

Modellierung betrieblicher Informationssysteme

*Manfred Sommer**

- 1 Einleitung
- 2 Modellierung nach Beschreibungssichten
 - 2.1 Datensicht
 - 2.2 Funktionssicht
 - 2.3 Organisationssicht
 - 2.4 Prozesssicht
 - 2.5 Exkurs zur Geschäftsprozess- und Objektmodellierung
- 3 Modellierung nach Beschreibungsebenen
 - 3.1 Fachkonzeptuelle Ebene
 - 3.2 DV-konzeptuelle Ebene
 - 3.3 Implementierungsebene
- 4 Modellierung von Struktur und Verhalten
- 5 Modellierung nach Geltungsanspruch
 - 5.1 Istmodelle
 - 5.2 Sollmodelle
- 6 Modellierung nach Geltungsbereich
 - 6.1 Unternehmensweites Modell
 - 6.2 Partialmodell

* Prof. Dr. Manfred Sommer, Universität Hamburg

- 7 Modellierung nach Individualität
- 8 Modellierung nach Abstraktionsgrad
- 9 Modellierung nach Darstellungstechnik
 - 9.1 Grafisches Modell
 - 9.2 Natürlichsprachliches Modell
 - 9.3 Computersprachliches Modell
 - 9.4 Mathematisches Modell
- 10 Modellierung nach Modellzweck

Literatur

1 Einleitung

Informationssysteme (IS) sind für die Gestaltung und Durchführung von Geschäftsprozessen inzwischen bis in die Gruppe der Kleinstunternehmen vorgedrungen (vgl. Abschnitt 5.1). Obwohl die Vorzüge, sowohl die Geschäftsprozesse selbst als auch die diese zumindest partiell steuernden Informationssysteme zu modellieren, auf der Hand liegen, basieren bei weitem nicht alle Anwendungssysteme auf Modellen. Das mag bei kleinen, vom Anwender selbst mit den typischen Mitteln von Office-Paketen, den sog. Endbenutzerwerkzeugen, erstellten und nur für seine eigene Verwendung gedachten Anwendungen sinnvoll und auch wirtschaftlich sein, weil sich in diesen Fällen eine Modellierung als „Umwegproduktion“ nicht rechnen würde. Schon bei manchem von seiner Funktionalität her sehr eingeschränkten Standardsoftwareprojekt dürfte sich aber der Verzicht auf eine solide Modellbildung eines dann „quick and dirty“ entwickelten Systems im Nachhinein gerächt haben.

Den Gründen für den nicht immer vorteilhaften Verzicht auf die Modellierung betrieblicher Informationssysteme soll hier nicht vertieft nachgegangen werden. In der Praxis zumindest der semi-professionellen Softwareentwicklung wird Modellierung häufig – ähnlich wie Dokumentation – als lästiges Übel betrachtet, das zum unmittelbaren Projektnutzen wenig beiträgt und im Zweifelsfall nur die Fertigstellung verzögert. Die (Wirtschafts-)Informatik hat die Modellierungs-Skeptiker vielleicht dadurch bestärkt, indem sie mit Schlagworten wie „rapid application prototyping“ oder „Agilität“, die als Korrekturhilfe für starre Phasenmodelle gedacht waren, Hoffnungen auf eine beliebige Beschleunigung von IS-Entwicklungsprozessen geweckt hat, die in der Praxis hier und da als Aufforderung zum Verzicht auf Modellunterstützung missverstanden wurden.

Nicht zuletzt dürfte die Fülle der für die Modellierung betrieblicher Informationssysteme angebotenen Methoden, Notationen, Vorgehensweisen und Werkzeuge insbesondere in kleinen und mittleren Unternehmen eine abwartende und zögerliche Grundeinstellung bestärkt haben. Dabei zeichnet sich in der Wirtschaftsinformatik inzwischen ein breiter Basiskonsens darüber ab, wie Modelle betrieblicher Informationssysteme klassifiziert werden können, und welche Modellierungsansätze zu den aussichtsreichen zählen.¹

Modelle betrieblicher Informationssysteme lassen sich an Hand diverser Kriterien typisieren. Mit der an Schütte,² Fink, Schneiderei und Voß³ sowie Rautenstrauch und

¹ Mit der Konvergenz bei der Systematisierung der Modelle betrieblicher IS wurden weder individuelle Präferenzen für bestimmte Methoden, Notationen und Tools nivelliert, noch haben sich sämtliche methodischen Kontroversen erledigt. Stellvertretend für andere wird in diesem Beitrag auf das nicht abschließend geklärte Problem von Geschäftsprozess- und Objektmodellierung hingewiesen (vgl. Abschnitt 2.5).

² Schütte (1998), S. 64.

³ Fink/Schneiderei/Voß (2001), S. 92.

Schulze⁴ angelehnten Zusammenstellung (Abbildung 1) spannt sich das für die heutige Entwicklung betrieblicher Informationssysteme relevante Modellspektrum auf, das in den folgenden Abschnitten überblicksartig dargestellt wird.

Merkmal	Ausprägung			
	Beschreibungssicht	Datensicht	Funktionssicht	Organisationsicht
[„Objektsicht“]				
Beschreibungsebene	Fachkonzept		DV-Konzept	Implementierung
Struktur / Verhalten	Statische Modelle		Dynamische Modelle	
Geltungsanspruch	Istmodell		Sollmodell	
Geltungsbereich	Unternehmensweites Modell		Partialmodell	
Individualität	Unternehmensspezifisches Modell	Individualisiertes Referenzmodell	Referenzmodell	
Abstraktionsgrad	Ausprägungsebene		Metamodell	
Darstellungstechnik	Sprache		Grafische Notation	Mathemat. Formalismus
	natürliche	Computer-		
Modellzweck	Erklärungsmodell		Entscheidungsmodell	
			Gestaltungsmodell	Optimierungsmodell

Abbildung 1: Modelltypen für betriebliche Informationssysteme

2 Modellierung nach Beschreibungssichten

Eine wesentliche Aufgabe von Modellen ist es, bei der Abbildung realer Systeme deren Komplexität zu reduzieren. Als eines der wirkungsvollsten methodischen Instrumente zur Komplexitätsbewältigung hat sich in der Wirtschaftsinformatik die Sichtenbildung etabliert. Allgemein reduziert jede Sicht – wie aus Bauzeichnungen vertraut – die Komplexität dadurch, dass sie ein System nur aus einem bestimmten Blickwinkel betrachtet und alles weglässt, was aus diesem nicht sichtbar ist. Der Grundriss eines Gebäudes – genauer: der einzelnen Stockwerke – gibt nur die horizontale Ebene wieder, während die vertikalen Maße außer Acht bleiben. Genau umgekehrt verhält es sich mit den Ansichten auf und den Schnitten durch ein Gebäude. Während das Sichtenkonzept als solches sehr abstrakt und vielseitig einsetzbar ist, sind die der „Architektur integrierter Informationssysteme“⁵ (ARIS) entlehnten Beschreibungssichten semantikhaltiger, weil

⁴ Rautenstrauch/Schulze (2003), S. 230.

⁵ Scheer (1988).

sie jeweils bestimmte Modellierungsgegenstände fokussieren bzw. sich um eine Integration der so gebildeten Sichten bemühen.

In ARIS werden die Modelltypisierungsmerkmale Beschreibungssicht und Beschreibungsebene (vgl. Abschnitt 3) zum sog. ARIS-Haus kombiniert (Abbildung 2).

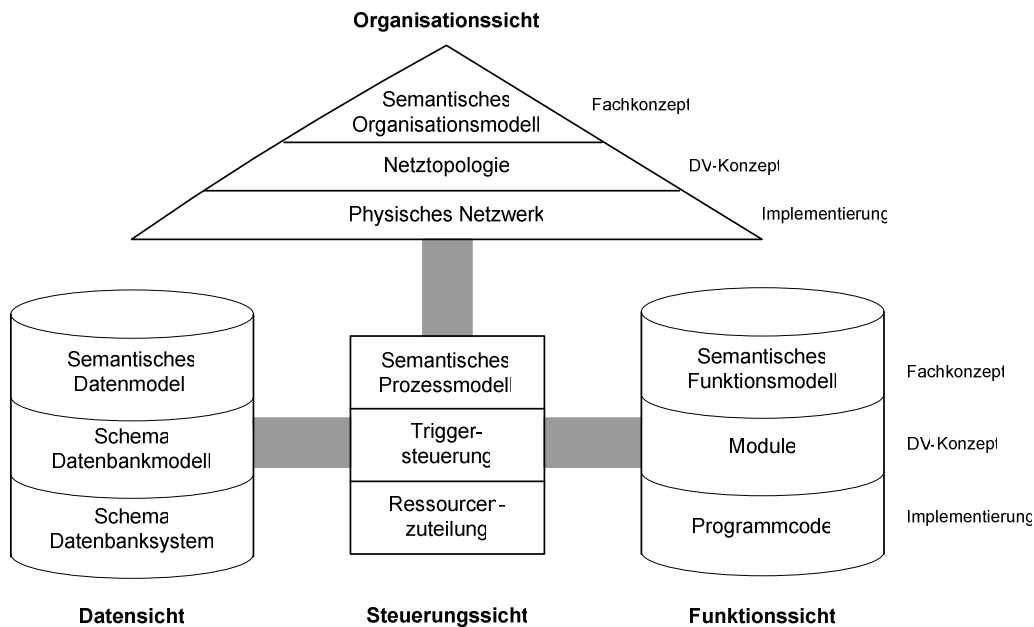


Abbildung 2: Architektur integrierter Informationssysteme⁶

2.1 Datensicht

Die Modellierung betrieblicher Informationssysteme aus der Datensicht ist identisch mit der klassischen Datenmodellierung. „Die Datensicht enthält die Beschreibung der Datenobjekte, die von Funktionen manipuliert werden“⁷. Auf der fachkonzeptuellen Ebene ist die Methode der ER-Modellierung (ER) so dominant etabliert, dass sich die einzelnen Spielarten und Werkzeuge nur hinsichtlich der von ihnen verwendeten Notationen unterscheiden.

Das ARIS-Konzept verwendet die sog. Chen-Notation (Abbildung 3), deren Symbole folgende Bedeutung haben:

- Rechtecke (wie Artikel): Entitäten,

⁶ Quelle: Schwarzer/Krcmar (2004), S. 97.

⁷ Scheer (1998), S. 67.

- Ovale (wie Name): Attribute,
- Rauten (wie Konditionenzuordnung): Beziehungen,
- Rechtecke mit Rauten (wie Vertriebsbeleg): als Entitäten interpretierte Beziehungen,
- Dreiecke (wie zwischen Vertriebsbeleg und Kundenanfrage, Kundenangebot, Kundenauftrag): Generalisierungsbeziehung.

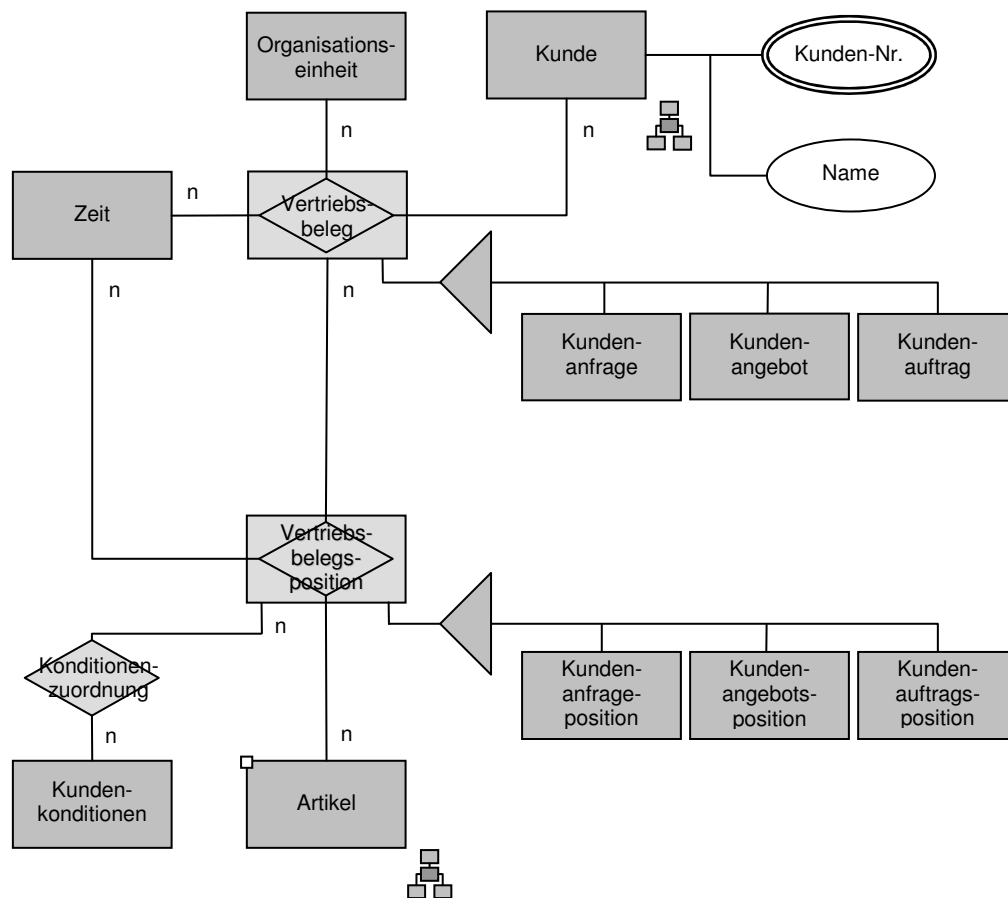


Abbildung 3: Datensicht in ARIS⁸

Für die ER-Modellierung gibt es auch andere Notationen wie die **Information-Engineering-Notation** – häufig auch salopp als Krähenfußnotation bezeichnet – oder

⁸ IDS Scheer (1999), Beispielmodelle „General Cars Coop.“.

die IDEF1X-Notation, die beide z.B. vom Datenmodellierungswerkzeug Erwin unterstützt werden. Obwohl diese Notationen einen gemeinsamen, sich aus der ER-Methode ergebenden Kern aufweisen, können sie je nach verwendetem Modellierungswerkzeug noch einmal unterschiedlich aussehen. Die Präferenz für eine der Notationen ist stark subjektiv geprägt.

2.2 Funktionssicht

Während die Daten häufig auch als passive oder statische Modellelemente bezeichnet werden, sind die in der Funktionssicht betrachteten Funktionen (fachliche Synonyme: Aktivitäten, Aufgaben, Tätigkeiten, Vorgänge)⁹ die aktiven oder dynamischen Modellelemente. In der Definition einer Funktion als einer Verrichtung an einem Objekt zur Unterstützung eines oder mehrerer Ziele kommt die doppelte Bindung von Funktionen zum Ausdruck:

- Aus fachlicher Sicht leiten sich insbesondere komplexe, hoch aggregierte Funktionen (sog. Funktionsbündel) aus den die Geschäftsprozessstrategie konstituierenden Unternehmenszielen ab.
- Aus DV-technischer Sicht sind die Funktionen eng an die Daten gebunden, weil insbesondere die feinkörnigen Funktionen (sog. Elementarfunktionen) die durch Daten beschriebenen Zustände der Informationsobjekte ändern, indem sie Input-Daten zu Output-Daten transformieren.¹⁰

Wie bereits angedeutet werden Funktionen in der Funktionssicht auf unterschiedlichen Verdichtungsstufen betrachtet, auf diesen mit zunehmendem Detaillierungsgrad als Funktionsbündel, Funktionen, Teilfunktionen und Elementarfunktionen bezeichnet und i. d. R. hierarchisierend in Funktionsbäumen¹¹ modelliert. Da vor allem hoch verdichtete Funktionen ihrerseits auch als Prozesse interpretiert werden können,¹² ist die Unterscheidbarkeit von Funktions- und Prozesssicht nicht unumstritten.¹³ Diese Kritik und insbesondere die These, die Funktionssicht sei der Verhaltenssicht zuzuordnen, weil Funktionen Aussagen über die Prozesslogik trafen, verkennt das Abstraktionsprinzip eines jeden Sichtenkonzepts.

Die Funktionssicht lässt die innere dynamische Prozesslogik der Funktionen außer Acht, überlässt deren Modellierung der Prozesssicht und konzentriert sich auf die statischen Aspekte der Funktionsstruktur.¹⁴ Anders als ereignisorientierte Prozessketten

⁹ Scheer (1998), S. 21.

¹⁰ Rautenstrauch/Schulze (2003), S. 227.

¹¹ Einem Funktionsbaum formal sehr ähnlich, jedoch inhaltlich nicht mit ihm identisch, ist ein in Alpar/Grob/Weimann/Winter (2002), S. 131 vorgestellter Aktivitätsbaum. Beide sind hierarchische und statische Modelle von Prozessen unterschiedlicher Granularität, der Funktionsbaum ist aber auf der fachlichen, der Aktivitätsbaum auf der DV-technischen Ebene (vgl. Abschnitt 3) angesiedelt.

¹² Rautenstrauch/Schulze (2003), S. 232.

¹³ Schütte (1998), S. 65.

¹⁴ Scheer (1998), S. 21.

(vgl. Abschnitt 2.4) betrachten Funktionsbäume nicht zeitlich-logische Abfolgen, sondern die strukturelle Zerlegbarkeit von Funktionen nach dem Kriterium der Zugehörigkeit zu

- gleichartigen Geschäftsprozessen,
- gleichartigen Informationsobjekten,
- gleichartigen Verrichtungen.¹⁵

Die Modellierung der Funktionssicht kann als von der Modellierung der Datensicht weitgehend unanhängig betrachtet werden – eine Trennung, die erst in der Objektmodellierung konsequent aufgehoben wurde (vgl. Abschnitt 2.5). Bei der toolgestützten Modellierung der physikalischen Sicht relationaler Datenmodelle wird diese Trennung ebenfalls bereits perforiert, da SQL Server aktive Datenbanken¹⁶ in dem Sinne sind, dass sie serverseitig gespeicherte Prozeduren, Funktionen und Triggern ausführen und damit Bestandteile der Funktionssicht implementieren können.

2.3 Organisationssicht

In der Organisationssicht werden die Organisationseinheiten der statischen Aufbauorganisation mit ihren Beziehungen zu einander ebenfalls durch Hierarchiediagramme, die dort als Organigramme bezeichnet werden, modelliert. Die Modellierung der Organisationssicht ähnelt also hinsichtlich des hierarchischen Herunterbrechens von Organisationseinheiten bis hin zu einzelnen Stellen sowie hinsichtlich der grafischen Darstellung stark der Funktionssicht, ist methodisch aber wesentlich einfacher, da Organisationseinheiten ein offensichtlicheres Pendant im betrieblichen Basissystem und damit eine höhere „Sichtbarkeit“ als Funktionen haben.

Die Strukturierung der Organisationseinheiten zumindest im Ist-Modell (vgl. Abschnitt 5.1) ist deshalb erheblich einfacher als die Zerlegung von Funktionen, weshalb die von verschiedenen Modellierern für dasselbe Informationssystem entwickelten Modelle der Organisationssicht erheblich ähnlicher sein dürften als die Modelle der Funktionssicht.

Weder mit der Funktions- noch mit der Datenmodellierung hat die Modellierung der Organisationssicht prima facie etwas zu tun. Dem widerspricht auch nicht, dass Elemente der Organisationssicht (Organisationseinheiten, Stellen oder Personen) zugleich als Entitätstypen in Datenmodellen auftreten können.

2.4 Prozesssicht

Das Innovative des ARIS-Ansatzes liegt in der Prozess- oder Steuerungssicht. Während die Daten-, Funktions- und Organisationssicht echte Partialsichten sind, soll die Steue-

¹⁵ Scheer (1998), S. 25 f.

¹⁶ Rautenstrauch/Schulze (2003), S. 140 f.

rungssicht diese zunächst getrennt behandelten Sichten wieder verbinden.¹⁷ Die Steuerungssicht ist also eine Integrationssicht, die entweder paarweise

- die Funktionssicht und die Organisationssicht,
- die Funktionssicht und die Datensicht,
- die Organisationssicht und die Datensicht

koppelt oder

- die Funktionssicht, die Datensicht und die Organisationssicht

insgesamt integriert.

ARIS stellt sowohl für die paarweise wie für die vollständige Sichtintegration spezifische Darstellungsmittel zur Verfügung, deren prominentestes die ereignisorientierten Prozessketten sind. Das Besondere an den ereignisorientierten Prozessketten ist, dass sie erstens die drei anderen Sichten integrieren und zweitens die statische Funktionssicht dynamisieren.

Integrieren: Erweiterte ereignisorientierte Prozessketten (eEPK) verknüpfen die Funktionssicht mit der Datensicht und/oder der Organisationssicht:

- Lesende und schreibende Zugriffe von Funktionen auf Daten.
- Zuständigkeiten von Organisationseinheiten für die Durchführung von Funktionen.

Dynamisieren: Bereits einfache ereignisorientierte Prozessketten (EPK) dynamisieren die Funktionssicht, indem sie unter Hinzunahme der Modellelemente Ereignis und logischer Konnektor die in Funktionsbäumen statisch betrachteten Funktionen in eine zeitlich-logische Abfolge bringen.¹⁸ Die Steuerungssicht ist deshalb mit der Funktionssicht besonders eng verwandt, enger jedenfalls als mit der Organisations- und der Datensicht, was im synonymen Begriff der Prozesssicht noch deutlicher wird. Die Prozesssicht wirft auf die Funktionen einen dynamischen Blick, während die Funktionssicht eine statische Perspektive einnimmt.¹⁹

¹⁷ Scheer (1998), S. 102.

¹⁸ Ausdrücklich weist Scheer (1998), S. 31 hin auf den fließenden Übergang von der „Ablaufolge der Funktionen“ in der Funktionssicht, die noch keine Ereignissteuerung kennt, zu „späteren Prozessbeschreibungen der Steuerungssicht“. Weniger deutlich wird, dass eine der Steuerungssicht zugeordnete einfache EPK eigentlich keine Sichtintegration bewirkt, sondern lediglich eine Partialsicht – die Funktionssicht – dynamisiert.

¹⁹ Deshalb überrascht es auch nicht, dass beide Sichten gelegentlich gemeinsam angehandelt werden. Vgl. z.B. Fink/Schneiderei/Voß (2001), S. 111-113.

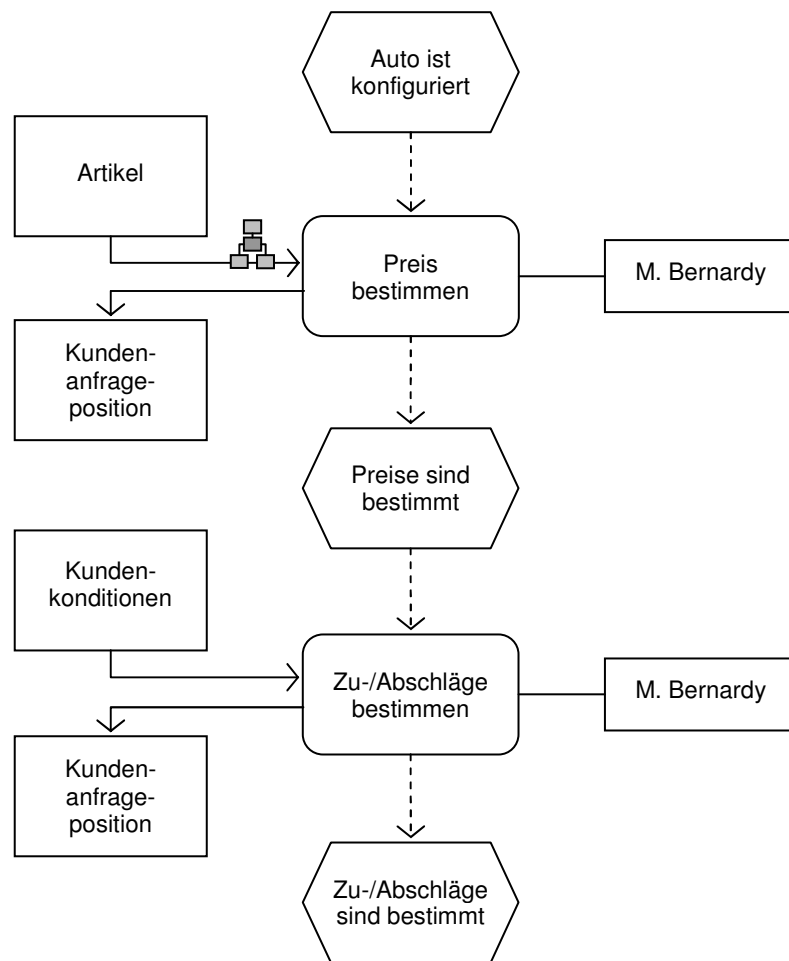


Abbildung 4: Prozesssicht in ARIS²⁰

Abbildung 4 zeigt einen Ausschnitt aus der eEPK einer Kundenanfragebearbeitung aus dem Beispielmodell „General Cars Coop.“ des ARIS-Tool-Sets. Die Symbole haben die folgende Bedeutung:

- Sechsecke (wie „Auto ist konfiguriert“): Ereignisse,
- Abgerundete Rechtecke (wie „Preis bestimmen“): Funktionen,
- Hohe Rechtecke (wie „Artikel“): Entitäten,
- Flache Rechtecke (wie „M. Bernardy“): Stelleninhaber.

Die Funktion „Preis bestimmen“ wird durch das Ereignis „Auto ist konfiguriert“ ausgelöst und resultiert im Ereignis „Preise sind bestimmt“. Diese Funktion wird von

²⁰ IDS Scheer (1999), Beispielmodelle „General Cars Coop.“.

M. Bernardy ausgeführt; sie greift auf Artikeldaten zu und modifiziert Daten von Kundenanfragepositionen.

Datenobjekte werden in der Prozesssicht nicht in ihrem sachlogischen Zusammenhang dargestellt – dafür ist die Datensicht (Abbildung 3) zuständig – sondern nur in ihrem Bezug zu den Funktionen, mit denen sie über Datenflüsse verbunden sind. Sie verweisen also lediglich auf Entitätstypen in einem korrespondierenden Datenmodell.²¹

2.5 Exkurs zur Geschäftsprozess- und Objektmodellierung

Die Gesamtheit der vier zuvor behandelten Sichten konstituiert das, was in der Wirtschaftsinformatik heute unter Geschäftsprozessmodellierung verstanden wird, zumindest dann, wenn man der breit akzeptierten ARIS-Philosophie folgt. Um anderen wirtschaftsinformatischen Ansätzen der Geschäftsprozessmodellierung²² nicht Unrecht zu tun, müsste man einschränkend von *ARIS-orientierter Geschäftsprozessmodellierung* sprechen, was hier jedoch der Einfachheit halber und auch deshalb unterlassen wird, weil sich das ARIS-Konzept in der Wirtschaftsinformatik inzwischen breiter Unterstützung erfreut.²³ Auch sei nicht verschwiegen, dass es mehr informatikaffine, aus der Objektmodellierung stammende Ansätze wie die *objektorientierte Geschäftsprozessmodellierung*²⁴ gibt, die auf der Unified Modeling Language (UML) aufbauend ARIS Paroli bietet.

Während Entitäten samt ihrer Attribute und Beziehungen untereinander die Struktur der Informationsobjekte lediglich statisch abbilden, betrachten Objekte zusätzlich das dynamische Verhalten.²⁵ „Vereinfacht formuliert sind Daten passiv, Objekte aktiv“.²⁶

Die **Objektmodellierung** ist ein umfassenderes Konzept als die Datenmodellierung:

- Klassen bzw. Objekte sind um Funktionen – genauer: die auf sie anzuwendenden Operationen – erweiterte Entitätstypen bzw. Entitäten.
- Klassendiagramme sind um Operationen erweiterte ER-Diagramme.
- Klassenmodelle werden durch objektrelationales Mapping auf (relationale) Datenmodelle abgebildet.

Datenmodellierung und Objektmodellierung stehen jedoch nicht in Konkurrenz zueinander. Vielmehr ist die Datenmodellierung eine echte Teilmenge der Objektmodellierung (Abbildung 5).

²¹ Rautenstrauch/Schulze (2003), S. 245.

²² Österle/Blessing (2005).

²³ Vgl. z.B. Fink/Schneiderei/Voß (2001), S. 94-101; Hansen/Neumann (2005), S. 178-185, 233-238; Schwarzer/Krcmar (2004), S. 97 f.

²⁴ Vgl. Oestereich/Weiss/Schröder/Weilkins/Lenhard (2003) und Oestereich (2005).

²⁵ Vgl. Abschnitt 4 zur Unterscheidung statischer und dynamischer Modelle.

²⁶ Stahlknecht/Hasenkamp (2005), S. 271.

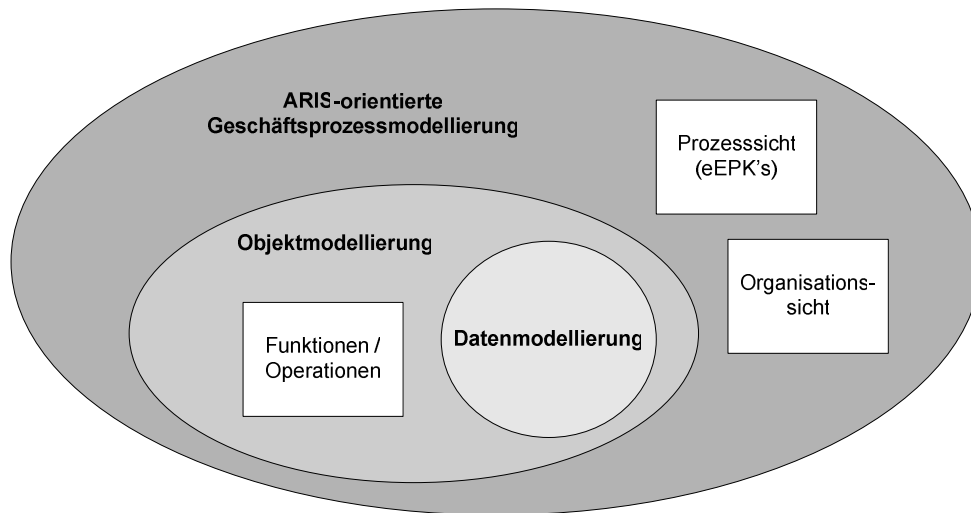


Abbildung 5: Geschäftsprozess-, Objekt- und Datenmodellierung

Nicht ganz so einfach ist das Verhältnis von ARIS-Geschäftsprozessmodellierung zur Objektmodellierung zu fassen. Im ARIS-Konzept der Beschreibungssichten gibt es keine explizite „Objektsicht“, obwohl Objekte die Datensicht und die Funktionssicht bündeln, da sie Entitäten, deren Eigenschaften und Beziehungen untereinander um Funktionen (Methoden) erweitern. Dadurch werden aus den ER-Diagrammen der Datensicht Klassendiagramme der Objektmodellierung.

Objektorientierung im Allgemeinen und die UML im Besonderen werden in der ARIS-Literatur und im ARIS Tool Set berücksichtigt. Das Methodenhandbuch zum ARIS Tool Set erläutert das wechselseitige Mapping von ARIS- und UML-Objekten²⁷ und die Hinterlegungsmöglichkeiten von ARIS- und UML-Modellen²⁸ als Mechanismen, wie die beiden Modellierungswelten aufeinander bezogen werden können. Eine „Objektsicht“ – als Ersatz für die Daten- und Funktionssicht – wurde in das ARIS-Konzept damit jedoch nicht aufgenommen, weshalb dieser Begriff in Abbildung 1 in Anführungszeichen und eckige Klammern gesetzt ist.

Eine Objektsicht wäre im Gegensatz zur Datensicht und zur Funktionssicht nämlich keine Partialsicht, sondern eine Integrationssicht. Als solche fungiert aber in ARIS bereits die Steuerungssicht. Deshalb ist es systematisch korrekt, Klassendiagramme als

²⁷ IDS Scheer (2004), S. 5/14-5/17.

²⁸ IDS Scheer (2004), S. 5/19-5/21.

Kombination von Funktions- und Datensicht unter die Steuerungssicht zu subsumieren.²⁹

Die Frage der Existenz einer „Objektsichtsicht“ in ARIS ist keine begriffliche Haarspalterei, sondern von sehr praktischer Bedeutung. Bei Modellierung einer Objektsicht wäre eine gesonderte Modellierung der Datensicht und der Funktionssicht redundant und damit unter dem Aspekt der Modellierungseffizienz (→ Einfachheitsprinzip) obsolet. Das sich hier öffnende weite Feld, wie denn Objektorientierung und Prozessorientierung zusammenzubringen wären, kann hier nur mit dem Hinweis auf einige divergenten Entwurfsparadigmen angedeutet werden³⁰.

ARIS-Geschäftsprozessmodellierung und Objektmodellierung stehen paradigmatisch in einer nicht abschließend geklärten Konkurrenz. Aus der Perspektive der Beschreibungssichten erscheint die ARIS-Geschäftsprozessmodellierung als der umfassendere Ansatz, weil er eine gegenüber dem rudimentären Akteurskonzept der UML semantisch reichhaltigere Organisationssicht und mit der Prozesssicht über ein auch für Nicht-Informatiker intuitiv anschauliches Integrationskonzept verfügt.

3 Modellierung nach Beschreibungsebenen

Anders als die Beschreibungssichten sind die Beschreibungsebenen nicht vom Bestreben nach Komplexitätsreduktion motiviert. Hier geht es vielmehr um die Unterscheidung von Modellen nach ihrer Nähe zur informationstechnischen Implementierung³¹ [RaSc03: 227] bzw. um eine sehr grobe Strukturierung eines Projekts in Phasen [Seid02: 23]. Bezug nehmend auf das ARIS-Konzept unterscheidet man drei Beschreibungsebenen, denen im Großen und Ganzen drei Phasen korrespondieren:

²⁹ Scheer (1998), Abschnitt A.III.2.1.1.1. Im Gegensatz dazu provoziert die Zuordnung von Objekten = Daten + Funktionen zu den Beschreibungsebenen (Rautenstrauch/Schulze (2003), S. 230) das Missverständnis einer eigenständigen Objektsicht.

³⁰ Scheer (1998), S. 136 und weiterführend z.B. Nüttgens/Hoffmann/Feld (1998); Loos/Allweyer (1998); Borberg/Müller/Kick (2000); Bühler (2000); Scheuch (2004).

³¹ Das Kriterium wachsender Implementierungsnähe wird auch anderen Ansätzen der Ebenenbildung zugeschrieben (vgl. Alpar/Grob/Weimann/Winter (2002), S. 112 f.): Multiperspektivische Unternehmensmodellierung: - Strategische Perspektive, - Organisatorische Perspektive; Informationsperspektive Business Engineering: - Strategie, - Prozess, - Informationssystem; Semantisches Objektmodell: - Unternehmensplan, - Geschäftsprozessmodelle, - Anwendungssysteme.

Beschreibungsebene	Phase	Ziel
Fachkonzept	Analyse	Beschreibung der fachlichen Anforderungen
DV-Konzept	Entwurf (Design)	Vorgaben für die informationstechnische Umsetzung
Implementierung	Realisierung	Umsetzung in einer spezielle IT-Struktur

Abbildung 6: Phasenbezug der Beschreibungsebenen

Dabei sind die Übergänge zwischen den Phasen fließend. So beschränken einige Autoren die Analysephase nicht nur auf die fachlichen Anforderungen, sondern beziehen auch informationstechnische Aspekte mit ein. Z.B. subsumieren Stahlknecht und Hasenkamp den IT-Grobentwurf unter die Analysephase, den IT-Detailentwurf hingegen unter die Entwurfsphase³². Zusätzlich gerät die Abgrenzung von Entwurf und Implementierung beim Einsatz von Generatoren ins Schwimmen, die im Wege des forward engineering aus Entwurfsmodellen, z.B. Tabellen einer relationalen Datenbank (Datenhaltungsebene), kompilierbare Quelltexte von Prozeduren (Applikationsebene) oder Bildschirmmasken (Präsentationsebene) erzeugen. Aus Platzgründen wird hier lediglich auf die drei Beschreibungsebenen der Datensicht eingegangen.

3.1 Fachkonzeptuelle Ebene

Auf der Ebene des Fachkonzepts werden zunächst die relevanten Datenobjekte und die zwischen ihnen bestehenden Beziehungen in einem *semantischen Datenmodell* beschrieben. Der Begriff *semantisch* weist darauf hin, dass hier die fachliche Bedeutung der Daten im Mittelpunkt steht. Grundsätzlich könnte ein semantisches Datenmodell freitextlich oder tabellarisch mit einem Textverarbeitungs- oder Tabellenkalkulationsprogramm (semi-strukturierter Text) erstellt werden. Das ist aber überflüssig, da semantische Datenmodelle grundsätzlich visualisiert werden sollten, wofür zumindest ein Zeichenprogramm, am besten jedoch ein ER-Modellierungswerkzeug eingesetzt wird. In der Praxis spricht man deshalb statt von semantischen Datenmodellen direkt von *ER-Modellen* (ERM) oder *ER-Diagrammen* (ERD).

Häufig liest man noch, dass im nächsten Schritt aus dem semantischen Datenmodell ein **logisches Datenbankmodell** zu entwickeln sei, in dem man sich je nach verwendetem Datenbankverwaltungssystem für ein

- hierarchisches Datenbankmodell,
- Netzwerk-Datenbankmodell oder
- relationales Datenbankmodell

³² Stahlknecht/Hasenkamp (2005), S. 210 f.

zu entscheiden habe.³³ Da erstens bei neu zu entwickelnden datenbankgestützten betrieblichen Informationssystemen fast nur noch relationale Datenbanken zum Einsatz kommen, implizieren heutige logische Datenbankmodelle stets ein relationales Datenbankmodell. Da zweitens relationale Datenbankmodelle de facto stets durch ER-Modelle abgebildet werden, ist die gesamte feinsinnige Unterscheidung von semantischen Datenmodellen, logischen Datenbankmodellen und relationalem Datenbankmodell nur noch von theoretischem und historischem Interesse.

3.2 DV-konzeptuelle Ebene

Mit der Entscheidung für ein relationales Datenmodell hat man bereits die Ebene des DV-Konzepts betreten, der folgende weitere Arbeitsschritte zugeordnet werden.³⁴

- Umformung der Entitäten des Fachkonzepts (*logisches Modell*) in Relationen, sprich: in Tabellen (*physikalisches Modell*). Leistungsfähige Datenmodellierungswerkzeuge erlauben die parallele Verwaltung beider Modelle, so dass eine gesonderte „Umformung“ gar nicht stattfindet.
- Normalisierung der Tabellen, soweit dies nicht durch normalisierungskonforme Zuordnung der Attribute zu den Entitäten im Fachkonzept bereits implizit erledigt wurde.
- Hinzufügen von Integritätsbedingungen, vor allem der Entitätsintegrität und der referentiellen Integrität.
- Umformung des Relationenschemas in die Datenbeschreibungssprache des jeweils eingesetzten Datenbankverwaltungssystems, praktisch: Erstellung eines SQL-Skripts mit Datendefinitionsbefehlen (CREATE) aus dem grafischen Datenmodell. Bei Einsatz eines Datenmodellierungswerkzeugs muss der Modellierer nicht viel mehr tun als das spezifische Datenbankverwaltungssystem, z.B. den Microsoft SQL Server, Oracle, DB2 oder auch Microsoft Access, auszuwählen und dieses Skript automatisiert erstellen lassen.

3.3 Implementierungsebene

Vom DV-Konzept zur Implementierung ist es in der Datensicht nur ein kleiner Schritt, der im Wesentlichen darin besteht, mit einem Administrationswerkzeug des jeweiligen Datenbankverwaltungssystems das zuvor erzeugte SQL-Skript ablaufen zu lassen, m. a. W.: die im Datenmodell visuell entworfenen Datenobjekte als konkrete Datenbankobjekte zu erzeugen. Viele Datenmodellierungswerkzeuge automatisieren diesen Schritt, indem sie sich direkt mit der zuvor manuell anzulegenden, leeren Datenbank verbinden und die Generierung der Datenbankobjekte ohne Zuhilfenahme eines SQL Server-Administrationswerkzeugs in eigener Regie durchführen. Dieser Vorgang wird als *for-*

³³ Z.B. Stahlknecht/Hasenkamp (2005), S. 164.

³⁴ Scheer (1998), S. 80.

ward engineering bezeichnet. In der Terminologie des ANSI-Datenbankmodells kann man die Implementierung als Abbildung des auf der DV-Konzeptebene angesiedelten konzeptionellen Datenbankschemas auf ein *internes Datenbankschema* kennzeichnen.

In den anderen Sichten ist der Übergang vom DV-Konzept zur Implementierung aufwändiger und wird von Werkzeugen des Computer Aided Software Engineering (CASE) weniger unterstützt. Insbesondere in der Funktionssicht hat der Entwickler die meiste Arbeit, nämlich die Erstellung des Programmcodes für die Applikations- und Präsentationslogik, noch vor sich.

4 Modellierung von Struktur und Verhalten

Traditionell unterscheidet man nicht nur in der Softwareentwicklung, sondern auch in vielen anderen Fachwissenschaften zwischen statischen und dynamischen Modellen. Da jedes System letztlich statische und dynamische Aspekte aufweist, handelt es sich bei dieser Unterscheidung wiederum um ein spezielles Sichtenkonzept, bei dem das System entweder zeitpunktbezogen oder zeitraumbezogen modelliert wird.

Die Klassifizierung von Modellen als statisch oder dynamisch ist nicht immer ganz eindeutig. Im ARIS-Konzept bestehen die Datensicht, die Funktionssicht und die Organisationssicht aus statischen Modellen, die Prozesssicht hingegen aus dynamischen Modellen, weil diese – allen voran die ereignisgesteuerten Prozessketten – die Abläufe und damit das zeitbezogene Verhalten beschreiben. Das dynamisierende Modellelement sind dabei die Ereignisse.

Da viele Ereignisse in dem Sinne datenorientiert sind, als sie mit dem Erzeugen oder Löschen von Informationsobjekten bzw. dem Ändern ihrer Eigenschaften einhergehen, werden sie in der Literatur häufig der Datensicht zugeordnet.³⁵ Da ER-Modelle aber nur Modellelemente enthalten, die statische Zustände von Informationsobjekten und keine dynamischen Zustandsübergänge beschreiben, bleibt die Datensicht statisch.³⁶

Analoges gilt auch für die statischen Klassendiagramme der Objektmodellierung. Die Hinzunahme intern dynamischer, zustandsändernder Funktionen macht aus einem statischen ER-Modell kein dynamisches Klassenmodell. Die Auflistung der Funktionen einer Klasse sagt nichts über deren innere Dynamik aus, die durch dynamische Aktivitätsdiagramme abgebildet wird. Die Funktionen in einem Klassendiagramm sind eine statische Liste und nicht minder statisch als ein Funktionsbaum.

³⁵ Vgl. Seidlmeier (2002), S. 15 und IDS Scheer (1999a), S. 2, 4.

³⁶ Auch die explizite Modellierung der Zeit vor allem in sog. temporalen Datenmodellen ändert daran nichts. Die Abbildung z.B. einer Artikelpreishistorie durch die Attribute „Preis“ und „Gültig_seit“ erlaubt die Speicherung und Ermittlung des zu jedem beliebigen Zeitpunkt gültigen Preises, sagt aber nichts über den Preisänderungsmechanismus aus. Deshalb wird ein Modell, das Größen enthält, die sich auf verschiedene Zeiträume beziehen (Alpar/Grob/Weimann/Winter (2002), S. 22), nicht bereits zu einem dynamischen Modell.

Dessen ungeachtet bewegen sich Datenmodelle, die mit leistungsfähigen Datenmodellierungswerkzeugen angelegt werden, insofern an der Grenze von statischen zu dynamischen Modellen, da mit ihnen zustandsändernde Regeln der referentiellen Integrität (RI) definiert werden können. Schon der Oberbegriff „RI-Aktion“ für Operationen wie das kaskadierende oder das nullifizierende Löschen weist auf die inhärente Dynamik hin. Dies gilt erst recht für beliebige Trigger, mit denen ein fachlich noch erheblich weiteres Spektrum von Geschäftsregeln³⁷ implementiert werden kann. Ihre Formulierbarkeit als sog. ECA-Regeln (Event – Condition – Action)³⁸ charakterisiert sie bereits als dynamisch. Auch wenn solche Trigger meist erst bei der Implementierung in einer Datenbank in der Programmiersprache dieser Datenbank ausformuliert werden, so können sie mit einigen Datenmodellierungswerkzeugen in einer RDBMS-neutralen Meta-sprache bereits im ER-Modell – allerdings textuell und nicht grafisch – modelliert werden.

5 Modellierung nach Geltungsanspruch

Die Differenzierung von Modellen nach ihrem Geltungsanspruch in Istmodelle und Sollmodelle ist in der Wirtschaftsinformatik wahrscheinlich von größerer Bedeutung als in anderen Disziplinen. Istmodelle stehen in einem engen Zusammenhang mit dem Erklärungsanspruch der Modellierung, Sollmodelle mit deren Gestaltungsanspruch.

5.1 Istmodelle

Da ein Istmodell ein betriebliches Informationssystem im vorhandenen Zustand abbildet, setzt es die Existenz eines solchen Informationssystems voraus. In den beiden folgenden Konstellationen trifft dies aber nicht zu:

- Wenn man die Computernutzung als Indiz für ein vorhandenes Informationssystem betrachtet, dann war ein solches 2003 zwar in ca. 80% der deutschen Unternehmen vorhanden³⁹, was ex negativo impliziert, dass in 20% der Unternehmen kein Informationssystem und damit auch kein Istmodell desselben existieren kann. Im Grundstücks- und Wohnungswesen und im Gastgewerbe gilt das sogar für etwa die Hälfte der Unternehmen.
- In der Frühphase einer Unternehmensgründung existiert weder ein betriebliches Basissystem noch ein betriebliches Informationssystem.

Auch wenn betriebliche Informationssysteme heute eher die Regel als die Ausnahme sind, so dürften diese in vielen Unternehmen – insbesondere in KMU – nicht durch ein Ist-Informationsmodell abgebildet sein. Und wenn es überhaupt Istmodelle gibt, so ist

³⁷ Die Summe der Geschäftsregeln wird gelegentlich in Abgrenzung von der referentiellen Integrität auch als Anwendungsintegrität bezeichnet.

³⁸ Knolmayer/Herbst (1993).

³⁹ Statistisches Bundesamt (2004), S. 9.

deren Aktualität fraglich, wenn an diesen zwischenzeitlich vorgenommene Änderungen nicht kontinuierlich in den Istmodellen nachgehalten werden.

5.2 Sollmodelle

Sollmodelle stellen die Anforderungen an ein neu zu erstellendes oder zu modifizierendes Informationssystem dar. Obwohl es hierzu je nach Vorgehensmodell unterschiedliche Grundpositionen gibt, kann oder sollte nur bei sehr kleinen Softwareprojekten auf ein zuvor erstelltes Soll-Informationsmodell verzichtet werden. Bei der professionellen Entwicklung betrieblicher Individualsoftware spielen Sollmodelle hingegen eine wichtige Rolle. Bei der Einführung von betriebswirtschaftlicher Standardsoftware übernehmen meistens Referenzmodelle (vgl. Abschnitt 7) die Funktion von Sollmodellen.

Die Abbildung der fachlichen Anforderungen an die Anwendungssoftware eines betrieblichen Informationssystems kann das Sollmodell durch Merkmale einer vorhandenen und nicht oder nur begrenzt zur Disposition stehenden Systemplattform mehr oder weniger stark eingeschränkt werden. Man spricht dann zuweilen von restriktionsbehafteten oder restriktionsfreien Sollmodellen.⁴⁰ Da wirklich restriktionsfreie Sollmodelle in der Praxis so gut wie keine praktische Relevanz haben, ist es müßig, ob es Sinn macht, hierfür als dritte Kategorie den Begriff des Idealmodells einzuführen.⁴¹

6 Modellierung nach Geltungsbereich

Modelle betrieblicher Informationssysteme können prinzipiell unternehmensweit angelegt sein oder sich auf Ausschnitte der betrieblichen Realität beschränken. Der Geltungsbereich eines IS-Modells hängt davon ab, wie viele Funktionsbereiche und Geschäftsbereiche das modellierte Informationssystem abdeckt.

6.1 Unternehmensweites Modell

In der Datenmodellierung werden unternehmensweite Modelle als **Unternehmensdatenmodelle** (UDM) bezeichnet, die einst große Hoffnungsträger für die Schaffung hoch integrierter Informationssysteme waren. Auch wenn sich die Erwartungen – wie bei manch anderer IT-Hoffnung auch – nicht in jedem Fall und oft gar nicht erfüllt haben,⁴² so kommen die in jedem umfassenden, datenbankbasierten ERP-System zumindest implizit enthaltenen Datenmodelle dem Ideal des Unternehmensdatenmodells noch am nächsten. Je mehr sich das Informationssystem eines Unternehmens jedoch aus mehr oder weniger lose gekoppelten, heterogenen Einzelanwendungen zusammensetzt, umso weiter entfernt man sich von diesem Ideal.

⁴⁰ Vgl. Schütte (1998), S. 66.

⁴¹ Dafür: Rautenstrauch/Schulze (2003), S. 228. Dagegen: Schütte (1998), S. 66.

⁴² Rautenstrauch/Schulze (2003), S. 241 f.

6.2 Partialmodell

Informationssystemen, die nur einzelne Funktions- oder Geschäftsbereiche abdecken, liegen Partialmodelle zu Grunde. Bricht man diese bis auf einzelne Geschäftsobjekte herunter, so kommt man zu den kleinsten fachlich sinnvollen Modellierungseinheiten, die man als **Geschäftsobjektmodelle** bezeichnen kann.

Es mag dahinstehen, ob man auch die durch **modelltechnische**, hierarchische Zerlegung entstehenden **Bereichsmodelle** eines Gesamtdatenmodells (in ERwin heißen sie z.B. subject areas) zu den Partialmodellen zählen möchte. Dies ist fragwürdig, weil es hier nicht um den Abdeckungsgrad des betrieblichen Basissystems durch das betriebliche Informationssystem geht, sondern um unterschiedliche Abstraktionsebenen, ohne die bei sehr großen Datenmodellen der Überblick verloren gehen kann. Streng genommen handelt es sich um partielle Sichten auf ein ganzheitliches Modell.

7 Modellierung nach Individualität

Während *unternehmensspezifische Modelle* sowohl als Ist- wie auch als Sollmodelle nur für ein bestimmtes Unternehmen gelten, erheben *Referenzmodelle* den Anspruch einer größeren Allgemeingültigkeit für eine ganze Gruppe gleichartiger Unternehmen. Wissenschaftstheoretisch ist ein Referenzmodell wegen der von ihm beanspruchten Gültigkeit für viele Unternehmen eine Sammlung kühnerer betriebswirtschaftlicher Hypothesen als ein unternehmensspezifisches Modell. Da Modelle betrieblicher Informationssysteme aber keine Wahrheitsansprüche, sondern Gestaltungsansprüche verfolgen, ist hiermit keine negative Wertung verbunden.

Referenzmodelle sind per se Sollmodelle und nur in dem Grenzfall Istmodelle, wenn überhaupt keine Anpassungen an Besonderheiten des betrieblichen Basissystems des Anwenderunternehmens vorgenommen werden. Ein im Wege des **Customizing** an die Anforderungen eines bestimmten Unternehmens angepasstes Referenzmodell mutiert zu einem Mittelding zwischen Referenzmodell und unternehmensspezifischem Modell, das als **individualisiertes Referenzmodell** bezeichnet werden kann. Damit soll einerseits das Unternehmensspezifische durch das Customizing und andererseits die Herkunft aus einem Referenzmodell betont werden.

Referenzmodelle und unternehmensspezifische Modelle verhalten sich sowohl induktiv wie auch deduktiv zu einander:

- *Induktiv*, weil Referenzmodelle in der Praxis nicht zuletzt aus der Verallgemeinerung vieler unternehmensspezifischer Modelle entwickelt werden.
- *Deduktiv*, weil bei der Einführung von Standardsoftware das unternehmensspezifische Sollmodell aus dem Referenzmodell – so vorhanden – übernommen bzw. mehr oder weniger angepasst abgeleitet wird.

8 Modellierung nach Abstraktionsgrad

Schließlich kann man „Modelle über Modelle“ als so genannte *Metamodelle* von den „normalen“ *Modellen auf der Ausprägungsebene* abgrenzen. „Ein Metamodell beschreibt, wie Phänomene eines Realitätsausschnitts durch entsprechende Modellelemente abgebildet werden können. Als Modell wird dann ein bestimmter, mit Hilfe eines Metamodells abgebildeter Realitätsausschnitt bezeichnet.“⁴³

Grundsätzlich lassen sich alle Modellierungsansätze auf der Metaebene mit Entity-Relationship-Modellen beschreiben. Das gilt natürlich auch für die ER-Modellierung selbst.

Nicht nur die ER-Modellierung, sondern auch andere Modellierungsansätze können mit einem **ER-Metamodell** sehr anschaulich beschrieben werden, wie es Scheer für das ARIS-Konzept demonstriert hat.⁴⁴ Ähnliches gilt für die Beschreibung der UML-Objektmodellierung durch **UML-Metamodelle**, insbesondere Klassenmodelle, und für die **Beschreibung von Vorgehensmodellen mit ereignisorientierten Prozessketten**, die dann ebenfalls Metamodellcharakter haben.

9 Modellierung nach Darstellungstechnik

9.1 Grafisches Modell

Die dominierende Darstellungstechnik für IS-Modelle ist die grafische. Dies gilt

- für alle Beschreibungssichten (vgl. Abschnitt 2),
- vor allem auf der fachkonzeptuellen und auf der DV-technischen Beschreibungsebene (vgl. Abschnitt 3),
- für Ist- und Sollmodelle sowie (vgl. Abschnitt 5),
- für unternehmensspezifische wie Referenzmodelle (vgl. Abschnitt 7).

9.2 Natürlichsprachliches Modell

Natürlichsprachliche, textuelle Beschreibungen spielen bei Datenmodellen nur eine untergeordnete, ergänzende Rolle. Keinesfalls können sie grafische Datenmodelle ersetzen. Dem widerspricht nicht, dass wichtige Eigenschaften eines grafischen Datenmodells auch aus textuellen Anforderungsbeschreibungen für ein Informationssystem destilliert werden. Auch kommt die natürliche Sprache für die detaillierte Beschreibung von Informationsobjekten z.B. in Fachbegriffsmodellen oder bei der Kommentierung

⁴³ Alpar/Grob/Weimann/Winter (2002), S. 138.

⁴⁴ Scheer (1998).

von Entitäten, Attributen und Beziehungen zum Zug. Eine geschlossene, rein verbale Beschreibung von Datenmodellen ist jedoch unüblich, da sie unübersichtlich und nicht in Elemente eines Informationssystems transformierbar wäre.

Semi-formale Darstellungstechniken kommen bei der Modellierung der Ablauflogik von Funktionen (Pseudocode) und bei der Modellierung von Prozessen (Geschäftsprozessschablonen) zum Einsatz. Bei der Datenmodellierung sind sie hingegen unwichtig.

9.3 Computersprachliches Modell

Von großer Bedeutung ist natürlich die computersprachliche Modellierung von IS-Modellen, weil sie den Übergang zur Implementierungsebene nachhaltig erleichtert. Unter Automatisierungsaspekten besonders sind dabei Modelle der Datensicht, deren Elemente nahtlos in Elemente der Datenhaltungsschicht überführt werden können. Wird die Datenhaltung durch eine relationale Datenbank realisiert, dann erfolgt die computersprachliche Modellierung mit den CREATE-Befehlen der Standardsprache SQL. Ein großer Vorteil von Datenmodellierungswerkzeugen besteht darin, dass sie das grafische Modell sehr einfach in ein computersprachliches Modell übersetzen können.

9.4 Mathematisches Modell

Mathematische Modelle spielen für betriebliche Informationssysteme insbesondere dort eine Rolle, wo Optimierungsverfahren zum Zuge kommen. Für operative Informationssysteme bedient sich die Wirtschaftsinformatik vorwiegend beim Operations Research, für analytische Informationssysteme greift sie auf statistische Modelle und Verfahren z.B. des Data Mining zurück.

10 Modellierung nach Modellzweck

In der Modelltheorie unterscheidet man Erklärungs- und Entscheidungsmodelle. *Erklärungsmodelle* werden häufig mit dem Schlagwort „deklarativ“ versehen und sind vor allem für die Theoriebildung bedeutsam. Auch wenn es nicht ihr Hauptzweck ist, so ist aus IS-Modellen einiges über die strukturellen Zusammenhänge von Informationssystemen zu lernen. Sie tragen erheblich zu deren Verständnis bei. Noch prägnanter wird der theoriebildende Anspruch im maßgeblich von Scheer geprägten Paradigma der „EDV-orientierten Betriebswirtschaftslehre“⁴⁵ und in Referenzmodellen, die auch als in IS-Modelle geronnene BWL gelesen werden können.

Für die Modellierung betrieblicher Informationssysteme ist der pragmatische Anspruch eines **Entscheidungsmodells**, insbesondere in Form eines **Gestaltungsmodells**, noch bedeutsamer als der theoretische Anspruch eines Erklärungsmodells. Präskriptiv wirken Modelle für ein Informationssystem, indem sie diesem die Struktur seiner In-

⁴⁵ Scheer (1990).

formationsobjekte vorgeben, und mittelbar auch für ein betriebliches Basissystem selbst, wenn sie in der Gestalt von Referenzmodellen betrieblicher Standardsoftware in dessen Systemgestaltung eingreifen⁴⁶.

Literatur

- Alpar, P./Grob, H. L./Weimann, P./Winter, R. (2002): Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen, Wiesbaden.
- Borberg, B./Müller, H./Kick, U. (2000): Von der Geschäftsprozessmodellierung zur UML – eine pragmatische Methodik, in: OBJEKTSpektrum, Heft 2, S. 68-72.
- Bühler, F. (2000): Übergang von ARIS-Prozessmodellierung zu Objektorientierter Analyse/Design mit UML/Rose. Workshop “Wie funktioniert das Mapping”, Dresdner Bank, Frankfurt.
- Fink, A./Schneidereit, G./Voß, S. (2001): Grundlagen der Wirtschaftsinformatik, Heidelberg.
- Hansen, H. R./Neumann, G. (2005): Wirtschaftsinformatik, Stuttgart.
- IDS Scheer (1999): ARIS Version 4.1.1.1 - Beispieldatenbanken, Saarbrücken.
- IDS Scheer (1999a): ARIS Methode Version 4.1, Saarbrücken.
- IDS Scheer (2004): ARIS Methode (Handbuch zur ARIS 6 Collaborative Suite), Saarbrücken.
- Knolmayer, G./Herbst, H. (1993): Business Rules, in: Wirtschaftsinformatik, Heft 4, S. 386-390.
- Loos, P./Allweyer, Th. (1998): Process Orientation and Object-Oriented. An Approach for Integrating UML and Event-Driven Process Chains (EPC), in: Arbeitsberichte des Instituts für Wirtschaftsinformatik H. 144, Saarbrücken.
- Oestereich, B. (2005): Objektorientierte Geschäftsprozessmodellierung und modellgetriebene Softwareentwicklung, in: HMD - Praxis der Wirtschaftsinformatik, Heft 241, S. 27-34.
- Oestereich, B./Weiss, Ch./Schröder, C./Weilkins, T./Lenhard, A. (2003): Objektorientierte Geschäftsprozessmodellierung mit der UML, Heidelberg.
- Österle, H./Blessing, D. (2005): Ansätze des Business Engineering, in: HMD - Praxis der Wirtschaftsinformatik, Heft 241, S. 7-17.
- Rautenstrauch, C./Schulze, Th. (2003): Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker, Berlin.

⁴⁶ Fink/Schneidereit/Voß (2001), S. 92.

-
- Nüttgens, M./Hoffmann, M./Feld, Th. (1998): Objektorientierte Systementwicklung mit der Unified Modeling Language (UML), in: Scheer (1998), S. 197-203.
- Scheer, A.-W.(1990): EDV-orientierte Betriebswirtschaftslehre, Berlin.
- Scheer, A.-W. (1998): ARIS – Modellierungsmethoden, Metamodelle, Anwendungen, Berlin.
- Scheuch, R. (2004).: Mit der UML 2.0 Geschäftsprozesse modellieren: Ein Vergleich marktrelevanter Methoden und Alternativen, in: OBJEKTSpektrum, Heft 3, S. 29-33.
- Schütte, R. (1998): Grundsätze ordnungsmäßiger Referenzmodellierung, Wiesbaden.
- Schwarzer, B./Krcmar, H. (2004): Wirtschaftsinformatik. Grundzüge der betrieblichen Datenverarbeitung, Stuttgart.
- Seidlmeier, H. (2002): Prozessmodellierung mit ARIS, Wiesbaden.
- Stahlknecht, P./Hasenkamp, U. (2005): Einführung in die Wirtschaftsinformatik, Berlin 2005.
- Statistisches Bundesamt (2004): Informationstechnologie in Unternehmen. Ergebnisse für das Jahr 2003, Wiesbaden.

