bedingung an die Ausgangstransition der Oberaktivität erfolgen.

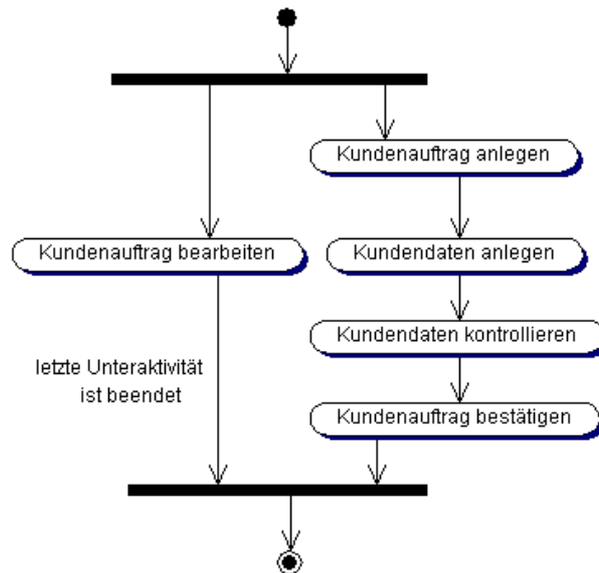


Abbildung 44: Aktivitätsdiagramm (prozessorientiert)

Eine Überführung von Aktivitätsdiagrammen in Funktionsbäume ist somit auch gegeben, wenn die Verwendung bei einstufigen durch mehrere interne jedoch geordnete Funktionen oder bei allgemeinen Funktionsbäumen die Hierarchisierung durch Nebenläufigkeiten erfolgt. Hierbei ist die Aktivität der höheren Ebene zum einen im jeweiligen Nebenläufigkeitsbereich allein vorhanden und zum anderen durch eine interne Aktion gekennzeichnet, die erst beendet ist, wenn die letzte Aktivität der Nebenläufigkeit vollendet ist. Tabelle 12 zeigt die wesentlichen Elemente der Transformation auf, wobei die Information über die Art des Funktionsbaumes zusätzlich in der Bezeichnung der Entscheidungsrauten angegeben werden muss.

⁷⁶ Oberaktivitäten werden in den Abbildungen stets im linken Ast der Nebenläufigkeit abgebildet.

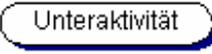
Funktionsbaum	Aktivitätsdiagramm
	
	
prozessorientiert	
objektorientiert	
verrichtungsorientiert	

Tabelle 12: Transformation Funktionsbaum und Aktivitätsdiagramm

7.5.2 Transformation von Zustandsdiagramm und Funktionsbaum

Da Funktionsbäume aus einer Ordnung von Funktionen bestehen, und diese auch als ein Zustand betrachtet werden können, bei dem sich die Funktion in Bearbeitung befindet, ist eine Transformation von Zustandsdiagrammen und Funktionsbäumen möglich.

Da ein Zustandsdiagramm sich auf ein Objekt bezieht, sollte es nur aus objektorientierten Funktionsbäumen abgeleitet werden. Damit werden Zustandsdiagramme zur Verfügung gestellt, die die wesentlichen Objektzustände wiedergeben. Jedoch wird dabei nicht jeder vorhandene Zustandsübergang aufgezeigt. Das Diagramm kann somit nicht als vollständig bezeichnet werden, sondern dient als ein Entwurf für mögliche Zustände. Aus diesem Grund ist eine Transformation zum Funktionsbaum möglich, aber Informationen des Zustandsdiagramms werden hierbei vernachlässigt. Die Funktion der höheren Ebene wird durch einen Oberzustand ausgedrückt. Ihr liegen die Unterfunktionen als innere Zustände vor, wobei jeder Unterzustand zu einem Verlassen des Oberzustandes führen, aber auch jeden inneren Zustand, sich selbst einschlossen, aufrufen kann. Damit die Darstellung durch die vielen Transitionen nicht unübersichtlich wird, ist der Verteilungszustand hinzugefügt, der die Steuerung übernimmt. Hierbei kann auch von einer Auslagerung der Entscheidung gesprochen werden.

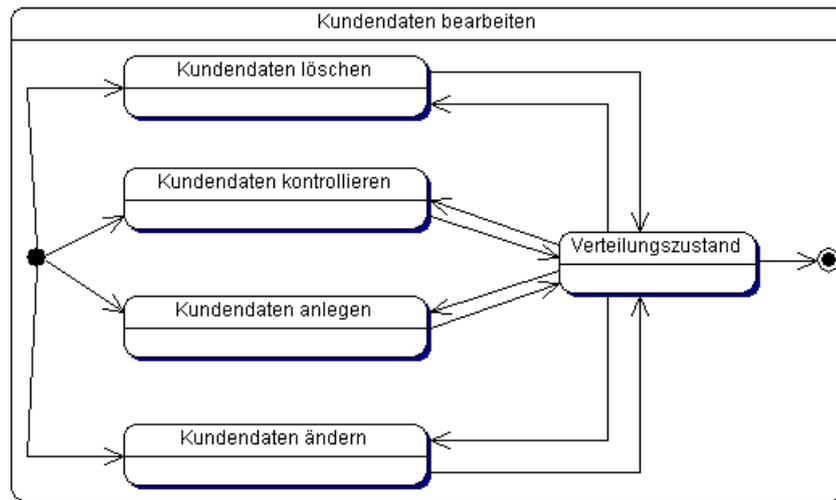


Abbildung 45: Zustandsdiagramm (objektorientiert)

Alle dargestellten Elemente lassen sich auch in den Funktionsbaum transformieren. Dabei gibt der Verrichtungszustand lediglich an, dass es sich um gleichwertige Funktionen handelt, also eine Ablauffolge nicht vorgegeben ist. Dieser Zustand wird nicht mit übertragen. Alle Unterzustände sind im Funktionsbaum durch Unterfunktionen gekennzeichnet. Der zugehörige Oberzustand bildet die Oberfunktion. Auch eine mehrstufige Hierarchisierung lässt sich dadurch effizient abbilden.

Die Verwendung von verrichtungsorientierten Funktionsbäumen kann zwar prinzipiell in analoger Weise erfolgen, jedoch ist diese Übertragung nicht sinnvoll und verwertbar, da nur gleiche Verrichtungsweisen, die mit unterschiedlichen Objekten zusammenhängen, abgebildet werden. Somit werden die abgebildeten Zustände zwar untereinander in Verbindung gebracht, jedoch ist diese für die weitere Verwendung im Zustandsdiagramm nicht ausreichend, da mehrere eventuell notwendige Zwischenschritte aus anderen Verrichtungen an dieser Stelle nicht modelliert sind.

Der prozessorientierte Funktionsbaum stellt eine Nacheinanderschaltung von Funktionen dar, die auch als innere Zustände betrachtet werden können, wobei sie durch ihre Hintereinanderschaltung den Prozess abbilden. Die Hierarchisierung kann dabei durch die Verwendung von Ober- und Unterzuständen erreicht werden, die Ablauffolge durch das Hintereinanderschalten der Unterzustände, wobei der Oberzustand erst bei Beendigung der Ablauffolge erreicht ist. Abbildung 46 zeigt den Sachverhalt des Funktionsbaumes der Abbildung 22. Hierbei wird der Prozess deutlich durch die Reihenfolge der Zustände herausgestellt.

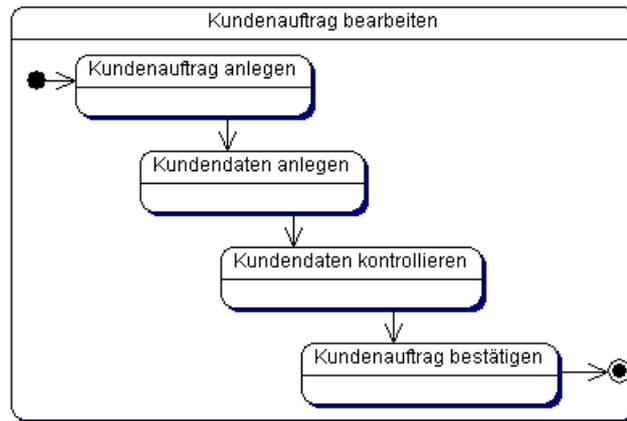


Abbildung 46: Zustandsdiagramm (prozessorientiert)

Die Transformation zum Funktionsbaum kann aus diesem Diagramm durchgeführt werden, wobei die Unterzustände gemäß ihrem Ablauf im Funktionsbaum hierarchisiert werden müssen.

Funktionsbaum	Zustandsdiagramm
Oberfunktion	Oberzustand
Unterfunktion	Unterzustand
prozessorientiert	U1 → U2
objektorientiert	Verteilungszustand
verrichtungorientiert	-

Tabelle 13: Transformation Funktionsbaum und Zustandsdiagramm

7.5.3 Fazit Transformation der Funktionsebene

Für die Transformation der Funktionsebene eignen sich sowohl das Aktivitätsdiagramm als auch das Zustandsdiagramm. Dabei sind durch die Hierarchisierungsmöglichkeiten des Zustandsdiagramms die dortigen Modelle intuitiver, es werden keine größeren zusätzlichen Erklärungen benötigt, die Modelle im Aktivitätsdiagramm sind im Gegensatz dazu verschieden interpretierbar. Auch die Transformation vom Zustandsdiagramm zum Funktionsbaum ist leichter durchzuführen. Wohingegen bei Aktivitätsdiagramm zusätzliche Restriktionen zu beachten sind, wie zum Beispiel die Verwendung von Ober- und Unteraktivitäten.

7.6 Transformation von eEPK und Aktivitätsdiagramm

Die direkte Transformation von eEPK und Aktivitätsdiagramm soll zunächst aus der Überführung beider Modelle in ein Petri – Netz abgeleitet werden. Dies ist deshalb sinnvoll, da die unterschiedlich verwendeten Sprachen leicht zu Fehlinterpretationen führen können. Durch den Zwischenschritt der Transformation in ein Petri – Netz lassen sich die möglichen Einschränkungen bei der Transformation leichter ermitteln und die direkte Transformation von eEPK und Aktivitätsdiagramm ableiten.

7.6.1 Transformation eEPK und Petri – Netz

Für die Abbildung von eEPKs bieten sich eine Reihe von Petri – Netzen an.

In einem ersten Schritt soll die schlanke EPK in einem Bedingungs – Ereignis – Netz abgebildet werden. Die Erweiterungen zur eEPK werden dann durch weitere Netze und somit zusätzliche Eigenschaften dargestellt. Die Verwendung von gefärbten Petri – Netzen⁷⁷ nach Chen und Scheer⁷⁸ wird in dieser Arbeit durch die Transformation der Inklusiven Oder Verknüpfung nicht notwendig. Dadurch erhöht sich jedoch bei komplexen Modellen die Anzahl der Komponenten drastisch. Dies kann durch die Umformung des an dieser Stelle benutzten Netzes zum gefärbten Petri – Netz neutralisiert werden. Jedoch kann dieses gefärbte Petri - Netz in ein äquivalentes Stellen – Transitions – Netz und somit in ein Bedingungs – Ereignis – Netz überführt werden.⁷⁹

Die Transformation von EPK und Petri – Netzen wird in der Literatur häufig durch die Zuordnung von Funktion und aktiver Komponente, sowie Ereignis und passiver Komponente durchgeführt. Dies ergibt sich aus dem Sachverhalt, dass Ereignisse vorliegen und Funktionen

⁷⁷ Gefärbte Petri - Netze gehören zur Klasse der Netze mit individuellen Marken.

⁷⁸ Siehe Chen (1994), S. 8 ff.

⁷⁹ Vgl. Rittgens (2000), S. 6.

aktiv den Prozess bilden.⁸⁰ Für die Transformation von prozessorientierter zu objektorientierter Methode bietet sich jedoch die entgegengesetzte Darstellung an.⁸¹

Ereignisse bezeichnen Zustandsübergänge und werden deshalb durch aktive Komponenten im Petri – Netz dargestellt.⁸² Funktionen können sowohl als passive wie auch aktive Komponente betrachtet werden. Als Zustand kann die Funktion betrachtet werden, wenn sie als „Funktion in Bearbeitung“ angesehen wird.⁸³ Somit wird die Funktion dann durch eine passive Komponente im Petri – Netz dargestellt. Die Verbindungen zwischen den Ereignissen und Funktionen werden analog im Petri – Netz durch den Fluss von Gegenständen und Informationen visualisiert. Die Namen der Ereignisse und Funktionen können auch direkt im Petri – Netz integriert werden. Ebenso werden die zusätzlichen Informationen in der Beschreibung der jeweiligen Komponente hinterlegt. Somit sind alle Informationen sowohl in der EPK wie auch im Petri – Netz vorhanden und eine Abbildung kann in die jeweils andere Methode erfolgen. Auch die beiden Verknüpfungen der EPK, das UND sowie das Exklusive Oder können durch die Elemente des Petri – Netzes dargestellt werden.

Die auch als Konjugation bezeichnete Aufteilung bzw. Zusammenführung von Nebenläufigkeiten wird durch Vervielfachung bzw. das Verschmelzen von Objekten erreicht.⁸⁴ Dafür wird die diesbezügliche Komponente entweder mit mehreren Eingangstransitionen (Verschmelzung) oder Ausgangstransitionen (Vervielfachung) verbunden. Somit ist die Und – Verknüpfung im Petri – Netz durch eine aktive Komponente darstellbar. Die Integration der Anschlussstellen erfolgt dabei abhängig von den vorhergehenden Komponenten durch Dummykomponenten.

Die auch als Exklusivdisjunktion⁸⁵ bezeichnete Verknüpfung wird durch eine willkürliche Verzweigung oder Synchronisationsstelle repräsentiert.⁸⁶ Dadurch kann die XOR – Verknüpfung durch eine passive Komponente ausgedrückt werden, die ebenfalls durch Dummykomponenten in Zusammenhang mit den restlichen Komponenten gebracht wird.

⁸⁰ Vgl. Moldt (2000) S.3 und Rittgens (2000), S. 4.

⁸¹ Siehe Anhang B.

⁸² Vgl. Chen (1994), S. 9.

⁸³ Vgl. Chen (1994), S. 9.

⁸⁴ Vgl. Balzert (2001), S. 358.

⁸⁵ Ein weiterer in der Literatur verwendeter Begriff ist Antivalenz.

⁸⁶ Vgl. Balzert (2001), S.358.

EPK_n beginnen und enden stets mit einem Ereignis. Da sie bei der Transformation in ein Petri – Netz durch aktive Komponenten repräsentiert werden, sollte sowohl den Startereignissen eine passive Komponente als Quelle für Objekte vorgestellt werden, wie auch dem Endereignis in der Petri – Netz – Notation eine tote Stelle nachgestellt werden. Dies ist nicht zwingend notwendig, aber für eventuelle Testläufe des Petri – Netzes sinnvoll, da hierdurch nicht ständig neue Marken erzeugt bzw. durchgelaufene Marken gelöscht werden. Die so erzeugten passiven Komponenten sollten durch eine Beschreibung (zum Beispiel automatisch erzeugt oder Dummy – Komponente) herausgestellt werden, damit sie bei der Rücktransformation in die EPK vernachlässigt werden können.

Die Einbeziehung zusätzlicher Informationen, wie Funktionsverantwortlichkeit oder Leistungselemente kann im Petri – Netz durch eigene passive Komponenten dargestellt werden, die mit dem nachfolgenden Ereignis verknüpft sind. Somit gehören alle Komponenten, die unmittelbar vor der aktiven Komponente liegen und markiert sind zur zugehörigen Funktion. Gleiche Organisationseinheiten können entweder durch eine Stelle gekennzeichnet, oder auch durch einen Fluss dargestellt werden, wobei die einzelnen Stellen dann die Organisationseinheit an Funktion X bezeichnen. Da aufgrund der Möglichkeit, einen Prozess mehrfach zu starten bei der letztgenannten Modellierungsweise ein Warteprozess modelliert wird, der den folgenden Durchlauf erst bei Vollendung des vorhergehenden beginnt, wird diese Modellierung hier nicht weiter verfolgt. Um Prozesse erfolgreich zu beenden, kann eine Priorisierung der einzelnen Kanten zum gewünschten Ziel führen.

Tabelle 14 stellt die jeweiligen Elemente der schlanken EPK und des Bedingungs - Ereignis – Netzes gegenüber. Da in beiden Modellen die gleichen Informationen vorhanden sind, ist eine Transformation sowohl vom Petri – Netz zur EPK, wie auch von der EPK zum Petri – Netz möglich.

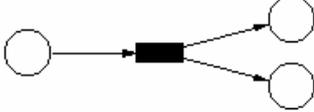
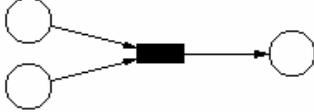
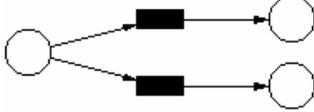
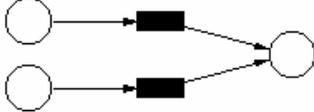
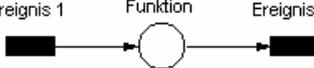
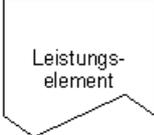
Schlanke EPK	Bedingungs – Ereignis – Netz
	Ereignis 
	Funktion 
	 
	 
	Ereignis 1 Funktion Ereignis 2 
	
	 oder 

Tabelle 14: Transformation eEPK und Petri – Netz

Die wesentlichen Informationen der EPK können in ein Petri – Netz überführt werden. Bei der eEPK sind zusätzliche Informationen vorhanden, die auch in einem Bedingungs – Ereignis – Netz abgebildet werden können. Dadurch kann es aber zu einer stark erhöhten Anzahl der Komponenten kommen. Zur Verringerung der Anzahl der Komponenten und zur Erleichterung der Übersichtlichkeit bieten sich andere Netze an. So können bei Stellen – Transitions – Netzen mehrere Marken in den durch Kapazitätsangaben beschränkten Komponenten verwaltet werden. Es ist möglich, die Information von Dokumenten und Organisationseinheiten der eEPK als passive Komponenten in das Stellen – Transitions – Netz zu integrieren und

dadurch eine gleichwertige Abbildung der eEPK in der Ausdrucksweise eines Petri – Netzes zu erhalten, bei denen jedes Element der eEPK einer Komponente im zugehörigen Stellen – Transitions – Netz zugeordnet werden kann. Dies erleichtert wiederum die Überführung eines Modells der Petri – Netz – Notation in die ARIS – Sichtweise. Beispiele für die Probleme von Bedingungs – Ereignis – Netzen im Zusammenhang mit der Transformation sind mehrfach besetzte Stellen oder mehrfach benötigte bzw. vorliegende Dokumente.

Abbildung 47 zeigt eine eEPK am Beispiel einer Antragsausfüllung. Hierbei wird nach dem Eintritt des Ereignisses Antragswunsch die Funktion Antrag ausfüllen durch den Antragsteller (modelliert als Person Herr Wunsch) ausgeführt. Dabei benötigt er einen leeren Antrag zur Durchführung. Ergebnis dieser Funktion sind der ausgefüllte Antrag und das Ereignis Antrag ausgefüllt. Bevor nun der Vorgang beendet wird, prüft Herr Wunsch nochmals den Antrag. Dadurch entdeckte Fehler werden korrigiert. Somit stellt der ausgefüllte Antrag sowohl einen Input, wie auch einen Output der Funktion Antrag überprüfen dar.

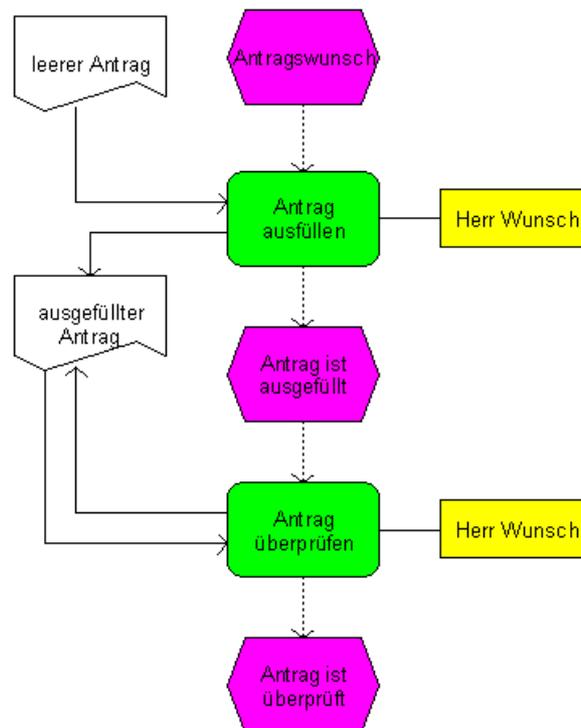


Abbildung 47: eEPK Antrag ausfüllen

Für die Darstellung dieses Sachverhaltes in der Notation eines Stellen – Transitions – Netzes werden zuerst die Ereignisse durch aktive Komponenten ersetzt. Die Funktionen werden durch passive Komponenten (Stellen) abgebildet. Die Verbindungen zwischen den Ereignissen und Funktionen, der Kontrollfluss wird übernommen. Für die Organisationseinheiten werden Stellen hinzugefügt, die entsprechend der realen Mitarbeiterzahl mit Marken gefüllt werden. Ebenso werden die vorliegenden Inputdokumente, die nicht gleichzeitig Output des

vorhergehenden beschriebenen Vorgangs sind, durch passive mit Marken gefüllte Komponenten ersetzt. Jedes Outputdokument wird durch eine Stelle repräsentiert. Da nur unterschiedliche Komponenten untereinander verbunden werden dürfen, kann die Verknüpfung nicht zwischen Funktion und Organisationseinheit bzw. Dokument erfolgen, sondern diese Elemente müssen mit der Komponente verbunden werden, die das Ereignis nach der Funktion darstellt. Dabei werden Organisationseinheiten sowohl als Eingang in die Transition wie auch als Ausgang aus der Transition dargestellt, da sie nach Durchführung der Funktion wieder vorliegen. Stellen die Inputdokumente lediglich Informationen zur Verfügung und werden durch die Funktion nicht verändert, sondern liegen unverändert vor, werden sie analog zu den Organisationseinheiten verbunden. Durch die Funktion genutzte Inputdaten werden durch eine Verbindung von Dokumentenkomponente in die der Funktion unmittelbar folgenden Transition dargestellt. Outputdaten werden hingegen durch eine Verbindung von der Transition modelliert. Bei der Beteiligung von logischen Konnektoren, die unmittelbar vor dem Ereignis liegen, erfolgen die Verbindungen nicht direkt mit dem Ereignis, sondern bereits mit der ersten aktiven Komponente des Konnektors. In Abbildung 48 werden den Start- und Endereignissen Quellen bzw. Archive vor bzw. nachgestellt, um eine Simulation zu vereinfachen und um eine leichtere Verständlichkeit des Modells zu gewähren. Die Startquelle erhält dabei eine Marke, so dass der Vorgang einen Durchlauf simulieren kann. Zusätzlich wird das Dokument leerer Antrag durch eine Stelle modelliert, in der sich beim Start drei Elemente befinden.

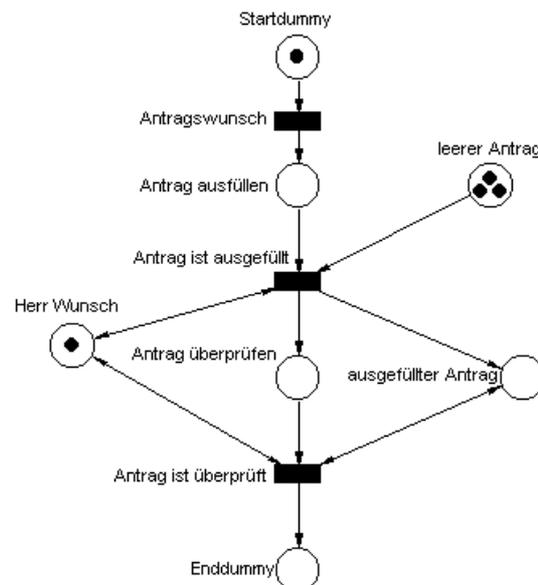


Abbildung 48: Stellen - Transitions – Netz Antrag ausfüllen

Eine Transformation des Stellen – Transitions – Netzes der Abbildung 48 in eine eEPK ist jedoch nicht möglich, da die Informationen, um welches Symbol der eEPK es sich bei den

einzelnen Komponenten handelt, fehlen. Diese können aber als Information im Petri – Netz hinterlegt werden. So zeigt Abbildung 49 ein in die eEPK der Abbildung 47 überführbares Stellen – Transitions – Netz. Hierbei sind die Informationen, welche Elemente der ARIS – Sicht durch die jeweilige Komponente dargestellt wird, integriert. Für eine vollständige und somit eindeutige Transformation müssen stets alle Elemente mit ihrer Gesamtheit an Informationen eindeutig abgebildet werden.

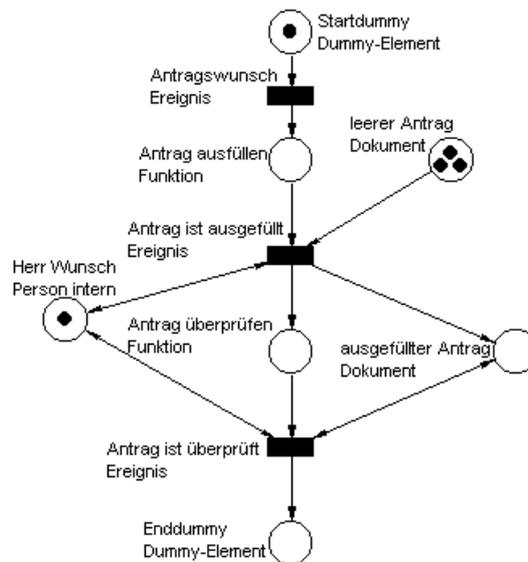


Abbildung 49: Stellen - Transitions - Netz Antrag ausfüllen (überführbar)

Für die Transformation in die eEPK müssen nun die einzelnen Symbolinformationen genutzt werden. Dabei werden Dummy-Elemente nicht mitübertragen. Diese können durch Start- und Endereignisse und logische Konnektoren entstanden sein. Organisationselemente, die im Petri – Netz mit Ereignissen verbunden sind, müssen in der eEPK mit der Funktion verbunden werden, die unmittelbar vor dem Ereignis ausgeführt wurde. Analog verhält es sich mit Stellen die Leistungselemente wie zum Beispiel Dokumente darstellen. Für den Fall der Beteiligung von Konnektoren ergeben sich die Verbindungen der Organisations- und Leistungssicht durch die oben beschriebene Vorgehensweise. Abbildung 50 und Abbildung 51 verdeutlichen den gleichen Sachverhalt einer logischen Verknüpfung.

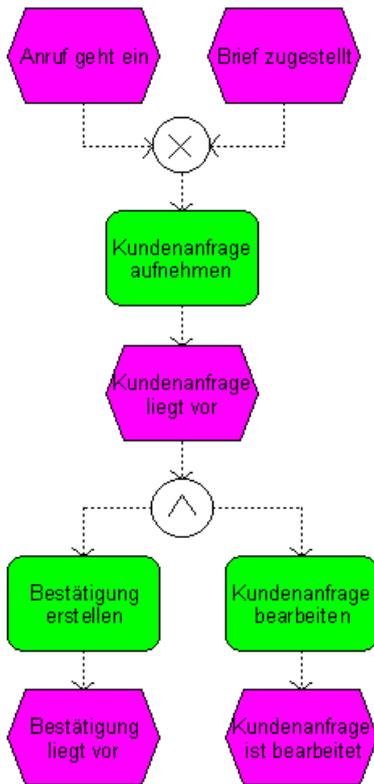


Abbildung 50: EPK mit Verknüpfung

Eine weitere Darstellungsmöglichkeit betriebswirtschaftlicher Prozesse stellen Netze mit individuellen Marken dar. Hierbei werden keine identischen Marken mehr verwendet, sondern die einzelnen Marken können unterschieden werden. Dadurch erhöht sich die Aussagekraft und die Komplexität des Modells wird verringert, weil die Anzahl der Komponenten reduziert wird. Da es sich aber um die gleichen Aussagen handelt und auch eine Überführung in andere Petri – Netze möglich sind, soll an dieser Stelle auf eine weitere Beschreibung verzichtet werden.

7.6.2 Transformation Aktivitätsdiagramm und Petri – Netz

Für die Überführung von Aktivitätsdiagrammen und Petri – Netzen müssen die sich gegenüberstehenden Elemente identifiziert werden.

Die wesentlichen Elemente des Aktivitätsdiagramms sind Aktivitäten und Transitionen. Da der Übergang von einer Aktivität zu einer weiteren als aktiv bezeichnet werden kann, ist seine Darstellung im Petri – Netz durch eine aktive Komponente möglich. Aktivitäten stellen auch aktive Einheiten dar. Betrachtet man jedoch die „Aktivität in Durchführung“ als Zustand, so

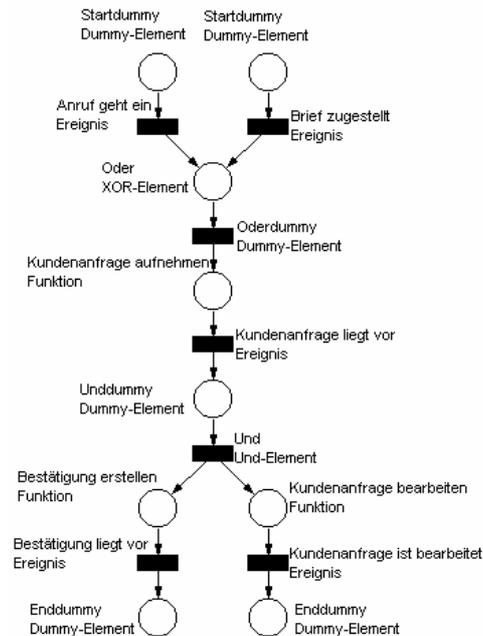


Abbildung 51: Stellen - Transitions - Netz mit Verknüpfung

ist es möglich die Darstellung als passive Komponente im Petri – Netz vorzunehmen.⁸⁷ Die Aktivität wird in der UML als Zustand mit interner Aktion definiert.⁸⁸

Die Entscheidungsraute kann als eigenständige Aktivität aufgefasst werden und ist somit als Aktivität mit mehreren Ein- bzw. Ausgängen darstellbar. Dies entspricht auch der Darstellung von konkurrierenden Nebenläufigkeiten im Petri - Netz.⁸⁹ Da die Entscheidungsraute sowohl Eingangstransitionen, wie auch Ausgangstransitionen besitzt und diese Transitionen durch aktive Stellen im Petri – Netz modelliert werden können, ist die gleichwertige Darstellung im Petri – Netz eine passive Stelle. Die Synchronisation des Aktivitätsdiagramms muss durch eine aktive Komponente im Petri – Netz dargestellt werden. Da jedoch der Synchronisationsbalken nur Eingangs- bzw. Ausgangstransitionen besitzt, müssen zusätzlich im Petri – Netz passive Komponenten zwischengeschaltet werden. Diese treten nur bei der Transformation in das Petri – Netz auf und können bei der Rücktransformation weggelassen werden.

Um eine Rücktransformation zu gewährleisten müssen die einzelnen Elemente des Petri – Netzes den Elementen des Aktivitätsdiagramms eindeutig zugeordnet werden. Dies kann durch die Informationshinterlegung des Elementes erfolgen. Abbildung 52 zeigt den gleichen Sachverhalt wie Abbildung 50 in der Darstellungsweise eines Aktivitätsdiagramms. Das zugehörige Petri – Netz ist in Abbildung 53 abgebildet. Da ein Aktivitätsdiagramm stets nur mit einem Startzustand beginnen kann, müssen in einem ersten Schritt Aktivitäten zur Erzeugung der Startereignisse angelegt werden. Diese stehen direkt nach dem Startzustand und werden je nach Zusammenführung der Ereignisse durch einen Split oder eine Raute aufgeteilt.⁹⁰ Die Aktivitäten ergeben sich dann analog zu den Funktionen, sowie der Fluss zwischen den einzelnen als gleiche Darstellung aus der eEPK. Zusätzlich müssen jedoch die verwendeten Splits wieder synchronisiert werden. Dies erfolgt vor der Beendigung der zugehörigen Aktivitäten und führt erst dann zum Endzustand.

⁸⁷ Vgl. hierzu Transformation eEPK und Petri – Netz, S. 71.

⁸⁸ Vgl. OMG (2001), S. 3-157.

⁸⁹ Vgl. Balzert (2001), S. 358.

⁹⁰ Da in der eEPK beide Startereignisse durch eine XOR - Verknüpfung integriert sind, wird an dieser Stelle eine Entscheidungsraute verwendet.

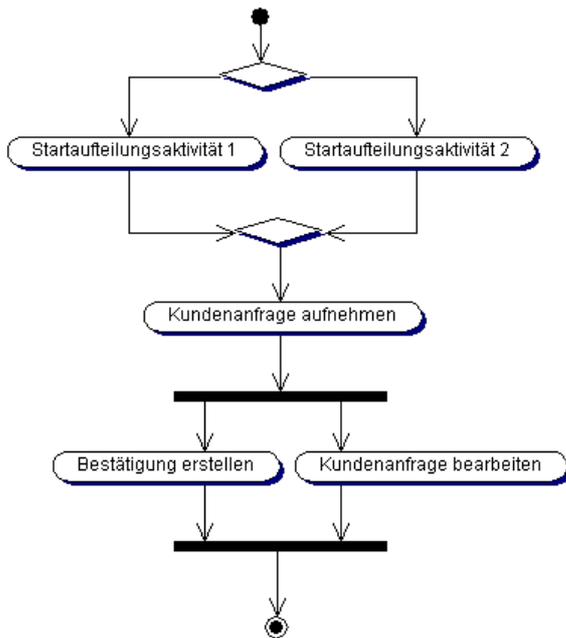


Abbildung 52: Aktivitätsdiagramm mit Konnektoren

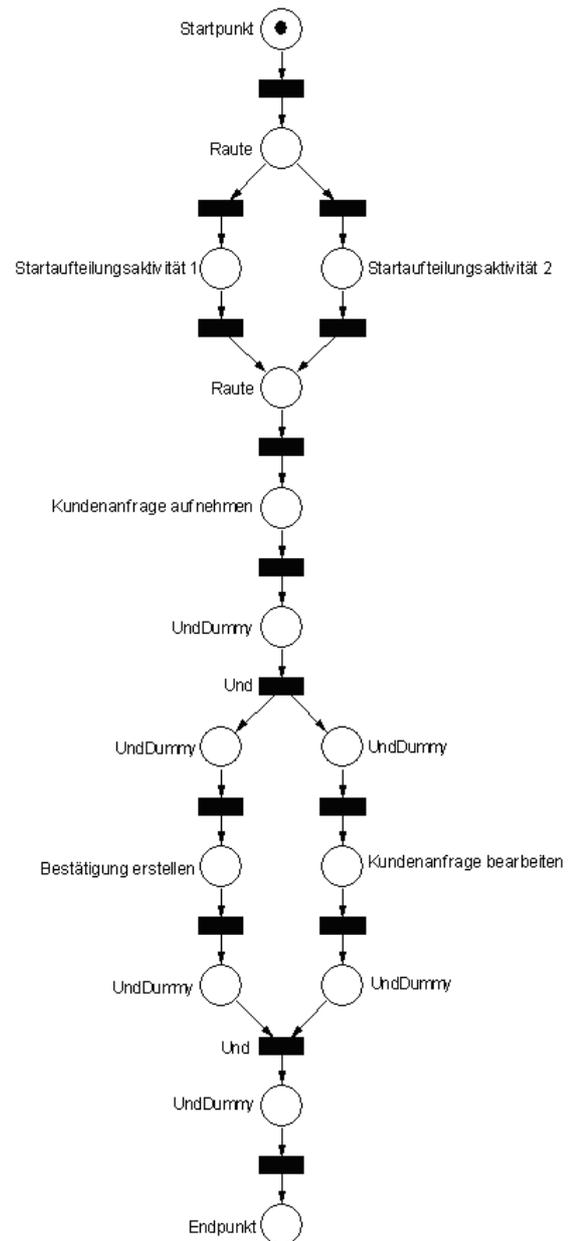


Abbildung 53: Petri Netz für Aktivitätsdiagramm mit Konnektoren

Zur Reduzierung der verwendeten Dummykomponenten des Petri – Netzes kann die oben beschriebene Transformation der Verteilungskomponenten abgewandelt werden. ...

Für die Darstellung von Verantwortlichkeiten für die einzelnen Aktivitäten können im Petri – Netz markierte passive Komponenten verwendet werden, die beidseitige Verbindungen mit den unmittelbar nachfolgenden aktiven Komponenten besitzen. Somit kann die Darstellung der Verantwortlichkeitsbereiche⁹¹ des Aktivitätsdiagramms auch im Petri – Netz dargestellt werden. Zusätzlich können Input- bzw. Outputobjekte für die einzelnen Aktivitäten analog

⁹¹ Im Englischen swim – lane.

transformiert werden. Dabei werden bereits vor dem Beginn des Ablaufs vorliegende Objekte durch eine Stelle mit Markierung ersetzt. Objekte, die im Prozess entstehen, werden durch unmarkierte passive Komponenten dargestellt. Der Informationsfluss verläuft bei Inputobjekten zur nachliegenden aktiven Komponente und bei Outputobjekten von ihr weg. Zu achten ist auf die Rückrichtung zum Objekt, wenn dieses als Input vorliegt, jedoch nicht verändert wurde. Die Darstellung in Abbildung 54 ergibt sich aus der Überführung von Abbildung 49, wobei die Organisationseinheiten durch Verantwortlichkeitsbereiche und Dokumente durch Objekte übersetzt werden. Petri – Netz – Elemente mit der Bezeichnung Funktion wurden als Aktivität und Ereignisse als Transition verstanden. Entscheidungsrauten stellen die XOR – Komponenten und Synchronisationen die UND – Komponenten dar. Für die eindeutige Beschreibung der Elemente reicht damit die Verwendung der ARIS – Begriffe aus.⁹²

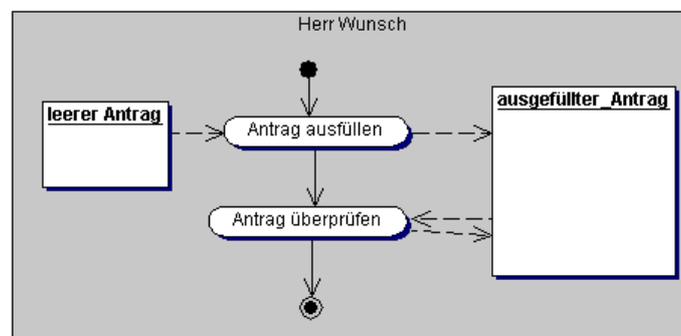


Abbildung 54: Aktivitätsdiagramm mit Objekten und Verantwortlichkeitsbereichen

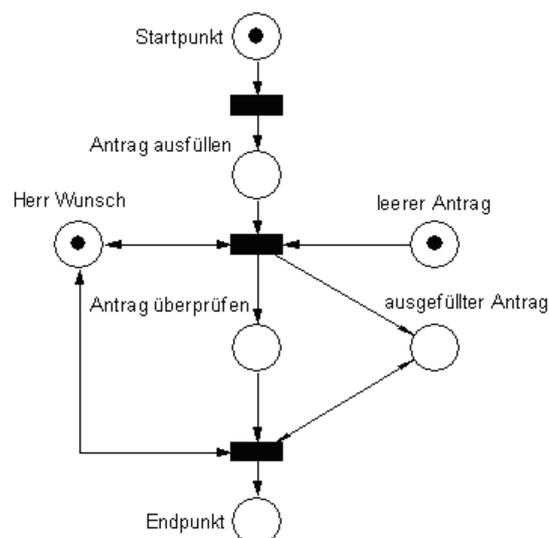


Abbildung 55: Petri - Netz für Aktivitäten mit Objekten und Verantwortlichkeiten

⁹² hierzu: Resultat der Transformation.

Für die Überführung von Aktivitätsdiagrammen und Petri- Netzen zeigt Tabelle 15 die sich gegenüberstehenden Elemente auf.

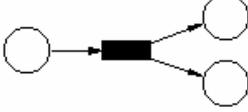
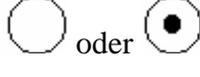
Aktivitätsdiagramm	Petri – Netz
	
	
	
	
	
	
	 oder 
	

Tabelle 15: Transformation Aktivitätsdiagramm und Petri - Netz

Analog zum Kapitel Transformation eEPK und Petri – Netz ist eine Verwendung von anderen Petri – Netzen möglich. Da aber die Überführung der einzelnen Netze in die hier verwendeten Netze möglich ist, wird eine weitere Betrachtung nicht vorgenommen.

7.6.3 Resultat der Transformation

Bei dem Vergleich der entstandenen Petri – Netze fällt die starke Ähnlichkeit auf. Beide beschreiben den gleichen Sachverhalt und können durch einfache Änderungen den jeweiligen anderen Modellausdruck erreichen. Änderungen hierbei sind Erweiterung bzw. Weglassen an den Netzen, dargestellt durch Start- und Endpunkte, Verfeinerung bzw. Abstraktion einzelner Komponenten im Netz. Durch das Hinzufügen von neuen Komponenten entstehen keine zusätzlichen Informationen, sondern die Überführung in das Aktivitätsdiagramm wird da-

durch ermöglicht. Das Ausblenden führt ebenfalls zu keinem Informationsverlust, sondern vereinfacht das dann entstehende Modell der eEPK.

Da es im Aktivitätsdiagramm zu einigen Einschränkungen bezüglich der Verwendung von Startpunkten und Synchronisationen kommt, müssen für die Transformation von eEPK und Aktivitätsdiagramm Hilfsfunktionen implementiert werden, die nur im Aktivitätsdiagramm auftreten. Diese Mittel werden bei der Transformation von der eEPK hinzugefügt und bei der Transformation in die eEPK herausgenommen bzw. ausgeblendet. Für die Überführung einer eEPK mit mehreren Startereignissen ist im Aktivitätsdiagramm ein Hinzufügen von Aktivitäten notwendig, die unmittelbar nach dem Startpunkt liegen. Diese werden vorher durch die entsprechenden Aufteilungsregeln getrennt, so dass eine sinnvolle Zusammenführung nach den Transitionen möglich ist. Die Aufteilungsregeln ergeben sich aus der jeweiligen ereigniszusammenführenden Verknüpfung der eEPK. Diese Erweiterung bezieht sich auf den Anfang des jeweils zugehörigen Petri – Netzes.

Des Weiteren müssen alle nicht wieder zusammengeführten logischen UND im Aktivitätsdiagramm geschlossen werden, bevor der Endzustand erreicht wird. Dies erweitert das zugehörige Petri – Netz am Ende. Start- und Endzustand treten nur im Aktivitätsdiagramm auf und werden nicht in die eEPK übersetzt (Vergleich der Start und Endstellen im Petri-Netz). Transitionen entsprechen den Ereignissen. Im Petri – Netz werden sie durch die aktiven Komponenten dargestellt, die jeweils eine Eingangs- und Ausgangstransition besitzen. Passive Stellen mit einer Eingangs- und Ausgangstransition von der vorgelagerten bzw. nachgelagerten aktiven Komponente werden in der eEPK durch Funktionen und im Aktivitätsdiagramm durch Aktivitäten dargestellt. Eine direkte Übersetzung ist somit an dieser Stelle möglich. Jedoch sollte beachtet werden, dass es sich bei der Aktivität um einen Zustand handelt. Aus diesem Grund sollte sie aus Korrektheitsgründen mit „Funktion in Bearbeitung“ übersetzt werden. Als interne Aktion liegt ihr dann die eigentliche Funktion vor. Dies kann im Aktivitätsdiagramm durch den Eintrag „on entry“ erfolgen. Da die Transformation in dieser Arbeit im Vordergrund steht, soll auf diese zusätzliche Gestaltung verzichtet werden, und sich auf die Bezeichnung der Aktivität durch den Funktionsnamen beschränkt werden, auch wenn die oben beschriebene Vorgehensweise eindeutiger ist. Da jede Aktivität nur einmal verwendet werden kann, muss die Einschränkung auch in der eEPK berücksichtigt werden. Ist es für die Darstellung in der eEPK zweckmäßiger, Funktionen mehrfach zu benutzen, so sollte im Aktivitätsdiagramm für jede Funktion eine eigene Aktivität verwendet werden, die einen anderen Namen erhält (z.B. durch zusätzliche Nummerierung), jedoch bei ihrer internen Aktion die gleiche Funktion wie in der eEPK zugewiesen bekommt. Dadurch wird auch die Rücktrans-

formation möglich. Jede Aktivität, der eine anders lautende innere Aktion zugewiesen ist, wird durch diese Aktion in der eEPK ersetzt. Die Nummerierung ist nur im Aktivitätsdiagramm aufzufinden.

Auch die logischen Konnektoren werden in beiden Petri – Netzen dargestellt, wobei sich hier einige Unterschiede ergeben, die für die Überführung jedoch unproblematisch sind. Aufgrund der unterschiedlichen Verwendungsweise der logischen Und – Verknüpfung im Petri – Netz müssen beide Modelle aneinander angepasst werden. Das kann entweder durch eine Verfeinerung des Petri – Netzes aus der eEPK (Abbildung 51) erfolgen, indem das Und-Element verfeinert dargestellt wird. Es kann aber auch eine Abstraktion des Teilnetzes bezüglich des Und-Elements (wie beispielsweise in Abbildung 53 benutzt) erfolgen. Dabei werden die Dummyelemente ausgeblendet. Die Aktivitäten beziehen sich dann auf den gesamten Sachverhalt des Teilnetzes. Abbildung 56 zeigt die Abstraktion bzw. Verfeinerung einer aktiven aufteilenden Komponente exemplarisch.

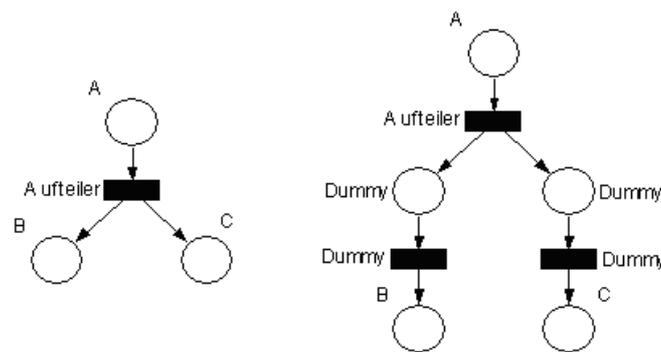


Abbildung 56: Und – Übertragung

Damit kann die Und – Verknüpfung der eEPK dem Split / Synchro des Aktivitätsdiagramms gleichgestellt werden. Die Entscheidungsraute wiederum kann durch die XOR – Verknüpfung interpretiert werden.

Verantwortlichkeitsbereiche werden für Organisationseinheiten verwendet.⁹³ Im Petri – Netz sind sie durch passive Stellen gekennzeichnet, die beidseitig mit den jeweiligen nachgelagerten aktiven Komponenten verknüpft sind. Die gleiche Darstellungsweise wurde im Netz bei der Transformation der eEPK verwendet.

Leistungselemente wie Input und Output können durch Objekte, die durch einen Objektfluss mit der Aktivität verbunden sind, modelliert werden. Tabelle 16 zeigt die sich gegenüberste-

⁹³ Vgl. OMG (2001), S. 3-162.

henden Elemente auf. Diese können jeweils in der entsprechenden Modellierungstechnik ersetzt werden, da ihr Aussagegehalt gleich ist.

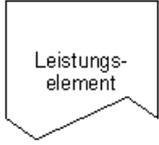
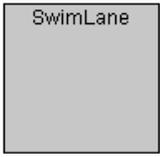
eEPK	Aktivitätsdiagramm
	
	
	
	
	
	
Kein Element	
Kein Element	

Tabelle 16: Transformation eEPK und Aktivitätsdiagramm

7.7 Transformation eEPK und Zustandsdiagramm

Analog zum vorhergehenden Kapitel soll die Transformation beider Modelle durch das Petri – Netz herausgestellt werden. Dabei wird nur die Transformation von Zustandsdiagramm und Petri – Netz durchgeführt, da die Beschreibung der eEPK und des Petri – Netzes bereits im vorhergehenden Abschnitt behandelt wurde. Der Vergleich soll dann eine direkte Transformation zwischen beiden Modellen liefern.

7.7.1 Transformation Zustandsdiagramm und Petri – Netz

Zustände werden als passive Komponenten dargestellt. Zustandsänderungen werden analog zur Vorgehensweise in Kapitel 7.6 durch eine aktive Komponente dargestellt. Damit wird der Transformationsprozess erleichtert. Auch die Verwendung der jeweiligen Komponente lässt sich schlussfolgern. Da der Zustand vorliegt, stellt er keine Aktivität dar, sondern ist als Träger von Informationen zu sehen. Zu beachten ist dabei, dass ein Zustand unterschiedliche Aktionen hervorrufen kann. Diese wiederum führen zu Ereignissen, die den Übergang zu anderen Zuständen auslösen. Der Übergang von Zuständen hingegen ist eine Veränderung, die als aktive Komponente betrachtet wird.

Ein Zustandsautomat besitzt zusätzlich Pseudo-Zustände, wie Start- und Endpunkt, die analog zum eigentlichen Zustand im Petri – Netz durch eine passive Komponente dargestellt werden. Dabei sind sie für die Rücktransformation durch ihre Namensgebung im Petri - Netz gekennzeichnet. Für Simulationsmöglichkeiten erhält der Startpunkt eine Markierung, zusätzlich besitzt er keinen Eingangspfeil. Der Endpunkt hingegen wird als unmarkierte Komponente dargestellt, die gleichzeitig keinen Ausgangspfeil besitzt.

Auch die Aufteilung bzw. Zusammenführung von Parallelitäten stellt einen Pseudozustand dar. Dieser hat jedoch einen aktiven Charakter, da er Marken im Netz vervielfacht bzw. auslöscht. Aus diesem Grund wird er als aktive Komponente dargestellt. Aufgrund der Verknüpfungsmöglichkeit im Petri – Netz müssen passive Komponenten vor- und nachgeschaltet werden. Somit lässt sich die Aufteilungsregel durch das Konstrukt der Und-Nebenläufigkeit und der Und-Zusammenführung im Petri – Netz darstellen. Abbildung 57 stellt den Sachverhalt von Abbildung 50 als Zustandsdiagramm dar. Hierbei sind beide Verknüpfungsarten integriert. Zusätzlich muss die Verwendung eines Aufteilungszustandes vorgenommen werden, da nur ein Startpunkt vorliegen darf. Auch die Aufteilung der Parallelität muss im Zustandsdiagramm wieder zusammengeführt werden. Abbildung 58 ist das gleichwertige Modell in der Notation eines Bedingungs – Ereignis – Netzes.

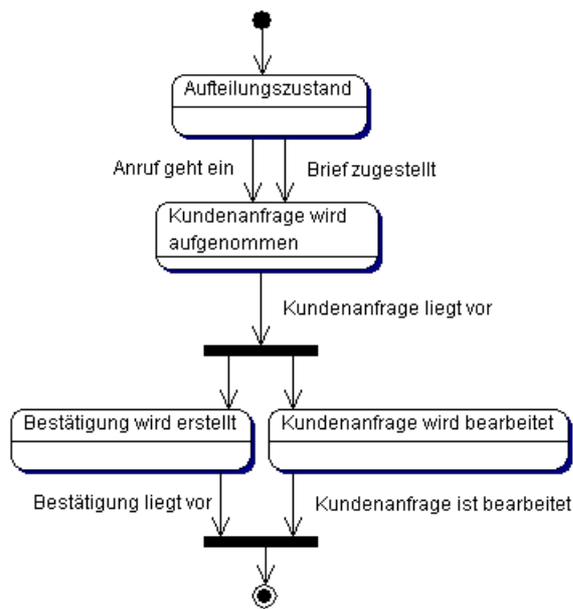


Abbildung 57: Zustandsdiagramm mit Konnektoren

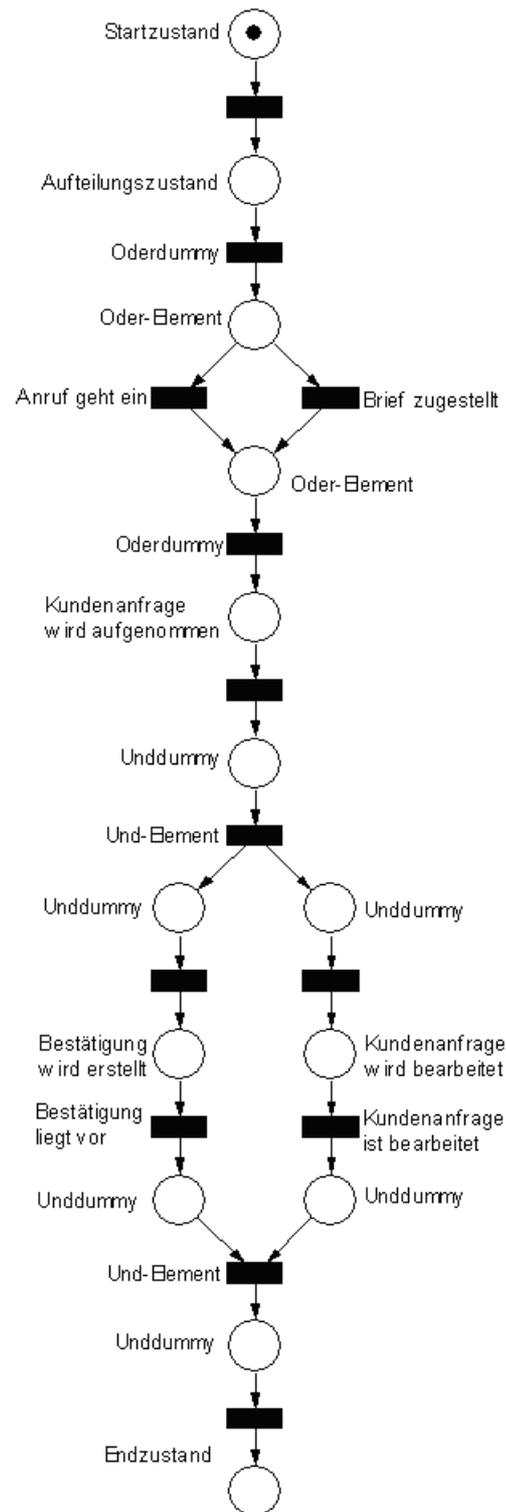


Abbildung 58: Petri - Netz für Zustandsdiagramm mit Konnektoren

Die direkte Verwendung von Objekten ist im Zustandsdiagramm nicht möglich. Um jedoch eine toolabhängige⁹⁴ Lösung für Zustandsdiagramme zu ermöglichen, die Geschäftsprozesse

⁹⁴ Bei Verwendung der Software von TogetherSoft.

abbilden, wird die Einbeziehung von Objekten mit Hilfe von Notizen gestaltet. Für die Zuweisung von Objekten zu einzelnen Zuständen, wird die in Kapitel „Das Zustandsdiagramm“ verwendete Verfahrensweise genutzt. Objekte stellen Informationsträger dar und sollten aus diesem Grund im Petri – Netz als passive Komponenten deklariert sein. Da eine Verknüpfung von gleichen Komponententypen im Petri – Netz nicht möglich ist, müssen die Objektkomponenten und die Zustandskomponenten durch ihr gemeinsames Ergebnis, die Ausgangstransition aus dem Zustand, verbunden sein. Die Notiz stellt nur die Möglichkeit der Verbindung beider Elemente zur Verfügung und wird nicht mit übertragen. Um eine Überführung von Petri – Netzen in ein Zustandsdiagramm zu gewährleisten, müssen die verwendeten passiven Komponenten unterschieden werden. Somit muss neben den Namen auch der Typ (Objekt oder Zustand) und die für das Objekt zugehörige Information über den Stereotyp integriert sein. Dabei ist es ausreichend, entweder die Zustände oder die Objekte mit der Information anzureichern. Wenn alle Objekte über Stereotypen verfügen, ist dies allein ausreichend für die Differenzierung zwischen den Komponenten.

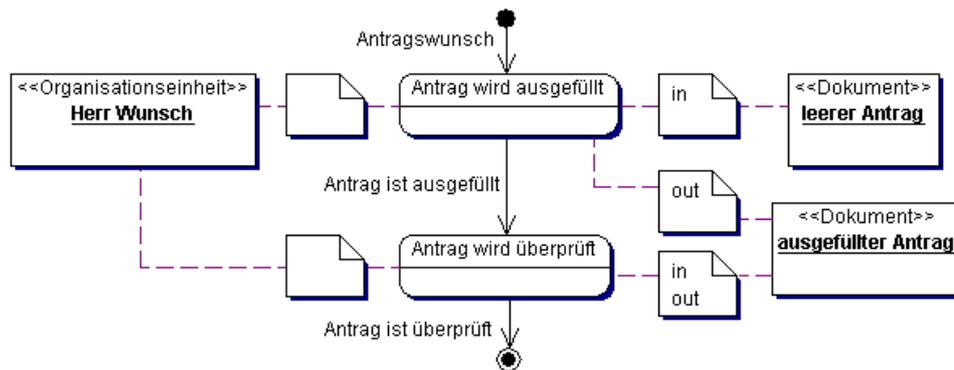


Abbildung 59: Zustandsdiagramm mit Objekten

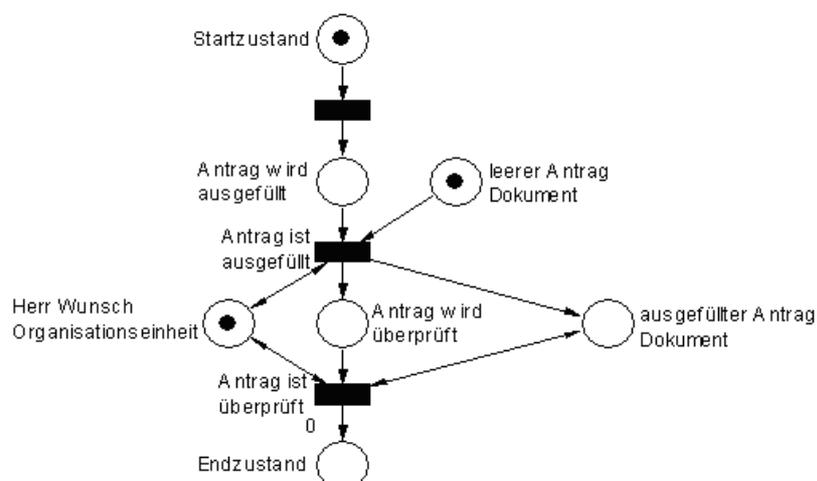


Abbildung 60: Petri - Netz für Zustandsdiagramm mit Objekten

Die einzelnen Elemente, die für die Transformation relevant sind, werden in Tabelle 17 gegenübergestellt.

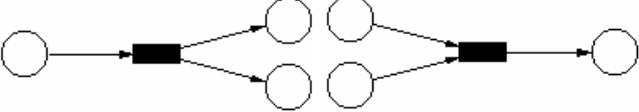
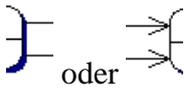
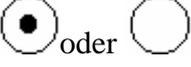
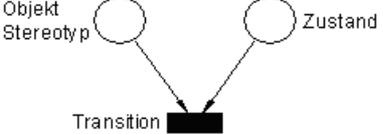
Zustandsdiagramm	Petri – Netz
	
	
	
	
	
	
	
	

Tabelle 17: Transformation Zustandsdiagramm und Petri - Netz

7.7.2 Resultat der Transformation

Ein Vergleich der Petri – Netze ergibt, dass sich die Zustandsübergänge analog zu den Ereignissen abbilden lassen. Die Zustände werden wie Funktionen der eEPK im Petri – Netz modelliert. Dass sich hier eine Analogie ergibt, wird dadurch unterstützt, dass bei Eintreten des Zustandes im UML – Diagramm Prozesse (Methoden) ausgeführt werden können. Daher kann der Zustand gleichgesetzt werden mit der Funktion. Als Vergleichspunkt ergibt sich die Betrachtung der „Funktion in Bearbeitung“ als Zustand.

Auch die Abbildung der logischen Konnektoren ist bei beiden Darstellungsformen integriert. So werden parallele Nebenläufigkeiten durch den Synchronisationsbalken im Zustandsdia-

gramm und der UND – Verknüpfung in der eEPK repräsentiert. Beide sind im Vergleich mit der Abbildung im Petri – Netz identisch.

Die Entscheidung wird hingegen von beiden Darstellungsformen unterschiedlich gehandhabt. Während im Zustandsdiagramm kein eigenes Symbol vorgesehen ist, sondern die Aufteilung unmittelbar am Zustand erfolgt, wird in der eEPK eine der Funktion ausgelagerte Verknüpfung verwendet. Somit ist auch die Darstellung beider Modelle im Petri – Netz unterschiedlich. Eine Umwandlung ist jedoch durch die Verfeinerung der jeweiligen aufteilenden Stellen möglich, die den Zustand von der Entscheidung abtrennen. Abbildung 61 stellt diesen Übergang dar, der reversibel ist, wenn die hinzugefügten Elemente im Petri – Netz ausreichend dokumentiert dargestellt sind.

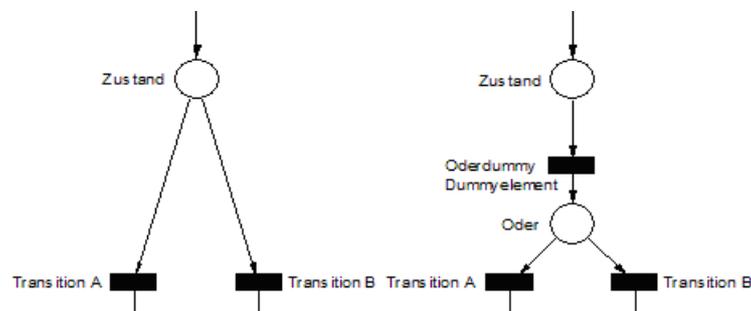


Abbildung 61: Oder - Verfeinerung

Organisationseinheiten und Leistungselemente können durch die Verwendung von Objekten klassifiziert werden. Auch der Vergleich der Petri – Netze zeigt die identische Darstellungskraft beider Methoden. Wichtig für die Reversibilität ist die Verwendung der Symbolbeschreibung in den Objekten. Somit ist eine Nutzung von Stereotypen zwingend notwendig, um zwischen den unterschiedlichen Objekten zu differenzieren.

Tabelle 18 zeigt zusammenfassend alle Elemente in ihrer direkten Transformation.

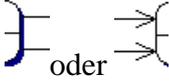
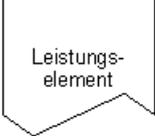
eEPK	Zustandsdiagramm
	
	
	
	
Kein Element	
Kein Element	
	
	

Tabelle 18: Transformation eEPK und Zustandsdiagramm

Im Vergleich mit dem Aktivitätsdiagramm empfiehlt es sich auf das Zustandsdiagramm für die Geschäftsprozessmodellierung zu verzichten. Das hat folgende Gründe: Zum einen ist die Darstellung beider Modelle relativ redundant, da das Aktivitätsdiagramm ein verändertes Zustandsdiagramm darstellt. Zum anderen sind für die Geschäftsprozessmodellierung wichtige Elemente wie Objektfluss und Aufteilungsregeln im Aktivitätsdiagramm besser bzw. überhaupt ausgeprägt und somit ist auch die Transformation leichter und verständlicher. Die Aussagekraft der so erstellten Modelle von Aktivitätsdiagrammen ist leichter zugänglich.

8 Zusammenführung beider Sichten

Nachdem beide Sichten erfolgreich einzeln transformiert werden konnten, ist die sichtenübergreifende Transformation, die vor allem durch das Klassendiagramm der UML bedingt ist, zu gewährleisten. Nachdem das Klassendiagramm bereits aus der Datensicht gewonnen ist, kann es durch die Informationen, die in der eEPK vorhanden sind angereichert werden.

Für die Erarbeitung der Operationen der Klassendiagramme bieten sich zwei Möglichkeiten an. Zum einem können die Elemente durch objektorientierte Modellierer aus den bereits erstellten Diagrammen ermittelt werden. So kann die Beschreibung der statischen Struktur, die durch das Klassendiagramm, das aus der Datensicht gewonnen wurde, mit den Elementen der dynamischen Betrachtungsweise, repräsentiert durch die Zustands- oder Aktivitätsdiagramme, gekoppelt werden. Bei dieser Form der Zusammenführung werden so gleichzeitig die eventuell automatisch gewonnen Modelle überprüft und andererseits eine Zuordnung der jeweiligen Beschreibung der dynamischen Elemente zu den geeignetsten Klassen vorgenommen. Nachteilig hierbei ist der dadurch verbundene Modellierungsaufwand.

Andererseits ist es möglich, die Zuordnung automatisch durchzuführen, wobei die dadurch entstehende Redundanz eventuell nachgebessert werden muss. Funktionen, die in der eEPK durch Organisationseinheiten oder Daten- bzw. Leistungselemente näher beschrieben werden, können zusätzlich in das Klassendiagramm integriert werden. Hierbei wird den beteiligten Klassen, die sich aus der Transformation der Datenelemente (sowohl Organisations- wie auch Leistungselemente können in die Datensicht überführt werden) ergeben, die Funktion als Operation zugewiesen. Damit ein Mehraufwand bei der Programmierung der einzelnen Operationen vermieden wird, sollte durch Notizen in den Klassendiagrammen auf die Redundanz hingewiesen werden. Dies sollte in der Form erfolgen, dass ersichtlich wird, welche Operationen identisch sind. Es könnte aber auch die Zuordnung jeweils nur zu einem beteiligten Element erfolgen, jedoch kann dabei nicht gewährleistet werden, dass diese Zuordnung stets sinnvoll ist. Durch die mehrfache Zuordnung kann sichergestellt werden, dass die Operation in einem sinnvollen Kontext zugeordnet ist. Der Programmierer kann dann die logisch zugehörige Klasse auswählen und die Operation aus den restlichen Klassen entfernen. **Abbildung 62** stellt die Information für Operationen bereit, die in **Abbildung 63** implementiert sind.

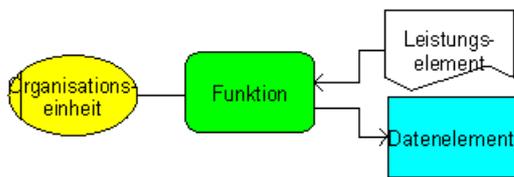


Abbildung 62: Funktion mit Datenelementen

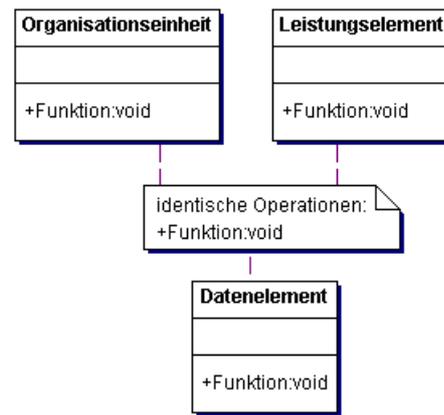


Abbildung 63: Klassendiagramm mit Operationen

Durch diese Zusammenführung werden keine neuen Informationen der prozessorientierten Modelle geschaffen. Dies wird auch dadurch deutlich, dass eine Gewinnung unabhängig von den Modellen der ARIS – Methode möglich sind. Dadurch ist eine Rücktransformation der Informationen von objektorientierter Modellierungsweise nicht notwendig. Es handelt sich somit um einen Schritt, der nur beim Forward – Engineering auftritt. Er kann jedoch genutzt werden zur Überprüfung der durch das Reverse – Engineering erhaltenen Modelle, denn die Funktionen müssen in der ePK zumindest den Elementen zugeordnet sein. Somit stellt die oben beschriebene Zusammenführung ebenfalls ein Mittel des Reverse – Engineering dar, bei dem die Güte der Modelle validiert wird.

9 Zusammenfassung und Ausblick

Die Geschäftsprozessmodellierung führt neben Anleitungsbeschreibungen für Mitarbeiter häufig auch zu Projekten, die eine Einbeziehung von Informationstechnologien in den Arbeitsablauf schaffen. Viele dieser Projekte erfordern eine starke Kommunikation zwischen den Beteiligten (Ersteller und Anwender des Systems). Dabei unterscheiden sich jedoch häufig die Ausdrucksmöglichkeiten. Dies kann zum Scheitern der potenziell erfolgreichen Projekte führen oder die Projektkosten stark erhöhen.

Mit dieser Arbeit soll eine möglichst einfache und eventuell automatische Überführung von prozessorientierter und objektorientierter Sprache erfolgen. Dabei soll vor allem das Business Process Reengineering betrachtet werden, das eine Überführung in beide Richtungen erzwingt. Da es neben diesem Ansatz noch weitere Entwicklungen gibt, sollen diese hier zuerst kurz skizziert werden.

9.1 Die UML 2.0

Die Informationen für die Entwicklung der UML 2.0 beziehen sich vorwiegend auf das DRAFT – Paper. Ob eine Einführung in dieser Art und Weise erfolgt, kann nicht mit Sicherheit garantiert werden, ist jedoch für die entscheidenden Elemente sehr wahrscheinlich.

Mit der Einführung der nächsten UML – Version werden für die Geschäftsprozessmodellierung wesentliche Elemente neu gestaltet. So erhält das Aktivitätsdiagramm ein vom Zustandsdiagramm losgelöstes Metamodell, das die Eigenschaften eines Petri – Netzes erhält. Somit ist eine Überführung in die jeweilig andere Modellierungsart einfacher möglich. Auch die Einschränkung der Wiederverwendbarkeit von Aktivitäten wird aufgehoben. Dadurch lassen sich Wiederholungsaktivitäten schneller ermitteln und Geschäftsprozesse einfacher abbilden. Das Aktivitätsdiagramm übernimmt in der neuen UML Version somit eine wichtige Rolle für die Gestaltung von Geschäftsabläufen. Inwieweit sich hieraus dann eine Verknüpfung zur Implementierungssicht ergibt, kann nur durch die Praxis aufgezeigt werden. Im Wesentlichen werden die Veränderungen jedoch nicht durch den Nutzer ersichtlich, sondern die Änderungen beziehen sich auf das MOF Metamodell.

9.2 Die ARIS - Weiterentwicklungen

Auch die IDS Scheer versucht durch ständige Verbesserungen die Architektur wettbewerbsfähig zu halten. Ob sich die Neuerungen durchsetzen, wird dann durch die Anwender bestimmt. Konzepte zur Integration der neuen Gegebenheiten stellen zum Beispiel die objektorientierte EPK (oEPK⁹⁵) dar. Hierbei wird die eEPK durch Elemente objektorientierter Modelle erweitert.⁹⁶ Diese können mit Funktionen und Ereignissen verknüpft werden. Somit lässt sich der Übergang bereits in ARIS durchführen. In der Praxis hat sich das Konzept jedoch nicht durchsetzen können. Dies liegt vor allem an den damit verbundenen Anforderungsprofilen der Modellierer, da sie mit dieser Methode beide Modellierungsarten beherrschen können. Zwar können Elemente durch Filter ausgeblendet werden, jedoch bleiben sie im Hintergrund vorhanden, was eine Modellierung beeinträchtigt.

Ein weiteres Integrationskonzept stellt der UML Designer⁹⁷ dar, der mit dem ARIS Toolset verknüpft ist. Hierbei können die objektorientierten Modellelemente mit den anderen ARIS Elementen in der Datenbank verknüpft werden. Somit lassen sich Änderungen in einem Modell dann auf die anderen automatisch übertragen. Auch hierbei wird die Strategie verfolgt, dass die Modellierung mit beiden Methoden erfolgt.

9.3 Andere Ansätze

Für die Verbindung von objektorientiertem und relationalem Ansatz muss nicht unbedingt eine Transformation erfolgen, sondern kann auch eine Integration beider Ansätze möglich sein. Hierzu bieten viele Datenbankhersteller und Drittanbieter Mapping Tools an, die Klassen auf die Tabellen eines relationalen Datenbankmanagementsystems abbilden.⁹⁸

Ein weiterer Ansatz ist die Überführung der EPK in ein Petri – Netz, um aus diesem dann Software zu entwickeln. Dabei handelt es sich um Workflowsysteme, die dabei verwendeten

⁹⁵ Hierzu Nüttgens (1997), S. 10.

⁹⁶ Hierzu Loos (1998), S. 9 ff.

⁹⁷ Der UML Designer ist Bestandteil des ARIS Paketes ab Version 6.2.

⁹⁸ Vgl. Schäfer (2003).

Petri – Netze werden als Workflow – Netze⁹⁹ bezeichnet. Diese haben einen starken Bezug zum Aktivitätsdiagramm, da sie zum einen nur einen Startpunkt aufweisen dürfen und durch die Zusicherung eines Endpunktes wird gleichzeitig sichergestellt, dass alle Nebenläufigkeiten wieder geschlossen werden. Der Freiheitsgrad ist jedoch beim Aktivitätsdiagramm größer, da nicht zwingenderweise ein Endpunkt vorgesehen ist. Die Transformation bezieht sich hierbei auf die schlanke EPK und wird durch Zusammenfassung von aufeinanderfolgenden gleichen Elementen erreicht¹⁰⁰, wobei Funktionen aktive Komponenten einnehmen und Ereignisse durch passive Komponenten dargestellt werden.¹⁰¹ Durch die Zusammenfassung von Elementen werden die Workflow – Netze lässt sich jedoch die Rücküberführung vom Workflow – Netz zur EPK nicht durchführen und stellt somit nur einen Prozess des Forward Engineerings dar. Für die Überführung wird weiterhin eine sukzessive Vorgehensweise vorgeschlagen, wobei jedoch jeweils zum Modell der EPK zurückgegangen wird.¹⁰²

Für die direkte Umsetzung von Arbeitsabläufen, die sich durch ein Workflowsystem unterstützen lassen, bietet sich die Schnittstelle von ARIS und Staffware (Process @ Work) an. Auch hierbei handelt es sich zum einen um eine beschränkte Anzahl von Geschäftsabläufen, die somit überführt werden können. Zum anderen kann nur eine Entwicklung der Software auf Basis des Forward Engineerings durchgeführt werden, da es keine Überführung von den Staffwareprozessen hin zur ARIS Darstellung gibt. Desweiteren unterliegen die Modelle in ARIS, die transformiert werden sollen einer hohen Modellierungsrestriktion, so dass die Freiheit des Modellierers stark eingeschränkt wird.¹⁰³

9.4 Zusammenfassung

Die Transformation zwischen prozess- und objektorientierter Methode lässt sich bei geeigneter Wahl der Modelle erreichen. Dabei kann eine Rücküberführung jedoch nur erfolgreich sein, wenn eine ausreichende Dokumentation durchgeführt wird.

⁹⁹ Workflow – Netze wurden von W.M.P. van der Aalst eingeführt. Siehe hierzu Aalst (1998).

¹⁰⁰ Vgl. Dehnert (2001a), S. 5.

¹⁰¹ Anhang B zeigt die Analogie zwischen den Überführungen.

¹⁰² Vgl. Dehnert (2001), S. 2.

¹⁰³ Zum ausführlichen Vergleich siehe Arnold (2000).

Die Transformation auf rein statischer Ebene kann zwischen dem eERM und dem Klassendiagramm erreicht werden. Diese beiden Modelle haben einen hohen Grad an Gemeinsamkeit, der die Überführung recht unkompliziert macht. Durch Umwandlung von Organisations- und Leistungsmodellen, die als Zwischenschritt im eERM dargestellt werden, ist die Überführung ins Klassendiagramm wiederum möglich. Aber auch die direkte Transformation kann erfolgen. Für das Klassendiagramm ist die Verwendung von Stereotypen zwingend erforderlich, da nur so eine gesicherte Zuordnung erfolgen kann. Die jeweiligen Stereotypen müssen wiederum ausreichend dokumentiert sein.

Auch eine Überführung von dynamischen Aspekten auf beiden Methoden ist möglich. So stehen sich der Funktionsbaum und die eEPK auf der einen Seite und das Aktivitäts- und das Zustandsdiagramm auf der anderen Seite. Für die Modellierung der eEPK bestehen im Wesentlichen keine Restriktionen, da durch einen Zwischenschritt der kritische ODER Konnektor aufgelöst werden kann. Zur Überführung bieten sich Petri – Netze an, die die beiden semi-formalen sich gegenüberstehenden Modelle abbilden. Für die Angleichung der beiden Methoden muss das Petri – Netz verfeinert werden, wobei nicht alle Verfeinerungen dann wieder mit übernommen werden. So kann sichergestellt werden, dass das Petri – Netz dann sowohl alle Elemente der eEPK, wie auch der UML – Diagramme abbilden und steht somit als formaler Vermittler zwischen beiden Methoden. Die direkte Überführung kann dann erfolgen, wenn die Abstraktionen und Verfeinerungen des Petri – Netzes angewendet werden. Durch die höheren Freiheitsgrade in der ARIS – Modellierung muss aber bei diesem Schritt besondere Aufmerksamkeit auf die Konnektoren vor allem auf den Split und Join und die Startpunkte gelegt werden. Für die Überführung von der eEPK eignet sich das Aktivitätsdiagramm besser als das Zustandsdiagramm.

Für die Transformation von Funktionsbäumen eignen sich sowohl Zustands- wie auch Aktivitätsdiagramme. Diese Überführung kann unmittelbar erfolgen. Somit können spezielle Aktivitätsdiagramme auch direkt in Funktionsbäume überführt werden. Wichtig für die Wahl des Überführungsmittels ist die Dokumentation der Modelle.

Auch die Überführung von Anwendungsfalldiagrammen kann durch das Modell der Prozessauswahlmatrix erfolgen. Da Anwendungsfalldiagramme vorwiegend zur ersten Analyse von Systemen genutzt werden, ist es fraglich bzw. redundant, diese Modelle zu transformieren. Sie sollten nur unterstützend eingesetzt werden. Anhang A zeigt zusätzlich die Herausarbeitung von Anwendungsfalldiagrammen aus den eEPKn. Dieser Schritt kann dann notwendig werden, wenn Anwendungsfalldiagramme durch die Entwickler gewünscht werden.

Da es in der UML zusätzlich die Verbindung von statischer und dynamischer Sichtweise gibt, und auch die eEPK diese Verbindungen aufzeigt, lässt sich in einem abschließenden Schritt die Überführung dieser Informationen bewerkstelligen. Dabei handelt es sich stets um den letzten Schritt, da die dann bereits vorhandenen Modelle durch die Informationen der Verbindung beider Sichten angereichert werden.

Im Vordergrund dieser Arbeit steht die Überführung der einzelnen relevanten Geschäftsprozessmodelle. Hierbei wird keine Überprüfung der Korrektheit der Modelle vorgenommen. Durch die Transformation kann aber beispielsweise die eEPK durch ein Petri – Netz verifiziert und validiert werden. Da jedes Element der eEPK einen Repräsentanten im Petri – Netz hat, kann das validierte und ggf. verbesserte Modell wiederum überführt werden, so dass ein korrektes Modell der eEPK – Notation vorliegt. Dies ist jedoch nicht Bestandteil dieser Arbeit. Zusätzlich gibt es für die Überprüfung der Modelle eine Vielzahl von Tools. Aus diesem Grund wird ein korrektes Ausgangsmodell angenommen, das in die jeweilige andere Modellwelt dann überführt wird.

Resultierend lässt sich sagen, dass eine Überführung von betriebswirtschaftlicher zu informationstechnologischer Sichtweise dargestellt durch die prozess- und objektorientierte Modellmethode möglich ist und bei sorgfältiger Dokumentation eine Rücküberführung ebenfalls erreicht werden kann.

10 Anhang

10.1 Anwendungsfalldiagramm und eEPK

Die Informationen für ein Anwendungsfalldiagramm können auch aus bereits existierenden eEPKs gewonnen werden. Da es sich dabei um einen Schritt des Forward Engineering handelt, der nicht reversibel ist, wird die Überführung in diesem Abschnitt erläutert.

Ob eine eEPK für die Abbildung in einem Anwendungsfalldiagramm geeignet ist, hängt von der Granularität der eEPK ab. Ist diese zu detailliert, so ist die Überführung in ein Aktivitätsdiagramm sinnvoller. Wenn der Abstraktionsgrad hingegen sehr groß ist, kann eine Abbildung in ein Anwendungsfalldiagramm für die objektorientierte Sicht hilfreich sein.

Funktionen der eEPK beschreiben in diesem Zusammenhang aggregierte Anwendungen und können den Anwendungsfällen zugeordnet werden. Die Reihenfolge, die für den Prozessablauf von Bedeutung ist, spielt im Anwendungsfalldiagramm keine Rolle. Hinterlegungen von Funktionen bzw. erläuternde Nebenläufigkeiten können durch include oder extend – Beziehungen dargestellt werden. Um eine include – Beziehung handelt es sich dann, wenn Teilfunktionen mehrfach auftreten und diese im beschriebenen Prozess eine untergeordnete Rolle spielen. Die extend – Beziehung wird für Spezialfälle angewendet, die zusätzliche Abläufe beschreiben. Häufig werden Nebenläufigkeiten diese Eigenschaft aufweisen, wenn ihnen gleiche Funktionselemente inne sind. So zeigen parallele Abläufe einzelne Teilschritte, die im Anwendungsfalldiagramm durch die include – Beziehung dargestellt werden. Die exklusive ODER – Aufteilung hingegen zeigt die unterschiedlichen Spezialfälle auf. Diese werden durch die extend – Beziehung im UML Modell beschrieben. Da jedoch die eEPK meist nur eine Granularität aufweist, kann es notwendig sein, die zugehörigen Anwendungen durch den Modellierer zu finden. Eine Automatisierung ist aus diesem Grund nicht möglich.

Die Generalisierung kann abgeleitet werden für Funktionen, die mit einem Funktionsbaum hinterlegt sind. Da aber auch abhängig von der jeweiligen Beschreibung include – oder extend – Beziehungen für die Überführung verwendet werden können, ist eine automatische Wahl des Mittels nicht möglich.

Die Beziehungen zu den Akteuren lassen sich durch die Organisationszuordnung ermitteln, wobei externe Personen ebenfalls in das Anwendungsfalldiagramm überführt werden, ohne

eine Differenzierung vorzunehmen. Beteiligte Systeme können ebenso in das UML – Modell überführt werden, da diese als Akteure identifiziert werden, die gegebenenfalls stereotypisiert sind.

Die Systemgrenze wird nur eine eindeutige Beschreibung des Modells ableitbar. So kann sich die Systemgrenze aus dem Namen des Modells der ARIS Notation ergeben, was nicht zwingend notwendig ist.

Die aufgezeigten Probleme verdeutlichen, dass eine Überführung der eEPK in das Anwendungsfalldiagramm nur dann sinnvoll ist, wenn eine Beschreibung auf einem hohen Abstraktionsniveau erfolgt. Diese Überführung kann nicht automatisiert werden, sondern ist abhängig vom jeweiligen Modell. Eine Nacharbeit ist zusätzlich notwendig. Die Überführung ist aus diesem Grund nur schlecht geeignet, wogegen sich die Prozessauswahlmatrix besser einsetzen lässt.

Tabelle 19 stellt den Elementen der eEPK die zugehörigen Elemente des Anwendungsfalldiagramms gegenüber, wobei eine Überprüfung durch den Modellierer auf Konsistenz notwendig ist.

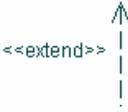
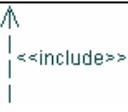
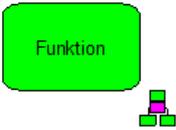
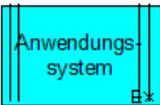
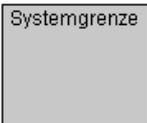
eEPK	Anwendungsfalldiagramm
	
	
	
	
	
	
das Modell	

Tabelle 19: Überführung eEPK und Anwendungsfalldiagramm

10.2 Petri – Netz – Transformation

In der Literatur werden für gleiche Sachverhalte unterschiedliche Verfahren genutzt. Das die jeweiligen Modelle in einander abbildbar sind, soll der folgende Abschnitt aufzeigen. Dabei wird die Darstellung von unterschiedlichen Petri-Netzen für den gleichen Darstellungswert verwendet. Die Modelle spiegeln somit die gleiche Realität wider.

Bei der Umwandlung von den unterschiedlichen Petri – Netzen ist auf die Verwendung von aktiven und passiven Elementen zu achten. Je nach Betrachtungsgegenstand können diese Komponenten gegeneinander ausgetauscht werden. Wichtig ist dabei jedoch die Durchgängigkeit des Austausches, das heißt alle Komponenten müssen dann entsprechend verändert werden. Die Transformation kann somit nicht lokal erfolgen, sondern muss im gesamten Modell angewendet werden. Schwieriger ist die Transformation von aufteilenden und zusammenführenden Elementen. Ein alleiniger Austausch würde den Informationsgehalt verändern, da aus parallelen Nebenläufigkeiten konkurrierende Nebenläufigkeiten werden. Somit darf die Aufteilungs- bzw. Zusammenführungskomponente nicht ersetzt werden. Allerdings ist das Resultat dann kein Petri – Netz mehr, da an diesen Stellen die Ablauffolge von aktiver und passiver Komponente nicht mehr eingehalten wird. Es lässt sich jedoch durch Hinzufügen oder Weglassen von unmittelbar vorhergehenden bzw. folgenden Komponenten der Fluss wiederherstellen. Das Ergebnis ist dann wieder ein Petri – Netz.

Da es sich bei der Überführung der vorhandenen Netze um einen reversiblen Prozess handeln soll, dürfen keine Informationen verloren gehen. Zusätzlich dürfen keine weiteren Informationen hinzugefügt werden, da diese sonst im Ausgangsmodell nicht vorhanden sind. Somit handelt es sich bei der Herauslassung und beim Hinzufügen von neuen Komponenten stets um Elemente, die keine Informationen tragen. Im Transformationsteil wurden solche Elemente bereits genutzt. Tritt an einer aufzweigenden bzw. zusammenführenden Komponente ein Dummyvorgänger bzw. –nachfolger auf, so wird dieser bei der Transformation entfernt. Tragen die Vorgänger- oder Nachfolgerkomponente Informationen, so wird die Überführung durch das Zwischenschalten von entsprechenden Dummykomponenten ermöglicht. Tabelle 20 zeigt die Darstellungsweisen der Verknüpfungen und Zusammenführungen, wobei die jeweils unmittelbaren Komponenten dargestellt sind.

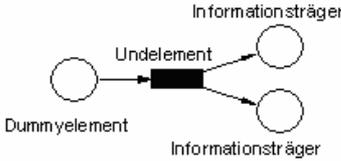
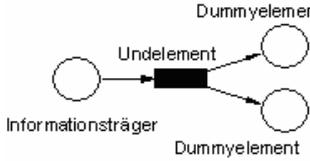
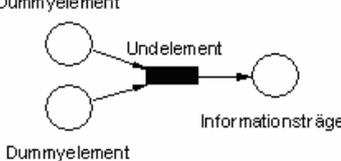
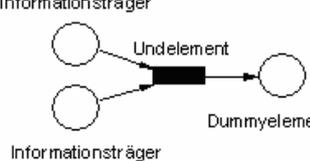
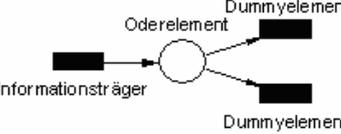
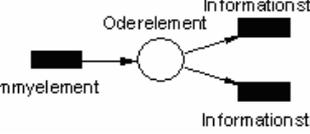
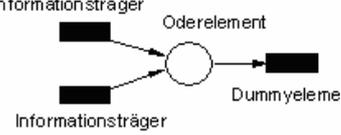
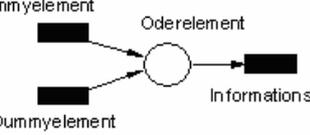
Funktion als passive Komponente	Funktion als aktive Komponente
	
	
	
	

Tabelle 20: Aufteilungs- und Zusammenführungsumwandlungen

Die Abbildung der Start- und Endpunkte eines Modells ist für die Simulation der Petri – Netze von Bedeutung. Da aktive Komponenten an diesen Stellen stets neue Marken erzeugen, bzw. entfernen, kann es zweckmäßiger sein, in den Modellen mit passiven Komponenten zu beginnen und zu enden. Handelt es sich bei den Komponenten um Informationsträger, so müssen Ihnen Dummykomponenten vor- bzw. nachgeschaltet werden. Tragen die umzuwandelnden Komponenten keine Informationen, so können diese weggelassen werden. Abbildung 64 zeigt den gleichen Sachverhalt wie Abbildung 51. Auch eine Rücktransformation ist durch den oben beschriebenen Ablauf möglich.

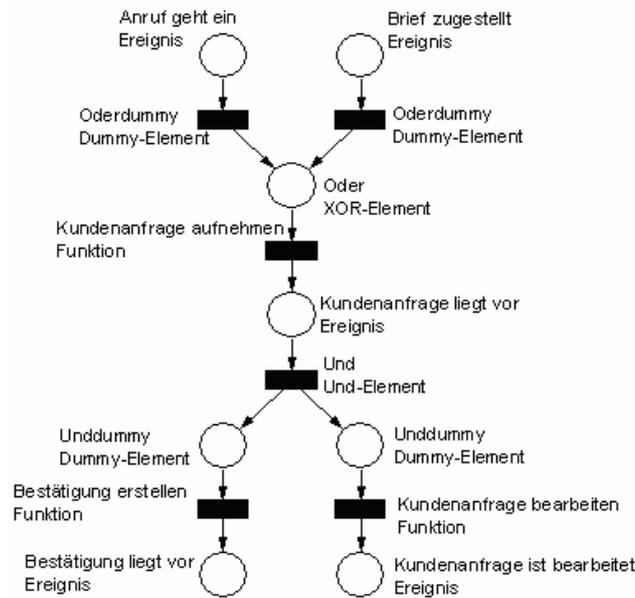


Abbildung 64: Petri - Netz der EPK mit Funktionen als aktiver Komponente

In der eEPK werden zusätzlich Informationen über Organisationseinheiten und Leistungselemente abgebildet. Auch diese Elemente sind in beiden Darstellungsformen des Petri – Netzes abbildbar. Sie können in beiden Netzdarstellungen als passive Komponente dargestellt werden. Hierzu muss jedoch der Fluss zwischen den anderen Komponentenabgewandelt werden. In der Darstellung der beschriebenen Transformation sind die Stellen, die Elemente der Organisations- oder Leistungseben abbilden mit den resultierenden Ereignissen verknüpft. Für die Umwandlung wird nun dieser Informationsfluss um ein Element vorverlegt, so dass eine Verbindung von Funktion und Organisations- bzw. Leistungseinheit vorhanden ist. Dies entspricht dann auch der Abbildung in der eEPK. Abbildung 65 bildet den Sachverhalt von Abbildung 49 ab. Der Zusammenhang zur eEPK (Abbildung 47) ist sofort ersichtlich.

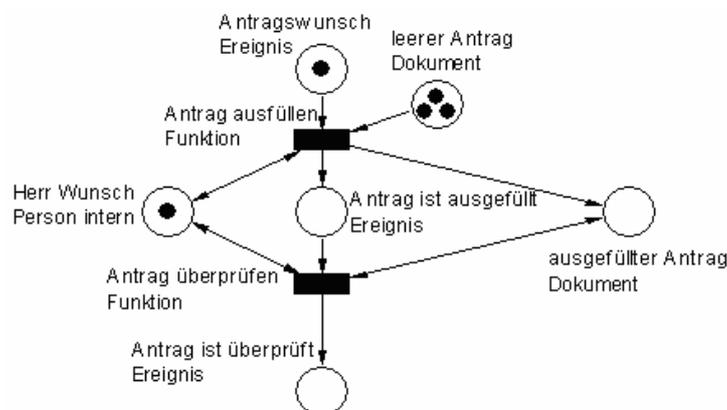


Abbildung 65: Petri - Netz der eEPK mit Funktionen als aktiver Komponente

Geschäftsprozesse können somit in Petri – Netzen abgebildet werden, unabhängig von der Definition des Zustandes und der Aktivität. Eine Überführungsvorschrift liefert die Tabelle 21.

Funktion als passive Komponente	Funktion als aktive Komponente
Funktion 	Funktion 
Ereignis 	Ereignis 
Organisations- element 	Organisations- element 
Leistungs- element 	Leistungs- element 
Daten- element 	Daten- element 

Tabelle 21: Überführung der Petri - Netze

Da im Petri – Netz der Kontrollfluss stets zwischen passiver und aktiver Komponente erfolgt, muss zusätzlich dieser bei der Überführung angepasst werden. So werden die Organisations-, Leistungs- und Datenelemente in der Darstellung mit Funktionen als passiver Komponente an das folgende Ereignis gesetzt. Im anderen Fall werden diese direkt mit der Funktion verbunden.

11 Literaturverzeichnis

Aaslt (1998) = W.M.P. van der Aalst: The Application of Petri Nets to Workflow Management, The Journal of Circuits, Systems and Computers 8(1), S. 21 – 66, 1998.

Aalst (1999) = W.M.P. van der Aalst: Formalisation and Verification of Event-driven Process Chains, Information and Software Technology 41(10), S. 639-650, 1999.

Arnold (2000) =

Balzert (2001) = Helmut Balzert: Lehrbuch der Softwaretechnik Band 1 Software - Entwicklung, 2. Auflage, Spektrum Akademischer Verlag, Heidelberg, Berlin, 2001.

Baumgarten (1990) = B. Baumgarten: Petri-Netze: Grundlagen und Anwendungen, BI-Wiss.-Verlag, Mannheim et.al. 1990.

Brücher (2001) = Heide Brücher, Rainer Endl: Erweiterung von UML zur geschäftsregelorientierten Prozessmodellierung, 2001.

Bullinger (2001) = Hans-Jörg Bullinger, Peter Schreiner (Hrsg.): Business Process Management Tools - Eine evaluierende Marktstudie über aktuelle Werkzeuge, Franhofergesellschaft IAO, Stuttgart, 2001.

Bungert (1995) = Winfried Bungert, Helge Heß: Objektorientierte Geschäftsprozessmodellierung, in Information Management I / 1995, S. 52-63.

Chen (1976) = P. P. Chen: The Entity-Relationship model: Toward a Unified View of Data, in: ACM Transactions on Database-Systems, Vol. 1 (1976), No. 1, S. 9 - 36.

Chen (1994) = R. Chen, A.-W. Scheer: Modellierung von Prozessketten mittels Petri-Netz-Theorie, Heft 107 Institut für Wirtschaftsinformatik Universität des Saarlandes, Saarbrücken 1994.

- Dandl (1999) = Jörg Dandl: Objektorientierte Prozessmodellierung mit der UML und EPK, Arbeitspapiere WI Nr. 12/1999 der Universität Mainz, 1999.
- Dehnert (2000) = Juliane Dehnert, Wijnand Derks: Ein pragmatischer Korrektheitsbegriff für die Modellierung von Geschäftsprozessen mit Petrinetzen, in: Holger Giese, Stephan Philippi (Hrsg.): Visuelle Verhaltensmodellierung verteilter und nebenläufiger Softwaresysteme, 8. Workshop des Arbeitskreises GROOM, Universität Münster, 2001.
- Dehnert (2001) = Juliane Dehnert: Four Systematic Steps towards Sound Business Process Models in: H. Weber, H. Ehrig, W. Reisig: Int. Colloquium on Petri Nets Technologies for Modelling Communication Based Systems: S. 55-64, Springer-Verlag, 2001.
- Dehnert (2001a) = Juliane Dehnert, Peter Rittgen: Relaxed Soundness of Business Processes in: K.L. Dittrich, A. Geppert, M.C. Norrie: Advanced Information System Engineering, CAISE 2001, vol. 2068 of LNCS, S. 157-170, Springer-Verlag, 2001.
- Fowler (2000) = Martin Fowler, Kendall Scott: UML konzentriert – Eine strukturierte Einführung in die Standard-Objektmodellierungssprache, 2. Auflage, Addison-Wesley, München 2000.
- Gartner (2003) = www.gartner.com/gc/webletter/idsscheer/issue1/article1.html, Stand: 15.06.2003.
- Göhring (2002) = Daniel Göhring, Frank Legler, Stephan Weißleder, Anatolij Zubow: Automatische Transformation von ARIS-Modellen nach UML, Seminararbeit, Humboldt – Universität Berlin, 2002.
- Harel (1987) = David Harel: Statecharts: A Visual Formalism for Complex Systems, in Science of Computer Programming, Elsevier, S. 231-274, 1987.

- Hausmann (2000) = Jan Hendrik Hausmann, Reiko Heckel, Stefan Sauer: Ein Konzept zur anwendungsbezogenen UML-Semantikbeschreibung durch dynamische Metamodellierung, Universität Paderborn, 2000.
- Holl (2000) = Alfred Holl, Thomas Krach, Roman Mnich: Geschäftsprozessdekomposition und Gestalttheorie, Georg-Simon-Ohm-Fachhochschule Nürnberg, 2000.
- IDS (2000) = Methodenhandbuch im ARIS-Toolset 5.0 E-Business Suite, IDS Scheer AG, Saarbrücken, 2000.
- IDS (2001) = Methodenhandbuch im ARIS-Toolset 6.0 Collaborative Suite, IDS Scheer AG, Saarbrücken, 2001.
- Jacobson (1994) = Ivar Jacobson, Maria Ericson und Agneta Jacobson: The Object Advantage Business Process Reengineering with Object Technology, 1. Auflage, Addison Wesley Wokingham, England et al. 1994.
- Köppen (2002) = Veit Köppen, Sebastian Rix: Vergleich statischer und dynamischer Geschäftsprozessmodellierung am Beispiel von ARIS und UML, Seminararbeit, Humboldt Universität zu Berlin, 2002.
- Loos (1998) = Peter Loos, Thomas Allweyer: Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains, in IWI Heft 144, Saarbrücken, 1998.
- Maurer (1996) = Gerd Maurer: Von der Prozeßorientierung zum Workflow Management, Arbeitspapiere WI Nr. 10/1996 der Universität Mainz, 1996.
- Mehner (2000) = Katharina Mehner, Tim Schattkowsky: Methodische Aspekte von Modellierungssprachen für nebenläufige Software, Universität Paderborn, 2000.
- Moldt (2000) = Daniel Moldt, Jörg Rodenhagen: Ereignisgesteuerte Prozeßketten und Petri-netze zur Modellierung von Workflows, in: Holger Giese und Stephan Philippi: Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme,

8. Workshop des Arbeitskreises GROOM der GI Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung, Seiten 57-63, Universität Münster, 2000.

Nüttgens (1997) = Markus Nüttgens, Thomas Feld, Volker Zimmermann: Business Process Modeling with EPC and UML: Transformation or Integration?, in: M. Schader; A. Korthaus (Hrsg.): The Unified Modeling Language - Technical Aspects and Applications, Proceedings (Mannheim, Oktober 1997), S. 250-261, Heidelberg 1998.

Nüttgens (1998) = Markus Nüttgens, Michael Hoffmann, Thomas Feld: Objektorientierte Systementwicklung mit der Unified Modeling Language, 5. Workshop der Fachgruppe 5.1.1: Vorgehensmodelle, IABG. Frankfurt am Main, 1998.

Oestereich (2001) = Bernd Oestereich: Developing Software with UML: Object-oriented analysis and design in practise, 5. Auflage, Addison Wesley Longman, Harlow 2001.

OMG (2001) = www.omg.org: OMG Unified Modeling Language Specification, 2001.

OMG (2003) = www.omg.org: Introduction to OMG's Unified Modeling Language (UML), Stand: 15.06.2003.

OMG (2003a) = www.omg.org: OMG Unified Modeling Language Specification, 2003.

Reisig (1985) = Wolfgang Reisig: Petri Nets: An Introduction, 1. Auflage, Springer-Verlag, Berlin et al. 1985.

Reisig (1985a) = Wolfgang Reisig: Systementwurf mit Netzen, 1. Auflage, Springer Verlag, Berlin et al. 1985.

Rittgen (2000) = Peter Rittgen: Quo vadis EPK in ARIS? - Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik., Wirtschaftsinformatik 42 (2000) S. 27 – 35, Koblenz, 2000.

- Rumbaugh (1999) = Jim Rumbaugh, Ivar Jacobson, Grady Booch: The Unified Modelling Language: Reference Manual, Addison Wesley Longman, Massachusetts 1999.
- Schäfer (2003) = Alexandra Schäfer: Der "Impedance Mismatch": Im Spannungsfeld zwischen objektorientiertem und relationalem Ansatz, in OBJEKTSpektrum Nr.3, 2003.
- Scheer (1992) = August-Wilhelm Scheer: Architektur integrierter Informationssysteme – Grundlagen der Unternehmensmodellierung, 2. Auflage, Springer-Verlage, Berlin, 1992.
- Scheer (1997) = August-Wilhelm Scheer: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse, 7. Auflage, Springer Verlag Berlin et al. 1997.
- Scheer (1998) = August-Wilhelm Scheer: ARIS – vom Geschäftsprozess zum Anwendungssystem, 3. Auflage, Springer-Verlag, Berlin 1998.
- Scheer (1998a) = August-Wilhelm Scheer: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen, 3. Auflage, Springer-Verlag, Berlin 1998.
- Stammen (2001) = Markus Stammen, Johannes Wolf: Vier Stufen bis zur Erleuchtung Ein vierstufiges Vorgehensmodell erleichtert den Umgang mit Prozessmodellierungstools, in: it - Industrielle Informationstechnik - Zeitschrift für Software, Organisation und Prozessengineering, Heft 12/2001, S. 22-24, München, 2001.
- Wehler (2000) = Joachim Wehler: Visuelle Verhaltensmodellierung in kommerziellen Projekten, in: Holger Giese, Stephan Philippi (Hrsg.): Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme. 8. Workshop des Arbeitskreises GROOM der GI-Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung, Universität Münster, 2000.

Krohn, F.; Stammen, M.; Wolf, J.: Geschäftsprozessmodellierung und -optimierung als Darstellungs- und Gestaltungsinstrument, in: Industrie Management, Heft 2/2002