

Heft 114

W. Hoffmann, A.-W. Scheer,
M. Hoffmann

Überführung strukturierter Modellierungsmethoden
in die Object Modeling Technique (OMT)

März 1995

Inhalt

1 INTEGRATION OBJEKTORIENTIERTER ENTWURFSTECHNIKEN IN DIE SOFTWAREENTWICKLUNG	1
2 MODELLE DER OMT UND IHRE GRAPHISCHE NOTATION	2
2.1 DAS OBJEKTMODELL	3
2.2 DAS DYNAMISCHE MODELL	4
2.3 DAS FUNKTIONALE MODELL	6
2.4 ZUSAMMENWIRKEN DER DREI MODELLTYPEN	7
3 ANALOGIEN AUSGEWÄHLTER STRUKTURIERTER METHODEN MIT DER OMT NACH RUMBAUGH ET AL.	7
3.1 ENTITYRELATIONSHIP-MODELL (ERM) UND OBJEKTMODELL	7
3.1.1 <i>Das erweiterte ERM (eERM)</i>	7
3.1.2 <i>Analogien eERM - Objektmodell</i>	9
3.1.3 <i>Beispiel einer Überführung</i>	10
3.1.3.1 Sachverhalt für ein reines Handelsunternehmen	10
3.1.3.2 Darstellung des Sachverhalts als ERM	12
3.1.3.3 Darstellung als Objektmodell	13
3.2 EPK UND DYNAMISCHES MODELL	14
3.2.1 <i>Beschreibung der EPK und der graphischen Notation</i>	14
3.2.1.1 Die elementaren Bausteine der EPK	14
3.2.1.2 Modellierungsregeln	15
3.2.1.3 Verknüpfung der real-time-erweiterten EPK mit der EPK	15
3.2.2 <i>Analogien rEPK - dynamisches Modell</i>	16
3.2.2.1 Ereignisse	16
3.2.2.2 Zustände	17
3.2.2.3 Funktionen	17
3.2.2.4 Verknüpfungen zwischen Ereignissen und Funktionen	17
3.2.2.5 Anmerkung	19
3.2.3 <i>Beispiel einer Überführung</i>	20
3.2.3.1 Verbale Beschreibung des Sachverhalts	20
3.2.3.2 Beschreibung des Sachverhalts durch eine EPK	21
3.3 HIPO UND FUNKTIONALES MODELL	23
3.3.1 <i>HIPO</i>	23
3.3.2 <i>Analogien HIPO - funktionales Modell</i>	24
3.3.2.1 Der Funktionsbaum	24
3.3.2.2 Die Funktionsbeschreibung	24
3.3.2.3 Ebenendiagramme	24
3.3.3 <i>Beispiel einer Überführung</i>	25
3.4 ERGÄNZUNG DER OPERATIONEN IM OBJEKTMODELL	27
4 ANMERKUNGEN	27
LITERATURVERZEICHNIS	28

1 Integration objektorientierter Entwurfstechniken in die Softwareentwicklung

In vielen Unternehmen wurden strukturierte Modellierungsmethoden und geeignete Werkzeuge mit erheblichem Kostenaufwand eingeführt. Diese Konzepte sind weit verbreitet, stoßen auf eine breite Akzeptanz, und es wird eine Vielzahl von CASE-Tools, die diese Entwurfsmethoden unterstützen, auf dem Markt angeboten.

Objektorientierte Ansätze wurden zuerst auf der Implementierungsebene mit der Entwicklung objektorientierter Programmiersprachen eingesetzt. Mit dem Bestreben, Entscheidungen hinsichtlich der Softwarequalität in frühere Entwurfsphasen zu verlagern, Softwarequalität kalkulierbarer zu machen, hält der objektorientierte Gedanke auch Einzug in frühere Stadien des Software-Lifecycles, wodurch die Softwarequalität erheblich verbessert werden soll.

Zur Unterstützung der Wiederverwendung von Teilkomponenten sowie der einfachen Modifikation und Erweiterbarkeit von Software, entwickelt sich der objektorientierte Ansatz nach Meyer zur "besten bekannten Methode", dieser Zielsetzung gerecht zu werden.¹

Ein weiteres Ziel des Softwareengineering ist die kostengünstige Erstellung qualitativ hochwertiger Software. Um Rationalisierungspotentiale zur Kostenminimierung zu finden, müssen die Kosten in den verschiedenen Phasen des Softwareentwicklungsprozesses ermittelt werden. Abbildung 1 zeigt diese Kostenaufteilung nach Scheer.²

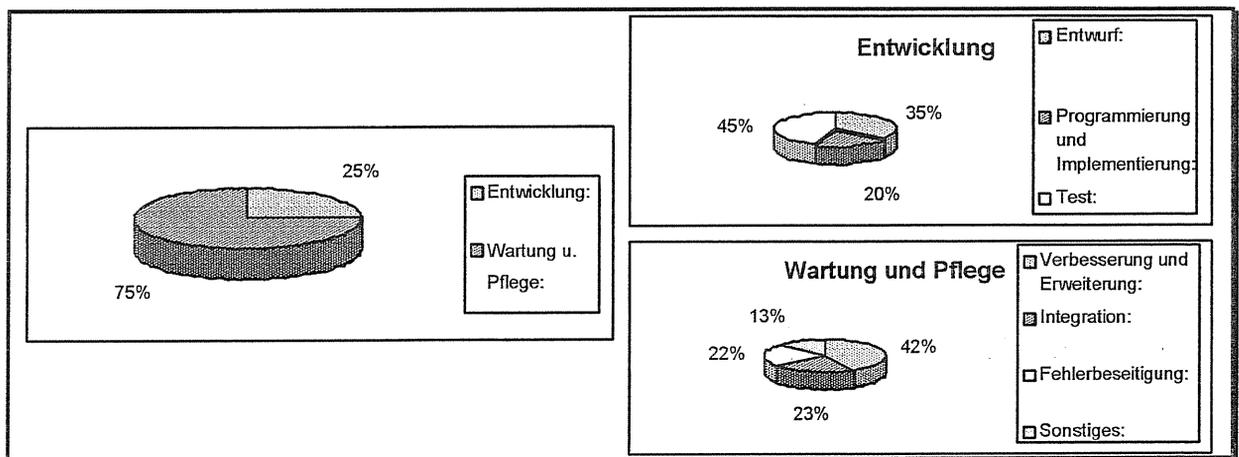


Abbildung 1: Kostenaufteilung in der Softwareentwicklung

Im Bereich der Wartung und Pflege von Software fallen beim Einsatz strukturierter Methoden in der Softwareentwicklung zwei Drittel der Kosten an.

Endres und Uhl vergleichen den Kostenverlauf von Softwareprojekten mit strukturierten Entwurfsmethoden und objektorientierten Entwurfsmethoden gemäß Abbildung 2.³

¹ Vgl. MEYER, B.: Objektorientierte Softwareentwicklung. Übers.: Simonsmeier, W. München-Wien-London 1990, S. 9.

² Vgl. SCHEER, A.-W.: EDV-orientierte Betriebswirtschaftslehre. 4. Auflage, Berlin-Heidelberg-New York-Tokyo 1990, S. 140.

³ Vgl. ENDRES, A., UHL, J.: Objektorientierte Software-Entwicklung. Informatik Spektrum, (1992) 15, S. 255-263, s. bes. S. 261

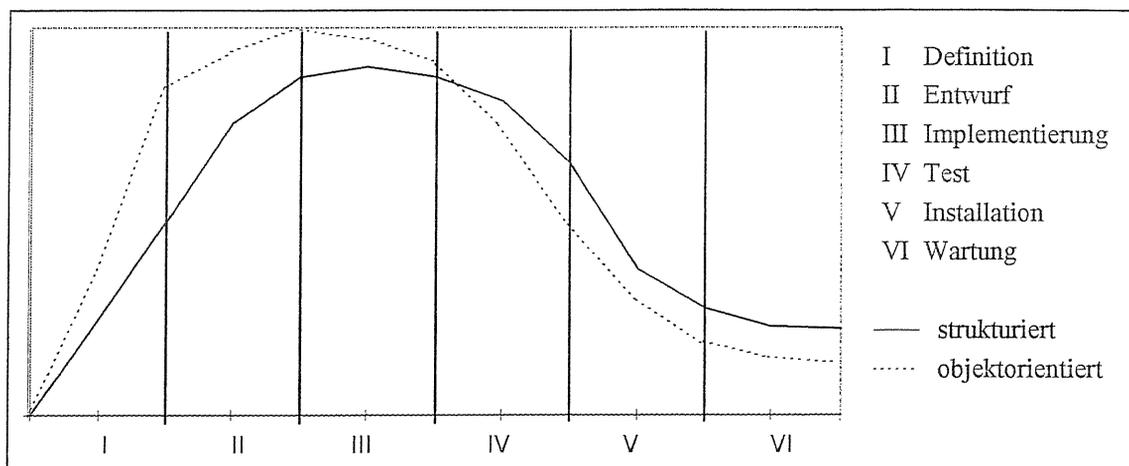


Abbildung 2: Kostenvergleich strukturierter / objektorientierter Softwareprojekte

Es zeigt sich, daß Softwareprojekte, in denen objektorientierte Methoden eingesetzt werden, in den ersten Entwicklungsphasen kostenintensiver sind, als Projekte mit strukturierten Methoden. In den Phasen nach der Implementierung erweist sich der Einsatz objektorientierter Methoden jedoch als kostengünstiger.

Mit der Absicht, Kosten in späteren Phasen des Software-Lifecycles einzusparen, besteht ein Anreiz, objektorientierte Methoden beim Fachkonzeptentwurf einzusetzen. Es stellt sich die Frage, ob beim Einsatz objektorientierter Konzepte eventuell bestehende, mit strukturierten Methoden entwickelte Konzeptionen in einen objektorientierten Modellansatz überführt werden können. Die im folgenden durchgeführte Untersuchung dieses Aspekts verwendet als objektorientierte Modellierungsmethode die "Object Modeling Technique (OMT)" nach J. Rumbaugh. Sie stellt neben einem Phasenkonzept auch die entsprechenden Modelltypen bereit. Eine übersichtliche graphische Notation ist entscheidend für eine leichte Einarbeitung und Nutzung dieser Methode. Nach Kavanagh entwickelt sich die OMT zu der "dominierenden objektorientierten Methode", also eine Art "Quasi-Standard". Kavanagh belegt diese These durch die Beobachtung, daß für die OMT derzeit mehr als ein Dutzend CASE-Tools mit unterschiedlichem Unterstützungsgrad auf dem Markt angeboten werden, während für die besten konkurrierenden objektorientierten Methoden jeweils vier bis fünf Werkzeuge erhältlich sind.⁴

2 Modelle der OMT und ihre graphische Notation

Im Mittelpunkt von OMT steht das *Objektmodell* des Anwendungssystems. Durch die Konzentration auf Objekte und Assoziationen zwischen Objekten wird eine zeitgemäße Sicht auf die Entwicklung von Anwendungssystemen eingenommen. Diese kommt den Anforderungen an Verständlichkeit, Ausdrucksstärke, Wartungsfreundlichkeit und Wiederverwertbarkeit realer Systeme wesentlich näher als Ansätze, die vorwiegend auf funktionale Dekomposition und Datenflüsse ausgerichtet sind. Das Objektmodell wird durch das *dynamische Modell* und das *funktionale Modell* ergänzt. Das dynamische Modell ist für die Modellierung von zeitlichen Reihenfolgeaspekten einer Anwendung zuständig, während das funktionale Modell die vom Anwendungssystem zu bewältigende Verarbeitungsleistung beschreibt.⁵ Im folgenden werden lediglich die Konstrukte der Modelle dargestellt und ihre graphische Notation vorge-

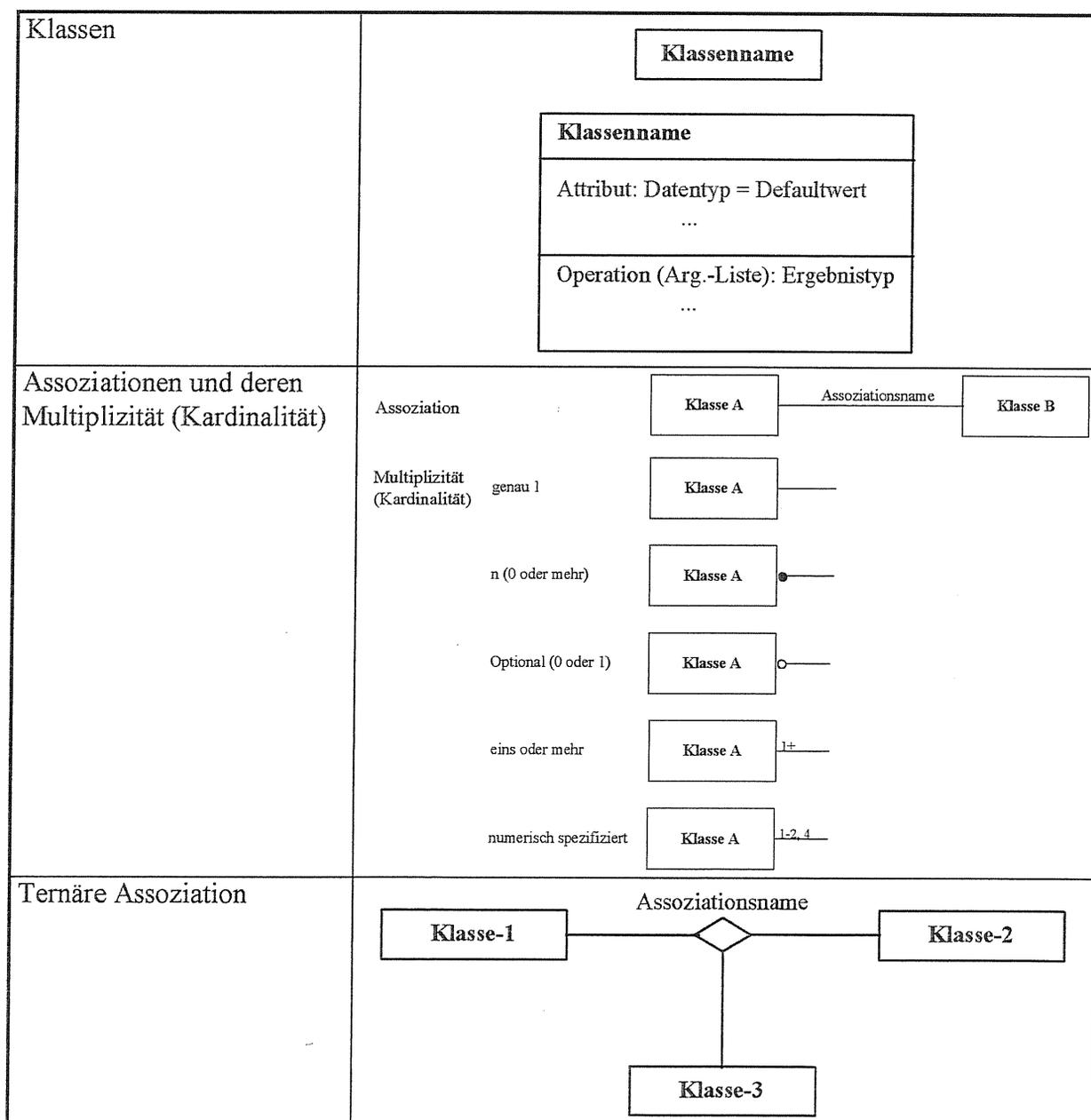
⁴ Vgl. KAVANAGH, D.: Der OMT-Entwicklungsprozeß im Jahr 1994. OBJEKTspektrum 4/94, S. 59-65, s. bes. S. 65.

⁵ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. Übers.: Martin D, Martin C. München-Wien-London 1993, S. xv.

stellt. Die Herleitungen und tiefergehenden Erläuterungen können der Literatur entnommen werden.⁶

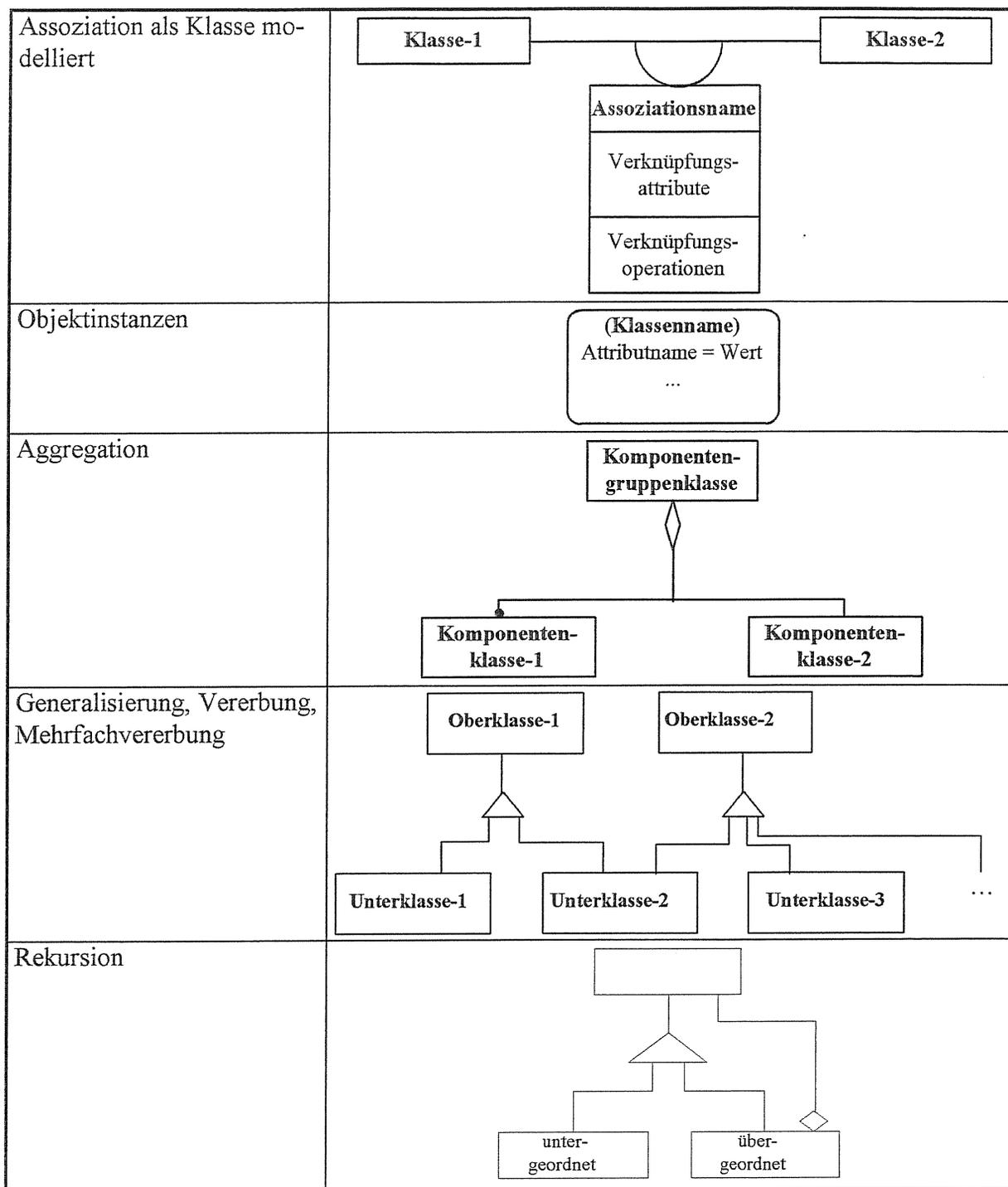
2.1 Das Objektmodell

Im Objektmodell wird die statische Struktur eines Systems beschrieben. Die Objekte, die Relationen zwischen den Objekten, die Attribute und Operationen, die jede Objektklasse charakterisieren, werden dargestellt. Rumbaugh et al. beschreiben es als das wichtigste Modell des Ansatzes, da es als Ausgangspunkt für das *funktionale Modell* und das *dynamische Modell*, die später noch beschrieben werden, gilt. Außerdem unterstützt es die Dokumentation von Systemstrukturen und erleichtert die Einarbeitung in das System.⁷ Abbildung 3 zeigt die wichtigsten Konstrukte der Objektmodellierung.



⁶ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. Übers.: Martin D, Martin C. München-Wien-London 1993.

⁷ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 27.

Abbildung 3: Symbolik des Objektmodells⁸

2.2 Das dynamische Modell

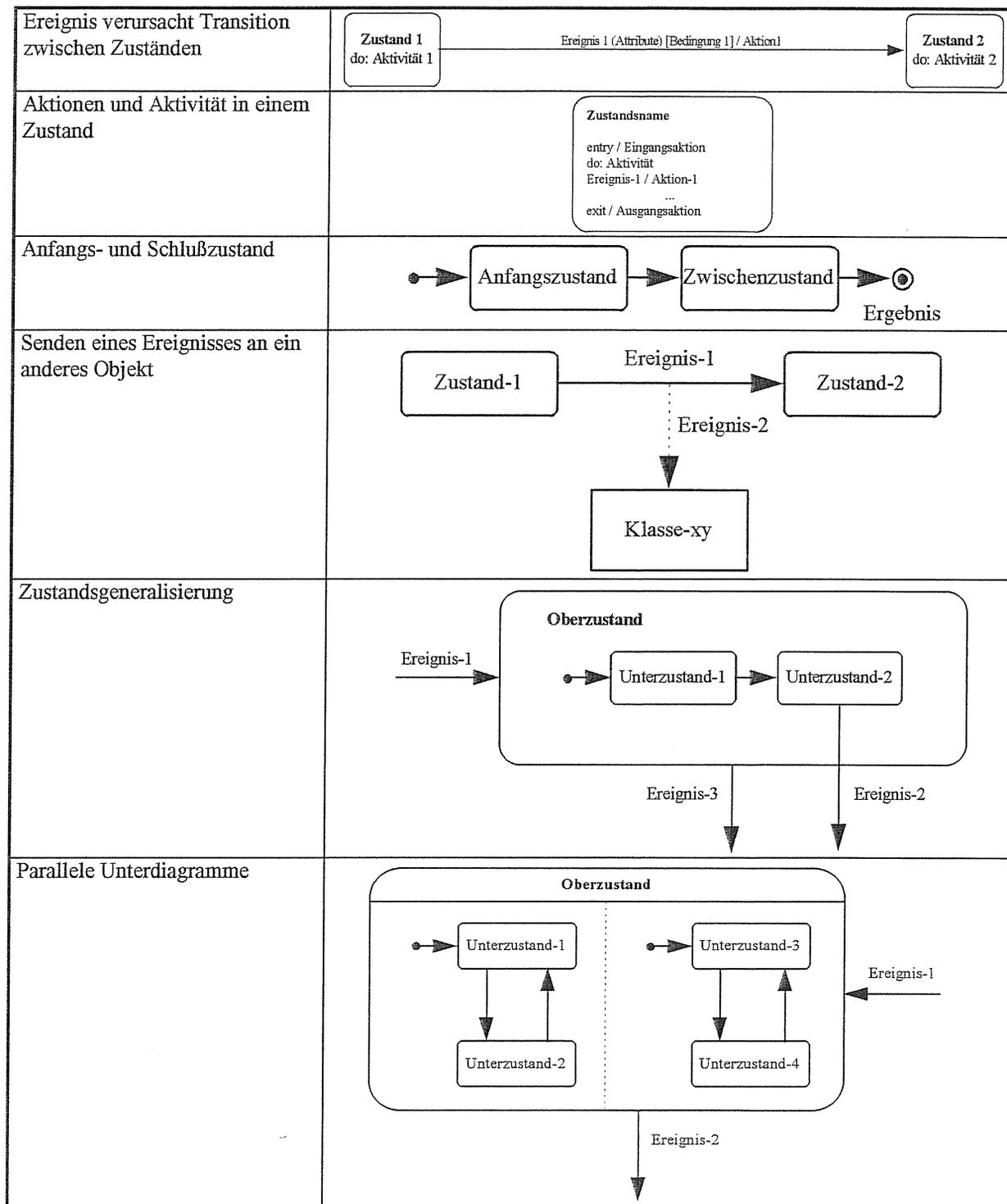
Eine graphische Beschreibung des dynamischen Modells erfolgt anhand von Zustandsdiagrammen, in einer auf Harel zurückgehenden Notation.⁹

Ziel der dynamischen Modellierung ist es, das Netzwerk von Ereignissen, Zuständen und Zustandsübergängen zu abstrahieren und in einem Zustandsdiagramm zu fixieren. Zu diesem

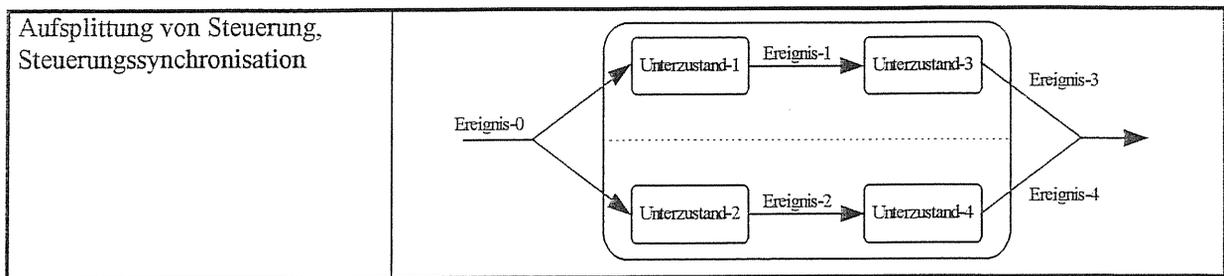
⁸ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 548 f.

⁹ Vgl. HAREL, D.: Statecharts. Science of Computer Programming, (1987), S. 231-274.

Zweck wird jede Klasse mit relevantem dynamischen Verhalten durch ein eigenes Zustandsdiagramm repräsentiert, deren Summe das "Geflecht der Aktivitäten für ein ganzes System", das dynamische Modell, darstellt.¹⁰ Die gemeinsam genutzten Ereignisse bilden dabei die Schnittstellen für das Kumulieren der einzelnen Zustandsdiagramme. Abbildung 4 zeigt die wichtigsten Konstrukte des dynamischen Modells.

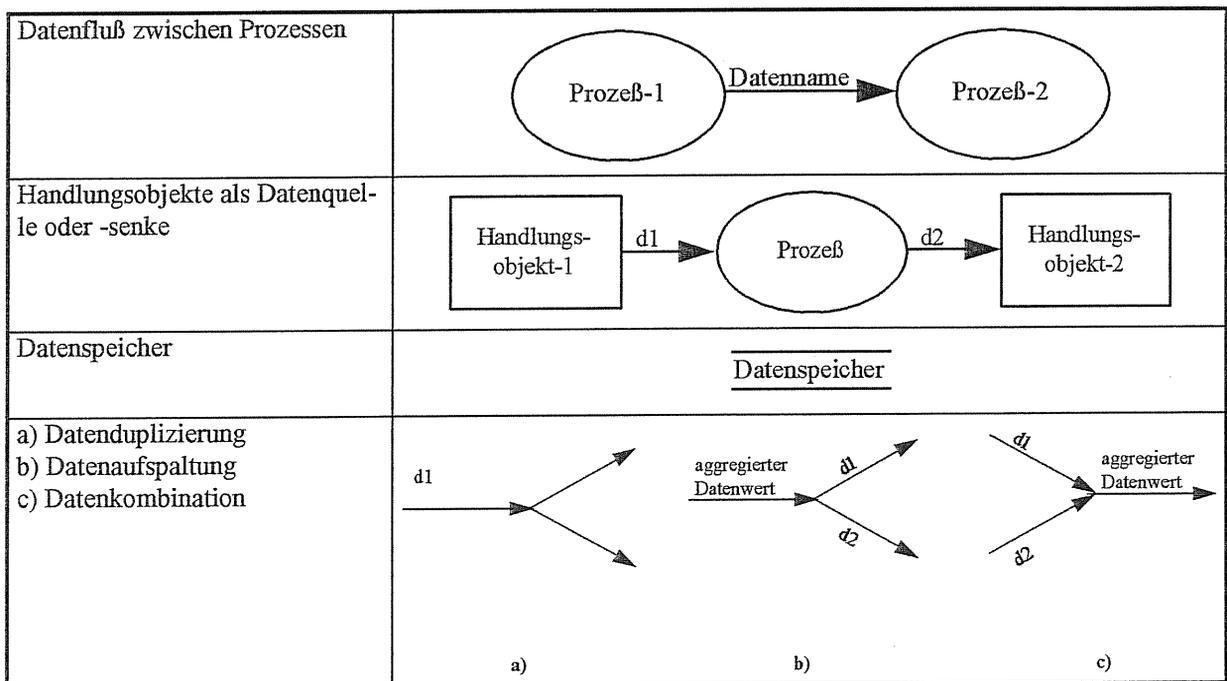


¹⁰ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 103.

Abbildung 4: Notation für Zustandsdiagramme¹¹

2.3 Das funktionale Modell

Aufgabe des funktionalen Modells ist die Beschreibung der Berechnungen innerhalb eines Systems. Es verdeutlicht die Herleitung der Ausgabewerte aus den Eingabewerten und setzt sich aus mehreren *Datenflußdiagrammen*, im folgenden mit DFD abgekürzt, zusammen. Diese sind eine graphische Veranschaulichung des Werteflusses von externen Eingaben über Operationen und internen Datenspeichern bis hin zu externen Ausgaben. Der Aufgabenbereich des funktionalen Modells umfaßt die Spezifikation der Bedeutungen von Operationen und gegebenenfalls die Einschränkungen im Objektmodell bzw. der Aktivitäten im dynamischen Modell. Wie und wann Ergebnisse einer Berechnung entstanden sind, wird nicht untersucht.¹² Abbildung 5 zeigt die wichtigsten Konstrukte des funktionalen Modells.

Abbildung 5: Notation für Datenflußdiagramme¹³

¹¹ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 550.

¹² Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 149.

¹³ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 551.

2.4 Zusammenwirken der drei Modelltypen

Eine vollständige Systembeschreibung erfolgt anhand eines Objektmodells, je einem Zustandsdiagramm für Klassen mit nichttrivialem dynamischem Verhalten, sowie Datenflußdiagrammen, in denen die Aktivitäten und Aktionen mit interessantem funktionalem Charakter definiert werden.¹⁴

- Prozesse im funktionalen Modell entsprechen den Operationen im Objektmodell. Bei der Überführung ist auf die entsprechende Schachteltiefe zu achten. Eine Elementarfunktion ist beispielsweise nicht in einem komplexen Objektmodell zu finden, sondern in der niedrigsten Abstraktionsstufe, während ein globaler Prozeß auf einer höheren Ebene anzusiedeln ist.
- Prozesse im funktionalen Modell vereinen Objekte, die in einer funktionalen Beziehung stehen. Die Beziehungen zwischen Datenlieferanten und den Zielobjekten stellen Implementierungsabhängigkeiten zwischen den Klassen her, indem Datenkonsumenten durch ihre Versorgerklassen implementiert werden.
- Handlungsobjekte sind explizite Objekte im Objektmodell. Deren Datenflüsse repräsentieren Operationen von Objekten. Der Zeitpunkt, wann das Handlungsobjekt aktiv wird, ist im dynamischen Modell beschrieben.
- Auch Datenspeicher stellen Objekte, manchmal auch Attribute im Objektmodell, dar. Jeder Datenfluß in einen Datenspeicher ist eine Aktualisierungsoperation, während jeder Datenfluß aus einem Datenspeicher heraus eine Anfrage charakterisiert, die keine Wirkung auf die im Datenspeicher befindlichen Werte hat. Da diese beiden Möglichkeiten das komplette Verhaltensrepertoire von Datenspeichern darstellt, ist ihr dynamisches Modell im Gegensatz zu dem der Handlungsobjekte trivial und nicht erforderlich.

3 Analogien ausgewählter strukturierter Methoden mit der OMT nach Rumbaugh et al.

3.1 Entityrelationship-Modell (ERM) und Objektmodell

3.1.1 Das erweiterte ERM (eERM)

Das Modell geht zurück auf Chen und wurde 1976 entwickelt. Es dient der Beschreibung sachlogischer Datenstrukturen. Unterschieden wird zwischen *Entities*, *Attributen* und *Beziehungen*, wobei eine Betrachtung sowohl auf Ausprägungs- als auch auf Typenebene erfolgen kann. Während die Typenebene Mengen repräsentiert, werden auf der Ausprägungsebene einzelne Elemente dieser Mengen beschrieben. *Entities* sind reale oder abstrakte Dinge, die für eine Unternehmung von Interesse sind. *Attribute* stellen Eigenschaften von *Entities* bzw. von Entitytypen dar, welche diese beschreiben. Eine Unterscheidung zwischen Entity und Attribut ist abhängig von dem jeweiligen Anwendungszweck. Als Unterscheidungsmerkmal ist festzuhalten, daß *Entities* Attribute besitzen, Attribute jedoch keine weiteren Attribute zur detaillierteren Darstellung. Eine *Beziehung* ist eine sachlogische Verknüpfung zwischen zwei oder mehreren *Entities*.¹⁵ Eine Form der graphischen Darstellung stellt das ERM-Diagramm

¹⁴ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 166 f.

¹⁵ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. 5. Auflage, Berlin et al. 1994, S. 31f.

dar:

Entitytypen werden als Rechtecke abgebildet, Beziehungstypen als Rauten. An den Kanten wird die Kardinalität, die mengentheoretische Verknüpfung der an dem Beziehungstyp beteiligten Entitytypen, eingetragen. Mögliche Varianten sind 1:1-, 1:n-, n:1- bzw. n:m-Beziehungen.

Das bisher dargestellte Grundmodell hat über die Jahre hinweg diverse Erweiterungen erfahren. Im folgenden werden diese lediglich kurz beschrieben.

Bei der *Klassifizierung* werden gleichartige Elemente (Entities) herausgearbeitet und einem Begriff (Entitytyp) zugeordnet. Zwei Elemente sind gleich, wenn sie durch die gleichen Attribute beschrieben werden können. Als Ergebnis erhält man eine Betrachtung auf Entitytypenebene.¹⁶

Bei der *Generalisierung* werden gleichartige Entitytypen einem übergeordneten Begriff zugeordnet. Dabei werden gemeinsame Eigenschaften auf den generalisierten Typ übertragen, so daß in den untergeordneten Typen lediglich davon abweichende Attribute beschrieben werden müssen. Bei umgekehrter Blickrichtung, der Zerlegung eines Oberbegriffs in Teilbegriffe, wird der Fall der *Spezialisierung* beschrieben. Die Eigenschaften des übergeordneten Objekttyps werden auf die untergeordneten vererbt. Außerdem ist es möglich, daß ererbte Eigenschaften abgeändert bzw. neue Eigenschaften hinzugefügt werden. Die Generalisierung wird eher bei der *Bottom-up-Vorgehensweise* der Datenstrukturierung angewendet, während die Spezialisierung der *Top-down-Vorgehensweise* zuzurechnen ist.¹⁷

Die *Aggregation* beschreibt die Möglichkeit durch Zusammenfassung vorhandener, unterschiedlicher Objekttypen neue Begriffe zu bilden. Diese findet im ERM zum einen in der Beschreibung von Beziehungen (a), zum anderen in der *Uminterpretation* von Beziehungen zu Entitytypen (b) Anwendung.¹⁸

- a) Eine *Gruppierung* kann im ERM durch eine 1:n-Beziehung dargestellt werden. Beispielsweise kann das Personal einer Unternehmung bestimmten Abteilungen zugewiesen werden.
- b) Eine *Uminterpretation* wird durch Umrandung der Raute graphisch dargestellt. Außerdem wird die vom uminterpretierten Beziehungstyp ausgehende Linie der neuen Beziehung nicht bis zur Rautenspitze, sondern nur bis zum äußeren Rand geführt. In der weiteren Vorgehensweise wird das Konstrukt als normaler Entitytyp betrachtet.

Es bleibt noch anzumerken, daß sich mehrere alternative graphische Darstellungsweisen im Laufe der Zeit herausgebildet haben, die völlig synonym verwendet werden können. Besonders gilt dies für die Abbildung der Kardinalitäten, der durch die Min-Max-Notation eine größere Aussagekraft verliehen wurde. Jede Multiplizitätsbeschreibung enthält ihr zufolge zwei Angaben, die besagen, wie oft eine Entität minimal und maximal an einem Beziehungstyp beteiligt sein kann. Krähenfußnotation, Pfeilnotation und Bachmannnotation seien hier nur der Vollständigkeit halber erwähnt. Auf eine tiefergehendere Beschreibung der Notationen soll an dieser Stelle mit einem Verweis auf die Literatur verzichtet werden.¹⁹

¹⁶ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 36.

¹⁷ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 36 f.

¹⁸ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 38 f.

¹⁹ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 31-47.

3.1.2 Analogien eERM - Objektmodell

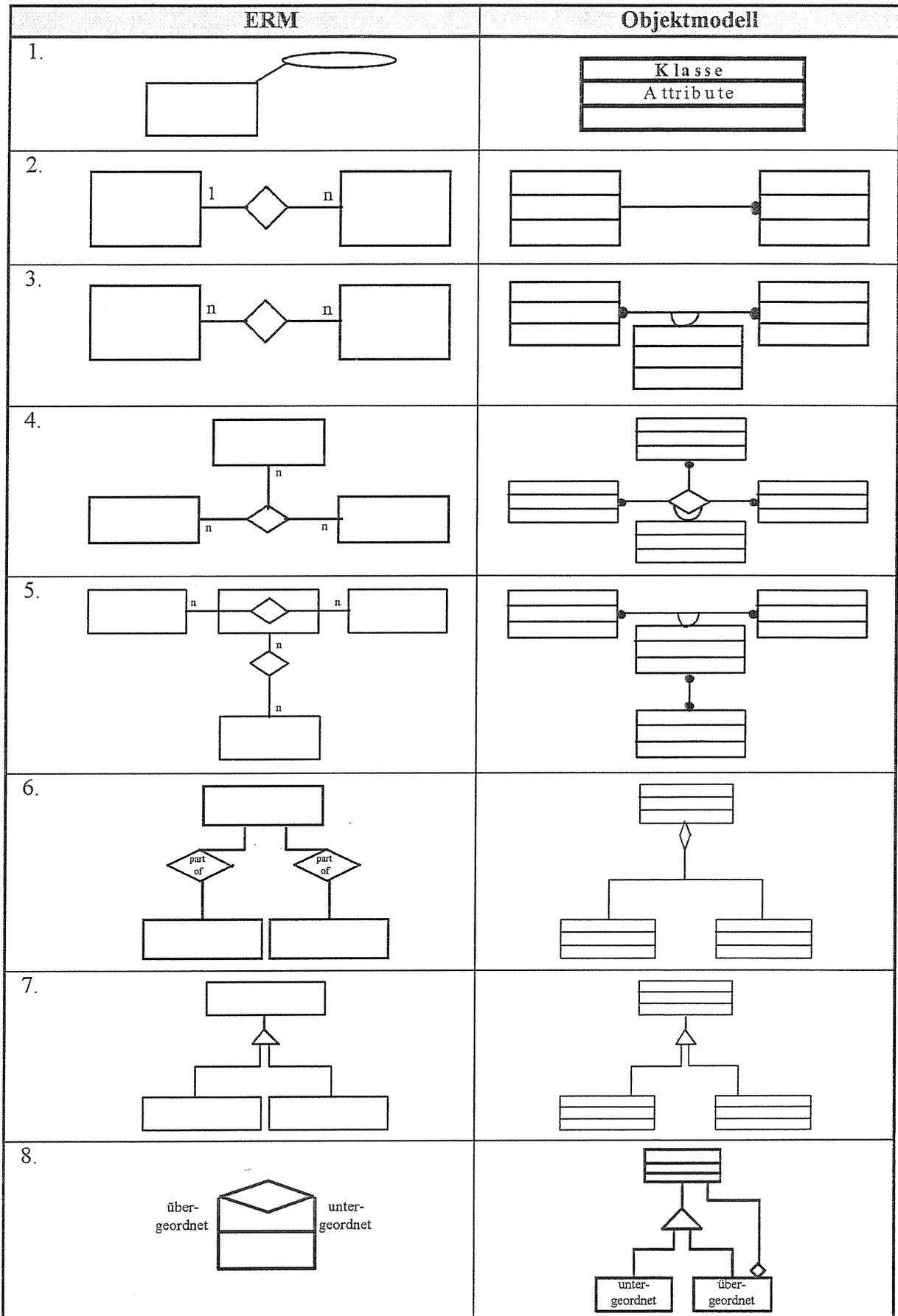


Abbildung 6: Gegenüberstellung der Konstrukte des ERMs und des Objektmodells

Abbildung 6 stellt die bei der Ableitung des Objektmodells aus einem vorhandenen ERM zu beachtenden Restriktionen zusammen. Die Tabellenelemente sind größtenteils selbsterklärend und stellen meist nur eine Transformation der Notation dar. Lediglich in den Zeilen 3 und 4 wird von der Freiheit Gebrauch gemacht, Verknüpfungen als Klassen zu modellieren. Eine Beziehung muß auf jeden Fall als Klasse modelliert werden, wenn sie wieder Ausgangspunkt weiterer Verknüpfungen ist. Dieser Fall ist in Zeile 5 dargestellt und entspricht der Uminterpretation im ERM. Werden einer Beziehung eigene Verknüpfungsattribute oder Operationen zugeordnet, ist eine Implementierung als Klasse ebenfalls sinnvoll. Zeile 8 zeigt die Überführung einer rekursiven Strukturbeziehung des ERM in das Objektmodell. Ein Anwendungsbeispiel hierfür wäre die Darstellung einer Erzeugnisstruktur.

Unter Berücksichtigung der in der Tabelle vorgeschlagenen Transformationsvorschriften sollte es möglich sein, ein ERM in ein Objektmodell ohne Operationen zu überführen.

Balzert gibt eine vergleichbare, jedoch nicht so detaillierte Spezifikation von Regeln zur Überführung ERM-spezifischer Konstrukte in die Symbolik des objektorientierten Ansatzes nach Coad/Yourdon an.²⁰

3.1.3 Beispiel einer Überführung

3.1.3.1 Sachverhalt für ein reines Handelsunternehmen

Ein Auftrag wird zwischen einem Kunden und einer Organisationseinheit der Unternehmung zu einer bestimmten Zeit angelegt. Der Auftragskopf ist durch die Schlüsselattribute KNR (vom Entitytyp KUNDE) ADATUM, (vom Entitytyp ZEIT) bzw. OENR (vom Entitytyp ORGANISATIONSEINHEIT) eindeutig identifiziert. Es handelt sich um eine ternäre Beziehung, wobei jeder Entitytyp n-mal am entstandenen Beziehungstyp beteiligt sein kann. Um dem AUFTRAGSKOPF Positionen zuordnen zu können, ist eine Uminterpretation des AUFTRAGSKOPFes erforderlich. Der Beziehungstyp AUFTRAGSPOSITION wird zwischen den Entitäten ARTIKEL, ZEIT (Erfüllungsdatum muß nicht gleich dem Datum des AUFTRAGSKOPFes sein) und dem durch Uminterpretation entstandenen Entitytyp AUFTRAGSKOPF angelegt. Es handelt sich wiederum um eine ternäre Beziehung, jeweils mit der Kardinalität n. Der ARTIKEL trägt das Schlüsselattribut ARTNR. Die Schlüsselattribute des Beziehungstyps AUFTRAGSPOSITION entstehen durch Vererbung und einem zweiten Datum (BDATUM).

Jeder ARTIKEL kann einem Lagerort zugewiesen werden, wobei sich ein ARTIKEL an mehreren LAGERORTen befinden kann und an jedem LAGERORT mehrere ARTIKEL lagern können (n:m-Beziehung). Der entstehende Beziehungstyp heißt LAGERBESTAND und trägt die Schlüsselattribute ARTNR und LONR. Jeder LAGERORT gehört eindeutig zu einem LAGERRAUM, der wiederum mehrere LAGERORTE beinhalten kann. Der LAGERRAUM trägt das Schlüsselattribut LRNR, so daß die LAGERZUORDNUNG die Schlüsselattribute LONR und LRNR erbt. RESERVIERUNG ist der Beziehungstyp zwischen AUFTRAGSPOSITION und LAGERBESTAND (Uminterpretation LAGERBESTAND erforderlich).

Zu bestimmten Zeiten kommt es vor, daß eine BEDARFSMELDUNG zwischen zwei ORGANISATIONSEINHEITen weitergeleitet wird. In Verbindung mit der ZEIT entsteht eine ternäre Beziehung. Analog zur Auftragsdefinition handelt es sich, exakter ausgedrückt, um den Kopf der Bedarfsmeldung. Dem uminterpretierten Beziehungstyp BEDARFSMELDUNG werden, in Verbindung mit ARTIKEL und ZEIT BEDARFSPOSITIONEN zugeordnet.

Ein BEDARFSDECKUNGS-AUFTRAGSKOPF entsteht durch eine Beziehung zwischen

²⁰ Vgl. BALZERT, H.: Datenmodellierung: Vom Entity-Relationship-Modell zu OOA. Datenbank FOKUS, 05/06 1993, S. 61-66, s. bes. S. 66.

ORGANISATIONSEINHEIT, LIEFERANT und ZEIT. Die Zuordnung von BEDARFSDECKUNGS-AUFTRAGSPOSITIONEN erfolgt durch eine Beziehung zwischen den uminterpretierten Beziehungstypen BEDARFSPOSITION, BEDARFSDECKUNGS-AUFTRAGSKOPF und der ZEIT.²¹

Es bestände die Möglichkeit, KUNDE und LIEFERANT als Spezialisierung des Entitytyps ORGANISATIONSEINHEIT zu betrachten. Das gleiche gilt für die verschiedenen Auftrags-typen, die unter dem Oberbegriff Auftrag zusammengefaßt werden könnten. Außerdem wäre die Darstellung einer Strukturbeziehung innerhalb der Entitytypen ORGANISATIONSEINHEIT oder ARTIKEL denkbar. Davon soll an dieser Stelle jedoch abgesehen werden, da es nicht unser Ziel ist, möglichst vollständige Datenmodelle zu entwickeln, sondern Ansätze zur Überführung eines ERMs ins Objektmodell zu liefern.

²¹ Im Text nicht angegebene Schlüsselattribute können der graphischen Darstellung des ERMs entnommen werden.

3.1.3.2 Darstellung des Sachverhalts als ERM

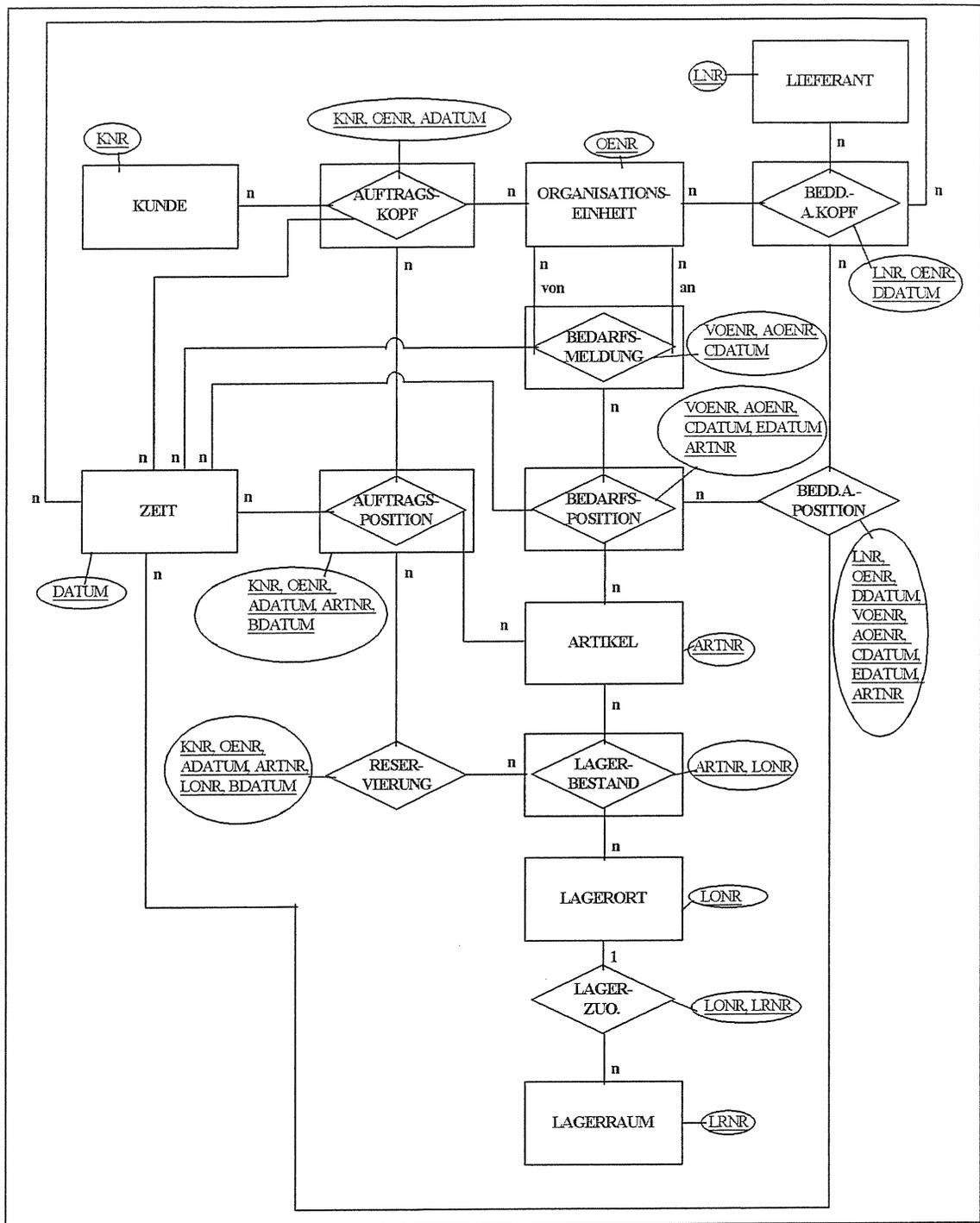


Abbildung 7: ERM Handelsunternehmen

Bei der Überführung in ein Objektmodell (zunächst ohne Operationen) müssen die Entitäten und die Beziehungen inklusive Kardinalitäten transponiert werden.

3.1.3.3 Darstellung als Objektmodell

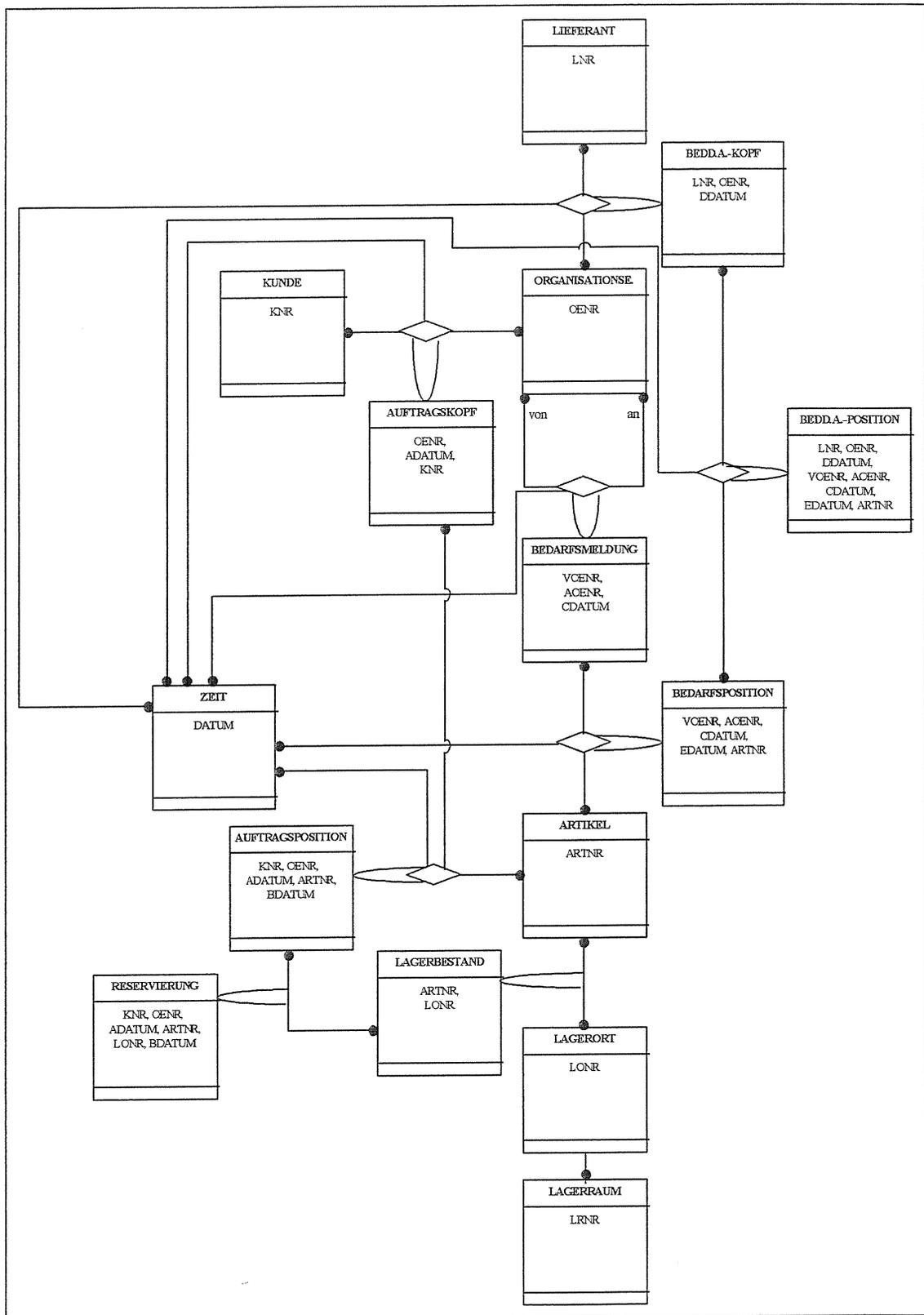


Abbildung 8: Objektmodell (ohne Operationen) für ein Handelsunternehmen

3.2 EPK und dynamisches Modell

3.2.1 Beschreibung der EPK und der graphischen Notation

3.2.1.1 Die elementaren Bausteine der EPK

Die wesentlichen Grundelemente der EPK sind *Ereignisse* und *Funktionen*. Da sich Ereignisse auf Änderungen von Attributen eines Objektes beziehen, sind sie der Datensicht zuzuordnen. Somit stellt die EPK eine Verbindung zwischen der Datensicht und der Funktionssicht her und ist in die Steuerungssicht des ARIS-Modells einzuordnen.²²

Funktionen beschreiben Transformationsprozesse, die zur Erreichung der Unternehmensziele durchgeführt werden. Bei einer Funktionszusammenstellung eines Unternehmens handelt es sich daher um das gesamte Handlungsrepertoire des Unternehmens bzw. seiner Angestellten, das eingesetzt werden kann, um den Unternehmensgewinn unter gegebenen Nebenbedingungen zu maximieren. Eine Darstellung der Prozesse auf unterschiedlichem Abstraktionsniveau ist möglich, es sollte jedoch darauf geachtet werden, daß man sich innerhalb einer EPK auf der selben Stufe bewegt.

In der graphischen Notation der EPK werden Funktionen graphisch als abgerundete Rechtecke dargestellt, wobei die Bezeichnung i. d. R. immer ein Bearbeitungsobjekt und eine Beschreibung der Tätigkeit enthalten sollte (Beispiele hierfür wären: "Angebot erstellen", "Auftrag prüfen"; es kann sich allerdings auch um aggregierte Objekte handeln, wie z. B. "Beschaffung durchführen"). Das Bearbeitungsobjekt muß im zugrundeliegenden Datenmodell (ERM, Clustermodell) wiederzufinden sein.²³

Ereignisse lösen Funktionen aus und sind wiederum Ergebnis von Funktionen.²⁴ Durch eine Aneinanderreihung, d. h. ein Ereignis ist Ergebnis einer Funktion und zugleich Starterereignis für eine weitere Funktion usw., entsteht ein Prozeß, der durch eine komplexe EPK dargestellt wird.

Ereignisse können als Ergebnis einer Zustandsänderung von Objekten angesehen werden, auf die mit einer Funktion reagiert wird, deren Durchführung eine weitere Zustandsänderung herbeiführt. Eine eindeutige Beschreibung eines Ereignisses erfolgt durch das Objekt, welches die Zustandsänderung erfahren hat und einer Tätigkeit im Partizip Perfekt, welche die Art der Änderung spezifiziert. Hier gilt, wie auch bei der Funktionsbeschreibung, daß das Objekt im Datenmodell als Informationsobjekt zu identifizieren sein soll.²⁵

Die logischen Verbindungen zwischen Ereignissen und Funktionen werden durch Pfeile, Ereignisse durch Sechsecke graphisch dargestellt.

²² Vgl. HOFFMANN, W., KIRSCH, J., SCHEER, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten. Hrsg.: Scheer, A.-W. Saarbrücken 1993 Veröffentlichungen des Institutes für Wirtschaftsinformatik (IWi), Heft 101, S. 9 f.

²³ Vgl. HOFFMANN, W., KIRSCH, J., SCHEER, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten. a. a. O., S. 4-5.

²⁴ Vgl. SCHEER, A.-W.: Architektur integrierter Informationssysteme. a. a. O., S. 114.

²⁵ Vgl. HOFFMANN, W., KIRSCH, J., SCHEER, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten. a. a. O., S. 4-7.

3.2.1.2 Modellierungsregeln

Jede Prozeßkette beginnt mit einem Ereignis und muß mit einem Ereignis enden. Nur so ist es möglich, die Anfangsbedingungen, die dazu geführt haben, daß ein Prozeß gestartet wurde sowie das Ergebnis des Prozesses, zu modellieren. Dies unterstützt die Eventualität, getrennt entwickelte EPKs zusammenzuführen, bzw. bestehende Ketten nötigenfalls um Komponenten zu ergänzen.

Da Funktionen von mehreren Ereignissen ausgelöst werden können und mehrere Ereignisse erzeugen können, wurden logische Konnektoren eingeführt, um die Auslöser-Erzeuger-Beziehungen zu modellieren.²⁶

3.2.1.3 Verknüpfung der real-time-erweiterten EPK mit der EPK

Um Zustände erfassen und darstellen zu können, wurde die EPK um eine Zustandslogik erweitert, die auf dem Real-Time-Konzept von Ward/Mellor basiert.²⁷ Die aktive Schnittstelle zum eigentlichen Prozeß bildet die Aktionslogik, die in der EPK Funktionen und Prozesse startet. Eine weitere Schnittstelle stellt die Ereignislogik dar. Ereignisse aus der EPK werden durch Messages an die Zustandslogik übertragen. Beide Schnittstellen werden als Steuerflüsse bezeichnet und graphisch durch strichpunktierte Pfeile symbolisiert.

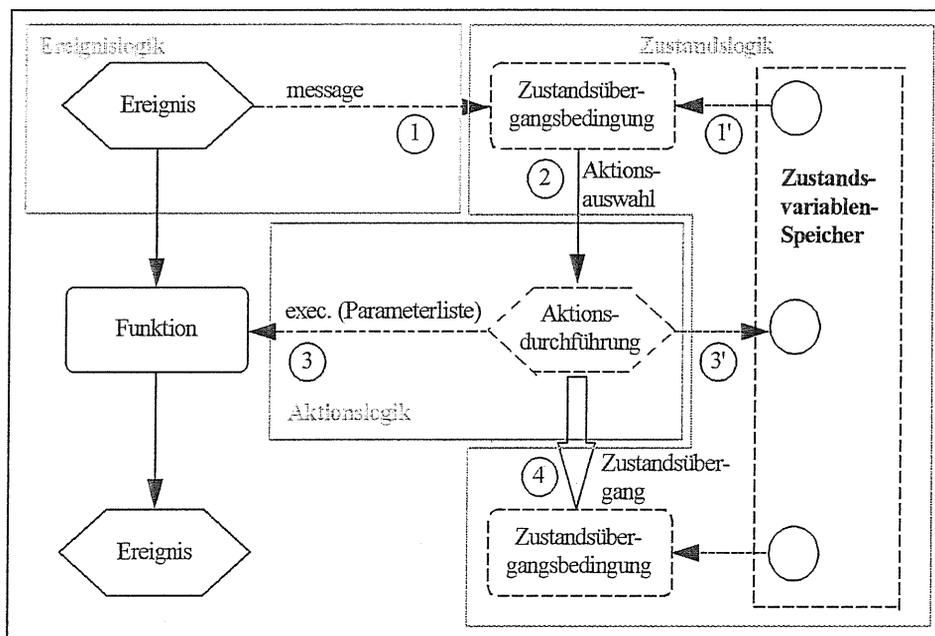


Abbildung 9: Die real-time erweiterte EPK (rEPK)²⁸

Abbildung 9 zeigt die integrierte Methode der real-time-erweiterten EPK (rEPK), wobei die Ablauffolge durch Zahlen verdeutlicht wird. Parallele Prozesse werden durch gleiche Zahlen, mit Hochkomma versehen, dargestellt. Eine vertiefende Beschreibung kann der Literatur entnommen werden.²⁹

²⁶ Vgl. HOFFMANN, W., KIRSCH, J., SCHEER, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten. a. a. O., S. 10-13.

²⁷ Vgl. WARD, P. T., MELLOR, S. J.: Strukturierte Analyse von Echtzeitsystemen. München et al. 1991.

²⁸ Vgl. HOFFMANN, W., WEIN, R., SCHEER, A.-W.: Konzeption eines Steuerungsmodells für Informationssysteme-Basis für die Real-Time-Erweiterung der EPK (rEPK). Hrsg.: Scheer, A.-W. Saarbrücken 1993, Veröffentlichung des Institutes für Wirtschaftsinformatik (IWi), Heft 106, S. 15.

²⁹ Vgl. HOFFMANN, W., WEIN, R., SCHEER, A.-W.: Konzeption eines Steuerungsmodells für Informationssysteme-Basis für die Real-Time-Erweiterung der EPK (rEPK), a. a. O.

3.2.2 Analogien rEPK - dynamisches Modell

3.2.2.1 Ereignisse

"Analysieren Sie die Szenarios, um alle externen Ereignisse zu identifizieren. Ereignisse sind unter anderem Signale, Eingaben, Entscheidungen, Unterbrechungen, Transitionen und Aktionen an oder durch den Benutzer oder externe Geräte. Interne Berechnungsschritte sind keine Ereignisse, Entscheidungspunkte ausgenommen, die mit der *externen Welt* interagieren."³⁰

Mit dieser Anweisung beschreiben Rumbaugh et al. die Vorgehensweise zur Identifikation der für Zustandsdiagramme relevanten Ereignisse, die dort als Auslöser einer Transition dargestellt werden. Interessant ist hierbei die Interpretation des Begriffs "externe Welt". Betrachtet man alle Objekte, die zum System gehören, als systemintern, bleibt für externe Objekte nur noch die Möglichkeit, einer Organisationseinheit als Benutzer des Systems anzugehören und über eine Schnittstelle (Terminal und andere Peripherie) mit dem System zu interagieren. Dies würde für Zustandsdiagramme bedeuten, daß nur solche Ereignisse aufgenommen werden, die von einem "Benutzer" ausgelöst werden und an das System adressiert sind bzw. solche, die vom System erzeugt werden und an Benutzer gerichtet sind. Damit würden Ereignisse, die von einer systeminternen Klasse an eine andere systeminterne Klasse gesendet werden, wie es beispielsweise bei der Triggersteuerung der Fall ist, vernachlässigt werden.

Geht man jedoch von dem Grundsatz des objektorientierten Entwurfs, der Objektkapselung aus und interpretiert in die Bezeichnung "externe Welt" alles, was nicht zu der betreffenden Objektklasse gehört, müssen alle Ereignisse, die von dieser Objektklasse gesendet und empfangen werden, als mögliche Kandidaten für das Zustandsdiagramm dieser Klasse aufgefaßt werden. Dies macht Sinn, wenn man bedenkt, daß eine Objektklasse auf das von einer anderen Objektklasse an sie gerichtete Ereignis durch das Starten einer Aktivität oder das Versenden eines anderen Ereignisses reagiert und so ihren Zustand ändern kann. Nur auf diese Weise kann das dynamische Verhalten eines Systems vollständig durch Zustandsdiagramme für jede Klasse (sofern diese nicht trivial sind) beschrieben werden.

Tatsächlich findet man in allen Beispielen für Zustandsdiagramme von Rumbaugh et al. nur Ereignisse, die infolge einer Benutzer-System-Interaktion aufgetreten sind.³¹ Dies kann jedoch auch auf den Aufbau der Beispiele, die nicht allzu komplex sind, zurückgeführt werden.

³⁰ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 211.

³¹ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 214 f.

Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 110,

Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., 121,

Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 130 f.

Im folgenden werden Regeln zur Ermittlung der Ereignisse für Zustandsdiagramme aus der EPK gegeben:

1. Isolation der Ereignisse aus der EPK
2. Sende- und Empfängerobjektklassen der Ereignisse ermitteln
3. Ein Ereignis wird in das Zustandsdiagramm des Sender- und Empfängerobjektes als Auslöser von Transitionen aufgenommen, sofern mit diesen Ereignissen eine Zustandsänderung einhergeht.

3.2.2.2 Zustände

Zustände werden in der rEPK durch Ausprägungen des Zustandsvariablenspeichers dargestellt. Eine Objektklasse befindet sich so lange im gleichen Zustand, wie die im Zustandsvariablenspeicher abgelegten Zustandsvariablen dieser Objektklasse bestimmte, diesen Zustand charakterisierende Werte annehmen. Diese Zustände liegen zeitlich zwischen zwei sukzessiv bei einer Objektklasse auftretenden Ereignissen, wodurch eine Aktion ausgelöst wird, die dann einen Zustandsübergang bewirkt. Laut Rumbaugh et al. werden die ermittelten Ereignisse zu einem Pfad angeordnet. Die Zustände liegen zwischen jeweils zwei nacheinander bei der betrachteten Objektklasse eintreffenden Ereignissen.³²

3.2.2.3 Funktionen

Rumbaugh et al. unterteilen Funktionen weiter in Aktionen und Aktivitäten. Während eine Aktion mit einem Ereignis assoziiert ist und eine theoretische Zeitdauer von "0" hat, handelt es sich bei einer Aktivität um einen Vorgang, der Zeit beansprucht und eher mit einem Zustand assoziiert ist.³³ Das Zeitkriterium entscheidet, ob eine Funktion der EPK als Aktion oder als Aktivität in Zustandsdiagramme zu überführen ist. Eine Funktion ist in die Zustandsdiagramme für die Objektklassen aufzunehmen.

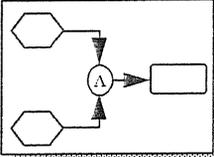
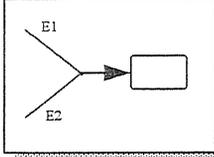
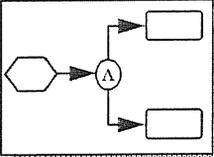
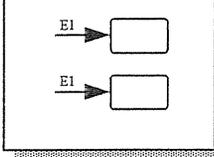
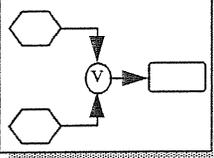
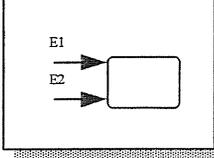
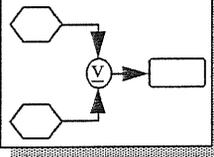
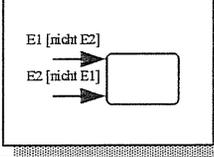
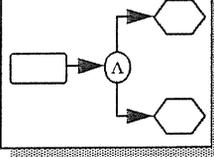
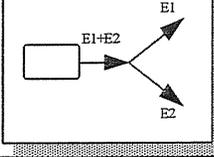
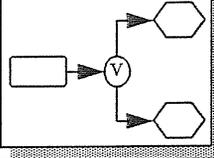
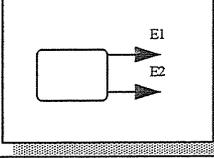
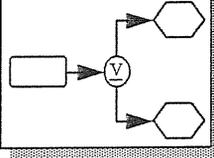
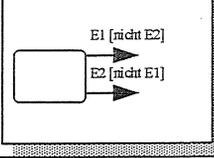
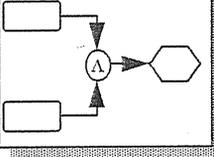
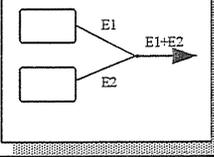
3.2.2.4 Verknüpfungen zwischen Ereignissen und Funktionen

Da Funktionen auch im objektorientierten Ansatz von den gleichen Ereignissen ausgelöst werden müssen, wie bei der Modellierung mit strukturierten Methoden und diese auch wieder zu den gleichen Ereignissen führen müssen, bleibt eine in der EPK dargestellte Verknüpfung erhalten. Wegen der Diskrepanz zwischen dem objektorientierten dynamischen Modell, das sich aus je einem Zustandsdiagramm pro Objektklasse zusammensetzt, und der Prozeßorientiertheit der EPK, lassen sich diese Verbindungen unter Umständen nicht mehr ohne weiteres nachvollziehen, da sie in mehreren unterschiedlichen Zustandsdiagrammen, verschiedene Objektklassen betreffend, realisiert sein können.

Für die Überführung der Symbolik läßt sich folgende Tabelle angeben:

³² Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 211-217.

³³ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 112-114.

EPK, rEPK	Zustandsdiagramm
<p>1</p> 	
<p>2</p> 	
<p>3</p> 	
<p>4</p> 	
<p>5</p> 	
<p>6</p> 	
<p>7</p> 	
<p>8</p> 	

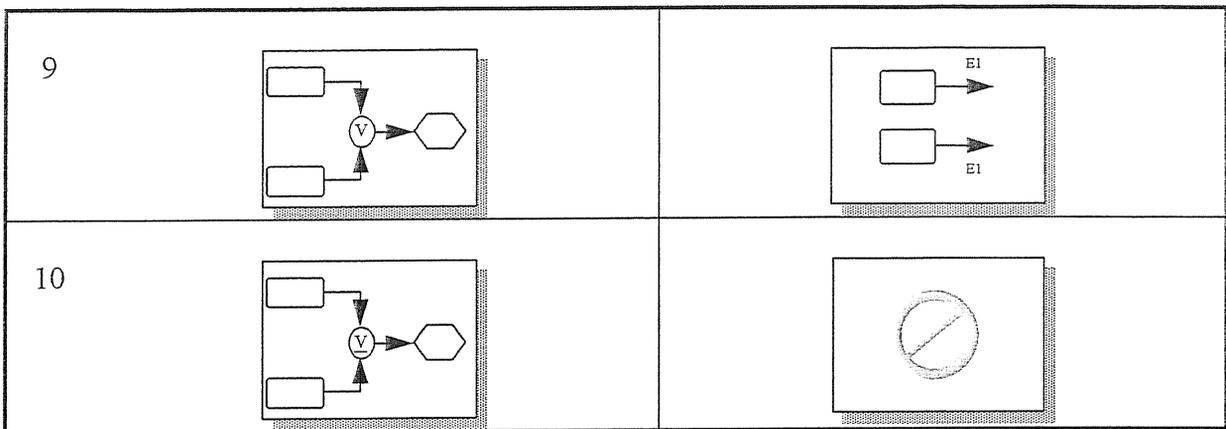


Abbildung 10: Überführung von Verknüpfungen

Bei obiger Abbildung ist zu beachten, daß Funktionen, in EPKs als abgerundetes Rechteck dargestellt, nicht in Zustände innerhalb des Zustandsdiagramms überführt werden. Sie werden zu Aktivitäten innerhalb eines Zustandes.

In Fall 1 werden die beiden konjunktiv verknüpften Ereignisse der EPK innerhalb des Zustandsdiagrammes zu einem zusammengesetzten Ereignis aggregiert. Dieses zusammengesetzte Ereignis ist Auslöser der Transition.

In Zeile 2 ist zu beachten, daß die beiden Transitionen und die dadurch ausgelösten Zustandsveränderungen sich auf zwei unterschiedliche Klassen beziehen können und somit in unterschiedlichen Zustandsdiagrammen Aufnahme finden.

Der 3. Fall spiegelt die Möglichkeit wider, daß unterschiedliche Ereignisse, die auch ihren Ursprung in unterschiedlichen Prozessen haben können, eine Transition auslösen, welche im gleichen Zustand endet. Es handelt sich um den klassischen Fall, daß aus verschiedenen Szenarien stammende Ereignisse, die auf eine Klasse einwirken, alternative Pfade im Zustandsdiagramm dieser Klasse anlegen.³⁴ Dies beschreibt die übergeordnete Situation der Zeile 4. Das "exklusive oder" ist in Zustandsdiagrammen nur durch Bedingungen für betreffende Ereignisse zu realisieren. Zeile 5 beschreibt den Fall, daß eine Funktion zwei konjunktiv verbundene Ereignisse erzeugt. Im Zustandsdiagramm läßt sich dies durch eine Aufspaltung des Steuerflusses beschreiben, indem ein von einer Aktivität erzeugtes, aggregiertes Ereignis in die Einzelkomponenten zerlegt wird. Bei Zeile 7 wird, da es nicht möglich ist ein "exklusives oder" durch Steuerflüsse darzustellen, die gleiche Aussage durch Bedingungen erreicht. Zeile 8 zeigt den umgekehrten Fall der Zeile 5. Hier werden zwei Ereignisse zu einem übergeordneten aggregiert. Dieses zusammengesetzte Ereignis ist Auslöser einer Transition. Der in Zeile 10 durch die EPK dargestellte Fall ist in einem Zustandsdiagramm nicht ohne weiteres darstellbar. Er sollte bei der Modellierung einer EPK, welche einmal in Zustandsdiagramme überführt werden soll, vermieden werden.

3.2.2.5 Anmerkung

Die Entwicklung dynamischer Modelle auf Basis einer EPK ist wegen ihrer Prozeßorientiertheit nicht ohne weiteres möglich. Eine Zerlegung in ihre Bestandteile, die anschließend den Objektklassen des statischen Modells zugeordnet werden müssen, ist erforderlich, um eine Grundlage für das objektorientierte dynamische Modell zu gewinnen. Aufgrund dieser Vorgehensweise ist es nicht ratsam, eine EPK zu modellieren, mit dem Ziel, diese in Zustandsdiagramme zu überführen. Es wäre in diesem Fall angebracht, sofort die von Rumbaugh et al. vorgeschlagene Vorgehensweise zur Erstellung eines dynamischen Modells zu

³⁴ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 213.

beherzigen, ohne den Umweg über die EPK einzuschlagen.

3.2.3 Beispiel einer Überführung

3.2.3.1 Verbale Beschreibung des Sachverhalts

Betrachtet wird ein Ausschnitt des Vertriebes. Geht bei einer Organisationseinheit eine Bestellung ein, ruft diese am Terminal die Auftragserfassung auf und beginnt mit der Eingabe des Auftragskopfes. Ist dieser angelegt, geht das Programm automatisch zu der Erfassung der Auftragspositionen über. Nach Eingabe der Artikelnummer und der Bestellmenge des betreffenden Artikels prüft das System automatisch nach, ob ein Artikel mit der eingegebenen Artikelnummer vorhanden ist. Existiert dieser nicht, so ist die eingegebene Position fehlerhaft. Die mit der Erfassung beschäftigte Organisationseinheit hat die Möglichkeit, eine Kundenrückfrage durchzuführen und kann nach erfolgreichem Abschluß die Position erneut anlegen. Die Auftragsposition wird angenommen, sofern die eingegebene Artikelnummer existent ist. Zusätzlich steht der Organisationseinheit die Möglichkeit offen, die Auftragspositionserfassung abubrechen.

Wird eine korrekte Auftragsposition eingegeben, so wird innerhalb des Systems automatisch eine Verfügbarkeitsprüfung durchgeführt, die als mögliche Ergebnisse "Position ist auf Lager bzw. "Position ist nicht auf Lager" haben kann.

3.2.3.2 Beschreibung des Sachverhalts durch eine EPK

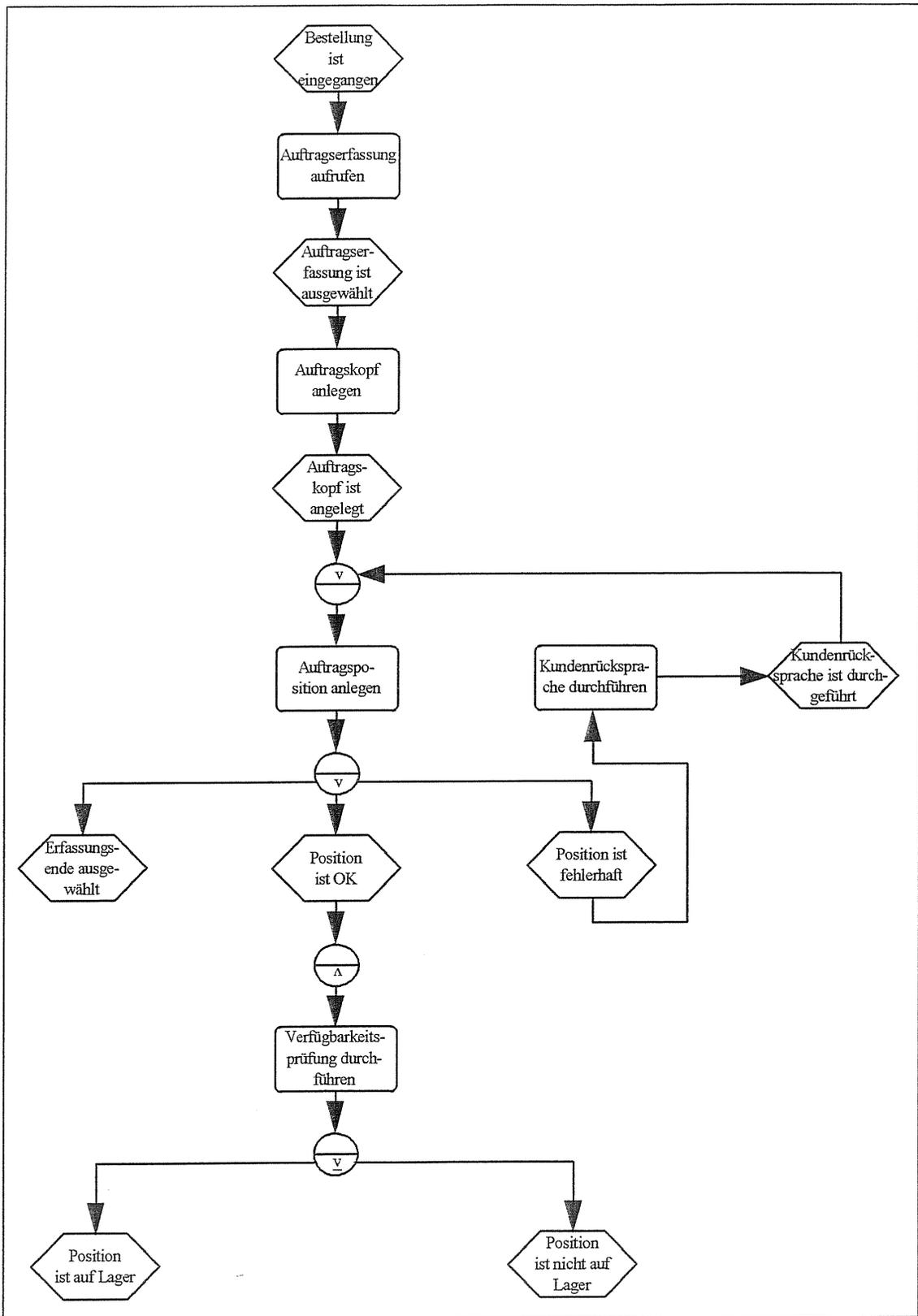


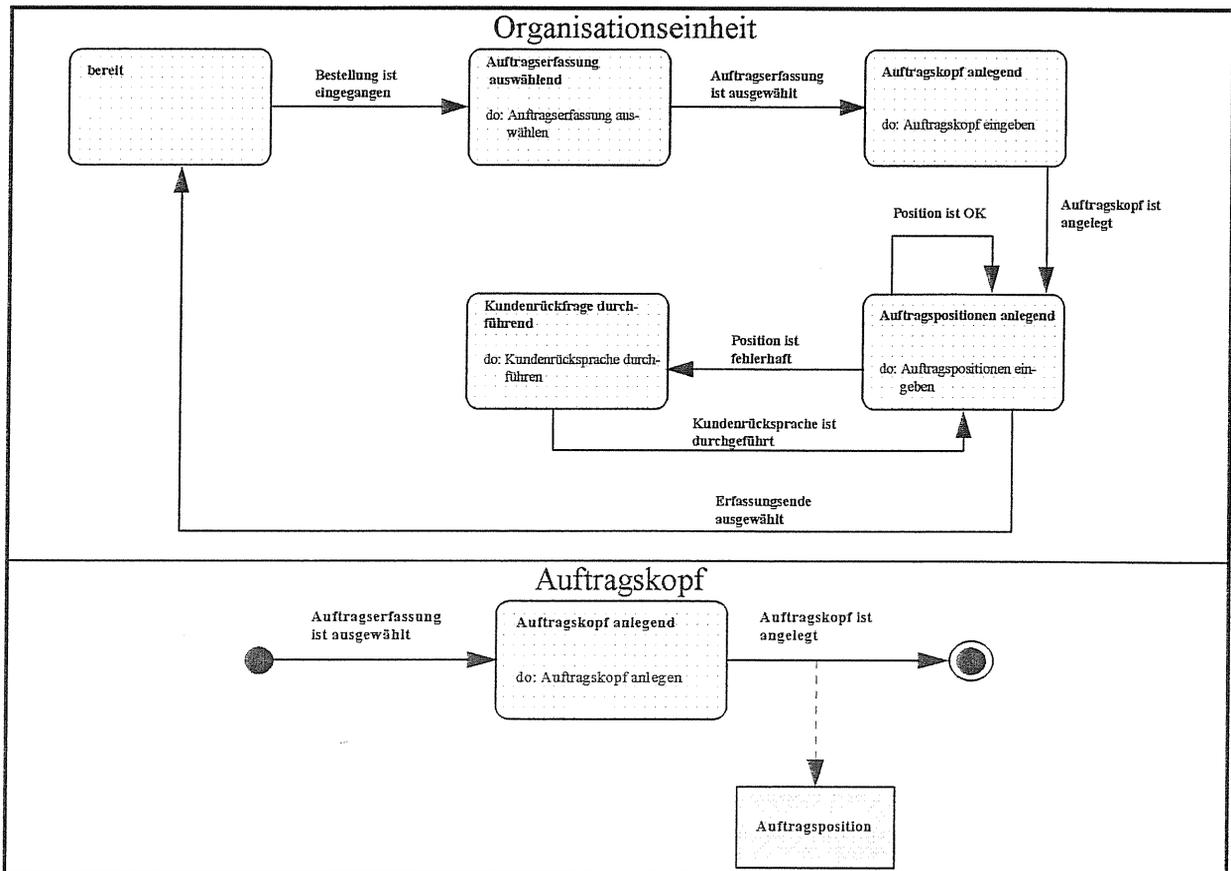
Abbildung 11: EPK Handelsunternehmen

Ereignis	Senderobjekt	Empfängerobjekt
1 Bestellung ist eingegangen	Kunde	OE
2 Auftrags erfassung ist ausgewählt	OE	Auftragskopf
3 Auftragskopf ist angelegt	Auftragskopf	OE, Auftragsposition
4 Position ist OK	Auftragsposition	OE
5 Position ist fehlerhaft	Auftragsposition	OE
6 Rückfrage ist durchgeführt	Kunde	OE
7 Erfassungsende	OE	Auftragsposition
8 Position an Lager	Lagerbestand	Auftragsposition
9 Position ist nicht an Lager	Lagerbestand	Auftragsposition

Abbildung 12: Sender- und Empfängerobjekte den Ereignissen zugeordnet

Zustandsdiagramm für Klasse	Ereignisnummern
OE	1, 2, 3, 4, 5, 6, 7
Kunde	1, 6
Auftragskopf	2, 3
Auftragsposition	3, 4, 5, 7, 8, 9
Lagerbestand	8, 9

Abbildung 13: Zustandsdiagramm-Ereignis-Zuordnung



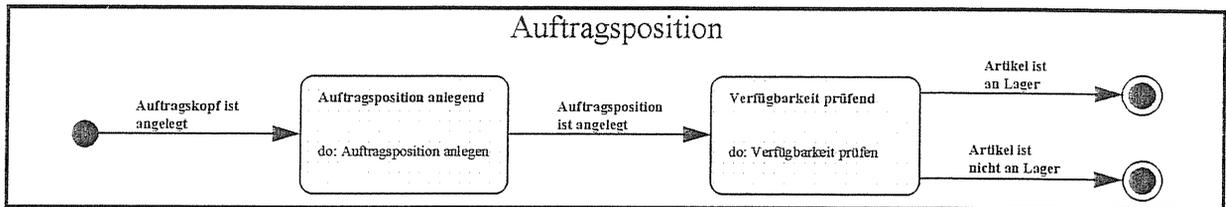


Abbildung 14: Zustandsdiagramme

In Abbildung 14 sieht man die Ereignisse, die gemeinsam von verschiedenen Klassen benutzt werden. Sie sind die Nahtstellen, an denen die Zustandsdiagramme zu einem gesamten dynamischen Modell zusammengefügt werden können. Ein Ereignis muß in das Zustandsdiagramm der sendenden und der empfangenden Klasse aufgenommen werden, wenn für diese damit eine Zustandsänderung einhergeht.

3.3 HIPO und funktionales Modell

3.3.1 HIPO

Bei HIPO handelt es sich um eine graphische Entwurfssprache, die dazu entwickelt wurde, den Entwurfsprozeß zu erleichtern, ein durchgängiges Beschreibungsmittel vom konzeptionellen Entwurf bis zum Detailentwurf zur Verfügung zu stellen und die Dokumentation zu verbessern.³⁵

HIPO-Diagramme bestehen aus zwei Komponenten. Als erste ist der *Funktionsbaum* oder das *Hierarchiediagramm* zu nennen. Hierarchiediagramme dienen dazu, eine komplexe Funktion in Teilfunktionen zu zerlegen, um auf diese Weise die Komplexität zu reduzieren. Eine Untergliederung kann über mehrere Ebenen stattfinden und endet, wenn man bei Elementarfunktionen, die nicht mehr sinnvoll aufgeteilt werden können, angelangt ist. Als Ergebnis erhält man die Funktionsstruktur, die den Funktionskomplex übersichtlicher macht. Diese ist statisch und sagt somit nichts über den zeitlichen Ablauf der Funktionen bzw. Teilfunktionen aus.³⁶

Bei der Entwicklung eines Funktionsbaumes wird sinnvollerweise eine Top-Down-Vorgehensweise angewendet.

Als zweite Komponente der Funktionsbeschreibung dienen sogenannte *Ebenendiagramme*, welche für jede Funktion die Eingabedaten (Input), eine algorithmische Funktionsbeschreibung (Process) und die Ausgabedaten (Output) enthalten. Sie können auf unterschiedlichen Detaillierungsstufen erstellt werden, so daß eine Einteilung in Übersichtsdiagramme und Detaildiagramme erfolgen kann.

Somit ermöglicht HIPO, die Abhängigkeiten zwischen Ein- und Ausgabedaten und die Funktionsstruktur des Systems zu beschreiben. Als wesentliche Vorteile von HIPO sind die leichte Erlernbarkeit und die Unterstützung des Top-Down-Entwurfs anzuführen. Ein Nachteil besteht darin, daß keine Sprachelemente zur Datenspezifikation und Verfeinerung der Datenelemente zur Verfügung gestellt werden, lediglich Datenträgertypen und Dateien sind dar-

³⁵ BALZERT, H.: Die Entwicklung von Software-Systemen. Hrsg.: Böhling, H., Kulisch, U. Mauer H. Mannheim-Leipzig-Wien-Zürich 1992 (Reihe Informatik, Band 34), S. 347.

³⁶ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 20,

siehe auch: SCHEER, A.-W.: Architektur integrierter Informationssysteme. a. a. O., S. 64-69.

stellbar.³⁷

Eine ausführlichere Beschreibung der Entwurfsmethodik kann der Literatur entnommen werden.³⁸

3.3.2 Analogien HIPO - funktionales Modell

3.3.2.1 Der Funktionsbaum

Für dieses Grundelement von HIPO kann zur Definition des funktionalen Modells nach Rumbaugh et al. keine direkte Verwendung gefunden werden. Möglicherweise kann es hilfreich bei der Verschachtelung von Datenflußdiagrammen (DFD) sein, als Grundlage für ein DFD ist der Funktionsbaum jedoch nicht geeignet.

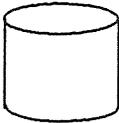
3.3.2.2 Die Funktionsbeschreibung

Sie kann von HIPO ohne Veränderung ins funktionale Modell nach Rumbaugh et al. übertragen werden. Wie in der Vorgangsweise bei der Erstellung des funktionalen Modells bei Rumbaugh et al. beschrieben, kann jede geeignete Form der Funktionsbeschreibung eingesetzt werden.³⁹

3.3.2.3 Ebenendiagramme

Ebenendiagramme sind als Grundlage zur Erstellung von DFDs nur bedingt geeignet. Ihr großes Manko ist, daß keine Sprachelemente zur Datenspezifikation bzw. keine Verfeinerungsmechanismen für Daten bereitgestellt werden.⁴⁰ Eine Darstellung der physischen Datenträger (Formulare, Dateien,...) ist möglich, nicht jedoch die Spezifizierung der einzelnen Datenelemente (Attributsebene). Wird ein Ebenendiagramm als Basis für ein DFD benutzt, so muß es zuerst um eine genaue Datenspezifikation ergänzt werden, auf einem Detaillierungsgrad, der es erlaubt, Attribute darzustellen. Anschließend sind die Handlungsobjekte zu bestimmen, zwischen denen die Daten (Attributswerte) ausgetauscht werden (Datenquellen, Datensenzen).

Sind diese Ergänzungen erfolgt, dürfte eine Umsetzung in die DFD-Notation keine Schwierigkeiten bereiten.

Elemente des Ebenendiagramms (HIPO)	Elemente des DFDs
	<p>Genauere Spezifikation der Daten, die in den Prozeß hinein- bzw. aus dem Prozeß herausfließen.</p> <p style="text-align: center;">Attribute →</p>
	<p>Genauere Spezifikation der Daten, die in den Prozeß hinein- bzw. aus dem Prozeß herausfließen.</p> <p style="text-align: center;">Attribute →</p>

³⁷ Vgl. BALZERT, H.: Die Entwicklung von Software- Systemen. Hrsg.: Böbling, H., Kulisch, U., Mauer, H. Mannheim-Leipzig-Wien-Zürich 1992 (Reihe Informatik, Band 34), S. 351.

³⁸ Vgl. BALZERT, H.: Die Entwicklung von Software- Systemen. a. a. O., S. 347-351.

³⁹ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 221.

⁴⁰ Vgl. BALZERT, H.: Die Entwicklung von Software- Systemen. a. a. O., S. 351.

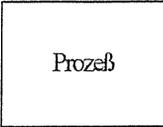
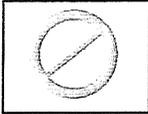
Elemente des Ebenendiagramms (HIPO)	Elemente des DFDs
	
	

Abbildung 15: Elemente des Ebenendiagramms und des DFDs im Vergleich

3.3.3 Beispiel einer Überführung

Mit Hilfe von HIPO könnte die Funktionalität des Handelsunternehmens folgendermaßen dargestellt werden:

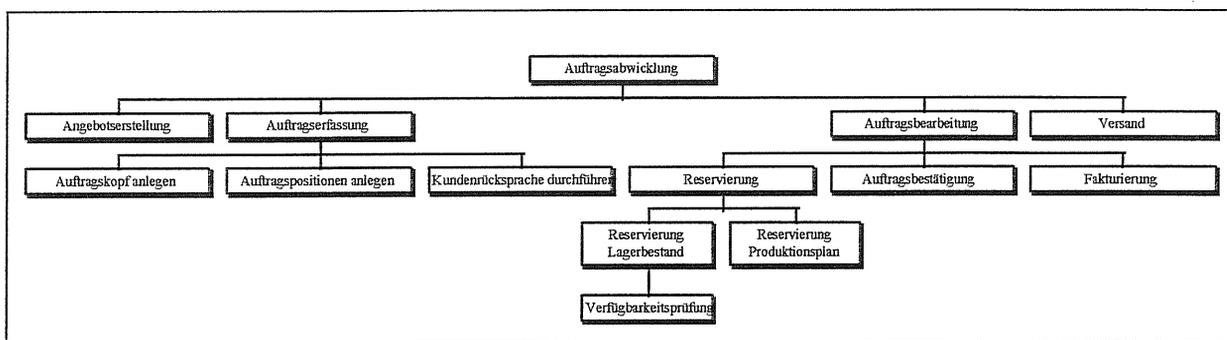
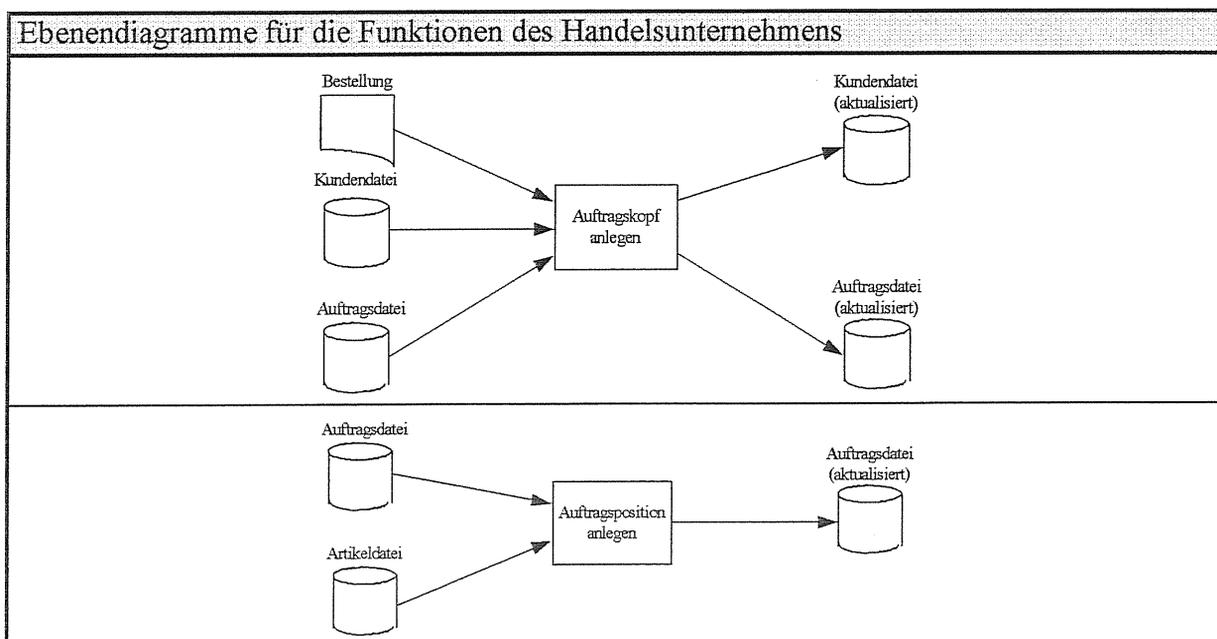


Abbildung 16: Funktionsbaum: Auftragsabwicklung⁴¹



⁴¹ Vgl. SCHEER, A.-W.: Wirtschaftsinformatik. a. a. O., S. 20.

3.4 Ergänzung der Operationen im Objektmodell

Zum Auffinden der Operationen zur Übernahme in das Objektmodell findet die von Rumbaugh et al. dargestellte Vorgehensweise Anwendung.⁴³

4 Anmerkungen

Viele Komponenten von Konzepten, die mit strukturierten Methoden erstellt sind, lassen sich in objektorientierte Modelle übertragen. Eine 1:1-Umsetzung wäre dabei zwar wünschenswert, läßt sich jedoch durch die Divergenz zwischen strukturierten und objektorientierten Modellierungsansätzen nicht verwirklichen. Häufig ist nur nach einer Zerlegung der Modelle strukturierter Methoden in Einzelbausteine und einer Zuordnung der so gewonnenen Elemente zu Objektklassen eine Überführung in objektorientierte Konzepte möglich. Schäfer warnt vor dem Versuch, möglichst alles in die neue Denkwelt herüberzuretten, da die Gefahr besteht, wichtige Vorteile des objektorientierten Ansatzes zunichte zu machen.⁴⁴

Der erhöhte Aufwand bei der Überführung strukturierter Methoden in objektorientierte Methoden stellt nicht den Hauptgrund dar, daß Unternehmen beim Einsatz objektorientierter Methoden noch zögernd reagieren. Immerhin besteht die Aussicht, mit qualitativ hochwertiger Software belohnt zu werden.

Nicht zu unterschätzen ist die Tatsache, daß der Entwicklungsprozeß bezüglich der Aufnahme objektorientierter Ansätze in die frühen Entwurfsphasen des Software-Lifecycles noch nicht abgeschlossen ist. Außerdem ist noch nicht abzusehen, welche der verschiedenen objektorientierten Methoden sich zu einem Quasi-Standard herauskristallisieren wird.

Erschwerend hinzu kommt, daß es den bestehenden Methoden häufig an einer qualifizierten und umfangreichen Dokumentation mangelt. Beispielsweise wäre es wünschenswert, daß Rumbaugh et al. in einer überarbeiteten Auflage ihres Werkes "Objektorientiertes Modellieren und Entwerfen" die teilweise unscharfen und mehrdeutigen Formulierungen beseitigen. Eine Beschreibung einer Methode sollte klar und eindeutig erfolgen und nicht einer Interpretation des Lesers bedürfen. Einige treffende Beispiele auf einer abstrakteren Detaillierungsstufe, wie sie typisch für erste Entwürfe auf Fachkonzeptebene sind, würden sicherlich zu einer höheren Akzeptanz beitragen.

⁴³ Vgl. RUMBAUGH, J. et al.: Objektorientiertes Modellieren und Entwerfen. a. a. O., S. 222-225.

⁴⁴ Vgl. SCHÄFER, S.: Klassische Entwurfsmethoden für die objektorientierte Softwareentwicklung. HMD, 170/1993, S. 47-54, s. bes. S. 53.

Literaturverzeichnis

- Balzert, H.:** Die Entwicklung von Software-Systemen. Hrsg.: Böbling, K. H., Kulisch, U. Mauer, H. Mannheim-Leipzig-Wien-Zürich 1992
- Balzert, H.:** Datenmodellierung: Vom Entity-Relationship-Modell zu OOA. Datenbank FOKUS, 05/06 (1993), S. 61-66
- Barth, G., Welsch, Chr.:** Objektorientierte Programmierung. Informationstechnik, it 6 (1988) S. 404- 420
- Becker, H.:** Objektorientierte Softwareentwicklung. Informationstechnik, it 34 (1992) 2, S. 92-101
- Endres, A., Uhl, J.:** Objektorientierte Software-Entwicklung. Informatik-Spektrum, (1992) 15, S. 255-263
- Harel, D.:** Statecharts. Science of Computer Programming, 8 (1987), S. 231-274
- Heß, H.:** Gestaltungsrichtlinien zur objektorientierten Modellierung. Hrsg.: Scheer, A.-W. Veröffentlichungen des Institutes für Wirtschaftsinformatik (IWi), Heft 99, Saarbrücken 1992
- Hoffmann, W., Wein, R., Scheer A.-W.:** Konzeption eines Steuerungsmodells für Informationssysteme - Basis für die Real-Time-Erweiterung der EPK (rEPK). Hrsg.: Scheer A.-W., Veröffentlichungen des Institutes für Wirtschaftsinformatik (IWi), Heft 106, Saarbrücken 1993
- Hoffmann, W., Kirsch, J., Scheer, A.-W.:** Modellierung mit Ereignisgesteuerten Prozeßketten. Hrsg: Scheer, A.-W., Veröffentlichungen des Institutes für Wirtschaftsinformatik (IWi), Heft 101, Saarbrücken 1993
- Kavanagh, D.:** Der OMT-Entwicklungsprozeß im Jahr 1994. OBJEKTspektrum 4/94, S. 59-65
- Meyer, B.:** Objektorientierte Softwareentwicklung. Übers.: Simonsmeier, W. München-Wien-London 1990
- Raasch, J.:** Systementwicklung mit Strukturierten Methoden. München Wien 1991
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.:** Objektorientiertes Modellieren und Entwerfen. Übers.: Märtin, D., Märtin, Chr. München-Wien-London 1993
- Schäfer, S.:** Klassische Entwurfsmethoden für die objektorientierte Softwareentwicklung. HMD, 170/193, S. 47-54
- Scheer, A.-W.:** EDV-orientierte Betriebswirtschaftslehre. 4. Auflage, Berlin-Heidelberg-New York-Tokyo 1990
- Scheer, A.-W.:** Architektur integrierter Informationssysteme. 2. Auflage, Berlin-Heidelberg-New York-London-Paris-Tokyo-Hong Kong-Barcelona-Budapest 1992
- Scheer, A.-W.:** Wirtschaftsinformatik. 5. Auflage, Berlin-Heidelberg-New York-Paris-Tokyo-Hong Kong-Barcelona-Budapest 1994
- Scheer, A.-W., Hoffmann, W., Wein, R.:** Customizing von Standardsoftware mit Referenzmodellen. HMD 31/1994 Heft 180
- Ward, P. T., Mellor, S. J.:** Strukturierte Analyse von Echtzeitsystemen. München et al. 1991.

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Institut für empirische Wirtschaftsforschung an der Universität des Saarlandes erscheinen in unregelmäßiger Folge.

* Die Hefte 1 - 31 werden nicht mehr verlegt.

- Heft 32: A.-W. Scheer: Einfluß neuer Informationstechnologien auf Methoden und Konzepte der Unternehmensplanung, März 1982, Vortrag anläßlich des Anwendergespräches "Unternehmensplanung und Steuerung in den 80er Jahren in Hamburg vom 24. - 25.11.1981
- Heft 33: A.-W. Scheer: Disposition- und Bestellwesen als Baustein zu integrierten Warenwirtschaftssystemen, März 1982, Vortrag anläßlich des gdi-Seminars "Integrierte Warenwirtschafts-Systeme" in Zürich vom 10. - 12. Dezember 1981
- Heft 34: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS - Ein Ansatz zur Entwicklung prüfungsgerechter Software-Systeme, Mai 1982
- Heft 35: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Konzept einer computergestützten Prüfungsumgebung, Juli 1982
- Heft 36: A.-W. Scheer: Rationalisierungserfolge durch Einsatz der EDV - Ziel und Wirklichkeit, August 1982, Vortrag anläßlich der 3. Saarbrücker Arbeitstagung "Rationalisierung" in Saarbrücken vom 04. - 06. 10.1982
- Heft 37: A.-W. Scheer: DV-gestützte Planungs- und Informationssysteme im Produktionsbereich, September 1982
- Heft 38: A.-W. Scheer: Interaktive Methodenbanken: Benutzerfreundliche Datenanalyse in der Marktforschung, Mai 1983
- Heft 39: A.-W. Scheer: Personal Computing - EDV-Einsatz in Fachabteilungen, Juni 1983
- Heft 40: A.-W. Scheer: Strategische Entscheidungen bei der Gestaltung EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 41: H. Krcmar: Schnittstellenprobleme EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 42: A.-W. Scheer: Factory of the Future, Vorträge im Fachausschuß "Informatik in Produktion und Materialwirtschaft" der Gesellschaft für Informatik e. V., Dezember 1983
- Heft 43: A.-W. Scheer: Einführungsstrategie für ein betriebliches Personal-Computer-Konzept, März 1984
- Heft 44: A.-W. Scheer: Schnittstellen zwischen betriebswirtschaftlicher und technische Datenverarbeitung in der Fabrik der Zukunft, Juli 1984

- Heft 45: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Ein Werkzeug zur Messung der Qualität von Software-Systemen, August 1984
- Heft 46: H. Krcmar: Die Gestaltung von Computer am-Arbeitsplatz-Systemen - ablauforientierte Planung durch Simulation, August 1984
- Heft 47: A.-W. Scheer: Integration des Personal Computers in EDV-Systeme zur Kostenrechnung, August 1984
- Heft 48: A.-W. Scheer: Kriterien für die Aufgabenverteilung in Mikro-Mainframe Anwendungssystemen, April 1985
- Heft 49: A.-W. Scheer: Wirtschaftlichkeitsfaktoren EDV-orientierter betriebswirtschaftlicher Problemlösungen, Juni 1985
- Heft 50: A.-W. Scheer: Konstruktionsbegleitende Kalkulation in CIM-Systemen, August 1985
- Heft 51: A.-W. Scheer: Strategie zur Entwicklung eines CIM-Konzeptes - Organisatorische Entscheidungen bei der CIM-Implementierung, Mai 1986
- Heft 52: P. Loos, T. Ruffing: Verteilte Produktionsplanung und -steuerung unter Einsatz von Mikrocomputern, Juni 1986
- Heft 53: A.-W. Scheer: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung, Juli 1986
- Heft 54: U. Leismann, E. Sick: Konzeption eines Bildschirmtext-gestützten Warenwirtschaftssystems zur Kommunikation in verzweigten Handelsunternehmungen, August 1986
- Heft 55: D. Steinmann: Expertensysteme (ES) in der Produktionsplanung und -steuerung (PPS) unter CIM-Aspekten, November 1987, Vortrag anlässlich der Fachtagung "Expertensysteme in der Produktion" am 16. und 17.11.1987 in München
- Heft 56: A.-W. Scheer: Enterprise wide Data Model (EDM) as a Basis for Integrated Information Systems, Juli 1988
- Heft 57: A.-W. Scheer: Present Trends of the CIM Implementation (A qualitative Survey) Juli 1988
- Heft 58: A.-W. Scheer: CIM in den USA - Stand der Forschung, Entwicklung und Anwendung, November 1988
- Heft 59: R. Herterich, M. Zell: Interaktive Fertigungssteuerung teilautonomer Bereiche, November 1988
- Heft 60: A.-W. Scheer, W. Kraemer: Konzeption und Realisierung eines Expertenunterstützungssystems im Controlling, Januar 1989

- Heft 61: A.-W. Scheer, G. Keller, R. Bartels: Organisatorische Konsequenzen des Einsatzes von Computer Aided Design (CAD) im Rahmen von CIM, Januar 1989
- Heft 62: M. Zell, A.-W. Scheer: Simulation als Entscheidungsunterstützungsinstrument in CIM, September 1989
- Heft 63: A.-W. Scheer: Unternehmens-Datenbanken - Der Weg zu bereichsübergreifenden Datenstrukturen, September 1989
- Heft 64: C. Berkau, W. Kraemer, A.-W. Scheer: Strategische CIM-Konzeption durch Eigenentwicklung von CIM-Modulen und Einsatz von Standardsoftware, Dezember 1989
- Heft 65: A. Hars, A.-W. Scheer: Entwicklungsstand von Leitständen^[1], Dezember 1989
- Heft 66: W. Jost, G. Keller, A.-W. Scheer: CIMAN - Konzeption eines DV-Tools zur Gestaltung einer CIM-orientierten Unternehmensarchitektur, März 1990
- Heft 67: A.-W. Scheer: Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1: Logisches Informationsmodell), März 1990
- Heft 68: W. Kraemer: Einsatzmöglichkeiten von Expertensystemen in betriebswirtschaftlichen Anwendungsgebieten, März 1990
- Heft 69: A.-W. Scheer, R. Bartels, G. Keller: Konzeption zur personalorientierten CIM-Einführung, April 1990
- Heft 70: St. Spang, K. Ibach: Zum Entwicklungsstand von Marketing-Informationssystemen in der Bundesrepublik Deutschland, September 1990
- Heft 71: D. Aue, M. Baresch, G. Keller: **URMEL**, Ein UnternehmensMODELlierungsansatz, Oktober 1990
- Heft 72: M. Zell: Datenmanagement simulationsgestützter Entscheidungsprozesse am Beispiel der Fertigungssteuerung, November 1990
- Heft 73: A.-W. Scheer, M. Bock, R. Bock: Expertensystem zur konstruktionsbegleitenden Kalkulation, November 1990
- Heft 74: R. Bartels, A.-W. Scheer: Ein Gruppenkonzept zur CIM-Einführung, Januar 1991
- Heft 75: M. Nüttgens, St. Eichacker, A.-W. Scheer: CIM-Qualifizierungskonzept für Klein- und Mittelunternehmen (KMU), Januar 1991
- Heft 76: Ch. Houy, J. Klein: Die Vernetzungsstrategie des Instituts für Wirtschaftsinformatik - Migration vom PC-Netzwerk zum Wide Area Network (noch nicht veröffentlicht)
- Heft 77: W. Kraemer: Ausgewählte Aspekte zum Stand der EDV-Unterstützung für das Kostenmanagement: Modellierung benutzerindividueller Auswertungssichten in einem wissensbasierten Controlling-Leitstand, Mai 1991

- Heft 78: H. Heß: Vergleich von Methoden zum objektorientierten Design von Softwaresystemen, August 1991
- Heft 79: A.-W. Scheer: Konsequenzen für die Betriebswirtschaftslehre aus der Entwicklung der Informations- und Kommunikationstechnologien, Mai 1991
- Heft 80: G. Keller, J. Kirsch, M. Nüttgens, A.-W. Scheer: Informationsmodellierung in der Fertigungssteuerung, August 1991
- Heft 81: A.-W. Scheer: Papierlose Beratung - Werkzeugunterstützung bei der DV-Beratung, August 1991
- Heft 82: C. Berkau: VOKAL (System zur Vorgangskettendarstellung und -analyse), Teil 1: Struktur der Modellierungsmethode - Dezember 1991 (wird nicht verlegt)
- Heft 83: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - Theoretical Foundations - 1991
- Heft 84: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - A Survey - 1991
- Heft 85: W. Hoffmann, M. Nüttgens, A.-W. Scheer, St. Scholz: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 1: Produktionsplanung), Oktober 1991
- Heft 86: A.-W. Scheer: Koordinierte Planungsinself: Ein neuer Lösungsansatz für die Produktionsplanung, November 1991
- Heft 87: M. Nüttgens, G. Keller, S. Stehle: Konzeption hyperbasierter Informationssysteme, Dezember 1991
- Heft 88: W. Hoffmann, B. Maldener, M. Nüttgens, A.-W. Scheer: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 2: Produktionssteuerung), Januar 1992
- Heft 89: G. Keller, M. Nüttgens, A.-W. Scheer: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)", Januar 1992
- Heft 90: C. Berkau, A.-W. Scheer: VOKAL (System zur Vorgangskettendarstellung), Teil 2: VKD-Modellierung mit Vokal, Dezember 1991 (wird nicht verlegt)
- Heft 91: C. Berkau: Konzept eines controllingbasierten Prozeßmanagers als intelligentes Multi-Agent-System, Januar 1992
- Heft 92: A. Hars, R. Heib, Chr. Kruse, J. Michely, A.-W. Scheer: Approach to classification for information engineering - methodology and tool specification, August 1992
- Heft 93: M. Nüttgens, A.-W. Scheer, M. Schwab: Integrierte Entsorgungssicherung als Bestandteil des betrieblichen Informationsmanagements, August 1992
- Heft 94: Chr. Kruse, A.-W. Scheer: Modellierung und Analyse dynamischen Systemverhaltens, Oktober 1992

- Heft 95: R. Backes, W. Hoffmann, A.-W. Scheer: Konzeption eines Ereignisklassifikationssystems in Prozeßketten, November 1992
- Heft 96: P. Loos: Die Semantik eines erweiterten Entity-Relationship-Modells und die Überführung in SQL-Datenbanken, November 1992
- Heft 97: Chr. Kruse, M. Gregor: Integrierte Simulationsmodellierung in der Fertigungssteuerung am Beispiel des CIM-TTZ Saarbrücken, Dezember 1992
- Heft 98: R. Heib: Konzeption für ein computergestütztes IS-Controlling, Dezember 1992
- Heft 99: H. Heß: Gestaltungsrichtlinien zur objektorientierten Modellierung, Dezember 1992
- Heft 100: P. Loos: Representation of Data Structures Using the Entity Relationship Model and the Transformation in Relational Databases, January 1993
- Heft 101: W. Hoffmann, J. Kirsch, A.-W. Scheer: Modellierung mit Ereignisgesteuerten Prozeßketten (Methodenbuch, Stand: Dezember 1992), Januar 1993
- Heft 102: P. Loos: Konzeption einer graphischen Rezeptverwaltung und deren Integration in eine CIP-Umgebung - Teil 1, Juni 1993
- Heft 103: wird noch nicht verlegt
- Heft 104: A. Traut; T. Geib; A.-W. Scheer: Sichtgeführter Montagevorgang - Planung, Realisierung, Prozeßmodell, Juni 1993
- Heft 105: A. Hars; V. Zimmermann; A.-W. Scheer: Entwicklungslinien für die computergestützte Modellierung von Aufbau- und Ablauforganisation, Dezember 1993
- Heft 106: W. Hoffmann; R. Wein; A.-W. Scheer: Konzeption eines Steuerungsmodells für Informationssysteme - Basis für die Real-Time-Erweiterung der EPK (rEPK), Dezember 1993
- Heft 107: R. Chen, A.-W. Scheer: Modellierung von Prozeßketten mittels Petri-Netz-Theorie, Februar 1994
- Heft 108: J. Galler, A.-W. Scheer: Workflow-Management: Die ARIS-Architektur als Basis eines multimedialen Workflow-Systems, Mai 1994
- Heft 109: Th. Allweyer, P. Loos, A.-W. Scheer: An Empirical Study on Scheduling in the Process Industries, July 1994
- Heft 110: M. Remme, A.-W. Scheer: Konzeption eines leistungsketteninduzierten Informationssystemmanagements, September 1994
- Heft 111: A.-W. Scheer: ARIS-Toolset: Die Geburt eines Softwareproduktes, Oktober 1994
- Heft 112: A.-W. Scheer, M. Nüttgens, A. Graf v. d. Schulenburg: Informationsmanagement in deutschen Großunternehmen - Eine empirische Erhebung zu Entwicklungsstand und -tendenzen, November 1994

- Heft 113: P. Hirschmann, A.-W. Scheer: Konzeption einer DV-Unterstützung für das überbetriebliche Prozeßmanagement, November 1994
- Heft 114: W. Hoffmann, A.-W. Scheer, M. Hoffmann: Überführung strukturierter Modellierungsmethoden in die Object Modeling Technique (OMT), März 1995