Praktische Informatik für Wirtschaftsmathematiker, Ingenieure und Naturwissenschaftler I (PIWIN I, 3 V + 1 Ü) WS 2002/03

13. Vorlesungswoche

Softwaretechnik: Beschreibungsnotationen im Softwareentwurf

Konzepte der Strukturierten Analyse

Entity-Relationship Modell zur Datenmodellierung

Unterlagen: Gumm/Sommer, Kapitel 11

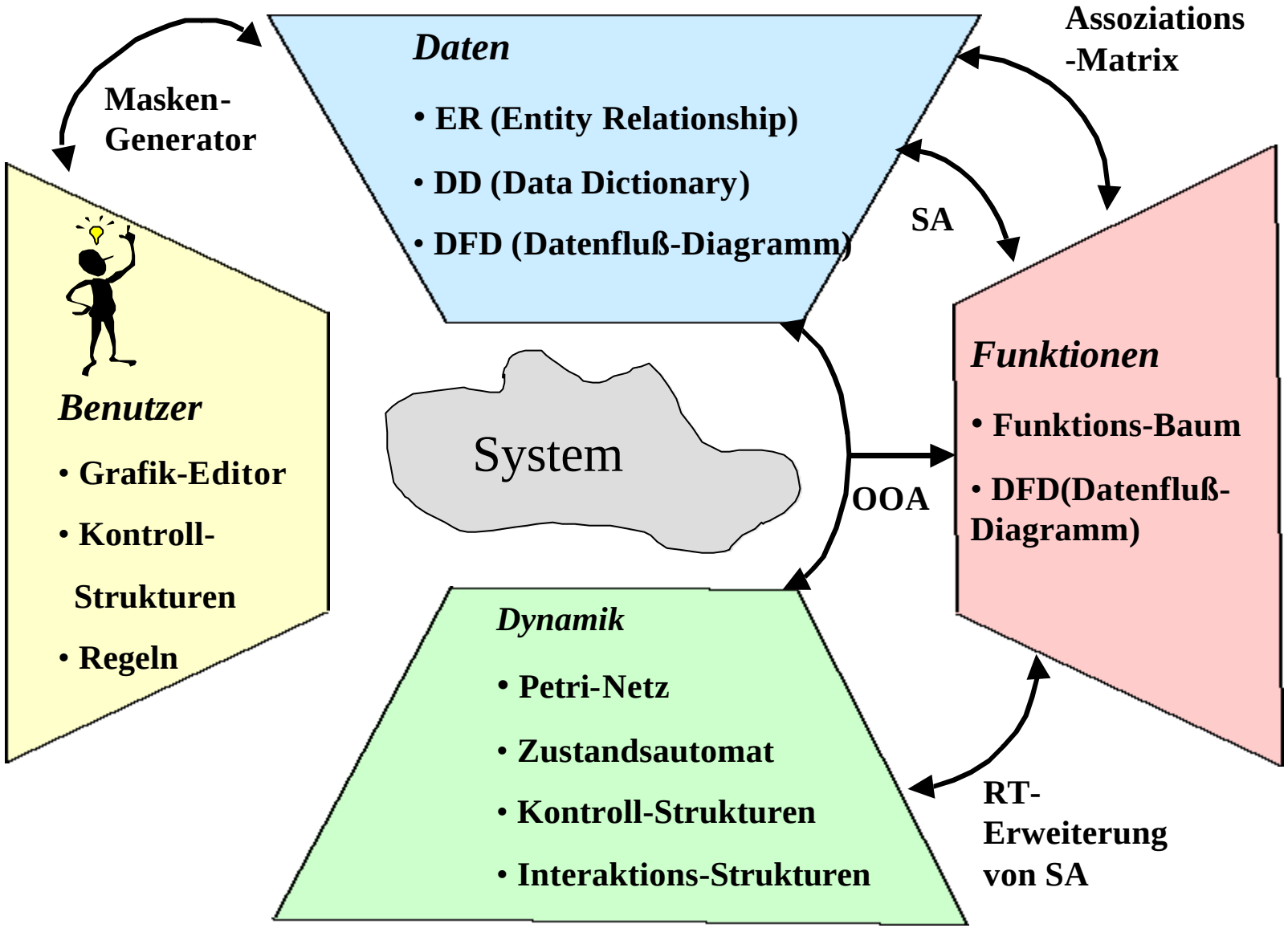
Helmut Balzert: Lehrbuch der Software-Technik 1 / 2,

Spektrum Akademischer Verlag, Heidelberg u.a. 2001

UB: Lehrbuchsammlung L Sr 389-1

Folien nach Heinecke (FH Gelsenkirchen), Heyer (Uni Leipzig)

Sichten und ihre Konzepte



Strukturierte Analyse

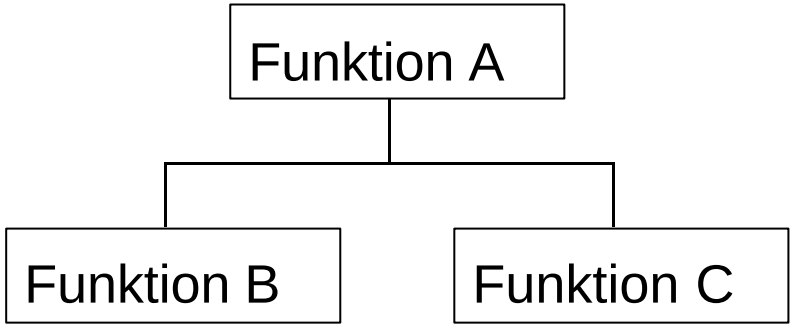
traditionelle Methodik,

enthält Notationen zur Beschreibung von

- Funktionalität: Funktionsbaum
- Wechselwirkung Funktionen&Daten: Datenflußdiagramm
- Daten: Data Dictionary
- Dynamik: Kontrollstrukturen, Entscheidungstabellen

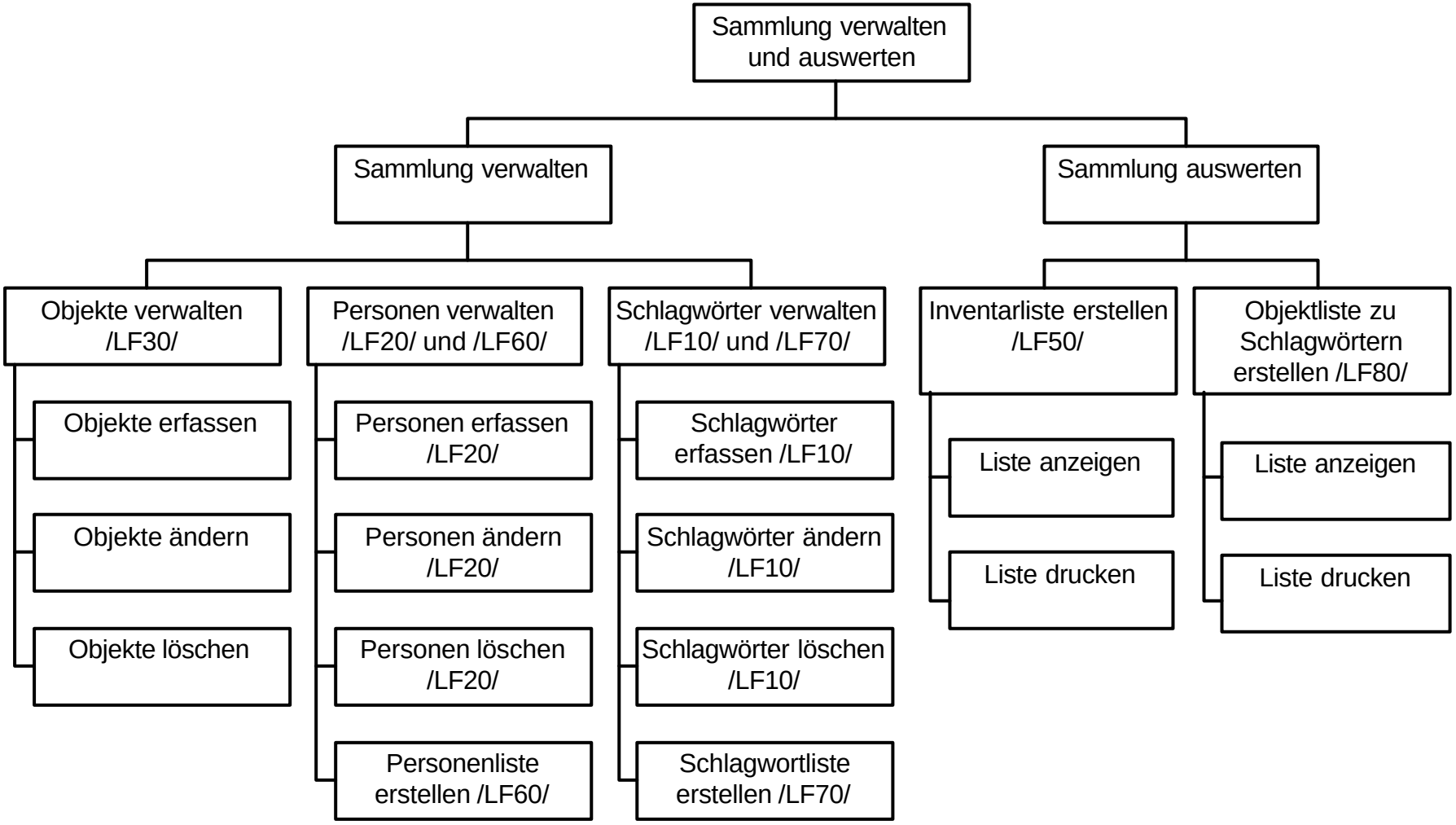
Funktionsbaum

- Hierarchie der Funktionen



- A ist B und C übergeordnet
  - "besteht aus" (meist in der Definitionsphase)
  - "ruft auf" (meist in der Entwurfsphase)
- Erstellungsregeln
  - nur fachlich eng verwandte Funktionen unter eine übergeordnete Funktion stellen ⇒ Fachwissen erforderlich
  - gleiche Hierarchieebene ⇔ gleiches Abstraktionsniveau

Beispiel: Funktionsbaum, Inventarverwaltung eines Museums

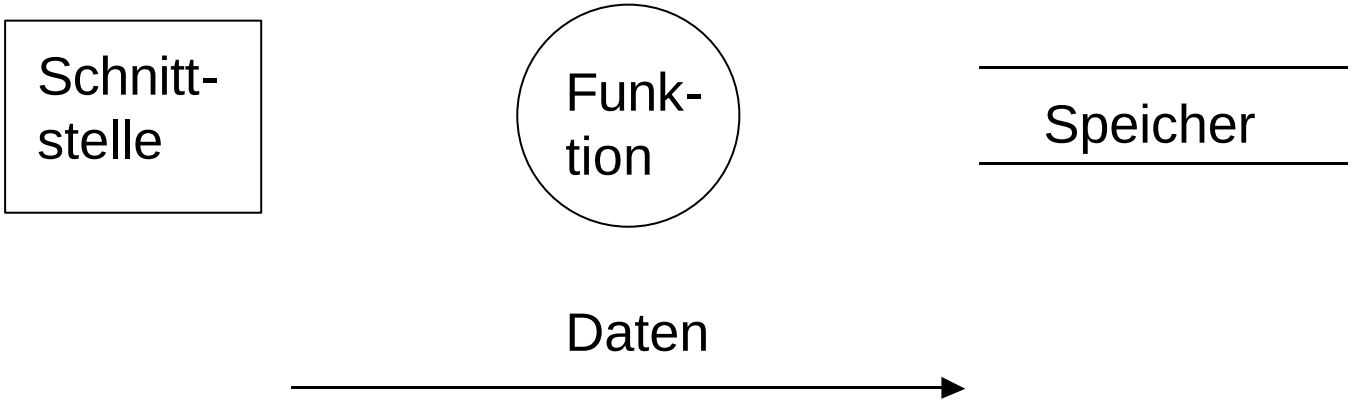


Bewertung des Konzeptes: Funktionsbaum

- 👍 bewährt zur systematischen Gliederung von Funktionen
- 👍 auch in anderen Kontexten verwendet, z.B. Organigramm
- 👍 mögliche Vorstrukturierung für Dialoggestaltung
- 👎 beschränkt auf funktionale Sicht

Datenflußdiagramm

- Weg der Daten / Informationen zwischen Funktionen, Speichern, Schnittstellen
  - genormte Darstellung nach DIN 66001 unüblich
  - hier Darstellung nach DeMarco (1979)
- Graphische Symbole
  - beschriftet mit aussagekräftigen Namen
  - beschreiben das System, als liefe es bereits



Datenflußdiagramm

Syntaktische Regeln

- mindestens eine Schnittstelle im Diagramm
- jede Schnittstelle i.d.R. nur einmal im Diagramm
- zwischen Schnittstellen keine direkten Datenflüsse
- zwischen Speichern keine direkten Datenflüsse
- zwischen Speicher und Schnittstelle keine direkten Datenflüsse
- Datenflüsse von und zu Speichern benötigen keinen Namen,
  wenn alle gespeicherten Daten transportiert werden

Semantische Regeln

- Beschreibung des Datenflusses, nicht der Kontrollstruktur:
  keine Entscheidungen oder Schleifen, keine Ablaufstruktur
- Angabe der tatsächlichen Datenquelle bzw. -senke:
  z.B. “Kunde” statt “Kundensachbearbeiter”
- Schnittstellen abstrahieren vom E/A-Medium:
  keine Schnittstelle “Drucker” oder “Tastatur”
- Datenflußname kann kein Verb enthalten, aber Adjektiv: z.B. “stornierte Buchungen”
- Funktionsname muß Verb und Objekt enthalten:
  z.B. “erstelle Quittung” oder “Quittung erstellen”
- Namen sollen aussagekräftig sein:
  nicht “Daten”, “Information”, “verarbeite ...”, “bediene...”

Datenflußdiagramm

- Bewertung des Konzepts
  - Leicht zu erstellen und leicht zu lesen
  - gut zu vermitteln an Auftraggeber, Benutzer etc.
  - mehr Information als im Funktionsbaum
  - für ganzes System schnell groß und unübersichtlich
  - kein einheitliches Abstraktionsniveau gewährleistet
  - keine detaillierte Beschreibung der Daten
- Kombination mit Funktionsbäumen
  - mehrere Datenflußdiagramme
  - jeweils nur Funktionen gleicher Hierarchiestufe zu einer gemeinsamen übergeordneten Funktion

Data Dictionary

- Datenkatalog mit Angaben über
  - Struktur der Daten
  - Eigenschaften der Daten
  - Verwendung der Daten
- Data Dictionary
  - angelegt in der Definitionsphase
  - weiter verwendet, ergänzt und verfeinert in der Entwurfs- und Implementierungsphase
  - erlaubt Konsistenz- und Redundanzüberprüfung
- Datenbeschreibung
  - Top-down
  - auf ausreichendem Abstraktionsniveau
  - mit Hilfe einer modifizierten Backus-Naur-Form (BNF)
- Beispiel

Kundendatei = {Kundeneintrag}
Kundeneintrag = Name + Adresse + (Geburtsdatum) + Funktion
Adresse = [Straße + Hausnr | Postfach] + (Länderkennz) + PLZ
+ Ort + 0{Telefon}2 + (Fax)

Bewertung des Konzepts Data Dictionary

- 👍 kompakte formale Beschreibung
- 👍 hierarchische Verfeinerung
- 👍 Ähnlichkeit mit Datenstrukturen in Programmiersprachen
- 👎 relativ schwer lesbar, vor allem für Anwender
  ⇒ Syntaxdiagramme als mögliche Alternative

Strukturierte Programmierung

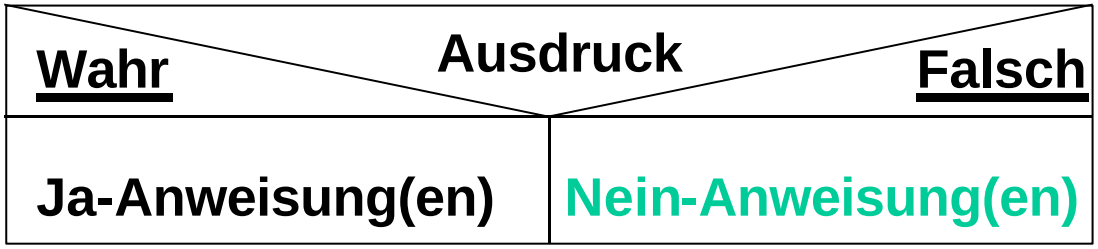
- Kontrollstrukturen für Algorithmen
  - Sequenz
  - Auswahl
  - Wiederholung
  - Aufruf anderer Algorithmen
- Strukturierte Programmierung
  - nur diese vier Kontrollstrukturen
  - Linearität und Lokalität
- Notationen für Kontrollstrukturen
  - Pseudo-Code
  - Struktogramme nach Nassi-Shneiderman
  - Programmablaufpläne (Flußdiagramme) nach DIN 66001
  - Warnier-Orr-Notation
  - Jackson-Diagramme

Beispiele für Notationen zu IF-THEN-ELSE

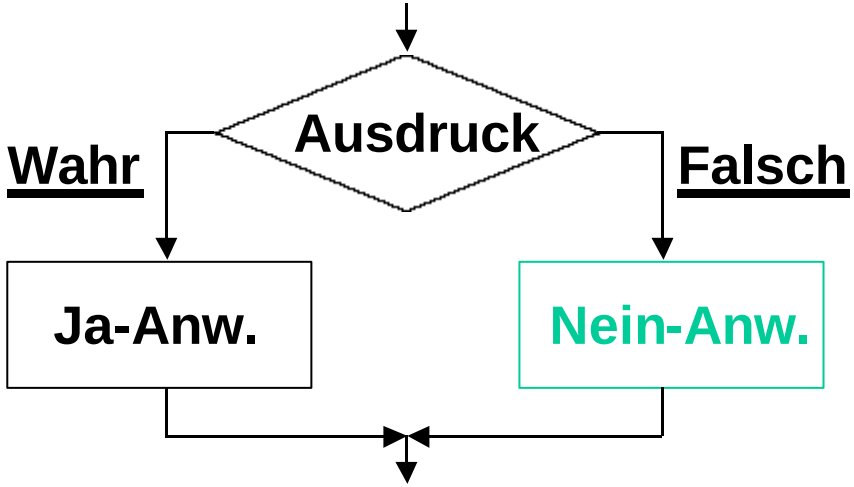
- Pseudocode

```
if Ausdruck
    then Ja-Anweisung(en)
    else Nein-Anweisung(en)
end if;
```

- Struktogramm



- Programmablaufplan



Strukturierte Analyse

- Hierarchiekonzept
- Kontextdiagramm
- Verfeinerte Datenflußdiagramme
- Einträge im Data Dictionary
- Datenintegrität
- Mini-Spezifikationen
- Methodik
- Bewertung

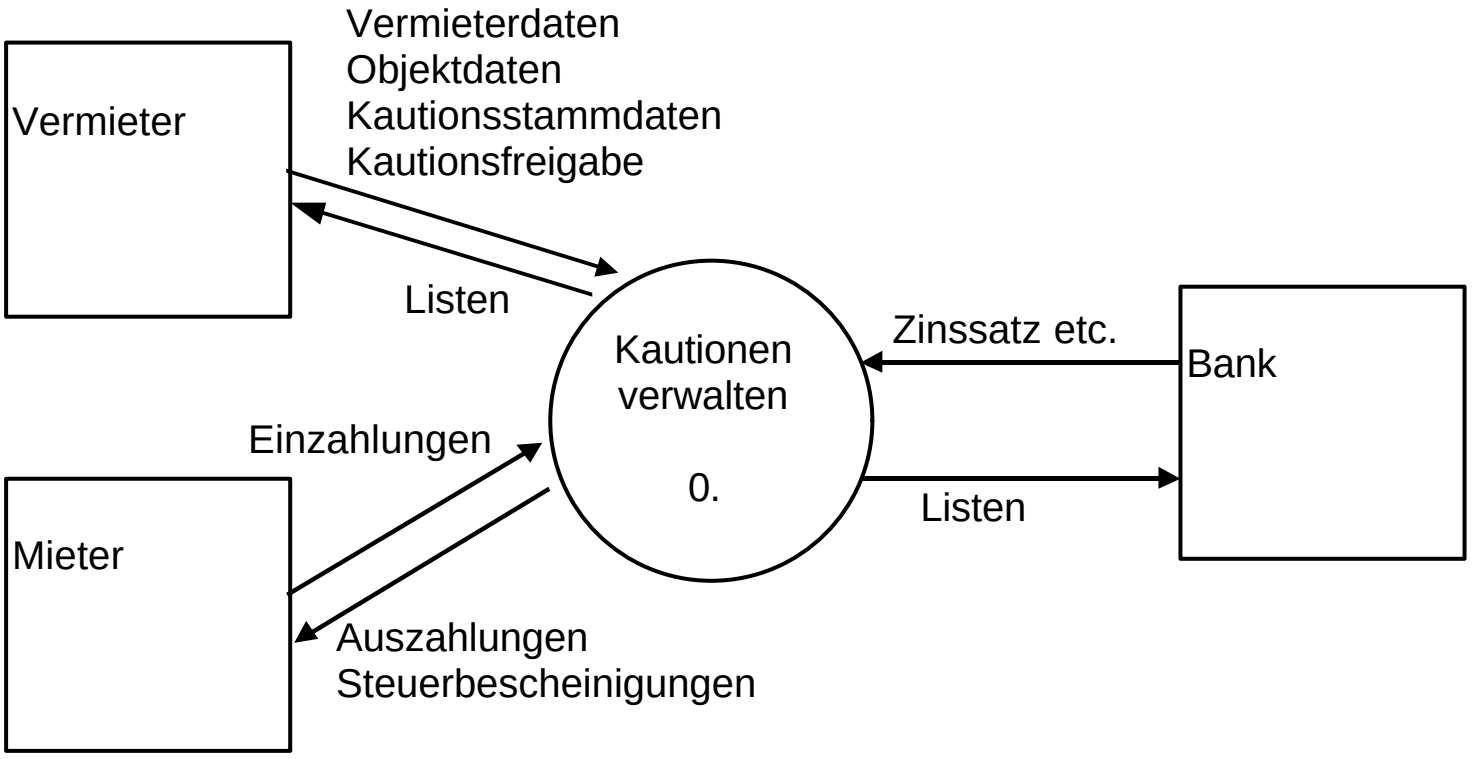
Hierarchiekonzept

- Kontextdiagramm auf höchster Ebene
  - abstraktes Datenflußdiagramm
  - beschreibt Schnittstellen zur Umwelt
- Verfeinerung durch Datenflußdiagramme
  - fortschreitende Unterteilung in Prozesse
  - fortschreitende Verfeinerung der Datenflüsse mit Hilfe eines Data Dictionary
- Spezifikation durch MiniSpec auf niedrigster Ebene
  - Prozeßspezifikation durch Entscheidungstabellen
  - Prozeßspezifikation durch Pseudo-Code

Kontextdiagramm

Beispiel Kautionsverwaltung

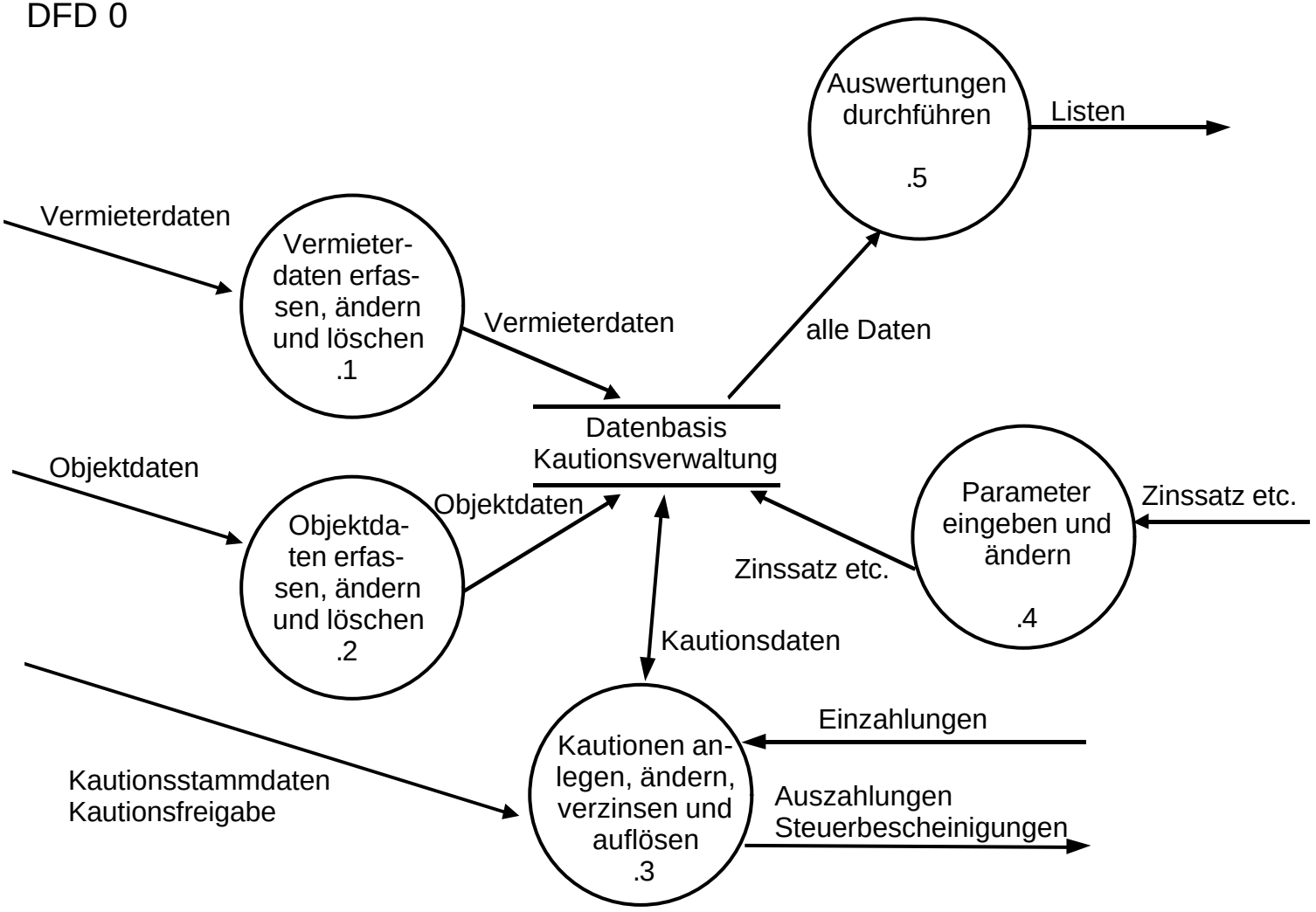
Kontextdiagramm



Verfeinerte Datenflußdiagramme

Beispiel Kautionsverwaltung

DFD 0



Einträge im Data Dictionary

- Zusammenhang der Datenflüsse in den Diagrammen
  - jeder Datenflußpfeil trägt einen Datenflußnamen
  - jeder Datenflußname ist im Data Dictionary definiert
  - jeder Speicher trägt einen Namen
  - jeder Speichername ist im Data Dictionary definiert
- Datenintegrität
  - Datenflüsse untergeordneter Diagramme im übergeordneten Diagramm unter gleichem Namen oder als Teilkomponente
  - Teilkomponenten-Eigenschaft im Data Dictionary beschrieben
  - keine neuen Datenflüsse auf tieferen Ebenen
  - Nachträge durch Ergänzung als Teilkomponente im DD oder auf allen übergeordneten Ebenen

Mini-Spezifikationen

- MiniSpec
  - an den Blättern des Hierarchie-Baumes
  - in Pseudocode oder als Entscheidungstabelle
  - keine Implementierungsvorschrift
  - Algorithmen abstrahiert
    (z.B. keine Angabe über Sortierverfahren)
- Aufgabe
  - beschreibt für einen Prozeß, wie Eingabe-Datenflüsse in Ausgabe-Datenflüsse transformiert werden
  - ermöglicht die Überprüfung, ob alle Ausgaben aus den vorhandenen Eingaben erstellt werden können
  - ergänzt DFD

Methodik in der Strukturierten Analyse

- Festlegung der Schnittstellen zur Umwelt
- Identifizierung aller Ein- und Ausgabedatenflüsse für Prozeß 0
- Ermitteln der Prozesse, die Eingaben in Ausgaben transformieren
- Überlegen, welche / wieviele Speicher gebraucht werden
  - ggf. Entity-Relationship-Modell erstellen
- Verfeinern der Datenflüsse und Zuordnung zu Prozessen/Speichern
- Definition der Datenflüsse und Speicher im Data Dictionary
- Überarbeiten des SA-Modells
  - Initialisierung / Terminierung ignorieren
  - ggf. vorhandene Kontrollflüsse aus DFD entfernen
  - triviale Fehlermeldungen weglassen, nur Roll-Backs behalten
  - Namen der Datenflüsse überarbeiten, ggf. Neustrukturierung
  - Namen der Prozesse überarbeiten, ggf. Neustrukturierung
- Verfeinerung ausgehend von überarbeitetem Diagramm 0
  - rechtzeitig beenden
  - MiniSpecs für alle nicht weiter verfeinerten Prozesse

Bewertung der SA

- 👍 Vorteilhafte Kombination bewährter Grundkonzepte
- 👍 verbesserte Übersicht durch DFD-Hierarchie
- 👍 Qualitätssicherungsmöglichkeiten durch Quervergleiche der Aspekte in den verschiedenen Grundkonzepte
- 👍 leicht erlernbar, leicht nachvollziehbar
- 👍 Unterstützung durch CASE-Werkzeuge
- 👍 Zusammenhang zu ER-Modellen über Speicher möglich
- 👎 keine Verfeinerung von Schnittstellen, daher Probleme bei umfangreichen Schnittstellen
- 👎 keine Verfeinerung von Speichern (aber Modellierung eines Gesamtspeichers über ER möglich)

SA ist traditionelle Entwicklungsmethodik,

Objektorientierte Analyse und Objektorientiertes Design ist moderner!

Entity-Relationship-Modell

- Grundbegriffe im ER-Modell
- Graphische Darstellung
- Kardinalitäten von Assoziationen
- Rekursive Assoziationen
- Semantische Datenmodellierung
- Tabellendarstellung
- Bewertung

Entity-Relationship-Modell (kurz: ER-Modell)

P. Chen 1976

weitere Bezeichnungen:

Gegenstands-Beziehungs-Modell,

Information Modeling

Ziel

Beschreibung permanent gespeicherten Daten und ihrer Beziehungen untereinander.

Analyse der Information erfolgt aus fachlogischer Sicht.

Resultat: konzeptionelles Modell, das gegen Veränderungen der Funktionalität weitgehend stabil ist.

Entitätsmengen und Assoziationen

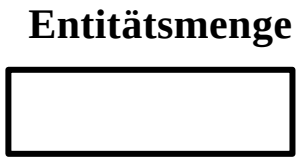
Ausgangspunkt des ER-Modelles ist der Begriff der *Entität.*

Entität (entity): individuelles, identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt, wird durch Eigenschaften beschrieben.

synonymer Begriff in diesem Kontext: Objekt

Entitätsmenge (entity set), auch Entitätstyp, Objekttyp: Zusammenfassung von Entitäten mit gleichen Eigenschaften unter einem eindeutigen gemeinsamen Oberbegriff.

Grafische Darstellung Entitätsmenge:



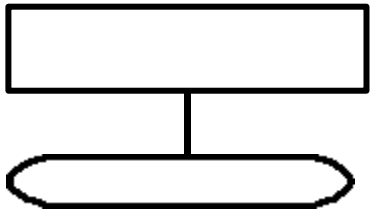
Attribut (attribut, property)

beschreibt eine fachliche Eigenschaft, die allen Entitäten einer Entitätsmenge gemeinsam ist.

definiert durch seinen Namen, der seiner fachlichen Bedeutung entsprechen soll, und seinen Wertebereich.

Wertebereich (auch Domäne): Menge aller möglichen bzw. zugelassenen Werte für ein Attribut

Attribute in ER-Diagrammen als Ovale dargestellt, durch ungerichtete Kanten mit jeweiliger Entitätsmenge verbunden.



Unterscheide: beschreibende Attribute, die anwendungsrelevante Eigenschaften der Entitäten festhalten, und identifizierende Attribute, die Schlüssel zur eindeutigen Identifikation einer konstanten Entität innerhalb ihrer Entitätsmenge bilden.

Schlüssel: kann aus einem oder mehreren identifizierenden Attributen zusammengesetzt sein.

Für eine Entitätsmenge läßt sich oft auch mehr als ein Schlüssel angeben, z. B.

Entitätsmenge: Stadt

Attribute: PLZ, Staat, Einwohnerzahl, Vorwahl

Schlüssel: PLZ und Staat oder Vorwahl

In einem solchen Fall wird stets ein Schlüssel als Primärschlüssel ausgezeichnet.

Von einem Schlüssel wird Minimalität verlangt.

Ein Schlüssel K ist stets eine minimale, identifizierende Attributskombination, jede echte Obermenge von K ist ein Schlüssel-Kandidat. Schlüssel werden unterstrichen dargestellt.

Beispiel: Entitätsmenge Kunden

Alle Kunden der Firma Teachware bilden die Entitätsmenge Kunde.

Diese Entitätsmenge hat die Attribute

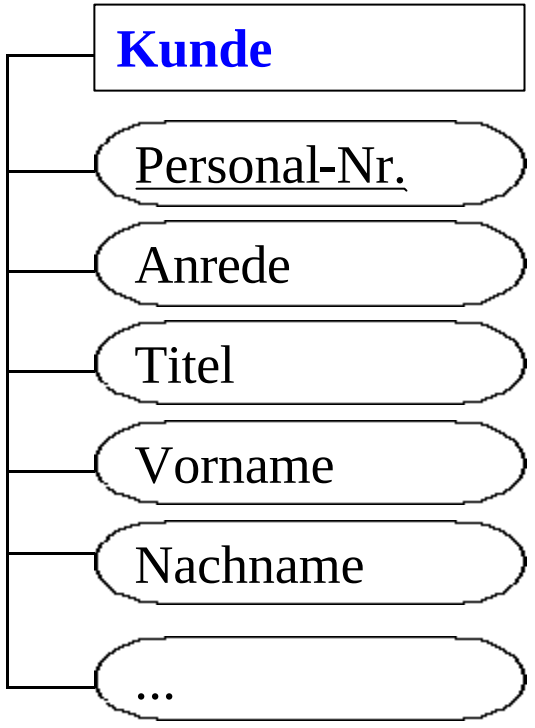
Personal-Nr. , Anrede, Titel, Vorname, Nachname, Straße, PLZ, Ort, ...

Die Personal-Nr. hat als Wertebereich die Menge aller ganzen Zahlen zwischen 0 und

100 000.

Die Anrede kennt nur die zwei Werte Herr und Frau.

Die Personal-Nr. ist der Schlüssel.

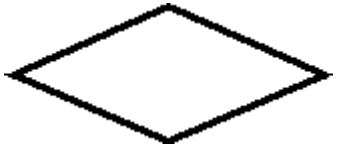


Assoziationen

Wechselwirkungen und Abhängigkeiten zwischen Entitäten werden durch Beziehungen (Relationen) dargestellt.

Die Zusammenfassung gleichartiger Beziehungen zwischen Entitäten erfolgt durch Beziehungsmengen, Assoziationen genannt.

graphische Darstellung:



Kardinalitäten

Die Kardinalität (auch Komplexitätsgrad genannt) gibt an, mit wie vielen anderen Entitäten eine Entität einer bestimmten Entitätsmenge in einer konkreten Beziehung stehen muß bzw. stehen kann.

Die Angabe der Kardinalität erfolgt durch eine Zahl bzw. einen Buchstaben an den Linien, die die Raute mit den Entitätsmengen verbindet.

Drei prinzipiell mögliche Mengenverhältnisse zwischen zwei Entitäten:

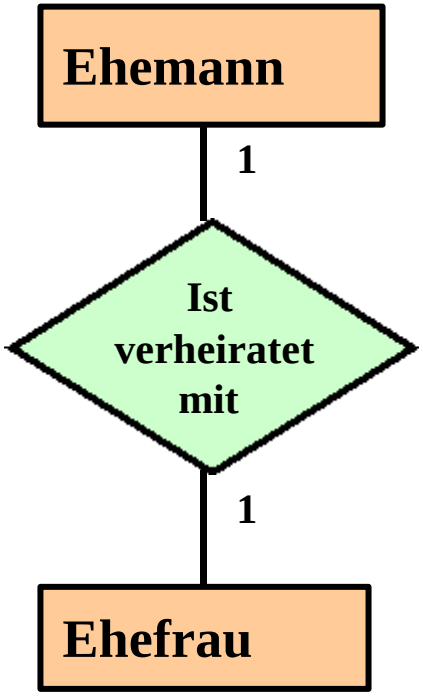
Wir betrachten jeweils zwei Entitätsmengen A und B

1:1 -Assoziation

jede Entität aus A ist mit genau einer Entität aus B verbunden ist und umgekehrt.

Im westlichen Kulturkreis besteht zwischen Ehepaaren eine 1:1-Beziehung.

- Als Name für eine Entitätsmenge sollte ein Substantiv im Singular gewählt werden, z.B. Ehemann, Kunde, ...
- Als Name für eine Assoziation sollte ein Verb im Singular gewählt werden, z.B. ist verheiratet mit,...



1:M -Assoziation

zu jeder Entität aus A existiert eine oder mehrere Entitäten in B, zu jeder Entität aus B existiert genau eine Entität in A.

M:N -Assoziation

zu einer Entität A existiert eine oder mehrere Entitäten in B und umgekehrt.

Anmerkung:

M, N bezeichnen beliebige Obergrenzen.

N:N-Assoziation bedeutet nicht, dass auf beiden Seiten jeweils die gleiche Anzahl von Entitäten miteinander in Beziehung stehen.

M:N bedeutet nicht, dass die Obergrenzen unterschiedlich sein müssen.

Konditionelle Beziehung:

Beziehung muss nicht sondern kann bestehen**.**

=>

1:C-Assoziation

zu jeder Entität aus A existiert eine oder keine Entität in B

zu jeder Entität aus B existiert genau eine Entität in A.

1:MC-Assoziation

zu jeder Entität aus A existiert keine, eine oder mehrere Entitäten in B
zu jeder Entität aus B existiert genau eine Entität in A.

N:MC -Assoziation

zu jeder Entität aus A existiert/en keine, eine oder mehrere Entitäten in B

zu jeder Entität aus B existieren N-Entitäten, d. h. mindestens eine in A

Insgesamt 16 verschiedene Kombinationen:

		muß		**kann**	
	B \ A	**1**	**N**	**C**	**NC**
muß	**1**	**1:1**	**1:N**	**1:C**	**1:NC**
	M	**M:1**	**M:N**	**M:C**	**M:CN**
kann	**C**	**C:1**	**C:N**	**C:C**	**C:NC**
	MC	**MC:1**	**MC:N**	**MC:C**	**MC:NC**

Weitere äquivalente Notationen:

MC-Notation: C = choice und M = mutiple interpretieren

Numerische Notation

Kardinalitäts als (min , max) -Tupel.

besagt, in wie vielen konkret vorhandenen Beziehungen eine Entität einer Entitätsmenge mindestens (min) und höchstens (max) vorkommt, (Angabe evtl. ohne Klammern).

Ist min=max, dann wird nur ein Wert angegeben.

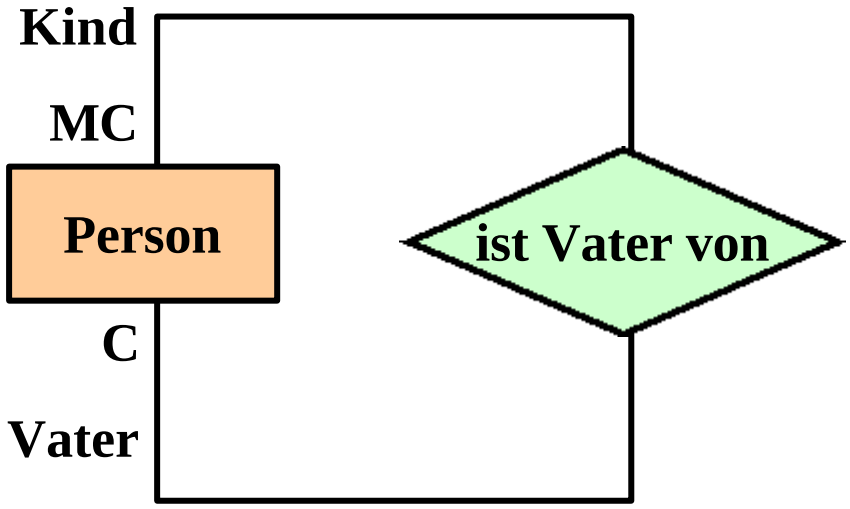
MC-Notation	**Numerische Notation**
C	**(0,1)**
1	**(1,1)**
MC	**(0,n)**
M	**(1,n)**

Alternative Darstellungsmöglichkeiten von Kardinalitäts-Angaben

Rekursive Assoziationen und Rollen

Eine Entitätsmenge kann auch mit sich selbst in Beziehung stehen.

Eine solche Beziehung heißt rekursive Assoziation.



Sind in einer Firma von einer Familie sowohl der Vater als auch seine Kinder beschäftigt, dann gibt es eine rekursive Assoziation „ist Vater von“.

Zu einem Vater können M Kinder-Beziehungen bestehen.

Ein Kind kann in einer Vater-Beziehung stehen (nicht zu jedem Mitarbeiter ist auch der Vater in der Firma beschäftigt).

Kommt eine Entitätsmenge in einer Assoziation mehrfach vor (rekursive Assoziation), werden die Rollen, die die Entitätsmenge in der Assoziation übernimmt, zusätzlich durch Rollennamen konkretisiert

Eine Rolle beschreibt, welche Funktion eine Entität in einer Assoziation innehat.

Der Rollenname wird jeweils an ein Ende der Assoziation geschrieben, und zwar bei der Entitätsmenge, deren Bedeutung in der Assoziation sie näher beschreibt. Im Beispiel sind die Rollen Vater und Kind.

Die Verwendung von Rollennamen ist optional. Die geschickte Wahl von Rollennamen kann die Verständlichkeit jedoch verbessern. Bei rekursiven Assoziationen müssen die Rollen angegeben werden, um die Verständlichkeit sicherzustellen.

Beispiel für Rollen:

Semantische Datenmodellierung

Erweiterung des ER-Modells zur semantischen Datenmodellierung

Wichtige zusätzliche Konzepte:

- Aggregation (is-part-of-Beziehung, ist-Teil-von-Beziehung)

  häufig benötigten Spezialfall einer Assoziation

- Generalisierungshierarchie

Assoziation beschreibt eine gleichrangige Beziehung zwischen Entitäten.

Aggregation beschreibt eine Über-Unterordnungs-Beziehung zwischen den Entitäten.

Kardinalitäten: wie bei Assoziationen

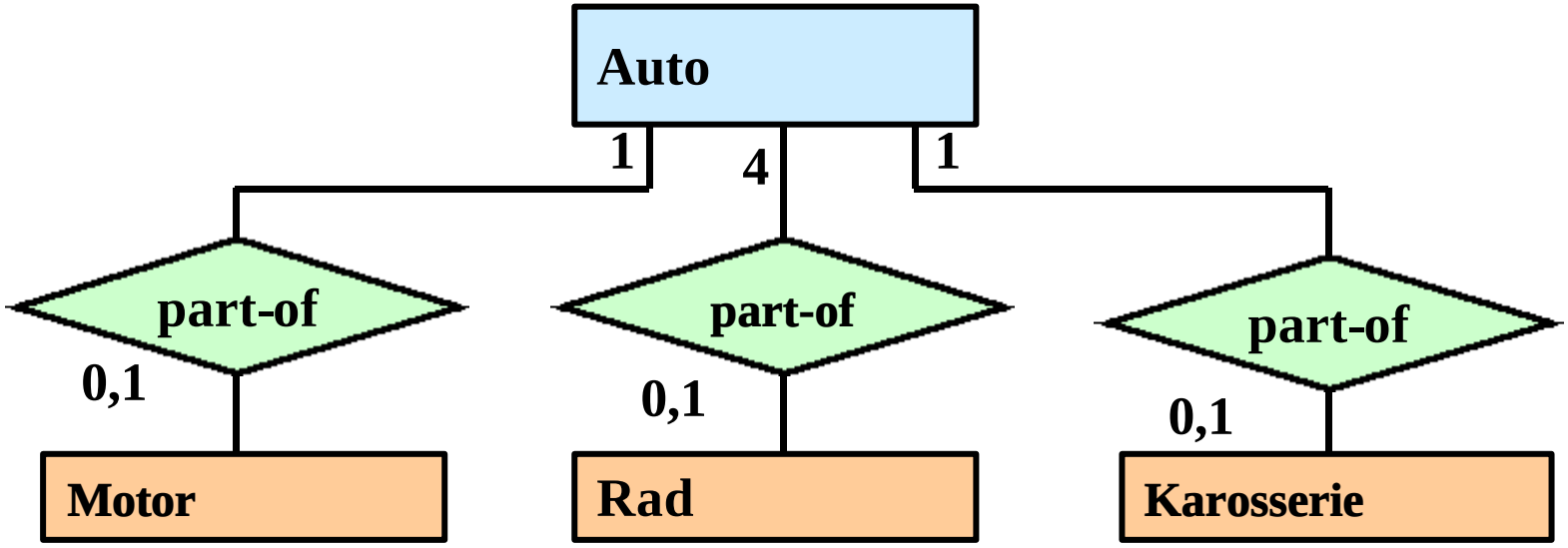
Aggregationen werden durch die Namen part-of oder eine geeignete Bezeichnerwahl ausgedrückt.

Beispiel für eine Aggregation (numerische Notation)

Ein Auto läßt sich als eine Aggregation, bestehend aus einem

Motor, 4 Rädern und einer Karosserie, beschreiben.

Ein Motor kann für sich allein stehen (noch nicht eingebaut) oder Teil eines Autos sein, daher die Kardinalität 0,1 (numerische Notation).



Eine Aggregation beschreibt einen semantisch engeren Zusammenhang zwischen Entitätsmengen als eine Assoziation.

Das Konzept der Generalisierungshierarchie

ermöglicht Zuordnung gemeinsamer Attribute von Entitätsmengen zu einer übergeordneten Entitätsmenge

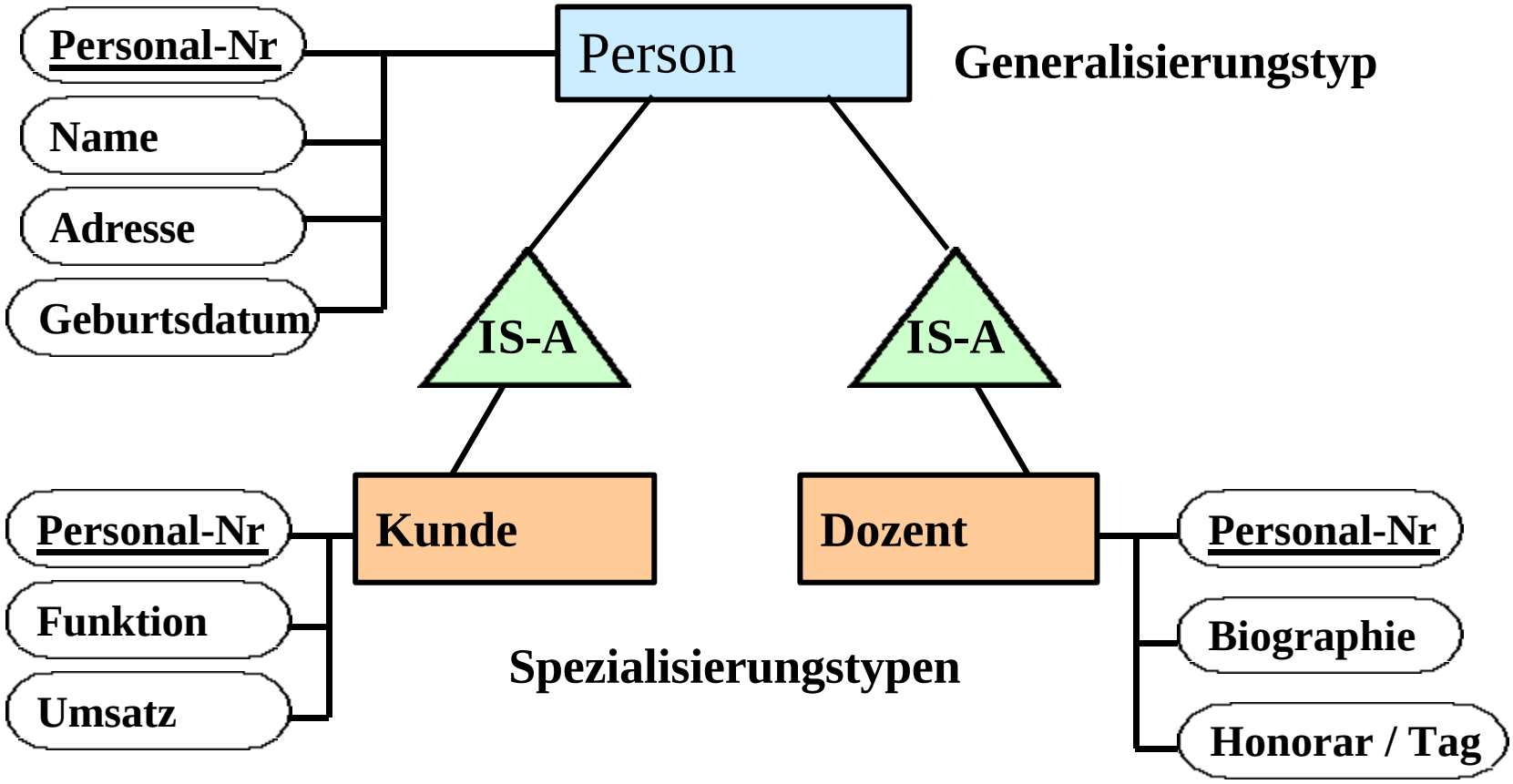
Beispiel : In einer Fallstudie werden sowohl Kunden als auch Dozenten modelliert.

Die jeweiligen Attribute lauten:

Kunde:	**Dozent:**
Personal-Nr	**Personal-Nr**
Name	**Name**
Adresse	**Adresse**
Geburtsdatum	**Geburtsdatum**
Funktion	**Biographie**
Umsatz	**Honorar pro Tag**

Es liegt nahe, die den beiden Entitätsmengen gemeinsamen Attribute einer übergeordneten Entitätsmenge „Person“ zuzuordnen (Generalisierungstyp). Die Entitätsmengen „Kunde“ und „Dozent“ stehen in einer besonderen Beziehung zu „Person“, oft als IS-A-Beziehung (Ist-Ein) bezeichnet.

Beispiel einer Generalisierungshierarchie



Im Gegensatz zu einer Assoziation und einer Aggregation, bei der eine Beziehung zwischen Entitäten besteht, verknüpft eine Generalisierungs-hierarchie die Entitätsmenge und nicht die Entitäten.

Eine übergeordnete Entitätsmenge bezeichnet man als Generalisierungstyp (supertype),

die in einer IS-A-Verknüpfung damit verbundenen Entitätsmengen als Spezialisierungstyp (subtypes).

Jeder Spezialisierungstyp erbt vom Generalisierungstyp automatisch alle Attribute.

Die identifizierenden Attribute in einer IS-A-Beziehung müssen gleich sein (hier: Personal-Nr)

Durch das Vererbungs-Konzept entsteht ein Generalisierungshierarchie.

Damit werden semantisch ähnliche Entitätsmengen zu einer übergeordneten Entitätsmenge zusammengefaßt. Die übergeordnete Entitätsmenge beschreibt die gemeinsamen Eigenschaften der untergeordneten Entitätsmengen.

Jede untergeordnete Entitätsmenge kann eigene Attribute besitzen.

Abbildung auf Dateien

Mit dem ER-Modell werden die permanent gespeicherten Daten und ihre Beziehungen modelliert.

Es muß sich also ein Zusammenhang zu den Speichern eines Datenflußdiagramms (DFD) herstellen lassen.

Es gelten folgende Regeln:

1. Für jede Entitätsmenge wird ein Speicher bzw. eine Datei benötigt. Jede Entität der entsprechenden Entitätsmenge stellt einen Eintrag in diese Datei dar.

2. Sind zwei Entitätsmengen A und B durch eine 1:1- oder M:1-Assoziation verbunden, dann wird der Schlüssel von B als sogenannter Fremdschlüssel in A eingetragen, d. h. als zusätzliches Attribut.

3. Sind zwei Entitätsmengen A und B durch eine M:N-Assoziation verbunden, dann wird für die Assoziation eine eigene Datei angelegt. Als Attribute werden die Schlüssel der Entitätsmengen verwendet, die die Assoziation verbindet.

Konsistenzprüfung eines semantischen Datenmodells

• Besitzt jede Entitätsmenge mindestens ein Attribut?

Nein => keine Entitätsmenge

• Sind die Entitätsmengen durch Substantive, die Assoziationen durch Verben beschrieben ?

Nein => Beziehungen überprüfen

• Erben in einer IS-A-Beziehung alle Spezialisierungstypen alle Attribute des Generalisierungstyps?

Nein => IS-A-Beziehung überprüfen.

Konsistenzprüfung eines semantischen Datenmodells

Sind zwei Entitätsmengen identisch? Identität kann vorliegen, wenn eine oder mehrere der folgenden Bedingungen erfüllt sind:

- 1:1-Assoziation.
- durch dieselben Assoziationen mit der Umgebung verbunden
- dieselben Schlüsselattribute.
- dieselben Attribute.

Jede Assoziation ist zu überprüfen auf

- ihre Notwendigkeit, d. h. bringt sie neue Informationen in das Modell,
- korrekte Darstellung des Sachverhalts.

Liegt eine Entitätsmenge oder ein Attribut vor?

- Entität: eindeutig identifizierbar, durch Attribute beschrieben.
- Attribut: besitzt selbst keine weiteren Attribute.

In der Systemanalyse kann man auf einem angemessenen Abstraktionsniveau aufhören z. B. bei Adresse und ein solches Attribut als elementar ansehen.

Abhängig vom Blickwinkel können Attribute zu Entitätsmengen werden und umgekehrt.

semantische Datenmodellierung

- im kaufmännischen Anwendungsbereich erforderlich.
- Voraussetzung für einen relationalen Datenbankentwurf.

Schwierigkeit:

- Semantische Datenmodelle können sehr umfangreich werden

=> schwer zu überblicken.

- Verfeinerungsmechanismus. um mehrere Abstraktionsebenen bilden zu können, fehlt.

Unternehmens-Datenmodelle und Weltmodelle

Bei größeren Unternehmen gibt es eine Vielzahl von Anwendungssystemen, die Teilbereiche verwalten. Betrachtet man jeweils nur den Bereich eines Unternehmens, der durch ein entsprechendes Anwendungssystem bearbeitet wird, dann führt dies zu folgenden Problemen:

- Mehrfachverwaltung der gleichen Datenbestände
- Brüche bei der Abwicklung von übergreifenden Geschäftsvorgängen
- Inkompatible Informationsflüsse

=> Ziel: umfassendes Unternehmens-Datenmodell

- mit Informationsstrukturen und Vorgangsketten aller Unternehmensbereiche
- mit Berücksichtigung der Schnittstellen zueinander
- in einheitlicher und ganzheitlicher Form darstellen.

=> ER-Modelle

strategisches Datenmodell

• nur die wichtigsten Entitätsmengen, Assoziationen, Aggregationen und Generalisierungen.

• nur grundsätzliche Branchenunterschiede, keine detaillierten Unterschiede.

• umfaßt i.d.R. rund 20 bis 30 Entitätsmengen, Assoziationen, Aggregationen und Generalisierungen.

Mehrere Abstraktionsstufen.

Nächste Stufe:

• unterschiedliche Strukturen innerhalb einer Branche.

• umfaßt ca. 200 bis 500 Entitätsmengen und Verknüpfungen.

Beispiel: Referenzmodell von A. W. Scheer mit 300 Entitäten,

(graphische Darstellung auf DIN A1)

Modellierung von Unternehmens-Datenmodellen nutzt Generalisierungshierarchie

ER-Modell fokussiert einseitig auf Daten-Sicht

Beispiel: Modellierung eines Roboters, weitere Sichten:

- die Umweltsicht (oberste Beschreibung der Roboterzelle)
- die Robotersicht (Modellierung des Betriebsmittels Roboter)
- die Montagesicht (Beschreibung von Montagefolge und Bewegungssegment)
- die Sicht der Handlungsobjekte (Beschreibung von Produkten)
- die Konstruktionssicht (Konstruktionsmodell physikalischer Objekte)

Die Zusammenfassung dieser verschiedenen Einzelsichten ergibt ein Weltmodell der Fertigungsumgebung, der Betriebsmittel und Werkstücke sowie der Produktionsdaten (ca. 100 Entitätsmengen).

Bei der Zusammenfassung von Teilmodellen zu einem Gesamtmodell muß folgendes beachtet werden:

- Erkennung und Auflösung von Synonymen

  (verschiedenen Benennungen für den gleichen Sachverhalt)

- Erkennung und Auflösung von Homonymen

  (Gleichbenennung unterschiedlicher Sachverhalte)

Diese beiden Punkte beziehen sich sowohl auf Entitätsmengen und Assoziationen als auch auf Attribute.

- Ermittlung nicht erfaßter Verknüpfungen zwischen den Einzelsichten
- Zusammenfassen unterschiedlicher Entitätsmengen für identische

  Informationseinheiten.

Bei der Entwicklung eines Unternehmens-Datenmodells oder eines Weltmodells kann man bei den generellen Begriffen beginnen und durch eine zunehmende Spezialisierung die Begriffe in feinere Begriffe aufspalten

top down-Methode

Bei der Generalisierung werden zunächst die auf einer detaillierten Ebene eingeführten Begriffe verallgemeinert

bottom up-Methode

In der Praxis wird oft eine Mischung beider Methoden verwendet.

Bewertung des ER-Modell

- 👍 semantische Datenmodellierung ist Standard in kaufmännischen Anwendungen
- 👍 bereitet relationalen Datenbankentwurf weitgehend vor
- 👍 strukturierter als Data Dictionary
- 👍 Verknüpfung mit Strukturierter Analyse möglich (Speicher im Datenflußdiagramm, Data Dictionary)
- 👍 Unterstützung durch CASE-Werkzeuge
- 👎 relativ abstrakt
- 👎 bei umfangreichen Datenmodellen unübersichtlich, da keine Verfeinerung

Zusammenfassung

- Vorgehensweise: Strukturierte Analyse
- Notationen zur Beschreibung von
  - Funktionalität
  - Daten
  - Verbindung Daten&Funktionalität
  - Dynamik
- Einsatz:
  - Definitionsphase
- traditioneller Methodik im Softwareentwurf
- Beobachtung:

  neben Schwierigkeiten, die in den Notationen selbst liegen, fällt deutlicher Bruch zur Umsetzung in Klassenhierarchien etc auf.

  Beim nächsten Mal:

  Objektorientierte Analyse
  - OOA greift Konzepte der OO Programmierung auf,
  - OOA hält zusätzliche Konzepte bereit,
  - OOA eng mit Unified Modeling Language (UML, Zusammenstellung „normierter“ Notationen für Funktionalität, Daten, Dynamik, ...) verbunden