

3 Analyse und Einführung betrieblicher Informationssysteme

3.1 Grundlagen der Systemanalyse

In diesem Kapitel lernen Sie

- die Bestandteile eines *Systems*,
- die Bedeutung des *Systemansatzes* für die Wirtschaftsinformatik und
- die Vorteile einer systemorientierten Betrachtung von Unternehmen kennen.

Unter dem Begriff *Systemanalyse* werden Aktivitäten zusammengefasst, die bei der Konzeption, Entwicklung und Einführung betrieblicher Informationssysteme zu durchlaufen sind. Die theoretische Grundlage dafür bildet der sog. *Systemansatz*.

Ein System wird definiert als eine Menge von Elementen, die zur Erreichung eines bestimmten Ziels interagieren.

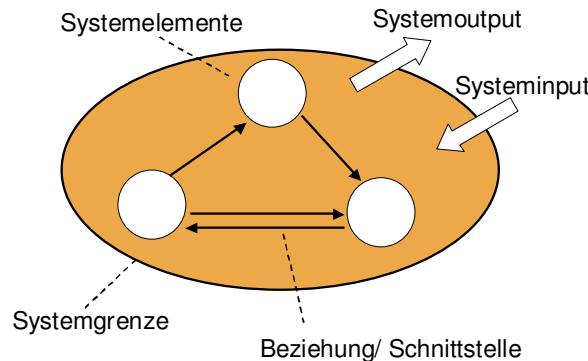


Abb. 3-1: Merkmale eines Systems

Aus Gründen der Zweckmäßigkeit oder zur Komplexitätsreduktion kann ein System in *Subsysteme* unterteilt werden. Unter (atomaren) *Elementen* werden nicht weiter zu zerlegende Bestandteile eines Systems verstanden. *Beziehungen* (Schnittstellen) sind Verbindungen zwischen den Elementen bzw. Subsystemen, die das Verhalten des gesamten Systems beeinflussen. Ein System kommuniziert bzw. interagiert mit seiner Umwelt (*Systeminput*/ -*output*), ist aber von dieser klar abgegrenzt. Die Systemstruktur wird bestimmt durch die Menge der Elemente sowie die Menge und Art der Beziehungen zu einem bestimmten Zeitpunkt.⁶⁵

⁶⁵ Vgl. Krallmann, H., Frank, H.; Gronau, N. (1999), S. 21; Schwarze, G. (2000), S. 127 f.

Warum ist nun in der Wirtschaftsinformatik eine solche systemorientierte Betrachtung von Bedeutung?

Häufig geht es darum, bestimmte Sachverhalte, Organisationen, Programme etc. zu analysieren, auf Schwachstellen zu untersuchen und Verbesserungen bzw. Veränderungen herbeizuführen. Aus den Erkenntnissen der allgemeinen Systemtheorie können dafür wertvolle Anregungen bzw. Hilfestellungen gewonnen werden. Wesentliche Vorteile sind dabei:

- *Problemstrukturierung*: Eine Betrachtung des jeweiligen Untersuchungsbereiches als System zwingt i. d. R. zu einer strukturierten Vorgehensweise.
- *Problemadäquate Abstraktionsebene*: Durch die Bildung von Subsystemen besteht die Möglichkeit von „Black-Box-Betrachtungen“, d. h. Nebensächliches kann ausgeblendet werden, und es findet eine Konzentration auf das Wesentliche statt.
- *Systemabgrenzung*: Eine systemorientierte Betrachtungsweise setzt stets eine eindeutige Abgrenzung des jeweiligen Untersuchungsbereiches voraus. Diese Kernaktivität sollte zu Beginn eines Analyseprojektes durchgeführt werden, um einen zielgerichteten Projektablauf zu gewährleisten.

Was letztlich als System, Systemelement, Subsystem bzw. Umwelt zu verstehen ist, wird durch den jeweiligen Analysezweck bestimmt. So ist beispielsweise denkbar, dass bei der Einführung eines unternehmensweiten Informationssystems das gesamte Unternehmen betrachtet wird, die einzelnen Unternehmensbereiche somit Subsysteme oder Elemente darstellen, dagegen bei der Reorganisation einer einzelnen Abteilung allein diese als System mit den Mitarbeitern als Systemelementen aufzufassen ist.

Mit der Systemanalyse sind zwei wesentliche Zielsetzungen verbunden:

- *Erklärungsaspekt*: Zielsetzung ist hier die Untersuchung bzw. Abbildung der Struktur und des Verhaltens bestehender Systeme. Dabei erfolgt eine Konzentration auf die relevanten Bereiche des betrachteten Systems. Ein Beispiel dafür ist die Ist-Analyse in der Vertriebsabteilung eines Unternehmens als Vorbereitung für die Entwicklung eines Auftragsabwicklungssystems.
- *Gestaltungsaspekt*: Hier wird die Veränderung bestehender bzw. die Entwicklung neuer Systeme angestrebt. Ein Beispiel dafür ist die Soll-Konzeption oder Realisierung des vorgenannten Auftragsabwicklungssystems.

Ein wesentliches Hilfsmittel der Systemanalyse sind Modelle, die dazu verwendet werden, komplexe Systeme abzubilden bzw. vereinfacht darzustellen und damit erst analysierbar zu machen. Grundlegende Modellierungsmethoden werden daher nachfolgend näher erläutert.

Wiederholungsfragen

1. Erläutern Sie die wesentlichen Kennzeichen eines *Systems*.
2. Welche Vorteile ergeben sich aus einer systemorientierten Betrachtungsweise für Aufgabenstellungen der Wirtschaftsinformatik?
3. Stellen Sie zur Vorbereitung der Einführung eines unternehmensweiten Informationssystems ein Unternehmen exemplarisch als System dar. Identifizieren Sie geeignete Subsysteme und deren Beziehungen. (Lösung s. Anhang).

3.2 Modelle

In diesem Kapitel lernen Sie

- die Bedeutung von *Modellen* im Rahmen der Systemanalyse,
 - grundlegende Aspekte zur Datenmodellierung,
 - Verfahren zur Modellierung von Geschäftsprozessen und
 - die Unified Modelling Language (UML) als objektorientierten Modellierungsansatz kennen.
-

Modelle stellen ein vereinfachtes Abbild der Realität dar und werden verwendet, um Systeme bzw. Sachverhalte zu beschreiben bzw. darzustellen.⁶⁶ Auf der Grundlage von Modellen sind Simulationen möglich, so dass Erkenntnisse gewonnen werden können, die über eine reine Systembeobachtung nicht erlangt werden können. Weitere Vorteile von Modellen sind:

- *Transparenz*: Modelle können dazu dienen, Zusammenhänge aufzuzeigen und sorgen somit für eine größere Transparenz komplexer Sachverhalte.
- *Komplexitätsreduktion*: Durch die Verwendung von Modellen wird ein tiefergehendes Verständnis für den Aufbau betrieblicher Systeme geschaffen, da die Komplexität des realen Systems reduziert wird.
- *Problembezug*: Durch Modellierungsaktivitäten wird die Konzentration auf die jeweils wesentlichen Aspekte gerichtet.
- *Mitarbeitereinbindung*: Vor allem grafische Modellierungsmethoden, die intuitiv verständlich sind, erlauben die frühzeitige Einbeziehung von Mitarbeitern aus Fachabteilungen in Analyse- bzw. Entwicklungsprojekte. Neben dem Know-how-Transfer fördert dies i. d. R. auch die Akzeptanz des jeweiligen Projektes durch die unmittelbar Betroffenen.

⁶⁶ Vgl. Scheer, A.-W. (2001a), S. 4. Zum Modellbegriff vgl. auch Ferstl, O.K.; Sinz, E.J. (1998), S. 117 ff., sowie Krallman, H.; Frank, H.; Gronau, N. (1999), S. 29 ff.

Wesentliche Kennzeichen von Modellen sind zum einen der *Zweckbezug*, d. h. Modelle werden stets im Hinblick auf eine ganz bestimmte Zielsetzung erstellt, zum anderen die *Verkürzung*, d. h. das Original wird nicht vollständig abgebildet, sondern es findet eine Beschränkung auf relevante Aspekte statt.

Die Anwendungsbereiche von Modellen sind vielschichtig. Wesentliche Einsatzmöglichkeiten ergeben sich im Rahmen von *organisatorischen Fragestellungen*, wie z. B. Reorganisationsmaßnahmen, Neuausrichtung von Geschäftsprozessen (Business Process Reengineering), Qualitätsmanagement, ISO-Zertifizierungen oder beim Aufbau neuer Geschäftsfelder. Weiterhin sind Modelle relevant bei der Gestaltung bzw. beim *Aufbau betrieblicher Informationssysteme*, d. h. bei Standardsoftware-Einführungen oder der Entwicklung von Individualsoftware. Ein weiterer Einsatzbereich ist das *Wissensmanagement*, z. B. wenn es darum geht, relevantes Wissen strukturiert zu dokumentieren.

Modelle sind hinsichtlich des jeweiligen Modellierungsgegenstandes zu differenzieren:

- *Daten*: In Datenmodellen erfolgt eine Beschreibung von physikalischen oder abstrakten Dingen (Gegenstände, Personen, abstrakte Begriffe etc.), ihren Eigenschaften sowie ihren Beziehungen. Im Gegensatz zu Prozessmodellen handelt es sich um eine statische Betrachtungsweise. Ein Beispiel ist das Modell zum Entwurf eines Datenbanksystems.
- *Prozesse*: Im Vordergrund steht hier die Darstellung der logischen Abfolge von Aktivitäten. Ein Anwendungsbeispiel sind Prozessmodelle im Rahmen eines Reorganisationsprojektes.
- *Objekte*: Objektmodelle beschreiben wie Datenmodelle auch physikalische oder abstrakte Dinge, ihre Eigenschaften sowie ihre Beziehungen, zusätzlich verdeutlichen sie auch deren dynamisches Verhalten. Anwendungsbeispiele sind Modelle im Rahmen des objektorientierten Software-Entwurfs.

Andere Modelle fokussieren auf *Funktionen*, d. h. die statische Darstellung von Aktivitäten, Organisationseinheiten oder Leistungen.

Zur Modellierung ist es erforderlich, eine geeignete Modellierungsmethode auszuwählen. Eine *Methode* ist ein auf einem System von Regeln aufbauendes Problemlösungsverfahren. Eine *Modellierungsmethode* kennzeichnet dementsprechend eine Methode zur Erstellung eines Modells in einer speziellen Form. Aus den o. g. Zielsetzungen von Modellen resultieren unterschiedliche Anforderungen an die zu wählende Modellierungsmethoden:

- *Verständlichkeit*: Modellierungsmethoden und die damit erstellten Modelle müssen allgemein verständlich sein, beispielsweise um die Mitarbeiter aus den Fachabteilungen sinnvoll mit einbinden zu können.
- *Vielseitige Einsetzbarkeit*: Eine Modellierungsmethode sollte so flexibel aufgebaut sein, dass ein Einsatz in möglichst vielen Projektphasen (Ist-Analyse und Soll-Konzept) und Unternehmensbereichen möglich ist. Eine ähnliche Flexibi-

lität sollte auch hinsichtlich der darstellbaren Modellierungsgegenstände (Daten, Prozesse, Objekte) bestehen.

- **Änderbarkeit der Darstellungen:** Eine Modellierungsmethode sollte es erlauben, Modelle zu ändern und an neue Anforderungen anzupassen. Ebenfalls sollte eine Erweiterbarkeit der Darstellungen sichergestellt sein.
- **Auswertungs-/Analysemöglichkeiten:** Modelle erfüllen keinen Selbstzweck sondern dienen bestimmten Zielsetzungen wie z. B. der Auswertung hinsichtlich möglicher Schwachstellen eines Ist-Zustands. Eine Modellierungsmethode sollte diesen Aspekt daher unbedingt berücksichtigen.

Auch wenn aus der o. g. Definition bereits eine tabellarische Darstellung oder ein strukturierter Text als Modell aufgefasst werden kann, so werden als Modellierungsmethoden im engeren Sinne grafische Methoden verstanden. Vorteile einer grafischen Darstellung sind z. B. die damit verbundene Eindeutigkeit und Übersichtlichkeit des Modells, die gute Analysierbarkeit, sowie die Beschränkung auf relevante Informationen. Als nachteilig kann sich erweisen, dass grafische Methoden i. d. R. entsprechende Methodenkenntnisse voraussetzen, dass sie stark formalisiert und nicht immer so einfach zu erweitern und zu ergänzen sind wie textuelle tabellarische Darstellungen.

Die folgende Übersicht zeigt eine Auswahl gängiger Modellierungsmethoden differenziert nach dem jeweiligen Modellierungsgegenstand:

Ausgewählte Methoden	Prozesse	Daten	Objekte
Entity Relationship Model (ERM)		X	
Datenflusspläne		X	
Systems Analysis and Design Technique (SADT)	X	X	
Structured (Systems) Analysis (SA)	X	X	
Hierarchie Plus Input Process Output (HIPO)	X	X	
Struktogramme	X		
Petri-Netze	X		
Vorgangskettendiagramme	X		
Ereignisgesteuerte Prozesskette (EPK)	X	(X)	
Unified Modelling Language (UML)	(X)	(X)	X

Abb. 3-2: Ausgewählte Modellierungsmethoden

3.2.1 Datenorientierte Sicht

Die Modellierung von Daten ist natürlich insbesondere im Zusammenhang mit dem Aufbau von Datenbanken bedeutsam. Da aber auch andere betriebliche Informationssysteme wie ERP-Systeme auf Datenbanken basieren, können entsprechende Modellierungsaktivitäten auch in solchen Fällen angebracht sein.

Das Datengerüst eines Informationssystems ist i. d. R. weitaus stabiler als die Funktionen bzw. Prozesse. Änderungen der Datenstruktur haben daher weitreichende Konsequenzen, die häufig zur Anpassung von Anwendungsprogrammen oder Schnittstellen führen. Vor diesem Hintergrund stellen die sorgfältige Planung und Modellierung der zugrundeliegenden Datenstrukturen einen zentralen Erfolgsfaktor für die Entwicklung von Informationssystemen dar.

Als Standardmethode zur Datenmodellierung haben sich die von Chen (1976) entwickelten Entity-Relationship-Diagramme durchgesetzt. Es handelt sich dabei grundsätzlich um einen konzeptionellen, d. h. Datenbank-unabhängigen Ansatz, der jedoch eng mit dem relationalen Datenbankmodell verknüpft ist. Kern der Modellierung ist die Abbildung von Datenobjekten (Entitäten), deren Attributen und Beziehungen (Relationships).⁶⁷

Ein *Entity* ist ein reales oder abstraktes Objekt (z. B. Konto 12345, Kunde X etc.), gleichartige Entities werden zu einem *Entity-Typ* zusammengefasst (z. B. Rechnung, Kunde).

Jeder Entity-Typ weist bestimmte *Eigenschaften* (Attribute) auf (z. B. Kunde: Name, Adresse etc.), jedes Entity nimmt für die Attribute des Entity-Typs bestimmte *Werte* an:

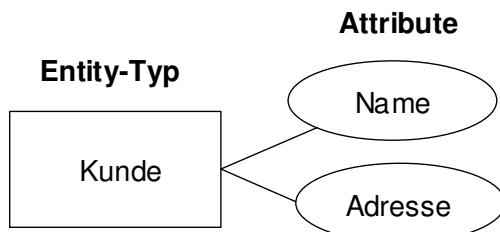


Abb. 3-3: ERM: Entity-Typ und Attribute

Entities bzw. Entity-Typen sind nicht isoliert, sondern stehen miteinander in Beziehungen, die in der grafischen Notation durch Rauten dargestellt werden:

⁶⁷ Zum ERM-Modell vgl. Ferstl, O.K.; Sinz, E. J. (1998), S. 127 ff.; Schwarze, J. (2000), S. 229 ff.; Stahlknecht, P.; Hasenkamp, U. (2002), S. 167 ff.

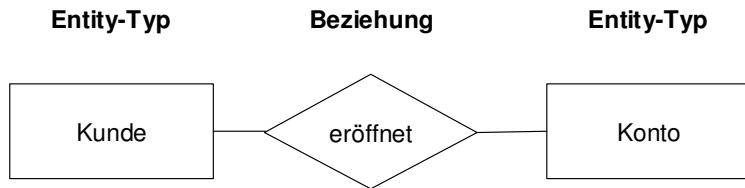


Abb. 3-4: ERM: Entity-Typen und Beziehung

Die Komplexität der jeweiligen Beziehung wird über *Kardinalitäten* gekennzeichnet. Eine gängige Darstellungsform ist dabei die (1,M,N)–Notation:

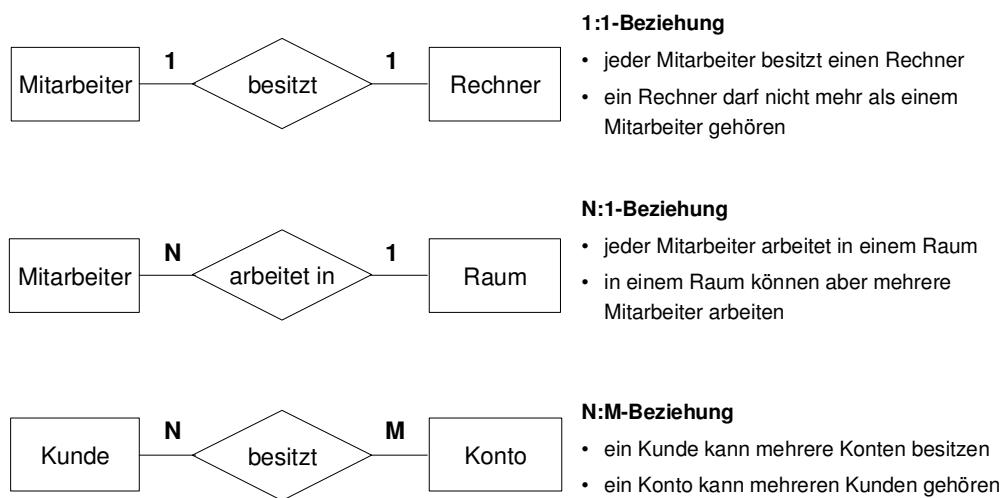


Abb. 3-5: ERM: Kardinalitäten

Neben den genannten Grundelementen und Modellierungsregeln existiert eine Reihe weiterer Notationen, die unterschiedliche Spezialfälle der Datenmodellierung aufgreifen. Beispiele dafür sind die Generalisierung bzw. Spezialisierung sowie die Aggregation.

Bei der *Generalisierung* bzw. *Spezialisierung* wird der Sachverhalt abgebildet, dass ein Entity-Typ ein Spezialfall eines anderen Entity-Typs ist (z. B. „Girokonto“ oder „Geldmarktkonto“ als Spezialfälle des Entity-Typs „Konto“). Identifizieren lässt sich eine solche Beziehung über die Formulierung: „Entity-Typ A ist ein Entity-Typ B“. In der grafischen Notation wird dies wie folgt abgebildet:

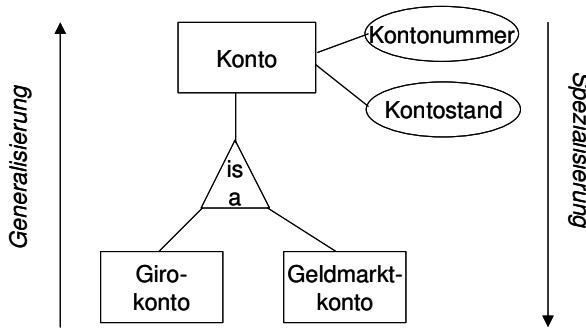


Abb. 3-6: ERM: Generalisierung / Spezialisierung

Eine Besonderheit besteht darin, dass alle Attribute des allgemeineren Entity-Typs auf die spezielleren übergehen („vererbt“ werden).

Die sog. *Aggregation* bezeichnet die Zusammenfassung einer Beziehung und den angrenzenden Entity-Typen zu einem neuen, komplexeren Entity-Typ. Zielsetzung ist die Vereinfachung der Darstellung sowie die Erfassung weiterer Beziehungen, die ansonsten nicht darstellbar wären (Beziehungen zwischen Beziehungen sind sonst nicht möglich). Ein Beispiel ist die nachfolgend dargestellte Zusammenfassung der Beziehung „Kunde eröffnet Konto“ zu einem aggregierten Entity-Typ „Antrag“, für den sich nun weitere Beziehungen wie z. B. die Überprüfung der Kontoeröffnung abbilden lassen:

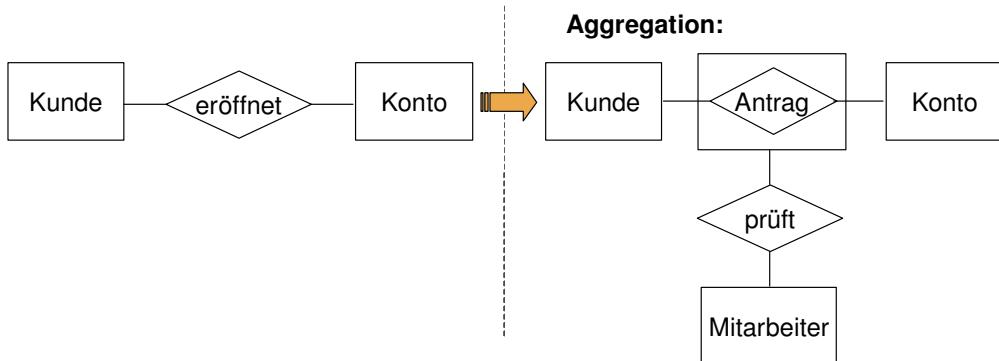


Abb. 3-7: ERM: Aggregation

3.2.2 Prozessorientierte Sicht

Ein *Prozess* ist gekennzeichnet durch eine Abfolge von Aktivitäten. Weiterhin existiert zu jedem Prozess ein Input, d. h. mindestens ein Objekt, das im Prozessverlauf eine Veränderung erfährt, sowie ein Output, d. h. mindestens ein Objekt, das als Prozessergebnis den Input für einen nachfolgenden Prozess bildet. Der Output resultiert aus einer Transformation eines oder mehrerer Inputobjekte

während der Prozessdurchführung. Die Input- und Outputobjekte können wie Rohstoffe materieller Art sein, aber auch immaterieller Art, wie z. B. Informationen. Zeitlich begrenzt wird ein Prozess einerseits durch ein auslösendes Ereignis, wie z. B. eine Umweltänderung oder den Abschluss vorgelagerter Aktivitäten, andererseits durch den Zeitpunkt, in dem das jeweilige Ergebnis erreicht ist:⁶⁸

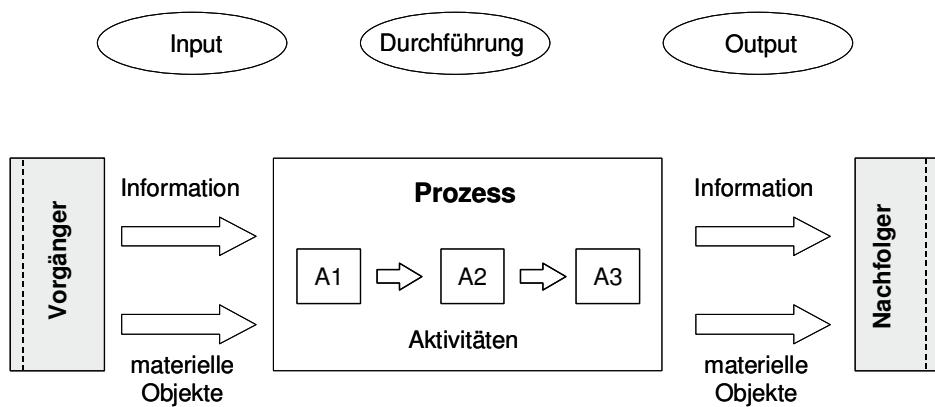


Abb. 3-8: Merkmale eines Prozesses

Zur Klassifizierung von Prozessen werden unterschiedliche Kriterien herangezogen:

- *Prozesshierarchie*: Aufteilung in Haupt- und Teilprozesse,
- *Beziehung zum Endprodukt*: Unterteilung in Führungs-, Leistungs- und Unterstützungsprozesse,
- *Prozessreichweite*: Unterscheidung von unternehmensinternen und –übergreifenden Prozessen,
- *Art der Inputobjekte*: Unterscheidung von materiellen und immateriellen Prozessen,

Die Zerlegung von Prozessen in Haupt- und Teilprozesse (und ggf. weiter in Arbeitsvorgänge und Aktivitäten) dient zum einen einer möglichst vollständigen Prozessbeschreibung, zum anderen aber auch der Identifikation überflüssiger Teilprozesse, an denen dann ggf. Reorganisationsmaßnahmen anknüpfen können.

Der Grad bzw. die Tiefe der Zerlegung ist an dem jeweiligen Zweck der Prozessanalyse auszurichten. Während bei einer unternehmensweiten Organisationsuntersuchung eine Betrachtung von Haupt- und Teilprozessen ggf. ausreicht, kann bei einer Softwareentwicklung eine detaillierte Analyse von Arbeitsvorgängen und Aktivitäten innerhalb der Prozesse erforderlich sein.

⁶⁸ Vgl. Schröder, H. (2001), S. 73 ff. und die dort angegebene Literatur.

Da Unternehmen häufig funktionsorientiert organisiert sind, werden Organisationseinheiten nach betrieblichen Funktionsbereichen wie Vertrieb oder Buchhaltung gebildet. Ein Geschäftsvorfall wie beispielsweise ein Kundenauftrag durchläuft dann mehrere Einheiten und wird von mehreren Personen bearbeitet (vgl. Abb. 3-9).

Dem Vorteil der Spezialisierung der jeweiligen Bearbeiter stehen die Nachteile gegenüber, dass bei Übergabe an die nachfolgende Einheit Informationen ausgetauschen sind, ggf. Rückfragen auftreten oder der Vorgang nicht zügig weiterbearbeitet wird. Derartige *Schnittstellen* sind oft der Grund für zu lange Prozesslaufzeiten und Bearbeitungsfehler.

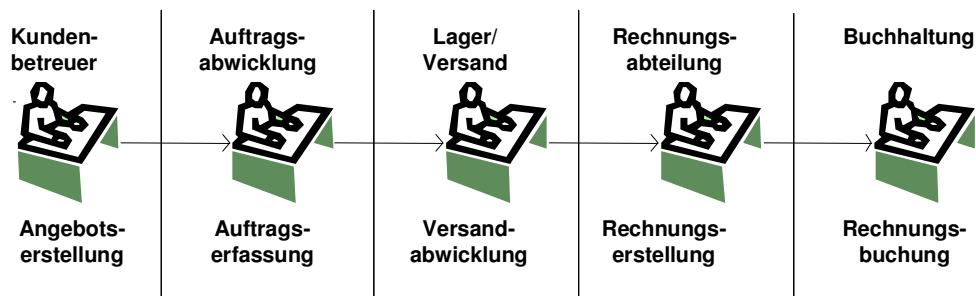


Abb. 3-9: Beispiel eines Geschäftsprozesses

Eine *Geschäftsprozessanalyse* untersucht den Prozessablauf unabhängig von den gegebenen betrieblichen Funktionsbereichen und versucht Optimierungspotenziale, wie überflüssige Prozessschritte oder vermeidbare Schnittstellen, aufzudecken. In Verbindung mit der Einführung betrieblicher Informationssysteme kommt diesem Aspekt insofern eine Bedeutung zu, als eine IT-Unterstützung des gesamten Prozesses anzustreben ist.

Zur Unterstützung einer prozessorientierten Analyse haben sich Modellierungsmethoden etabliert, die diesen Ansatz explizit unterstützen. Ein Beispiel dafür ist die Methode der *Ereignisgesteuerten Prozessketten* (EPK), die u. a. in der Modellierungssoftware ARIS® der IDS Scheer AG Verwendung findet.⁶⁹

Kernelemente dieser Methode sind Ereignisse und Funktionen, die durch Konnektoren miteinander verknüpft werden und auf diese Weise einen komplexen Geschäftsprozess darstellen.

Ein *Ereignis*, dargestellt als Sechseck, beschreibt einen eingetretenen betriebswirtschaftlich relevanten Zustand eines Objektes, der den weiteren Ablauf eines Geschäftsprozesses steuert oder beeinflusst.

⁶⁹ Vgl. zur EPK Scheer, A.-W. (1997), S. 49 ff.; Scheer, A.-W. (2001b), S. 124 ff. Zur ARIS®-Software s. auch die Homepage der IDS Scheer AG: www.ids-scheer.de.

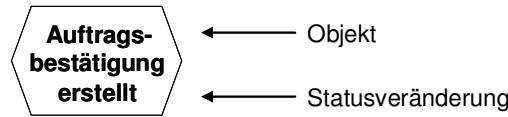


Abb. 3-10: Ereignis im Rahmen einer EPK

Eine *Funktion*, dargestellt als abgerundetes Rechteck, ist eine fachliche Aufgabe, ein Vorgang bzw. eine Tätigkeit an einem Objekt.

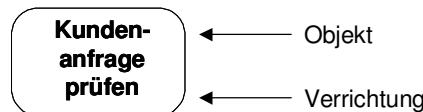


Abb. 3-11: Funktion innerhalb einer EPK

Im Gegensatz zu Ereignissen sind Funktionen Träger von Zeit und Kosten. Eine wichtige Grundregel der Modellierung von EPKs besagt, dass sich die beiden Elemente stets abwechseln, d. h. Ereignisse lösen Funktionen aus, die wiederum in Ereignissen münden.

Um Prozessverzweigungen und –zusammenführungen zu symbolisieren, werden die folgenden *Konnektoren* verwendet.

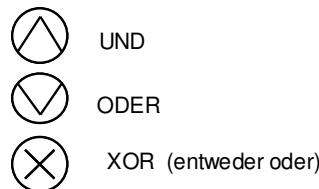


Abb. 3-12: Konnektoren innerhalb einer EPK

Während der „UND“-Konnektor zwei oder mehrere parallele Prozesspfade beschreibt, die beide durchlaufen werden, eröffnet der „XOR“-Konnektor zwei oder mehrere alternative Wege. Der „ODER“-Konnektor lässt beide Möglichkeiten, d. h. Parallelisierung und Alternative, zu.

Ein einfaches Beispiel einer EPK zeigt die folgende Abbildung:

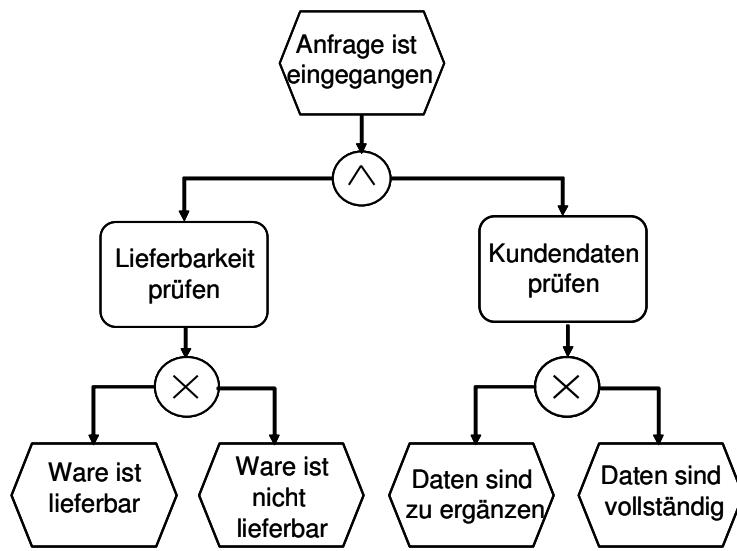


Abb. 3-13: Beispiel einer EPK

Auf diese Weise können sehr komplexe Modelle erstellt werden. Zudem kann die Darstellung durch weitere Symbole für Organisationseinheiten, IT-Systeme oder Daten ergänzt werden. Damit wird aber auch ein wesentliches Problem der Prozessmodellierung angesprochen: Bei einer detailgenauen Abbildung realer Prozesse werden die Modelle schnell unübersichtlich und schwer nachvollziehbar. Umso wichtiger ist es daher, bei der Modellierung den Blick auf das Wesentliche zu behalten. Weiterhin ist es zwingend erforderlich, Modellierungsregeln einzuhalten, damit die erstellten Modelle konsistent und lesbar bleiben. Im Falle der EPKs haben sich die folgenden Modellierungsregeln etabliert:⁷⁰

1. Jede EPK beginnt mit einem Ereignis (Startereignis) und endet mit einem Ereignis (Endereignis).
2. Ereignisse lösen Funktionen aus bzw. sind das Ergebnis von Funktionen. Daraus resultiert eine sequentielle Abfolge von Ereignissen und Funktionen, die einzuhalten ist.
3. Ereignisse besitzen keine Entscheidungskompetenz. Ereignisse können entweder eintreten oder nicht.
4. Funktionen sind aktive Elemente, d. h. sie können den Systemzustand verändern. In Funktionen können Entscheidungen getroffen werden, die den weiteren Prozessablauf beeinflussen.
5. Die einzelnen Elemente werden mit gerichteten Kanten verbunden.
6. Bei mehreren Prozesspfaden sind Verknüpfungsoperatoren (Konnektoren) zu verwenden.

⁷⁰ Vgl. Becker, J.; Kugeler, M.; Rosemann, M. (2000), S. 59 ff.

7. Verknüpfungsoperatoren haben entweder eine eingehende Kante und mehrere ausgehende oder mehrere eingehende und eine ausgehende Kante.
8. Wenn Verzweigungen wieder zusammengeführt werden sollen, muss dies mit dem gleichen Konnektor erfolgen.

Die Beachtung derartiger Regeln ist vor allem dann angebracht, wenn die Modellierung durch ein Team durchgeführt wird. Die erstellten Modelle müssen von allen Beteiligten und natürlich auch von Dritten einheitlich interpretiert werden können, und es muss ausgeschlossen werden, dass jeder Modellierer seine „individuellen“ Regeln zu Grunde legt.

3.2.3 Objektorientierte Sicht

Einen etwas anderen Ansatz verfolgt die Unified Modelling Language (*UML*). Es handelt sich hier um eine Modellierungssprache, die vorwiegend zur Unterstützung der objektorientierten Softwareentwicklung eingesetzt wird. Objektorientierte Programmiersprachen wie C++ oder Java fokussieren im Gegensatz zu klassischen prozeduralen Sprachen auf Objekte. Ein *Objekt* ist eine in sich abgeschlossene Einheit, die über *Attribute* beschrieben wird und über *Methoden* verfügt, die ausgeführt werden können. Aktiviert werden diese Methoden über *Nachrichten*.⁷¹

Die UML ist eine überwiegend grafische Notation zur Spezifikation, Visualisierung, Konstruktion und Dokumentation von Modellen für Softwaresysteme, Geschäftsprozesse und prinzipiell auch andere Systeme. Sie enthält jedoch keine Methode im Sinne einer Handlungsanleitung für den Entwicklungsprozess. Im Rahmen der UML werden unterschiedliche Diagrammtypen mit genau festgelegter Bedeutung (Semantik) der einzelnen Elemente verwendet, wobei teilweise mehrere Darstellungsarten für gleiche Sachverhalte möglich sind.

Die wesentliche Zielsetzung besteht in einer einheitlichen Diskussionsbasis für den Entwurf und die Entwicklung von Software. Eine Standardisierung bzw. Weiterentwicklung der UML erfolgt durch die Object Management Group (OMG).⁷²

Einen ersten Überblick über die unterschiedlichen Modelltypen und deren Verwendung im Software-Entwicklungszyklus liefert die folgende Abbildung.

⁷¹ Vgl. Hansen, H.R.; Neumann, G. (2001), S. 949 ff.; Schwarze (2000), S. 179 ff.

⁷² Vgl. <http://www.omg.org>.

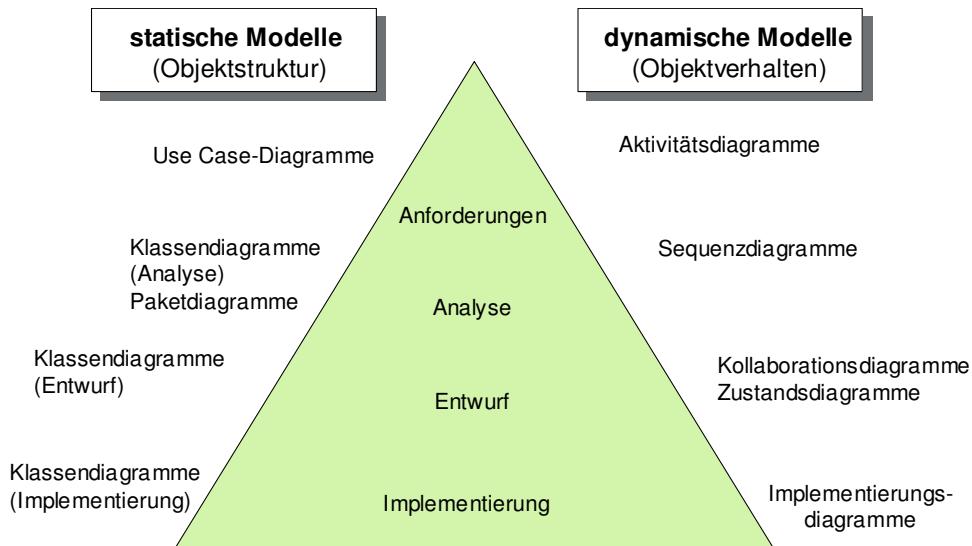


Abb. 3-14: Übersicht UML-Diagrammtypen⁷³

Demnach sind grundsätzlich statische und dynamische Modelltypen zu unterscheiden. Während *statische Modelle* die Struktur von Objekten, wie beispielsweise „Kunde“ oder „Konto“, beschreiben, wird mit den *dynamischen Modellen* deren Verhalten im Zeitablauf dokumentiert, beispielsweise die Veränderung des Objektes „Konto“ im Rahmen der Bearbeitung von Geschäftsvorfällen in einer Bank.

Ausgangspunkt der UML-Modellierung sind i. d. R. die Use Case- sowie die Klassendiagramme. Ein *Use Case-Diagramm* (Anwendungsfalldiagramm) besteht aus einer Reihe sogenannter Anwendungsfälle und stellt Beziehungen zwischen Akteuren und diesen Anwendungsfällen dar:

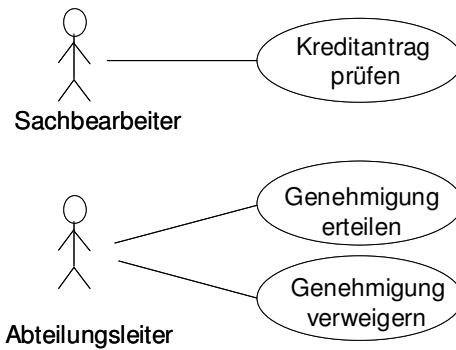


Abb. 3-15: UML Use Case-Diagramm

⁷³ Vgl. Seemann, J.; Wolff von Gudenberg, J. (2000), S. 8.

Beschrieben wird das Systemverhalten aus der Sicht eines Anwenders. Es wird verdeutlicht, was ein System leisten soll, jedoch nicht, wie dies geschehen muss. Es besteht die Möglichkeit zur Schachtelung, d. h. ein Anwendungsfall kann selbst wieder in mehrere Detailanwendungsfälle aufgegliedert werden. Wie in Abb. 3-15 exemplarisch dargestellt werden Anwendungsfälle durch Ellipsen, Akteure meist durch Personensymbole abgebildet.

Dieser auf den ersten Blick recht trivialen Darstellung ist insofern Bedeutung beizumessen, als damit eine erste Strukturierung des Problembereiches gelingen kann. Über weitere in der UML vorgesehene Features können zusätzlich Varianten von Anwendungsfällen oder Verschachtelungen modelliert werden.

Grundlegende Mechanismen der Objektorientierung werden in den *Klassendiagrammen* deutlich. Eine Klasse beschreibt eine Menge von Objekten, die ein gemeinsames Verhalten und gemeinsame Merkmale aufweisen. In der Klasse werden Eigenschaften (Attribute), Operationen (Methoden) und die Semantik (Bedeutung) der Objekte festgelegt.⁷⁴ Eine Klasse wird als Rechteck dargestellt, Attribute und Methoden in jeweils eigenen Feldern des Klassenrechteckes.

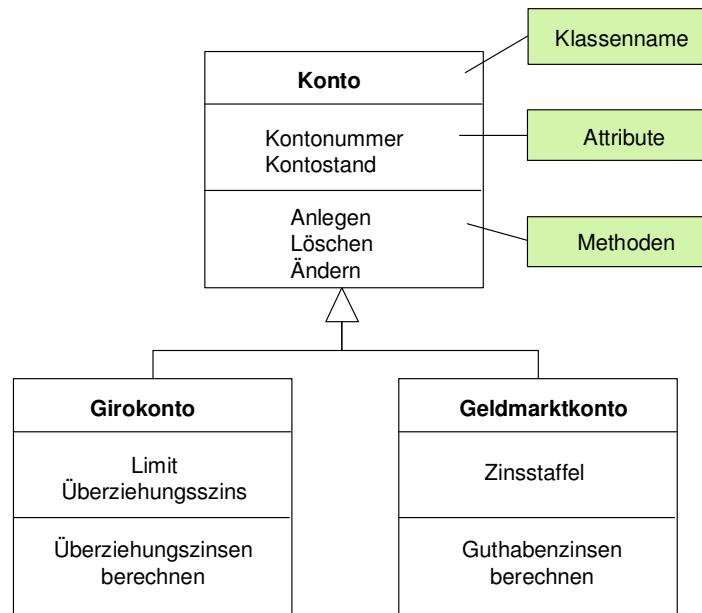


Abb. 3-16: UML Klassendiagramm

Ein Beispiel zeigt die Abb. 3-16. Dort wird die Klasse „Konto“ beschrieben durch die Eigenschaften „Kontonummer“ und „Kontostand“ und beinhaltet die Methoden „Anlegen“, „Löschen“ und „Ändern“. Ein Objekt dieser Klasse wäre beispielsweise das „Konto mit der Nummer 12345“.

⁷⁴ Vgl. Schwarze, J. (2000), S. 180; Stahlknecht, P.; Hasenkamp, U. (2002), S. 279 ff.

Zwischen den Klassen können vergleichbar den Entity-Typen im ERM-Modell (vgl. Kap. 3.2.1) Beziehungen bestehen. Weiterhin ist bei hierarchischen Beziehungen die Abbildung einer *Vererbung* möglich. Untergeordnete Klassen (*Subklassen*) erben die Attribute und Methoden der übergeordneten Klassen (*Superklassen*), können darüber hinaus aber eigene Attribute bzw. Methoden besitzen.

Klassendiagramme bilden den zentralen Bestandteil der UML und verbinden alle anderen Diagramme. Sie sind sowohl in der Analyse- als auch der Designphase einsetzbar und beschreiben die statische Struktur der Objekte und ihre Beziehungen. Bereits in der Analysephase werden erste Klassen konzipiert. Im sogenannten "groben Klassendiagramm" entstehen die wesentlichen Ausgangsklassen. Dabei werden zunächst nur die Attribute angegeben. Die Methoden, die eine Klasse besitzt, werden später in der Verfeinerung ergänzt. In den Folgeschritten ist es dann erforderlich, Beziehungen zwischen den identifizierten Klassen zu modellieren. Ein Beispiel dafür ist die *Generalisierung*, d. h. eine Klasse, z. B. „Konto“, fungiert als Oberbegriff für andere Klassen, z. B. „Girokonto“ und „Geldmarktkonto“.

Wiederholungsfragen

1. Erläutern Sie die wesentlichen Vorteile von Modellen. Warum werden Modelle im Rahmen einer Systemanalyse verwendet?
2. Differenzieren Sie Modelle hinsichtlich des jeweiligen Modellierungsgegenstandes. Nennen Sie jeweils ein Beispiel.
3. Erläutern Sie die *Generalisierung* sowie die *Aggregation* im Rahmen des Entity-Relationship-Modells mit selbst gewählten Beispielen.
4. Für den Aufbau eines Auftragsverwaltungssystems ist ein Entity-Relationship-Modell zu erstellen. Dabei sind folgende Sachverhalte zu berücksichtigen:

Für jeden Kunden werden die Kundennummer und der Name erfasst. Ein Kunde kann einen oder mehrere Aufträge erteilen, wobei Standardaufträge und Eilaufträge zu unterscheiden sind. Für jeden Auftrag wird eine Auftragsnummer vergeben. Ein Auftrag besteht aus einer oder mehreren Auftragspositionen (gekennzeichnet durch eine eindeutige Positionsnummer), für die jeweils ein Liefertermin ermittelt wird. In jeder Auftragsposition wird ein Artikel erfasst. Artikel werden durch die Artikelnummer und -bezeichnung näher gekennzeichnet.

Gleiche Artikel können in unterschiedlichen Lagern vorrätig sein, ein Auftrag wird aber aus genau einem Lager bedient. Die Auslieferung wird jeweils durch einen Transporteur vorgenommen, zu dem Name und Konditionen gespeichert werden.

(Lösung s. Anhang).

5. Warum ist die Betrachtung von *Prozessen* im Rahmen der Systemanalyse von Bedeutung?

6. Erläutern Sie die wesentlichen Elemente und Modellierungsregeln einer „Ereignisgesteuerten Prozesskette“ (EPK).
7. Ein Unternehmen möchte ein Shop-System einführen, das die Entgegennahme und Abwicklung von Kundenaufträgen über das Internet ermöglicht. Dazu ist der folgende Soll-Prozess beschrieben worden, der nun als EPK modelliert werden soll:

Wenn der Kunde die gewünschten Artikel ausgewählt und in seinen elektronischen Warenkorb gelegt hat, kann die Bestellung ausgelöst werden. Im ersten Schritt wird eine Verfügbarkeitsprüfung der gewählten Artikel im Lagersystem durchgeführt. Wenn aufgrund einer negativen Verfügbarkeitsprüfung eine termingerechte Lieferung eines oder mehrerer Artikel nicht möglich ist, wird dem Kunden für diese Waren ein späterer Liefertermin angeboten. Wird dieser Termin vom Kunden nicht akzeptiert und ist auch eine Teillieferung der übrigen verfügbaren Artikel der Bestellung nicht gewünscht, muss der komplette Bestellvorgang mit einer entsprechenden Mitteilung an den Kunden abgebrochen werden.

Bei Verfügbarkeit aller gewünschten Artikel bzw. Akzeptanz eines neuen Liefertermins oder von Teillieferungen verfügbarer Artikel kann die Bestellung weiterbearbeitet werden.

Systemseitig wird dann die Kreditkartennummer des Kunden abgefragt. Aufgrund dieser Information ermittelt das System, ob es sich um einen Neu- oder Bestandskunden handelt. Bei Neukunden wird automatisch eine externe Bonitätsprüfung bei der jeweiligen Kreditkartenorganisation angestoßen. Bei Bestandskunden wird dagegen das Konto im Buchhaltungssystem überprüft.

Bei negativen Ergebnissen der Bonitäts- oder Kontoüberprüfung erfolgt eine Absage an den jeweiligen Kunden. Der Bestellvorgang wird mit einer entsprechenden Meldung abgebrochen. Wird die Bestellung dagegen akzeptiert, wird eine elektronische Auftragsbestätigung für den Kunden generiert.

Erstellen Sie auf Basis dieser Informationen eine EPK in der Standard-Notation. Beachten Sie dabei die Modellierungsregeln und beschränken Sie sich auf die grundlegenden Modellierungselemente Ereignisse, Funktionen und Konnektoren. (Lösung s. Anhang).

8. Vergleichen Sie mit Hilfe von vier Anforderungen, die an eine Modellierungsmethode gestellt werden, eine tabellarische Darstellung von Prozessen mit der Methode EPK. (Lösung s. Anhang).
9. Worin besteht der Unterschied zwischen statischen und dynamischen Modelltypen im Rahmen der Unified Modelling Language (UML)? Nennen Sie jeweils ein Beispiel.