

Hrsg.: Prof. Dr. A.-W. Scheer

**Veröffentlichungen des
Instituts für Wirtschaftsinformatik (IWi),
Universität des Saarlandes**

Im Stadtwald, Gebäude 14.1, D - 66123 Saarbrücken,
phone: (+49) 681-302-3106, fax: (+49) 681-302-3696,
email: iwi@iwi.uni-sb.de

Heft 141

A.-W. Scheer,
M. Nüttgens, V. Zimmermann

**Objektorientierte
Ereignisgesteuerte Prozeßkette (oEPK)
- Methode und Anwendung**

Mai 1997

Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK) - Methode und Anwendung

August-Wilhelm Scheer, Markus Nüttgens, Volker Zimmermann*

Dieser Beitrag befaßt sich mit einem methodischen Ansatz zur integrierten Modellierung von Geschäftsobjekten und -prozessen unter betriebswirtschaftlichen Aspekten. Hierzu wird die am Institut für Wirtschaftsinformatik (IWi) an der Universität des Saarlandes entwickelte Methode der Ereignisgesteuerten Prozeßkette (EPK) um objektorientierte Konzepte auf der Grundlage der Unified Modeling Language (UML) erweitert. Anhand eines konkreten Anwendungsbeispiels wird aufgezeigt, wie betriebswirtschaftlich-organisatorische Szenarien in der Form einer „objektorientierten Ereignisgesteuerten Prozeßkette (oEPK)“ semiformal modelliert und DV-technisch spezifiziert werden können.

Inhaltsverzeichnis

1 Einführung	2
2 Prozeß- und objektorientierte Modellierungsmethoden	4
2.1 Ereignisgesteuerte Prozeßkette (EPK).....	4
2.2 Unified Modeling Language (UML)	8
2.2.1 Basismodell und statisches Modell	9
2.2.2 Dynamisches Modell.....	12
3 Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK)	16
3.1 Modellierungsmethode	16
3.1.1 Modellierung von Geschäftsobjekten	16
3.1.2 Modellierung von Geschäftsprozessen	18
3.2 Anwendungsbeispiel.....	22
4 Ausblick	23
5 Literatur.....	23

* Prof. Dr. A.-W. Scheer, Dr. M. Nüttgens, Dipl.-Kfm. V. Zimmermann, Institut für Wirtschaftsinformatik (IWi) an der Universität des Saarlandes, Im Stadtwald, Geb. 14.1, D-66123 Saarbrücken, Tel. ++49 681 9762 228, Fax ++49 681 77516, email {scheer; nuettgens, zimmermann}@iwi.uni-sb.de

1 Einführung

Die neuere betriebswirtschaftliche Literatur beschäftigt sich im Rahmen der Konzepte zum Business Engineering, Business Process Reengineering oder Process Innovation mit dem Konstrukt des Geschäftsprozesses.¹ Die dabei vertretenen Ansätze zum Verständnis des Geschäftsprozesses lassen sich auf zwei Denkrichtungen zurückführen. Zum einen wird die Aufgabe bzw. Funktion und ihre Abfolge in den Mittelpunkt der Betrachtung gestellt. Als Alternative hierzu wird zunehmend der Objektbezug und damit die betriebswirtschaftliche Leistungserstellung bzw. das Produkt als zentraler Beschreibungsgegenstand genannt. Dies wird nachfolgend als primär objektorientierter Ansatz bezeichnet.

- Bei der primär **funktionsorientierten Prozeßdefinition** wird der Prozeßbegriff synonym zur Funktionskette definiert und die Informationstransformation in den Vordergrund gestellt. So definieren Olle et al. den Prozeß als Transformation von Eingabedaten unter Einsatz von Steuerinformationen und Ressourcen in Ausgabedaten.² Hammer und Champy definieren den Prozeß beispielsweise als „a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer“.³ Sie ergänzen die Betrachtung um den betriebswirtschaftlichen Nutzenaspekt für den Kunden.
- Die primär **objektorientierte Prozeßdefinition** geht auf die Organisationslehre zurück und beschreibt den Prozeß als ein Verrichten an Objekten. Scheer definiert den Prozeß beispielsweise als die mit der Bearbeitung eines bestimmten Objektes verbundenen Funktionen.⁴ Ähnliche Ansätze finden sich auch bei Vertretern wie Becker, Ferstl/Sinz, Gaitanides und Oberweis.⁵

¹ Vgl. Davenport, T.H.: Process Innovation. Reengineering Work through Information Technology, Boston 1993. Hammer, M.; Champy, J.: Reengineering the Corporation. A Manifesto for Business Revolution, New York 1993. Österle, H.: Business Engineering. Prozeß- und Systementwicklung. Band 1: Entwurfstechniken, Berlin 1995. Scheer, A.-W.: Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 6. Aufl., Berlin et al. 1995.

² Vgl. Olle, T.W.; Hagelstein, J.; Macdonald, I.G.; Rolland, C.; Sol, H.G.; Van Assche, F.J.M.; Verrijn-Stuart, A.A.: Information Systems Methodologies: a framework for understanding, 2. Aufl., Workingham et al., 1991.

³ Vgl. Hammer, M.; Champy, J.: Reengineering the Corporation. A Manifesto for Business Revolution, New York 1993.

⁴ Vgl. Scheer, A.-W.: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2. Aufl., Berlin et al. 1992.

⁵ Vgl. Becker, J.: Strukturanalogien in Informationsmodellen. Ihre Definition, ihr Nutzen und ihr Einfluß auf die Bildung von Grundsätzen ordnungsgemäßer Modellierung (GoM), in: König, W. (Hrsg.): Wirtschaftsinformatik'95. Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit, Heidelberg 1995, S. 133-150. Ferstl, O.K.; Sinz, E.J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen, in: Wirtschaftsinformatik 37(1995)3, S. 209-220. Gaitanides, M. Prozeßorganisation: Entwicklung, Ansätze und Programme prozeßorientierter Organisationsgestaltung, München 1983. Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen, Stuttgart et al. 1996.

Je nach Definition überwiegt die Funktion oder das Objekt als bestimmender Faktor für den Prozeßablauf. Prozeßdefinitionen, die den Objektbezug in den Mittelpunkt der Betrachtung stellen, liegt die These zugrunde, daß die Objekte bzw. ihr Status die Transaktionsfolge und damit die Art der Verknüpfung der einzelnen Funktionen festlegt. Diese Sichtweise erfordert eine objektbezogene Ressourcenbereitstellung und hat somit gravierende Auswirkungen auf die Gestaltung der Ablauf- und Aufbauorganisation.⁶

Zu den Grundlagen des objektorientierten Entwurfsparadigmas sind in den vergangenen Jahren im Kontext des Software-Engineering umfassende Ansätze entwickelt worden. Diese Ansätze betonen jedoch primär DV-technische Aspekte bzw. Aspekte der Wiederverwendung von Softwarebausteinen und unterstützen nur unzureichend die Beschreibung betriebswirtschaftlich-organisatorischer Zusammenhänge.⁷ Dies hat auch zur Folge, daß bislang Aspekte der Ablauf- und Aufbauorganisation in objektorientierten Ansätzen weitgehend ausgeklammert werden bzw. eine untergeordnete Rolle spielen. Vielmehr geht man von der Annahme aus, daß ein korrektes Systemdesign korrekte Geschäftsprozesse gewährleistet.

Desweiteren erscheint es aufgrund der Komplexität betriebswirtschaftlicher Anwendungen nicht praktikabel, objektorientierte Paradigmen im Sinne eines top-down Vorgehensmodells in der betrieblichen Praxis anzuwenden und Endanwender bzw. Fachabteilungen mit der leseunfreundlichen Notation objektorientierter Ansätze zu konfrontieren. Andererseits liegt die Vermutung nahe, daß sich Modellierungsergebnisse nicht zwingend durch die Darstellungsmittel einer Beschreibungssprache ergeben, sondern durch den verfolgten Zweck und die Modellierungskonventionen begründet sind.⁸ Daher scheint es auch durchaus von Interesse, die Modellierungsregeln „puritanistischer“ objektorientierter Ansätze aufzuweichen und in Abhängigkeit zur verfolgten Zielsetzung zu variieren. Objektorientierte Modelle können dann - analog zur sichtenorientierten Modellierung - in Abhängigkeit zur Implementierungsnähe sowohl zur Wiedergabe betriebswirtschaftlich-semantischer Inhalte als auch implementierungsnaher Konzepte eingesetzt werden, ohne die dafür notwendigen syntaktischen Beschreibungsstrukturen zu ändern.⁹ Letzteres gilt insbesondere vor dem Hintergrund komponentenbasierter Informationssysteme, bei denen ein Geschäftsprozeß als Interaktion von ausführbaren Objekten implementiert ist.

Nachfolgend wird zunächst ein Überblick über die Geschäftsprozeß- und Geschäftsobjektmodellierung am Beispiel der „Ereignisgesteuerten Prozeßkette (EPK)“ und „Unified Modeling Language (UML)“ gegeben.

⁶ Vgl. Keller, G.; Nüttgens, M.; Scheer, A.-W.: Planungsinselformen: Vom Konzept zum integrierten Informationsmodell, in: Handbuch der modernen Datenverarbeitung (HMD), 29(1992)168, S. 25-39.

⁷ Vgl. Heß, H.: Wiederverwendung von Software - Framework für betriebliche Informationssysteme, Wiesbaden 1993.

⁸ Vgl. Nüttgens, M.: Dezentral-koordiniertes Informationsmanagement. Rahmenkonzept - Koordinationsmodelle - Werkzeug-Shell, Wiesbaden 1995, S. 106.

⁹ Vgl. Bungert, W.; Heß, H.: Objektorientierte Geschäftsprozeßmodellierung, in: Information Management, 10(1995)1, S. 52-63.

2 Prozeß- und objektorientierte Modellierungsmethoden

Die Darstellung zur Geschäftsprozeß- und Geschäftsobjektmodellierung am Beispiel der „Ereignisgesteuerten Prozeßkette (EPK)“ und „Unified Modeling Language (UML)“ schließt die zahlreichen existierenden Vergleiche von Prozeßmodellierungsmethoden¹⁰ und objektorientierten Modellierungsmethoden¹¹ in der Literatur ein und greift die dort erarbeiteten Ergebnisse auf. Rein funktions- oder aktivitätsorientierte Modellierungsansätze mit Blick auf die Informationstransformation werden nicht weiter betrachtet, da sie weder den Prozeß- noch den Objektbezug ausreichend wiedergeben.

2.1 Ereignisgesteuerte Prozeßkette (EPK)

Die Methode der Ereignisgesteuerten Prozeßkette (EPK) wurde im Rahmen der Architektur Integrierter Informationssysteme (ARIS) zur sichtenorientierten Modellierung von Geschäftsprozessen entwickelt.¹² Die EPK-Methode basiert im wesentlichen auf der Petri-Netz-Theorie und kann als eine Variante des Bedingungs-Ereignisnetzes, welches um logische Verknüpfungsoperatoren erweitert wurde, verstanden werden.

Wesentliches Kennzeichen ist die Abbildung der zu einem Prozeß gehörenden Funktionen in deren zeitlich-logischer Abfolge. Die Kontrollflußsteuerung zwischen den Funktionen kann über Geschäftsregeln (business rules) beschrieben werden. Sie lassen sich auf die Konstrukte Ereignis (event), Bedingung (condition) und Funktion/Methode (action) zurückführen.¹³ Es besteht auch die Möglichkeit, Ereignis- und Bedingungskomponenten zusammenzufassen. Diese Vereinfachung liegt auch der EPK zugrunde.

Da innerhalb eines Prozesses auch Entscheidungen auf Basis von Bedingungen und Regeln getroffen werden, existieren in der EPK Verknüpfungsoperatoren („und“, „oder“, „exklusiv oder“). Funktionen, Ereignisse und Verknüpfungsoperatoren stellen das Grundmodell der

¹⁰ Zum Überblick von Prozeßmodellierungsmethoden vgl. Hess, T.: Entwurf betrieblicher Prozesse: Grundlagen - Bestehende Methoden - Neue Ansätze, Wiesbaden 1996, S. 31-94. Keller, G.: Informationsmanagement in objektorientierten Organisationsstrukturen, Wiesbaden 1993, S. 71-105. Kruse: Referenzmodellgestütztes Geschäftsprozeßmanagement: Ein Ansatz zur prozeßorientierten Gestaltung vertriebslogistischer Systeme, Wiesbaden 1996, S. 90-106. Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden 1996, S. 48-84.

¹¹ Zum Überblick von objektorientierten Modellierungsmethoden vgl. Balzert, H.: Methoden der objektorientierten Systemanalyse, 2. Aufl., Heidelberg et al. 1996. Heß, H.: Wiederverwendung von Software - Framework für betriebliche Informationssysteme, Wiesbaden 1993, S. 60-71.

¹² Vgl. Scheer, A.-W.: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2. Aufl., Berlin et al. 1992. Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“, in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992. Scheer, A.-W.: Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 6. Aufl., Berlin et al. 1995.

¹³ Vgl. Knolmayer, G.; Herbst, H.: Business Rules, in: Wirtschaftsinformatik, 35(1993)4, S. 386-390.

Ereignisgesteuerten Prozeßkette dar. Dieses Grundmodell kann - wie in Abbildung 1 dargestellt - um weitere semantische Beschreibungselemente ergänzt werden.

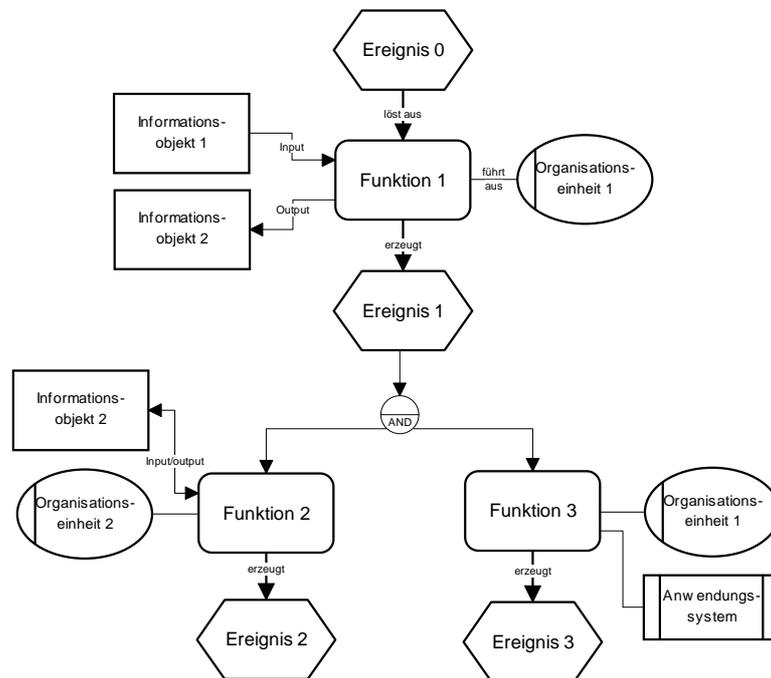


Abbildung 1: Erweiterte Ereignisgesteuerte Prozeßkette (eEPK)

Ein solcher Diagrammtyp wird auch als erweiterte Ereignisgesteuerte Prozeßkette (eEPK) bezeichnet. Beispiele für Erweiterungen sind die Abbildung von Datenflüssen, Organisationseinheiten oder Anwendungssystemen. Den Kanten zwischen den graphischen Objekten werden dabei Rollen zugewiesen. Beispielsweise drückt die Kante zwischen einer Organisationseinheit und einer Funktion die Rolle einer Organisationseinheit im Hinblick auf die Funktionsausführung aus (z.B. „führt aus“, „ist fachlich verantwortlich“, usw.).

Um den objektorientierten Gedanken stärker in die EPK einzubringen, wurde sie in jüngster Zeit um objektorientierte Konstrukte erweitert. Beim Ansatz von Bungert/Heß werden die Beziehungen der Methoden in der ARIS-Architektur und hier speziell der EPK zu den objektorientierten Konstrukten dargestellt.¹⁴ Abbildung 2 zeigt die vorgeschlagenen Erweiterungen und die mit diesen Erweiterungen erzielten semantischen Darstellungsergebnisse. Als Beispielprozeß wurde die Kundenauftragsabwicklung gewählt.

- Abbildung 2a) stellt die zugrundeliegenden objektorientierten Konstrukte in Anlehnung an das Entity-Relationship-Modell und die hier verfügbaren Elemente dar.

¹⁴ Vgl. Bungert, W.; Heß, H.: Objektorientierte Geschäftsprozessmodellierung, in: Information Management, 10(1995)1, S. 52-63.

- Abbildung 2b) beschreibt in einem Szenario die Objekte mit ihren Funktionen und die Ereignisse, die die Funktionen in den Objekten starten bzw. aus ihnen hervorgehen.
- Abbildung 2c) beschreibt den Geschäftsprozeß in Form der ereignisgesteuerten Prozeßkette, wie er sich im Gesamtzusammenhang ergibt und wie er aus der Abbildung 2b) abgeleitet wurde. Das Objekt ist hier implizit in den Funktionen enthalten, aber nicht explizit ausgewiesen.

Der Zusammenhang zwischen den beiden Darstellungen b) und c) ergibt sich über die Vorgehensweise. Zur Vorgehensweise werden sechs Schritte definiert:

1. Definition relevanter Objektklassen. Im Beispiel der „Kunde“, das „Produkt“ und der „Kundenauftrag“ (Schritt 1).
2. Zuordnung der Funktionen zu den Objekttypen (Schritt 2). Im Beispiel beim „Kundenauftrag“ die Funktionen „definieren“, „technische Machbarkeit prüfen“, „kaufmännische Machbarkeit prüfen“, „annehmen“ und „ablehnen“.
3. Detaillierung und Erweiterung der Objekttypstrukturen (Schritt 3). Hierbei werden aufgrund von bereits erfolgten Zuordnungen neue Objekte oder Funktionen definiert.
4. Definition der Startereignisse (Schritt 4). Den Funktionen werden Startereignisse zugeordnet. Beispielsweise der Funktion „definieren“ das Startereignis „Kunde ruft an“.
5. Definition der Endereignisse (Schritt 5). Es werden die Endereignisse je Funktion abgeleitet. Beispielsweise „Kundenauftrag ist definiert“ als Endereignis der Objektfunktion „definieren“ des Kundenauftrags.
6. Darstellung des Gesamtprozesses als Ereignisgesteuerte Prozeßkette (Schritt 6). Der Prozeßverlauf wird als Folge von Funktionen und Ereignissen, wie er der Grundstruktur der EPK entspricht, abgebildet.

Der vorgestellte Ansatz stellt eine erste konzeptionelle Grundlage für die Verbindung von Geschäftsprozeßmodellierung und Objektorientierung mit der Ereignisgesteuerten Prozeßkette dar. Vorteile der Wiederverwendung durch die Objektorientierung werden mit den Vorteilen der Prozeßdarstellung verknüpft. Damit können objektorientiert auch Aspekte der Aufbau- und Ablauforganisation betrachtet werden. Eine integrierte Betrachtung von Objekt und Prozeß in einem gemeinsamen Modell findet aber nicht statt. Beim Übergang von Schritt 5 nach Schritt 6 gehen somit die semantischen Informationen über das Objekt verloren.

Einen ergänzenden Ansatz wählt Rosemann, indem er das im jeweiligen Prozeßabschnitt relevante „Prozeßobjekt“ im Diagramm parallel aufführt. Eine Integration von Geschäftsprozeß und -objekt wird hierbei jedoch ebenfalls nicht explizit modelliert.¹⁵

¹⁵ Vgl. Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden 1996, 76-84.

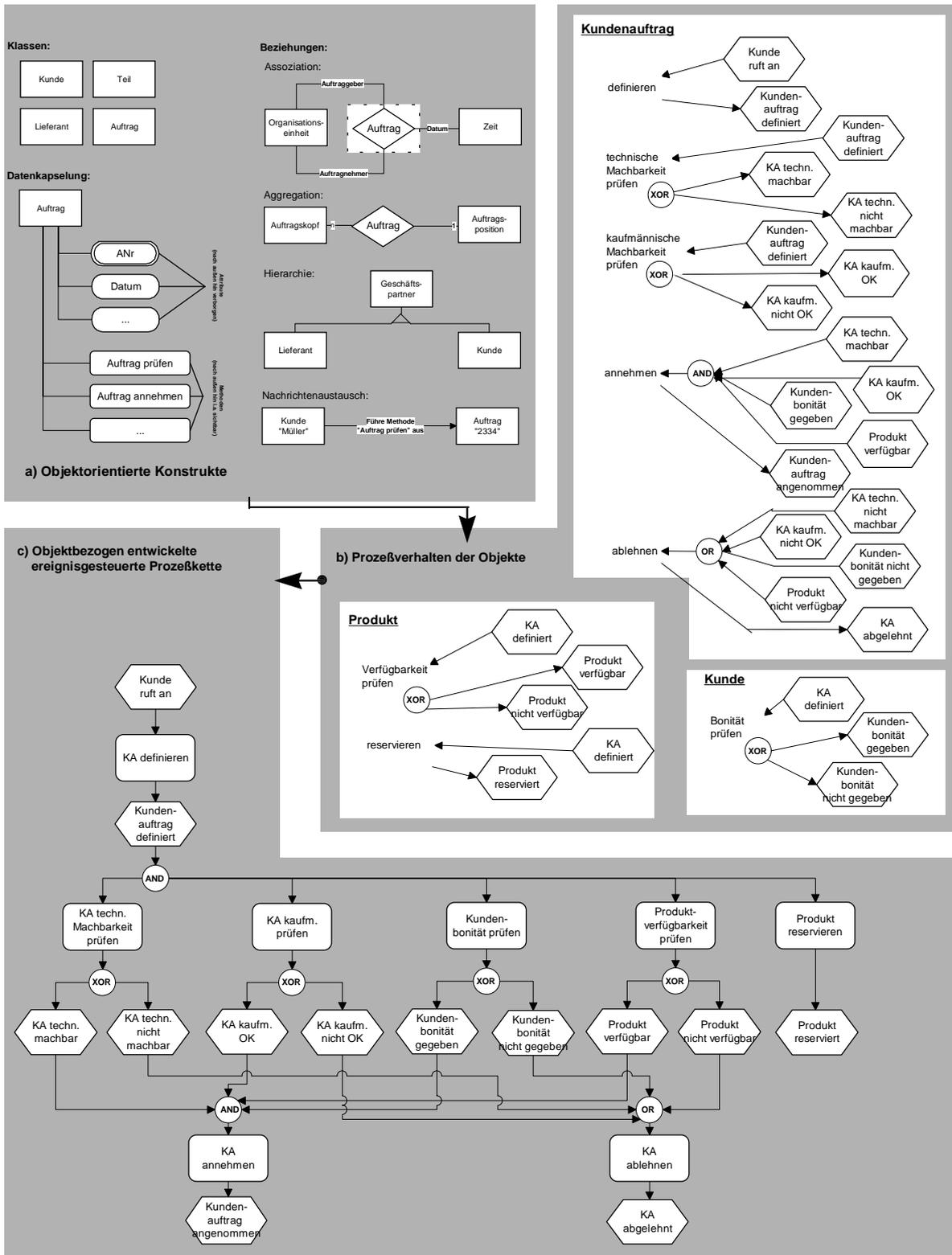


Abbildung 2: Objektorientierte Konstrukte in Verbindung mit der EPK¹⁶

¹⁶ Vgl. Bungert, W.; Heß, H.: Objektorientierte Geschäftsprozessmodellierung, in: Information Management, 10(1995)1, S. 53, 61 und 62.

2.2 Unified Modeling Language (UML)

In der Unified Modeling Language (UML) werden derzeit zahlreiche Methodenkonzepte zur objektorientierten Modellierung zusammengeführt¹⁷. Hierzu zählen insbesondere die Methoden OOD von Booch, OOSE von Jacobson, OMT von Rumbaugh et al., RDD von Wirfs-Brock et al. und bzgl. dynamischer Methoden die Entwicklungen von Harel in Form der State Charts (vgl. Abbildung 3).¹⁸

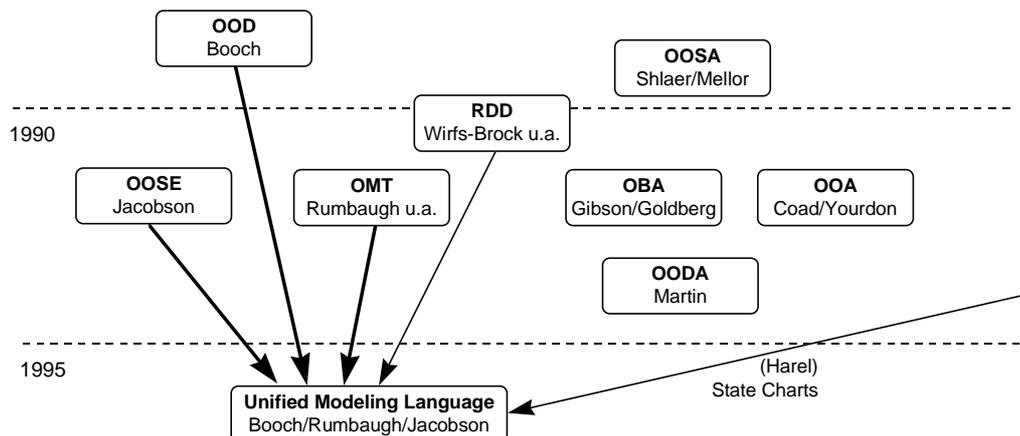


Abbildung 3: Entwicklung objektorientierter Modellierungsmethoden¹⁹

Eine mögliche Strukturierung zur Beschreibung objektorientierter Modellierungsmethoden ist eine Unterteilung in Basismodell, statisches Modell und dynamisches Modell.²⁰

- Im Basismodell werden die Notationen der Instanz, der Klasse, der Operation, des Attributs und des Polymorphismus beschrieben.
- Im statischen Modell sind die Prinzipien Assoziation, Aggregation, Vererbung und Abstraktion dargestellt.
- Im dynamischen Modell werden das Systemverhalten, die Interaktion und die Zustandsänderungen im Zeitverlauf abgebildet.

¹⁷ Vgl. Booch, G.; Rumbaugh, J. Jacobsen, I.: The Unified Method Language for Object-Oriented Developmant, Documentation Set 0.9 Addendum, Relational Software Corporation, Santa Clara 1996.

¹⁸ Vgl. Booch, G.: Object-Oriented Analysis and Design with Applications, 2nd Ed., Redwood City, CA, 1994. Jacobsen, I.; Christerson, M.; Jonsson, P.; Övergaard, G.: Object-Oriented Software-Engineering, A Use Case Driver Approach, Workingham 1992. Rumbaugh, J.; Blaha, M.; Prmerlani, W.; Eddy, F.; Lorensen, W.: Object-Oriented Modelling and Design, Englewood Cliffs 1991. Wirfs-Brock, R.; Wilkerson, B.; Wiener, L.: Objektorientiertes Software-Desing, München 1993. Harel, D.: Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming, 8(1987), S. 231ff.

¹⁹ Oestereich, B.: Objektorientierte Softwareentwicklung: Analyse und Design, 2. Aufl., München et al. 1997, S. 20.

²⁰ Vgl. Oestereich, B.: Objektorientierte Softwareentwicklung: Analyse und Design, 2. Aufl., München et al. 1997. Balzert, H.: Methoden der objektorientierten Systemanalyse, 2. Aufl., Heidelberg et al. 1996.

2.2.1 Basismodell und statisches Modell

Das Konstrukt des Objektes wird in den meisten Ansätzen einfach als Entity oder Sache charakterisiert und über Beispiele verdeutlicht. Rumbaugh definiert das Objekt als eine diskrete, unterscheidbare Entität.²¹ Die Entität als reale Sache wird in dieser Definition dadurch näher charakterisiert, daß es eindeutig von anderen unterscheidbar und eindeutig identifizierbar (diskret) ist. Objekte können Dinge (PKW, Fahrrad, Büro), Personen (Kunde, Mitarbeiter) und Begriffe (Programmiersprache, Krankheit) sein.²²

Im Rahmen der objektorientierten Modellierung gelten die folgenden Darstellungsprinzipien für die Beschreibung eines Objektes: Ein Objekt kapselt die *Daten*, die das Objekt charakterisieren. Diese nennt man *Attribute* oder *Instanzvariablen*. Es kapselt auch die *Operationen*, die das Verhalten des Objekts bestimmen. Sie heißen *Methoden*. Beispielsweise umfaßt ein Auftrag Daten wie Kundeninformationen, Auftragspositionen, Liefervereinbarungen usw. Er umfaßt auch Methoden wie erfassen, bearbeiten oder löschen. Kapseln bedeutet, daß nur das Objekt selbst die Struktur der Datenattribute kennt (information hiding). Auch weiß nur das Objekt, wie die Operationen aussehen, mit denen die Daten transformiert werden. Die Notation der UML bzgl. der Basiskonstrukte ist in Abbildung 4 dargestellt.

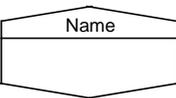
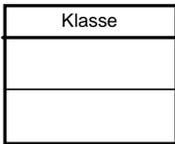
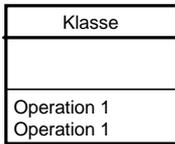
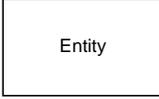
Basismodell				
Konzept	Objekt	Klasse	Attribute/ Instanzvariablen	Operation/ Methode
UML				
ERM	Kein grafisches Symbol: Entspricht einem Eintrag einer Instanz in relationalen Tabellen		 Unterscheidung in Schlüssel-, beschreibendes Attribut Fremdschlüsselattribut	Existiert bei Datenmodellierung nicht

Abbildung 4: Objektorientierte Basiskonstrukte (UML)

Eine *Instanz* eines Objektes ist als langgestrecktes Sechseck dargestellt. Eine *Klasse* ist als Rechteck mit Kopfbezeichnung beschrieben. *Attribute* und *Operationen* werden im Klassenobjekt abgetragen, sind hierbei aber durch eine dünne Linie voneinander optisch

²¹ Vgl. Rumbaugh, J.; Blaha, M.; Prmerlani, W.; Eddy, F.; Lorenson, W.: Object-Oriented Modelling and Design, Englewood Cliffs 1991, S. 1.

²² Vgl. Balzert, H.: Methoden der objektorientierten Systemanalyse, 2. Aufl., Heidelberg et al. 1996, S. 31.

abgetrennt. Da es der Verständlichkeit helfen kann und zeigt, daß die Veränderungen gegenüber den Ursprüngen der objektorientierten Methoden erkennbar sind, ist die Notation der datenorientierten Beschreibungssprache des Entity-Relationship-Modells (ERM) von Chen zum Vergleich abgetragen.

Die Kommunikation zwischen Objekten erfolgt über *Nachrichten* (Botschaften). Hierfür legt das Objekt seine Schnittstelle in Form eines *Methodenprotokolls* fest. Die Schnittstelle beschreibt im Prinzip die Zugriffsmethoden, die nach Außen gegenüber anderen Objekten freigegeben und offengelegt sind. Sie beschreibt außerdem die Variablen, die mit der Zugriffsmethode übergeben werden (*Parameter*).

In der objektorientierten Modellierung wird dieses Grundkonzept der Betrachtungsweise eines Objektes um weitere Prinzipien ergänzt. Rumbaugh nennt vier solcher Aspekte, die zu einem Objekt gehören:²³ Die Identifikation, die Klassifikation, Polymorphismus und Vererbung.

- *Identifikation* besagt, daß ein Objekt klar von einem anderen unterscheidbar und eindeutig identifizierbar ist, selbst wenn alle Attributwerte wie Name oder Größe identisch sind.
- *Klassifikation* meint, daß Objekte mit der gleichen Datenstruktur und den gleichen Methoden zu einer Klasse gruppiert werden können.
- *Polymorphismus* umfaßt den Aspekt, daß eine gleiche Funktion/Operation sich in unterschiedlichen Klassen unterschiedlich verhalten kann.
- *Vererbung* bezeichnet die gemeinsame Verwendung von Attributen und Operationen durch verschiedene Klassen. Die vererbten Eigenschaften und Methoden müssen der Unterklasse nicht mehr explizit hinzugefügt werden, sofern sie in die Oberklasse ausgelagert werden.

Die vier genannten Aspekte führen insgesamt zu folgenden Vorteilen bei der Softwareentwicklung auf Basis von Objekten:²⁴ Natürlichkeit der Realitätsabbildung, Modularität, Änderbarkeit, Erweiterbarkeit, Wiederverwendbarkeit, Verkürzung der Entwicklungszeiten, Sicherheit und Zuverlässigkeit. Die Konstrukte des statischen Modells sind - auch wieder mit Gegenüberstellung zum ERM - in Abbildung 5 dargestellt.

Von einer Oberklasse an die Unterklasse können sowohl Attribute als auch Operationen vererbt werden. In der Unterklasse können dann weitere Attribute oder Operationen hinzugefügt werden. Vererbte Eigenschaften oder Operationen können auch verändert werden. Man spricht von Einfachvererbung, wenn jede Klasse - mit Ausnahme der Wurzel - genau eine Oberklasse besitzt. Man spricht von Mehrfachvererbung (oder multipler Vererbung), wenn eine Unterklasse Attribute oder Operationen von mehreren Oberklassen erben kann. Oberklassen können

²³ Vgl. Rumbaugh, J.; Blaha, M.; Prmerlani, W.; Eddy, F.; Lorenson, W.: Object-Oriented Modelling and Design, Englewood Cliffs 1991, S. 2.

²⁴ Vgl. Heß, H.: Wiederverwendung von Software - Framework für betriebliche Informationssysteme, Wiesbaden, S. 38-44.

abstrakte Klassen sein, d.h. sie sind im Prinzip ein logisches, nicht reales Konstrukt. Sie können, müssen aber nicht vollständig spezifiziert werden. Im letzteren Fall werden sie dann genauer über ihre Unterklassen spezifiziert. Sie werden nur konzipiert, um ihre Eigenschaften oder Operationen zu vererben, nicht aber um softwaretechnisch realisiert zu werden.

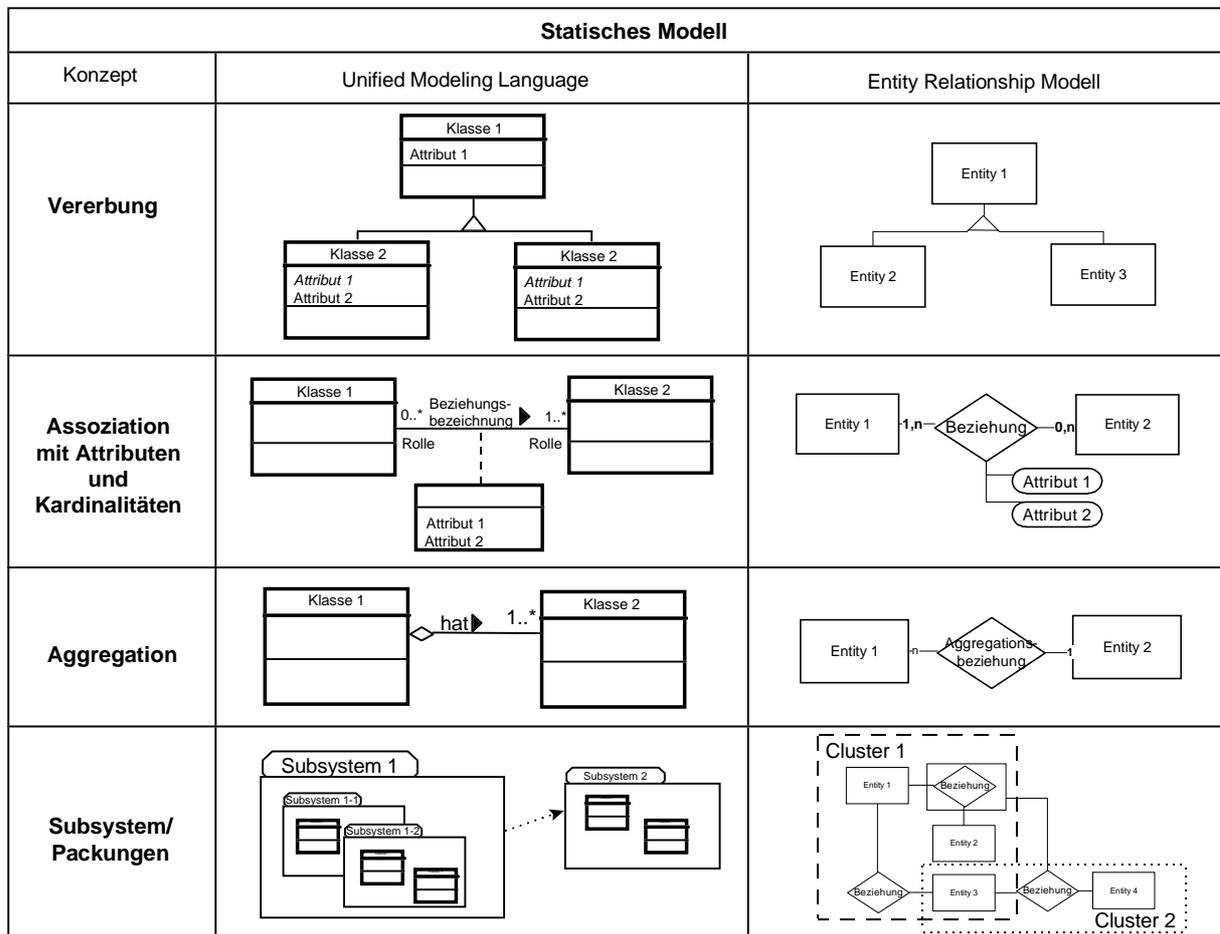


Abbildung 5: Statisches Modell (UML)

Bei einer *Assoziation* werden zwei oder mehrere Objektklassen in Beziehung zueinander gesetzt. Die Assoziation kann eine Bezeichnung haben, die Leserichtung sollte von links nach rechts sein. Beispielsweise kann der Auftragskopf als Assoziation von Kunde, Organisationseinheit und Datum definiert werden, die Auftragsposition als Assoziation wiederum von Auftragskopf und Produkt/Leistung (genauer: Artikel/Leistungskomponente). Es können auch Kardinalitäten und Rollen, wie z.B. Auftraggeber- oder Auftragnehmer-Rollen bei der Darstellung der Beziehung zwischen Kunde und Organisation, dargestellt werden. Diese Angaben sind aber optional.

Die *Aggregation* basiert auf dem Konstrukt der Assoziation. Sie bezeichnet eine stärkere Form der Assoziation im Sinne von „Ist-Teil-von“. Beispiele für Aggregationen sind der Auftrag, der aus dem Auftragskopf und der Auftragsposition besteht, oder die Stückliste bestehend aus den Komponenten eines Produkts oder einer Leistung. Wichtigste Eigenschaft der Aggregation ist die

Transitivität. Wenn Objektklasse A „Teil von“ Objektklasse B ist, dann ist B nicht gleichzeitig „Teil von“ A.

Ein *Subsystem* stellt eine Ansammlung von Modellelementen dar, mit denen das Gesamtsystem in überschaubarere, logische Einheiten gegliedert werden kann. Subsysteme können sich überlappen, d.h. eine Klasse kann in mehreren Subsystemen, aber auch in keinem vorkommen. *Packages* werden ähnlich wie der Begriff Subsystem verwendet, sie unterscheiden sich aber dadurch, daß bei ihnen eindeutige Klassenzuordnungen erfolgen.

2.2.2 Dynamisches Modell

Um zu klären, wie die Objekte des Systems optimal integriert werden, müssen die nachrichtenbasierten Objektinteraktionen modelliert werden. Diese ergeben sich aus Zustandsänderungen der Objekte. Die objektorientierte Modellierung stellt im wesentlichen drei Diagrammtypen zur Verfügung, die für eine Prozeßmodellierung verwendet werden können. Sie sind in Abbildung 6 dargestellt.

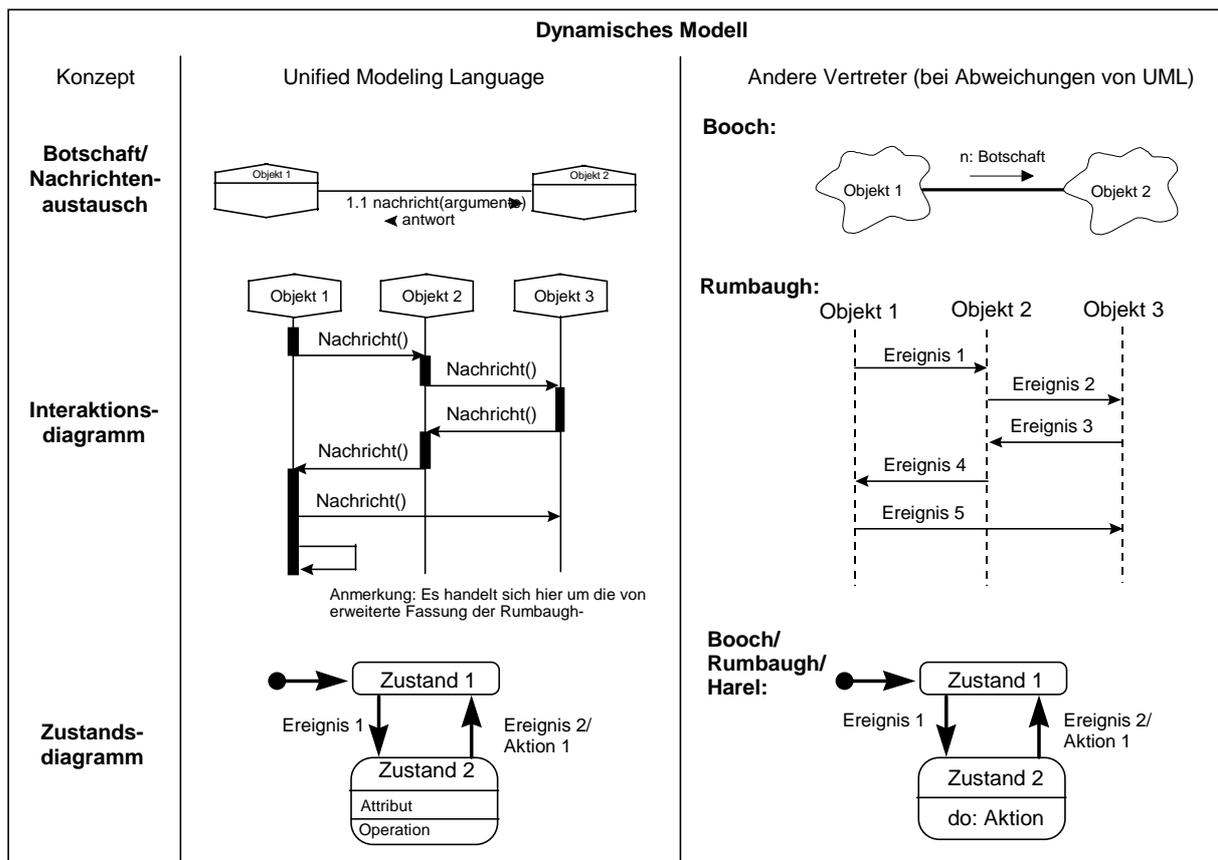


Abbildung 6: Dynamisches Modell (UML)

Botschaftsdiagramme (in UML Kollaborationsdiagramme genannt) stellen den Nachrichtenaustausch zwischen Objekten dar. Hierzu werden im Diagramm die relevanten

Objekte abgetragen und zwischen den miteinander kommunizierenden Objekten eine Kante dargestellt. An der Kante wird die ausgetauschte Nachricht spezifiziert. Ein Pfeil deutet dabei an, in welcher Richtung die Nachricht fließt. Antworten auf eine Anfrage werden ebenfalls ausgewiesen. Auch wird über die Angabe einer Nummer für eine Nachricht dargestellt, in welcher zeitlichen Abfolge die Informationen fließen.

Zur detaillierteren Spezifikation des Nachrichtenflusses stellen Botschaftsdiagramme zahlreiche Notationen zur Verfügung. So kann dargestellt werden, daß bestimmte Nachrichten Informationen über einzelne Objekte mitverschicken. Diese Angaben werden dann in einer Klammer hinter der Bezeichnung der Nachricht angegeben. Informationen an den Kanten der Nachricht deuten an, welche Art der Verbindung zwischen den Objekten über die Nachricht entsteht. A stellt beispielsweise eine Assoziations- oder Aggregationsverbindung dar. P bezeichnet eine Objektverbindungen, bei denen das empfangende Objekt ein Parameter einer Operation ist.

Abbildung 7 stellt ein Beispiel für ein Botschafts-/Kollaborationsdiagramm dar. Es wird gezeigt, wie die Reservierung eines Artikels auf Basis einer Bestellung und Bestellposition im Lager erfolgt.

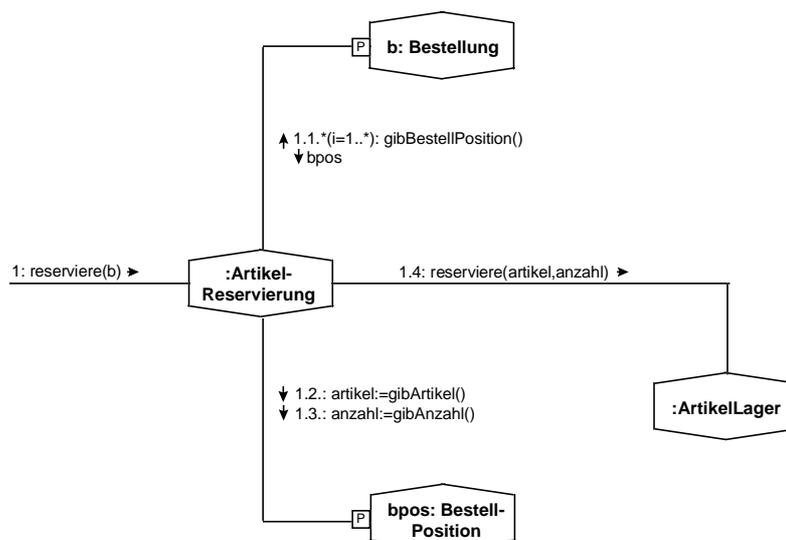


Abbildung 7: Beispiel eines Botschafts-/Kollaborationsdiagramms (UML)²⁵

Man erkennt, daß bei Botschafts- oder Kollaborationsdiagrammen immer nur ein begrenzter Ausschnitt des Systems dargestellt werden kann. Umfassende Abbildungen erscheinen unübersichtlich. Weiterhin wird nicht zwischen betriebswirtschaftlich sinnvollen Kommunikationsprozessen und rein systemtechnisch bedingten Prozessen differenziert. Insofern

²⁵ Oestereich, B.: Objektorientierte Softwareentwicklung: Analyse und Design, 2. Aufl., München et al. 1997, S. 223.

werden Nachrichten dargestellt, die implementierungstechnisch bedingt sind, betriebswirtschaftlich-organisatorisch aber keinen relevanten Sachverhalt darstellen.

Vor dem Hintergrund der eher technischen Ausrichtung werden die Botschaftsdiagramme heute zur Darstellung von betriebswirtschaftlich orientierten Abläufen durch *Interaktionsdiagramme* ersetzt. Auch werden mit Interaktionsdiagrammen häufig die Use-Cases beschrieben. Der Schwerpunkt der Interaktionsdiagramme liegt auf der Darstellung des Nachrichtenaustauschs zwischen Objekten im Zeitverlauf. Abbildung 8 zeigt ein Beispielszenario für ein Interaktionsdiagramm des Verkaufsprozesses.

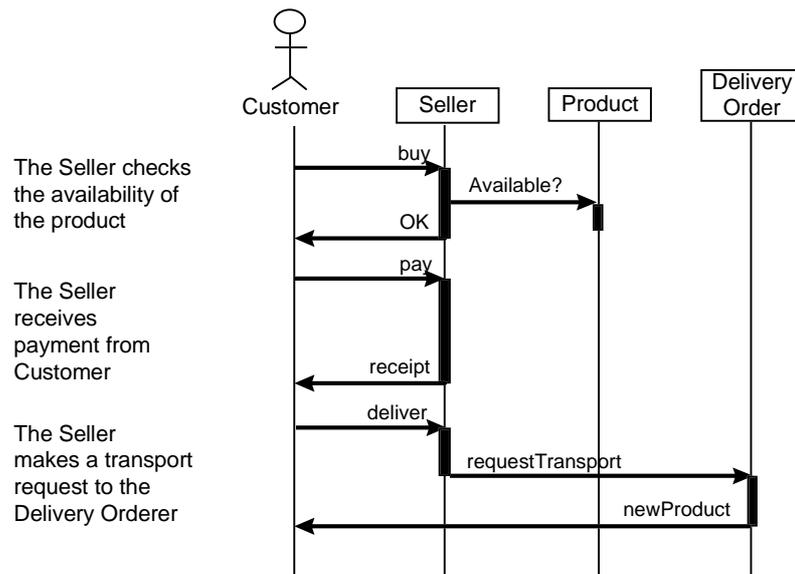


Abbildung 8: Beispiel für ein Interaktionsdiagramm (UML)²⁶

Interaktionsdiagramme sind übersichtlicher und verständlicher als Kollaborationsdiagramme. Die Art der Darstellungsweise erfordert aber für eine betriebswirtschaftlich verständliche Ausdrucksweise textuelle Erläuterungen. Das Fehlen von Entscheidungsoperatoren im Diagramm erzwingt es außerdem, daß jedes Ereignisszenario als eigenes Interaktionsdiagramm entwickelt wird. Zusammenhänge über mehrere Diagramme sind nicht darstellbar. Es fehlen desweiteren formale Regeln, wie die Kante als Nachricht verständlich und sinnvoll benannt werden kann.

Dritte zentrale Modellform zur Darstellung des Systemverhaltens ist das *Zustandsdiagramm*. Hierbei wird sich auf die Beschreibung eines einzelnen Objektes und seines Lebenszyklus konzentriert. Es wird das Verhalten des Objekts während seiner Lebensdauer dargestellt.

Somit ist ein Zustandsdiagramm eine Beschreibung eines Zustandsautomaten. Alle Objekte einer Klasse besitzen dabei denselben Zustandsautomaten, jedoch hat jede Objektinstanz einen eigenen Zustand. Zustandsdiagramme verfolgen das Ziel, die Zustände eines Objektes und seine

²⁶ Jacobson, I.; Ericsson, M.; Jacobson, A.: The Object Advantage - Business Process Reengineering with Object Technology, Workingham et al. 1995, S. 193.

Zustandsübergänge darzustellen. Ein Zustand beschreibt hier eine Menge von Attributausprägungen eines Objektes. Ein Zustandsübergang wird durch ein Ereignis ausgelöst. Der Übergang selbst wird als Transition bezeichnet.

Die Transition besteht solange, bis ein neues Ereignis eintritt, also eine bestimmte Attributmenge so verändert ist, daß sie aus abstrahierter Sicht für das Verhalten des Objektes weiter relevant ist. Insofern wird nicht jede Attributänderung, die implizit auch eine Zustandsänderung darstellt, als eine solche im Diagramm ausgewiesen.

Im Rahmen der Transition werden Aktionen bzw. Aktivitäten ausgeführt. Diese werden unterhalb der Zustandsänderung angeführt. Zustandsdiagramme haben einen Start- und einen Endzeitpunkt, die durch Kreise dargestellt werden (voller Kreis = Startereignis; doppelt umrandeter Kreis = Endereignis). Abbildung 9 stellt ein Beispiel für ein Zustandsdiagramm des Objektes „Produkt“ dar.

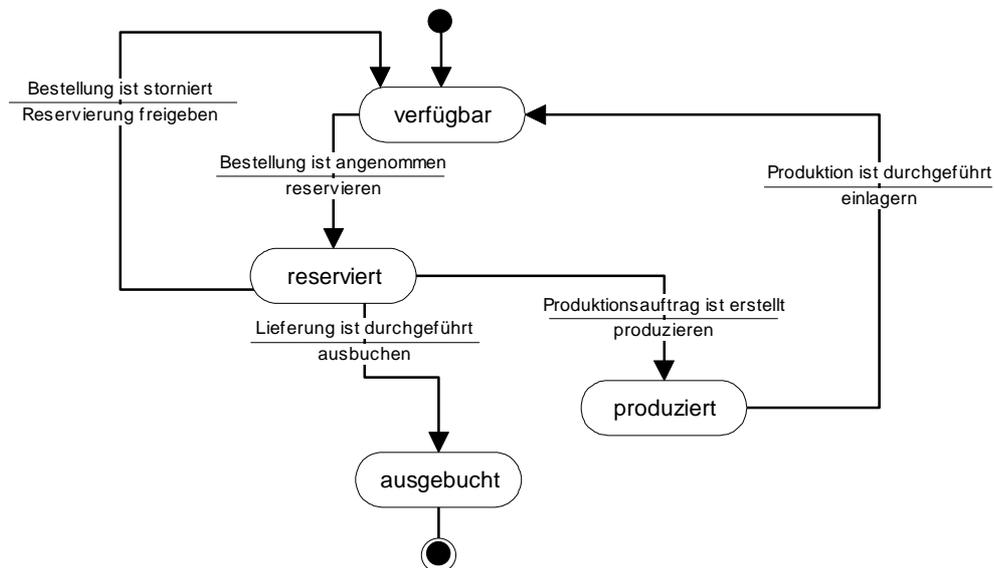


Abbildung 9: Beispiel für ein Zustandsdiagramm des Objekts "Produkt" (UML)

Zustandsdiagramme eignen sich besonders gut, um den Zusammenhang zwischen Zuständen und Ereignissen darzustellen. Aufgrund der Konzentration auf den Lebenszyklus eines einzigen Objektes zeigen sie aber keinen Prozeßzusammenhang und eignen sich damit nicht zur Darstellung eines Geschäftsprozesses.

3 Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK)

Der nachfolgend beschriebene Ansatz basiert auf der Methode der Ereignisgesteuerten Prozeßkette (EPK) und erweitert diese um objektorientierte Konzepte. Hierzu wird die EPK im Sinne des objektorientierten Paradigmas zur „objektorientierten Ereignisgesteuerten Prozeßkette (oEPK)“ weiterentwickelt. Damit werden insbesondere folgende Ziele verfolgt:

- In objektorientierten, komponentenbasierten Informationssystemen (Componentware) werden Geschäftsprozesse realisiert, indem Objekte ereignisgesteuert über Nachrichten miteinander interagieren. Diese Sichtweise sollte auf betriebswirtschaftliche Fragestellungen bei der prozeßorientierten Fachkonzeptanalyse übertragen werden. Dadurch kann der Übergang von der Analyse und dem Design hin zur Objektsteuerung in Workflow-Systemen und in komponentenbasierten Informationssystemen vereinfacht werden.
- Die Betrachtung der Leistung als Gegenstand des Informationssystems erfordert aus betriebswirtschaftlicher Sicht die integrierte Beschreibung von Prozessen (Verrichtungsaspekt der Leistung) und ihren Produkten/Leistungsergebnissen (Objektaspekt der Leistung). Beide Aspekte sollen in einer Modellierungsmethode auf Fachkonzeptebene integriert sein.

Nachfolgend werden die relevanten Beschreibungskonstrukte aufgeführt und ein Anwendungsbeispiel entwickelt. Hierbei werden neben den objektorientierten UML-Konstrukten (Objektbezug) auch sichtenorientierte Konstrukte zur Modellierung des betriebswirtschaftlichen Kontextes (z.B. Organisationseinheiten, Betriebsmittel etc.) eingeführt (Subjektbezug).

3.1 Modellierungsmethode

Ein Geschäftsprozeß wird im folgenden als eine ereignisgesteuerte Bearbeitung von Geschäftsobjekten mit dem Ziel der Leistungserstellung definiert. Ein Geschäftsobjekt (Objektklasse) ist eine betriebswirtschaftliche Leistung, welche zweckbezogen die zur Bearbeitung relevanten Funktionen (Methoden) und Daten (Instanzvariablen) kapselt. Die Steuerung des Kontroll- und Datenflusses zwischen den Geschäftsobjekten erfolgt über das Konzept der Ereignissteuerung (Nachrichtenaustausch).

3.1.1 Modellierung von Geschäftsobjekten

Daten können gemäß des objektorientierten Paradigmas nur über den Aufruf des Objektes und dessen Methoden erstellt, gelesen oder verändert werden. Beispielsweise kann ein Auftrag als Instanz nur über die Objektklasse Auftrag angelegt werden. Das Objekt legt sein Methodenprotokoll, also die Art und Weise, wie die Methode aufgerufen wird, gegenüber anderen Objekten offen. Andere Objekte können über das Methodenprotokoll erkennen, wie das

Objekt anzusprechen ist und welche Dienste das Objekt anbietet. Die Form der internen Implementierung der Methode ist dabei gegenüber anderen Objekten unsichtbar (information hiding).

Ein Objekt kann Methoden enthalten, die nicht im Methodenprotokoll offengelegt sind. Man spricht dann von „privaten“ Methoden. „Auftragsliste anzeigen“ kann beispielsweise eine solche private Methode sein, da sie kein definiertes Ergebnis gegenüber anderen Objekten liefert.

Leitet man aus diesen Sachverhalten die Struktur einer Geschäftsobjektklasse ab, so ergibt sich das in Abbildung 10 gezeigte Modell. Als Symbol für eine Objektklasse wird ein Rechteck mit Kopfteil verwendet. Im Kopfteil wird der Klassenname abgetragen. Um die im Objekt zusammengefaßte Daten- und Funktionssicht darzustellen, werden die Instanzvariablen links und die Methoden rechts vom Geschäftsobjekt abgetragen.

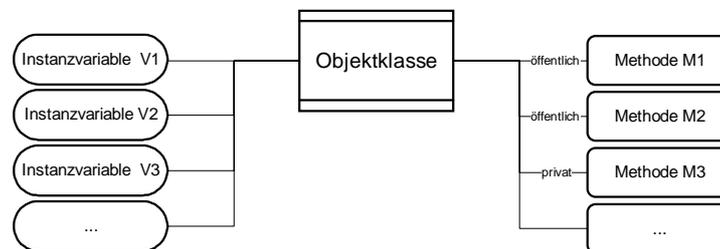


Abbildung 10: Modell eines oEPK-Geschäftsobjektes

Der Nachrichtenaustausch und damit die Interaktion zwischen Geschäftsobjekten erfolgt ereignisgesteuert. Da diese Ereignisse Nachrichten darstellen und es sich somit aus Meta-Sicht um Objekte handelt, werden sie als eigenes Symbol repräsentiert. Dies ist ein wesentlicher Unterschied zur Darstellung objektorientierter Interaktionsdiagramme, in denen Ereignisse nicht explizit dargestellt, sondern an Kanten zwischen zwei Objekten abgetragen werden. Dadurch gehen - ähnlich wie bei Datenflußdarstellungen - wichtige semantische Inhalte zur Beschreibung des dynamischen Systemverhaltens verloren.

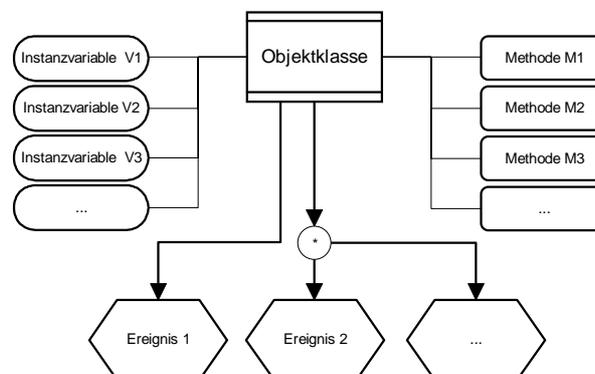


Abbildung 11: Modell eines oEPK-Geschäftsobjektes mit Ereignissen

In Abbildung 11 ist das Modell eines oEPK-Geschäftsobjektes mit Ereignissen dargestellt. Da ein Objekt gleichzeitig mehrere Nachrichten mit unterschiedlichem Inhalt - basierend auf einem einzigen Zustand - austauschen kann, wird ein Verknüpfungsoperator (*) oder Konnektor eingeführt. Dieser Konnektor enthält Konditionen (Mechanismen und Regeln) für die Wahl der entsprechenden Ereignisse.

3.1.2 Modellierung von Geschäftsprozessen

Mit der Beschreibung der Methoden, Attribute und Zustandsänderungen, die für andere Objekte relevant sind, ist das Modell eines Geschäftsobjektes und damit zunächst dessen prozeßunabhängiges Verhalten definiert. Die Interaktion zwischen Objekten erfolgt über einen Nachrichtenaustausch.

Es können zwei Typen von Kontrollfluß-Nachrichten unterschieden werden:

- a) **Ereignisgesteuerte Nachrichten:** Sie beschreiben den Kontrollfluß und damit die betriebswirtschaftliche Entscheidungslogik. Die Nachricht enthält Informationen über die betriebswirtschaftlich relevanten Zustandsänderungen (Ereignisse). Bildet man den durch ereignisgesteuerte Nachrichten definierten Kontrollfluß graphisch ab, so ergibt sich die in Abbildung 12 entwickelte Grundstruktur eines Geschäftsprozeßmodells.

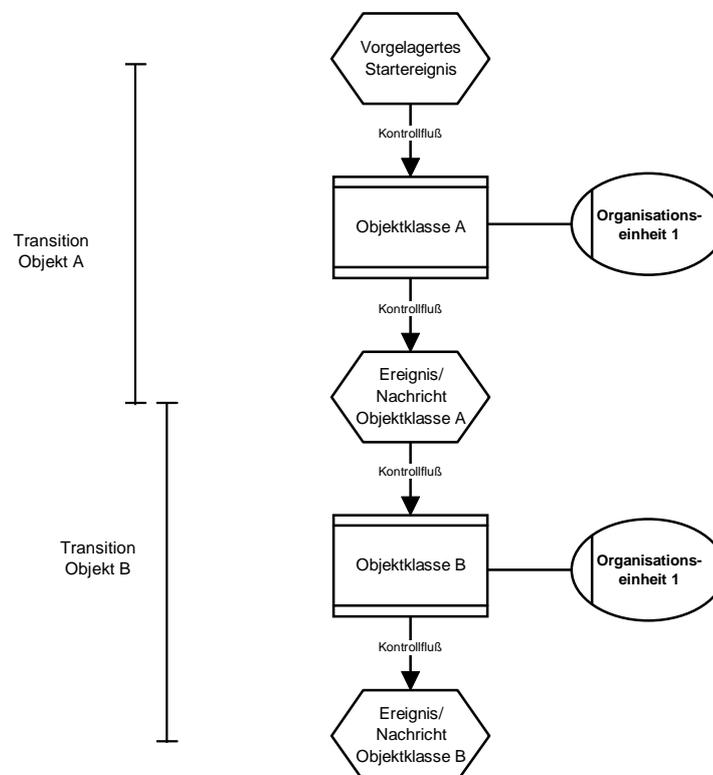


Abbildung 12: Grundmodell einer objektorientierten Ereignisgesteuerten Prozeßkette (oEPK)

Als ein wesentlicher Aspekt der Geschäftsprozeßmodellierung erfolgt hierbei die Zuordnung von Organisationseinheiten bzw. Aufgabenträgern zu den Geschäftsobjekten. Dadurch können im Kontext des jeweiligen Bearbeitungsstatus die relevanten Organisationseinheiten modelliert und deren Rolle im Prozeß abgebildet werden.

- b) **Auftrags-/Leistungsgesteuerte Nachrichten:** Sie beschreiben eine Auftraggeber-Auftragnehmerbeziehung. Ein sendendes Objekt verlangt von einem empfangenden Objekt eine Leistungserbringung, die das sendende Objekt zur weiteren Abwicklung seiner Transitionen benötigt. Abbildung 13 zeigt zwei alternative Darstellungsmöglichkeiten für Auftrags-/Leistungsgesteuerte Beziehungen. Sie sind besonders zur Darstellung der Beziehungen zu assoziierten Objekten im Prozeßverlauf relevant. Ihre Betrachtung steht aber bei der Geschäftsprozeßanalyse und Workflow-Steuerung nicht im Vordergrund.

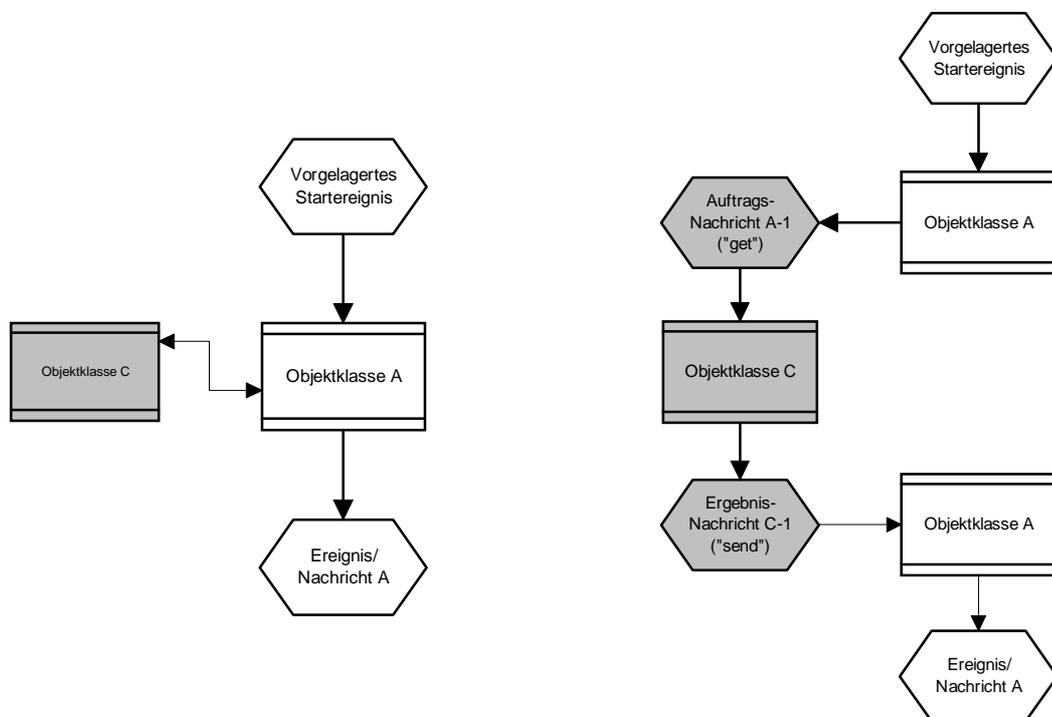


Abbildung 13: Auftrags-/Leistungsgesteuerte Beziehungen als gekapselter Kontrollfluß

Wie bereits bei der Beschreibung des Modells eines Geschäftsobjektes dargestellt, können mehrere Ereignisse aus einer Objektzustandsänderung resultieren. Analog kann es sein, daß ein Objekt erst dann eine Funktion/Methode ausführt, wenn mehrere Ereignisse eingetreten sind. Um diese und andere ähnlich gelagerten Zusammenhänge abzubilden, werden Verknüpfungsoperatoren bzw. Konnektoren in die Methode eingeführt. Abbildung 14 stellt in Anlehnung an die Methode der EPK die zulässigen Möglichkeiten der Verknüpfung von Objekten über Ereignisse und Verknüpfungsoperatoren dar.

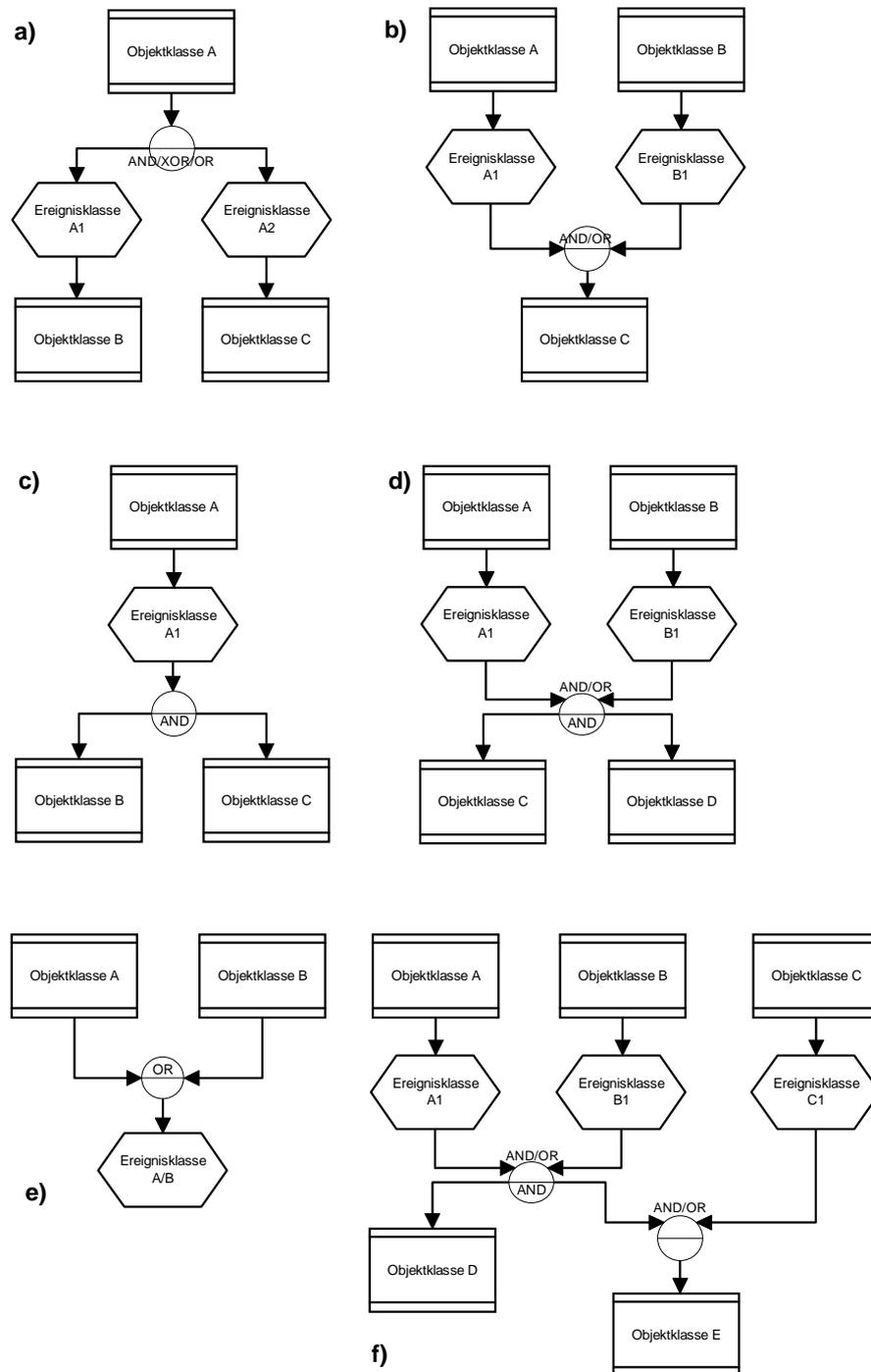


Abbildung 14: oEPK-Geschäftsprozessmodell (zulässige Verknüpfungen)

In Abbildung 14a werden zwei Objektclassen von einer Ereignisklasse ausgelöst. Als Operatoren sind möglich: AND, OR, XOR. Die Objektclass verfügt über die entsprechende Entscheidungskompetenz. In Abbildung 14b) lösen zwei Ereignisklassen eine Objektclass aus. Bei einer AND-Verknüpfung wartet die Objektclass bei Eintritt von einer der beiden Ereignisklassen solange, bis die noch fehlende Ereignisklasse eingetreten ist. Erst dann startet die Objektclass ihre Aktion. Bei einer OR-Verknüpfung reagiert die Objektclass, sobald eine der beiden

Ereignisklassen eintritt. Treten beide Ereignisklassen gleichzeitig ein, so feuert die Objektklasse zweimal. Eine XOR-Verknüpfung ist nicht möglich, da eine Objektklasse hinsichtlich der vorgelagerten Ereignisklasse nicht über Informationen verfügen kann, ob bei Eintritt eines konkreten Ereignisses noch ein weiteres Ereignis nachfolgt, um sich dann für eines der beiden zu entscheiden. Abbildung 14c) stellt dar, daß eine Ereignisklasse auch zwei Objektklassen gleichzeitig auslösen kann. Entscheidend ist hierbei, daß nur eine AND-Verknüpfung möglich ist. Die Ereignisklasse verfügt nicht über eine Entscheidungskompetenz. Abbildung 14d) zeigt, daß auch Verknüpfungen mehrerer Ereignisklassen mit mehreren Objektklassen erlaubt sind. Allerdings nur im Rahmen der genannten Restriktionen aus Abbildung 14a) bis Abbildung 14c). Entweder kann dann eine Ereignisklasse beide Objektklassen auslösen oder beide Ereignisklassen zusammen beide Objektklassen. Abbildung 14e) zeigt, daß zwei verschiedene Objektklassen die gleiche Ereignisklasse erzeugen können. Auch Abbildung 14f) ist möglich. Angenommen alle Operatoren stehen auf AND, so wird Objektklasse E nur ausgelöst, wenn alle drei Ereignisklassen eintreten. Steht der Operator zwischen A1 und B1 auf OR, der andere auf AND, so wird Objektklasse E ausgelöst, wenn entweder Ereignisklasse A1 oder B1 eintreten. Stehen alle oberen Operator auf OR, so wird Objektklasse E bei Eintritt eines der drei Ereignisklassen ausgelöst. Gleiches gilt für Objektklasse D.

Abbildung 15 stellt die unzulässigen Möglichkeiten der Verknüpfung von Objekten über Ereignisse und Verknüpfungsoperatoren dar. Abbildung 15a) stellt den oben bereits unter Fall Abbildung 15b) dargestellten unzulässigen XOR-Fall dar. Abbildung 15b) zeigt, daß es unzulässig ist, daß sich zwei Objektklassen hinsichtlich eines gemeinsamen Ereignisses „einigen“ (AND oder XOR sind nicht zulässig). Sie können allenfalls das gleiche Ereignis erzeugen. Semantisch kann alternativ jeder Objektklasse eine eigene, gleichnamige Ereignisklasse zugeordnet werden. Auch Abbildung 15c) ist unzulässig, da eine Ereignisklasse nicht über eine eigenständige Entscheidungskompetenz verfügt. Hier ist nur die UND-Verknüpfung zulässig (vgl. Fall c) in Abbildung 14).

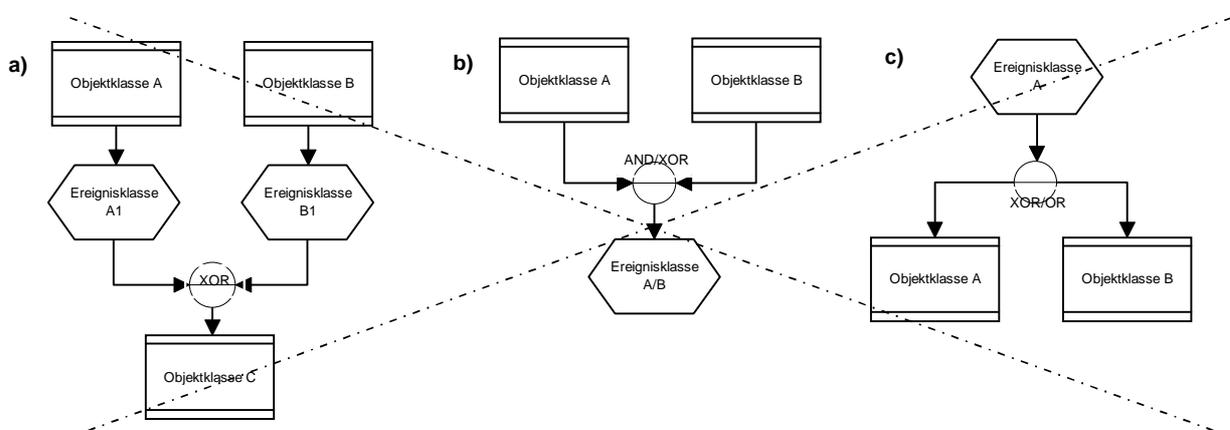


Abbildung 15: oEPK-Geschäftsprozeßmodell (unzulässige Verknüpfungsmöglichkeiten)

3.2 Anwendungsbeispiel

Abbildung 16 stellt anhand der entwickelten Methode der objektorientierten Ereignisgesteuerten Prozeßkette (oEPK) ein Szenario am Beispiel der Auftragserfassung dar. Auf Basis der vorangestellten Ausführungen kann auf eine Erläuterung verzichtet werden, da das Modell weitgehend selbsterklärend ist.

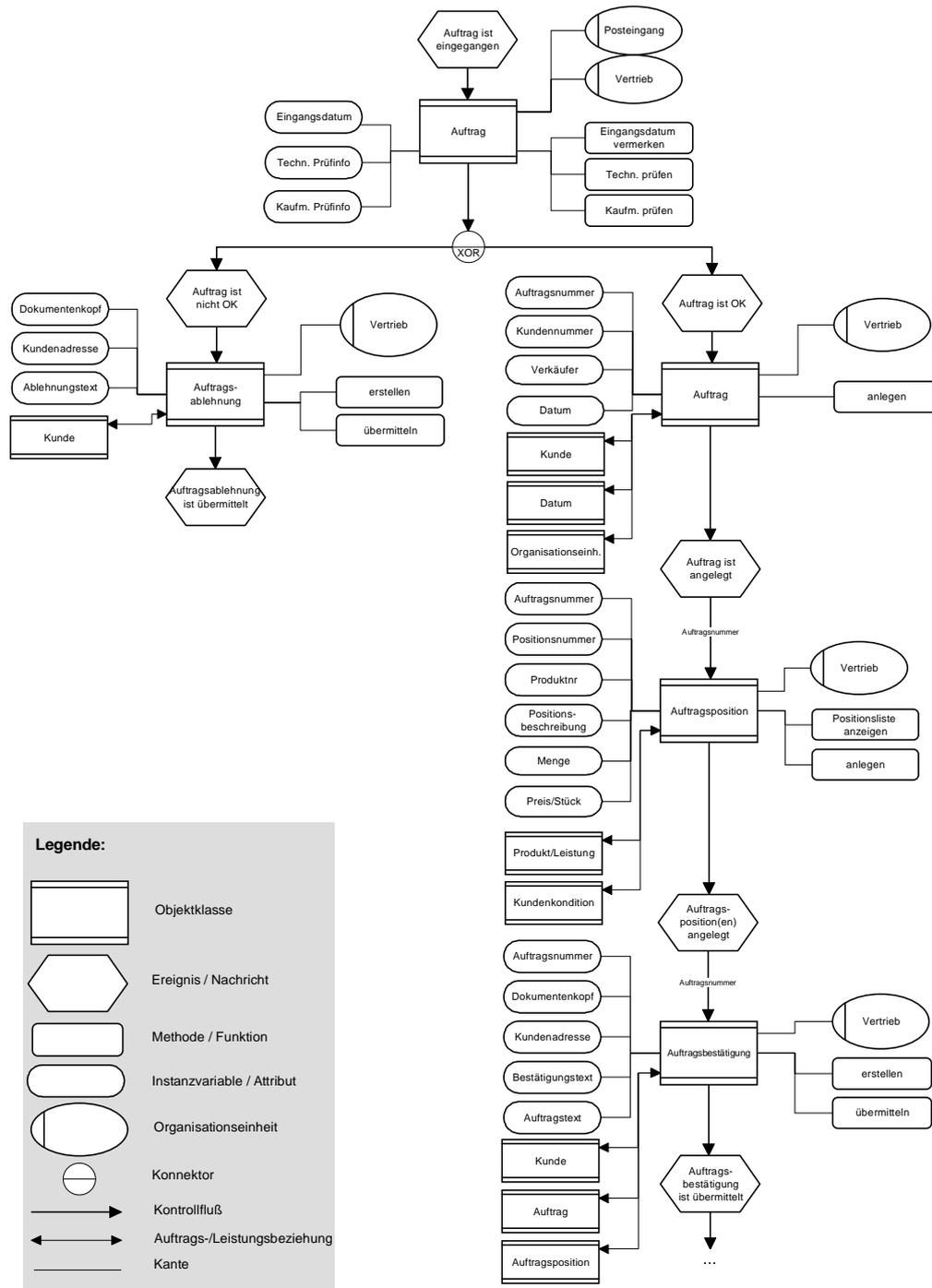


Abbildung 16: oEPK-Anwendungsszenario (Beispiel Auftragserfassung)

4 Ausblick

Der dargestellten Methode liegt eine Kombination sowohl strukturierter (sichtenorientierter) als auch objektorientierter Konzepte zugrunde. Diese Auffassung verändert die bislang eher dogmatische Methodendiskussion hin zu einer Integrationsdiskussion und wird auch zunehmend in der Literatur vertreten.

Die weitere wissenschaftliche Arbeit im Kontext der vorgestellten objektorientierten Ereignisgesteuerten Prozeßkette (oEPK) wird sich primär auf Fragestellungen der differenzierten Beschreibung der oEPK-Konstrukte, der Entwicklung eines konsistenten Vorgehensmodells und der Abbildung mittels Modellierungswerkzeugen befassen. Desweiteren werden Referenzmodelle zur integrierten Produkt/Leistungs- und Prozeßmodellierung und der Aspekt der Wiederverwendbarkeit von Geschäftsmodellen ein zentraler Gegenstand der Forschungsarbeiten sein.

5 Literatur

- Balzert, H.: Methoden der objektorientierten Systemanalyse, 2. Aufl., Heidelberg et al. 1996.
- Becker, J.: Strukturanalogien in Informationsmodellen. Ihre Definition, ihr Nutzen und ihr Einfluß auf die Bildung von Grundsätzen ordnungsgemäßer Modellierung (GoM), in: König, W. (Hrsg.): Wirtschaftsinformatik '95. Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit, Heidelberg 1995, S. 133-150.
- Bungert, W.; Heß, H.: Objektorientierte Geschäftsprozeßmodellierung, in: Information Management, 10(1995)1, S. 52-63.
- Booch, G.: Object-Oriented Analysis and Design with Applications, 2nd Ed., Redwood City, CA, 1994.
- Booch, G.; Rumbaugh, J. Jacobsen, I.: The Unified Method Language for Object-Oriented Developmant, Documentation Set 0.9 Addendum, Relational Software Corporation, Santa Clara 1996.
- Davenport, T.H.: Process Innovation. Reengineering Work through Information Technology, Boston 1993.
- Ferstl, O.K.; Sinz, E.J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen, in: Wirtschaftsinformatik 37(1995)3, S. 209-220.
- Gaitanides, M. Prozeßorganisation: Entwicklung, Ansätze und Programme prozeßorientierter Organisationsgestaltung, München 1983.
- Hammer, M.; Champy, J.: Reengineering the Corporation. A Manifesto for Business Revolution, New York 1993.
- Hars, A.: Referenzdatenmodelle - Gestaltung und Nutzung von Bibliotheken für semantische Datenmodelle, Wiesbaden 1994.
- Heß, H.: Wiederverwendung von Software - Framework für betriebliche Informationssysteme, Wiesbaden 1993.

- Hess, T.: Entwurf betrieblicher Prozesse: Grundlagen - Bestehende Methoden - Neue Ansätze, Wiesbaden 1996.
- Jacobson, I.; Christerson, M.; Jonsson, P.; Övergaard, G.: Object-Oriented Software-Engineering, A Use Case Driver Approach, Workingham 1992.
- Jacobson, I.; Ericsson, M.; Jacobson, A.: The Object Advantage - Business Process Reengineering with Object Technology, Workingham et al. 1995.
- Knolmayer, G.; Herbst, H.: Business Rules, in: Wirtschaftsinformatik, 35(1993)4, S. 386-390.
- Keller, G.: Informationsmanagement in objektorientierten Organisationsstrukturen, Wiesbaden 1993.
- Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“, in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992.
- Keller, G.; Nüttgens, M.; Scheer, A.-W.: Planungsinseln: Vom Konzept zum integrierten Informationsmodell, in: Handbuch der modernen Datenverarbeitung (HMD) - Theorie und Praxis der Wirtschaftsinformatik, 29(1992)168, S. 25-39.
- Kruse: Referenzmodellgestütztes Geschäftsprozeßmanagement: Ein Ansatz zur prozeßorientierten Gestaltung vertriebslogistischer Systeme, Wiesbaden 1996.
- Nüttgens, M.: Dezentral-koordiniertes Informationsmanagement. Rahmenkonzept - Koordinationsmodelle - Werkzeug-Shell, Wiesbaden 1995.
- Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen, Stuttgart et al. 1996.
- Österle, H.: Business Engineering. Prozeß- und Systementwicklung. Band 1: Entwurfstechniken, Berlin 1995.
- Oestereich, B.: Objektorientierte Softwareentwicklung: Analyse und Design, 2. Aufl., München et al. 1997.
- Olle, T.W.; Hagelstein, J.; Macdonald, I.G.; Rolland, C.; Sol, H.G.; Van Assche, F.J.M.; Verrijn-Stuart, A.A.: Information Systems Methodologies: a framework for understanding, 2. Aufl., Workingham et al., 1991.
- Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden 1996.
- Rumbaugh, J.; Blaha, M.; Prmerlani, W.; Eddy, F.; Lorensen, W.: Object-Oriented Modelling and Design, Englewood Cliffs 1991.
- Scheer, A.-W.: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2. Aufl., Berlin et al. 1992.
- Scheer, A.-W.: Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 6. Aufl., Berlin et al. 1995.
- Wirfs-Brock, R.; Wilkerson, B.; Wiener, L.: Objektorientiertes Software-Desing, München 1993.