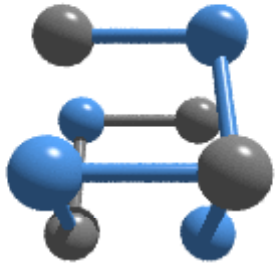
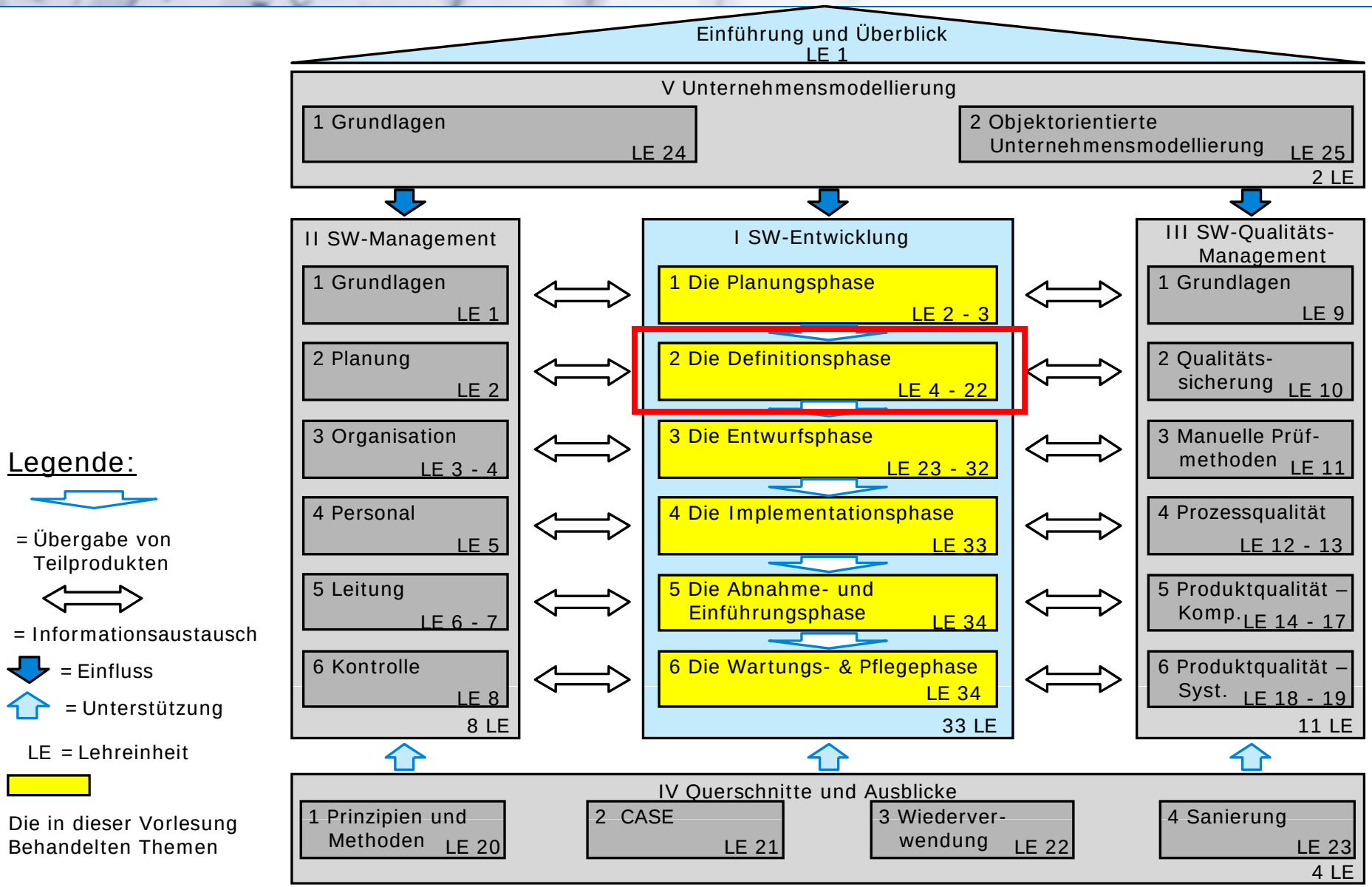


Vorlesung Softwaretechnik - Definitionsphase, funktionale und objektorientierte Sicht -

Prof. Dr.-Ing. habil. Klaus-Peter Fähnrich

Wintersemester 2009/2010


Einführung und Überblick
LE 1
V Unternehmensmodellierung
1 Grundlagen
LE 24
2 Objektorientierte Unternehmensmodellierung
LE 25
2 LE
II SW-Management
1 Grundlagen
LE 1
2 Planung
LE 2
3 Organisation
LE 3 - 4
4 Personal
LE 5
5 Leitung
LE 6 - 7
6 Kontrolle
LE 8
8 LE
I SW-Entwicklung
1 Die Planungsphase
LE 2 - 3
2 Die Definitionsphase
LE 4 - 22
3 Die Entwurfsphase
LE 23 - 32
4 Die Implementationsphase
LE 33
5 Die Abnahme- und Einführungsphase
LE 34
6 Die Wartungs- & Pflegephase
LE 34
33 LE
III SW-Qualitäts-Management
1 Grundlagen
LE 9
2 Qualitäts-sicherung
LE 10
3 Manuelle Prüf-methoden
LE 11
4 Prozessqualität
LE 12 - 13
5 Produktqualität – Komp.
LE 14 - 17
6 Produktqualität – Syst.
LE 18 - 19
11 LE
IV Querschnitte und Ausblicke
1 Prinzipien und Methoden
LE 20
2 CASE
LE 21
3 Wiederver-wendung
LE 22
4 Sanierung
LE 23
4 LE
Legende:
= Übergabe von Teilprodukten
= Informationsaustausch
= Einfluss
= Unterstützung
LE = Lehreinheit
Die in dieser Vorlesung Behandelten Themen

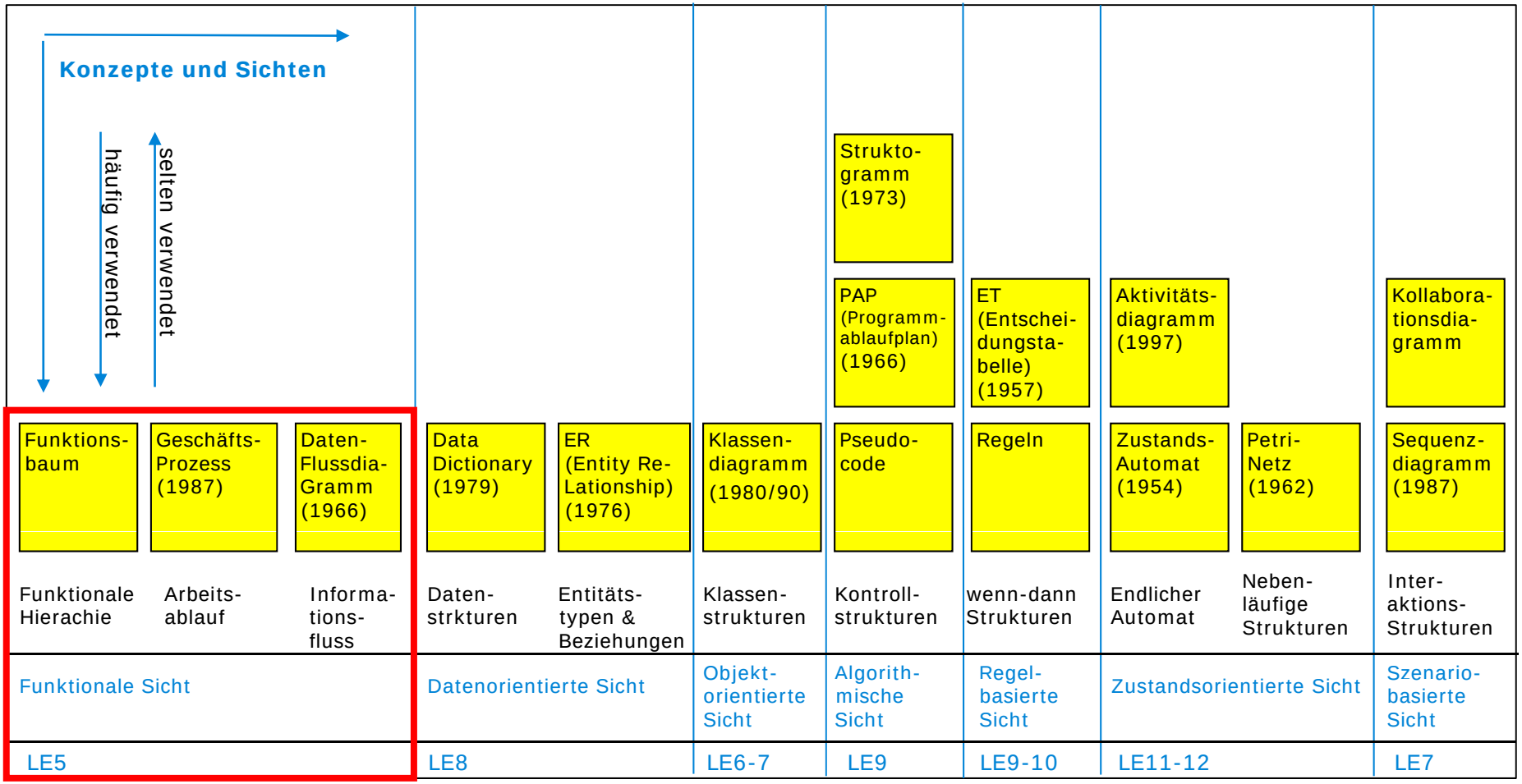
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Lernziele

1. Konzepte beschreiben für Funktionsbaum, Geschäftsprozess und Datenflussdiagramm
2. Hierarchische Gliederung von Funktionen und Konstruktion eines Funktionsbaumes für eine gegebene Problemstellung
3. Checklisten
4. Strukturierung von Geschäftsprozessen
5. Ermittlung von Schnittstellen, Funktionen, Speicher und Informationsfluss für gegebene Problemstellung
6. Regeln für die Syntax und Semantik von Datenflussdiagrammen (DFD)

Überblick

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

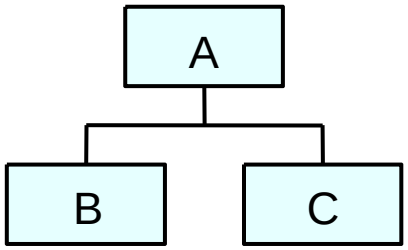


UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Funktionsbaum

- **Funktion** beschreibt Tätigkeit oder klar umrissene Aufgabe innerhalb eines größeren Zusammenhangs. Transformiert Eingabedaten in Ausgabedaten bzw. ändert die Struktur der Information.
- Funktionshierarchie entsteht wenn eine allgemeine Funktion in spezielle Teilfunktionen gegliedert wird.Es entsteht ein **Funktionsbaum**.

Grafische Darstellung



Hierarchie

Basis der Hierarchie

- **Besteht-aus** (meist in der Definitionsphase):
  A besteht aus B und C
- **Ruft-auf** (meist in der Entwurfsphase):

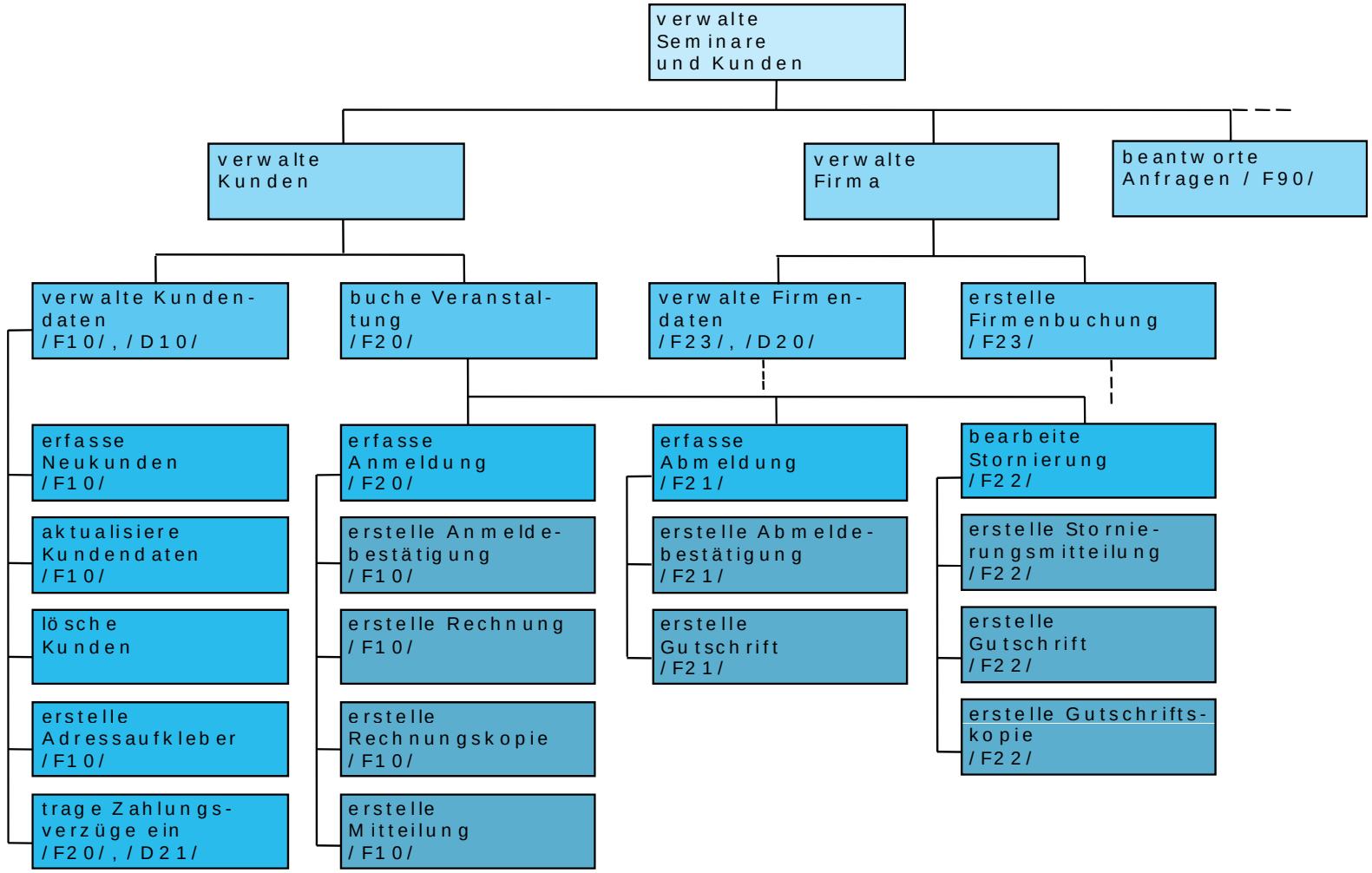
A ruft B und C auf.

Erstellungsregeln:

- Unter einer gemeinsamen Vaterfunktion sollen nur Funktionen (Kindfunktionen) angeordnet sein, die fachlich eng zusammengehörende Tätigkeiten beschreiben
- Was eng zusammengehört kann nur mit Fachwissen entschieden werden
- Auf einer Hierarchieebene sollen Funktionen angeordnet sein, die sich auf gleichem Abstraktions-niveau befinden
- Funktionsname ist i. d. R. Verb + Objekt oder Substantiv + Verb

Funktionsbaum-Beispiel: Seminarverwaltung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Legende: /.../ Bezüge zum Pflichtenheft

Geschäftsprozesse im Großen

- Ablauforganisation eines Unternehmens
- Unternehmensprozess (business process)
  - Besteht aus einer Anzahl von unternehmensinternen Aktivitäten, die durchgeführt werden, um die Wünsche eines Kunden zu befriedigen

Geschäftsprozesse im Kleinen

- Funktionalität eines Produkts
- Definiert einen Arbeitsablauf, der mit Hilfe von Software durchgeführt wird, aber manuelle und organisatorische Anteile besitzen kann
- **use case**
  - Teil eines Geschäftsprozesses, der die Benutzerkommunikation mit dem Software-System beschreibt
- Geschäftsprozess (*use case*) (Arbeitsablauf)
  - besteht aus mehreren zusammenhängenden Aufgaben, die von einem Akteur durchgeführt werden, um ein Ziel zu erreichen bzw. ein gewünschtes Ergebnis zu erstellen.

Akteure

- Rollen von Menschen oder Systeme, insbesondere Computersysteme, die als externe Beteiligte mit einem Unternehmen (Kunden) oder einem Software-Produkt (Benutzer) kommunizieren und Daten austauschen; stets außerhalb des Systems

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Schablone für Geschäftsprozesse

Geschäftsprozess: Name (was wird getan?)

- **Ziel**: Globale Zielsetzung bei erfolgreicher Ausführung des Geschäftsprozesses
- **Kategorie**: primär, sekundär oder optional
- **Vorbedingung**: Erwarteter Zustand, bevor der Geschäftsprozess beginnt
- **Nachbedingung Erfolg**: Erwarteter Zustand nach erfolgreicher Ausführung des Geschäftsprozesses
- **Nachbedingung Fehlschlag**: Erwarteter Zustand, wenn das Ziel nicht erreicht werden kann
- **Akteure**: Rollen von Personen oder andere Systeme, die den Geschäftsprozess auslösen oder daran beteiligt sind.
- **Auslösendes Ereignis**: Wenn dieses Ereignis eintritt, dann wird der Geschäftsprozess initiiert
- **Beschreibung**:

  1 Erste Aktion

  2 Zweite Aktion
- **Erweiterungen**:
  - 1a Erweiterung des Funktionsumfangs der ersten Aktion
- **Alternativen**:
  - 1a Alternative Ausführung der ersten Aktion
  - 1b Weitere Alternative zur ersten Aktion

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Schablone für Geschäftsprozesse - Beispiel

Geschäftsprozess: Buchen: Von Anmeldung bis Buchung

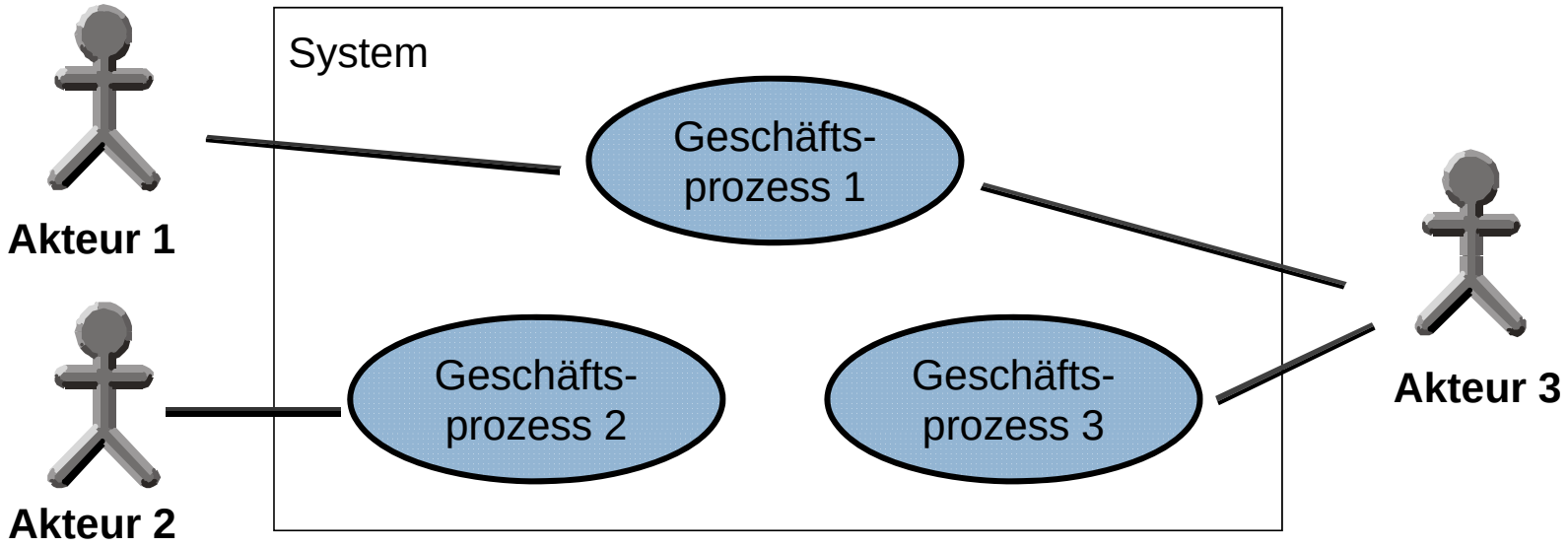
- **Ziel**: Anmeldebestätigung und Rechnung an Kunden geschickt
- **Kategorie**: primär
- **Vorbedingung**: -
- **Nachbedingung Erfolg**: Kunde ist angemeldet
- **Nachbedingung Fehlschlag**: Mitteilung an Kunden, dass Veranstaltung ausgebucht, ausfällt oder nicht existiert
- **Akteure**: Kundensachbearbeiter
- **Auslösendes Ereignis**: Anmeldung des Kunden liegt vor
- **Beschreibung**:
  - 1 Kundendaten prüfen
  - 2 Veranstaltung prüfen
  - 3 Anmeldebestätigung und Rechnung erstellen
- **Erweiterungen**:
  - 1a Kundendaten aktualisieren
  - 1b Wenn Kunde Mitarbeiter einer Firma ist, dann Firmendaten erfassen
- **Alternativen**:
  - 1a Neukunden erfassen
  - 2a auf alternative Veranstaltung hinweisen, wenn ausgebucht

Geschäftsprozessdiagramm

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

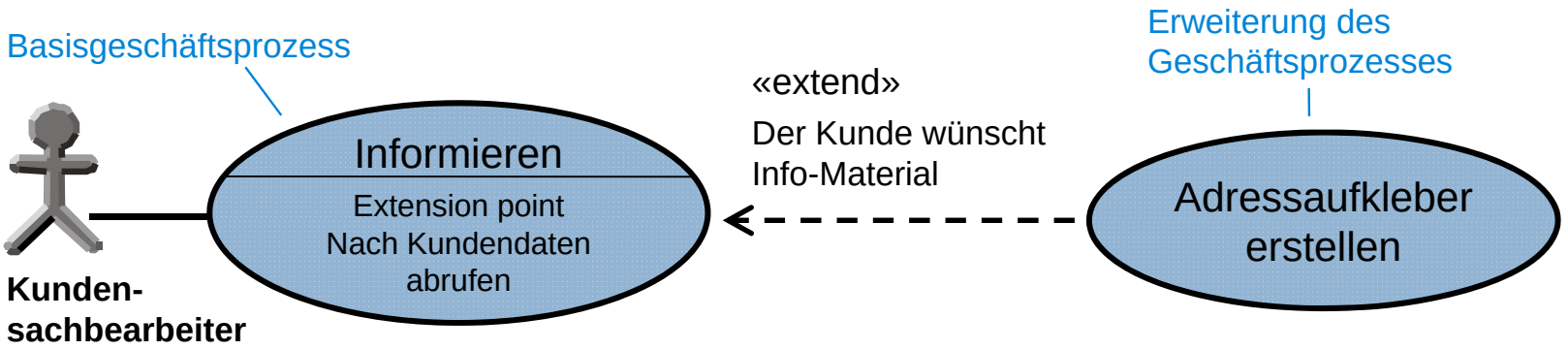
Geschäftsprozessdiagramm (use case diagram) :

- Beschreibt das Zusammenspiel mehrerer Geschäftsprozesse untereinander und mit den Akteuren
- Überblick über Produkt und Umgebung auf hohem Abstraktionsniveau
- In UML gibt es drei Möglichkeiten, Geschäftsprozesse zu strukturieren: die *extend*-Beziehung, die *include*-Beziehung und die Generalisierungsbeziehung



Struktur von Geschäftsprozessen

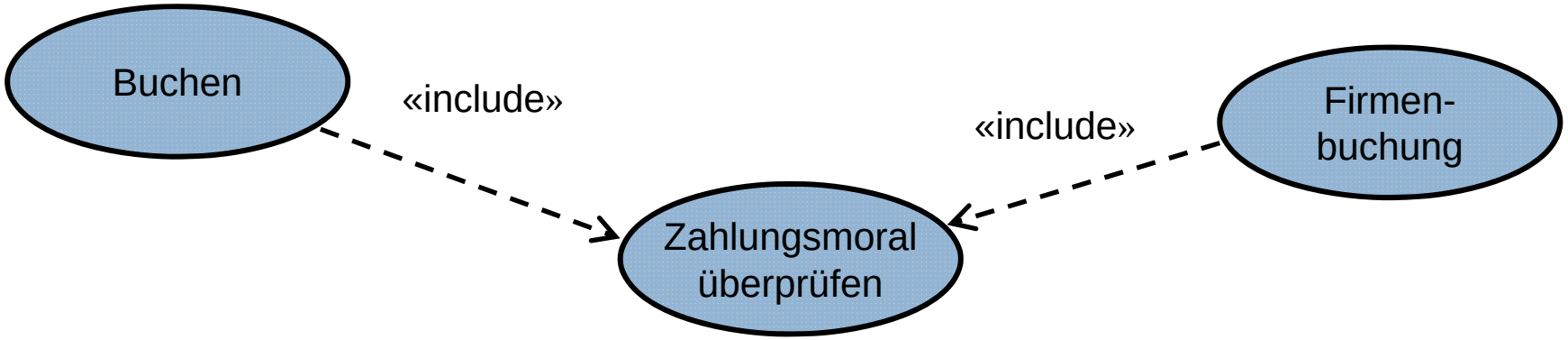
extend-Beziehung



- vom Basisprozess wird in den Unterprozess verzweigt, wenn bestimmte Bedingungen erfüllt sind
- Komplexitätsreduktion des Basis-Geschäftsprozesses

Struktur von Geschäftsprozessen

include-Beziehung



Mache Prozesse besitzen die gleichen Unterprozesse. Um diese nicht doppelt zu beschreiben werden sie als selbständiger Prozess modelliert. Mehrere Prozesse können den Unterprozess dann verwenden. Unterprozesse stehen nie für sich allein. (Analogie zu Unterprogramme in Programmiersprachen)

Der Pfeil zeigt vom Standard- zum Unterprozess, „Buchen“ ist also z. B. der Hauptprozess, „Zahlungsmoral überprüfen“ der Unterprozess.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

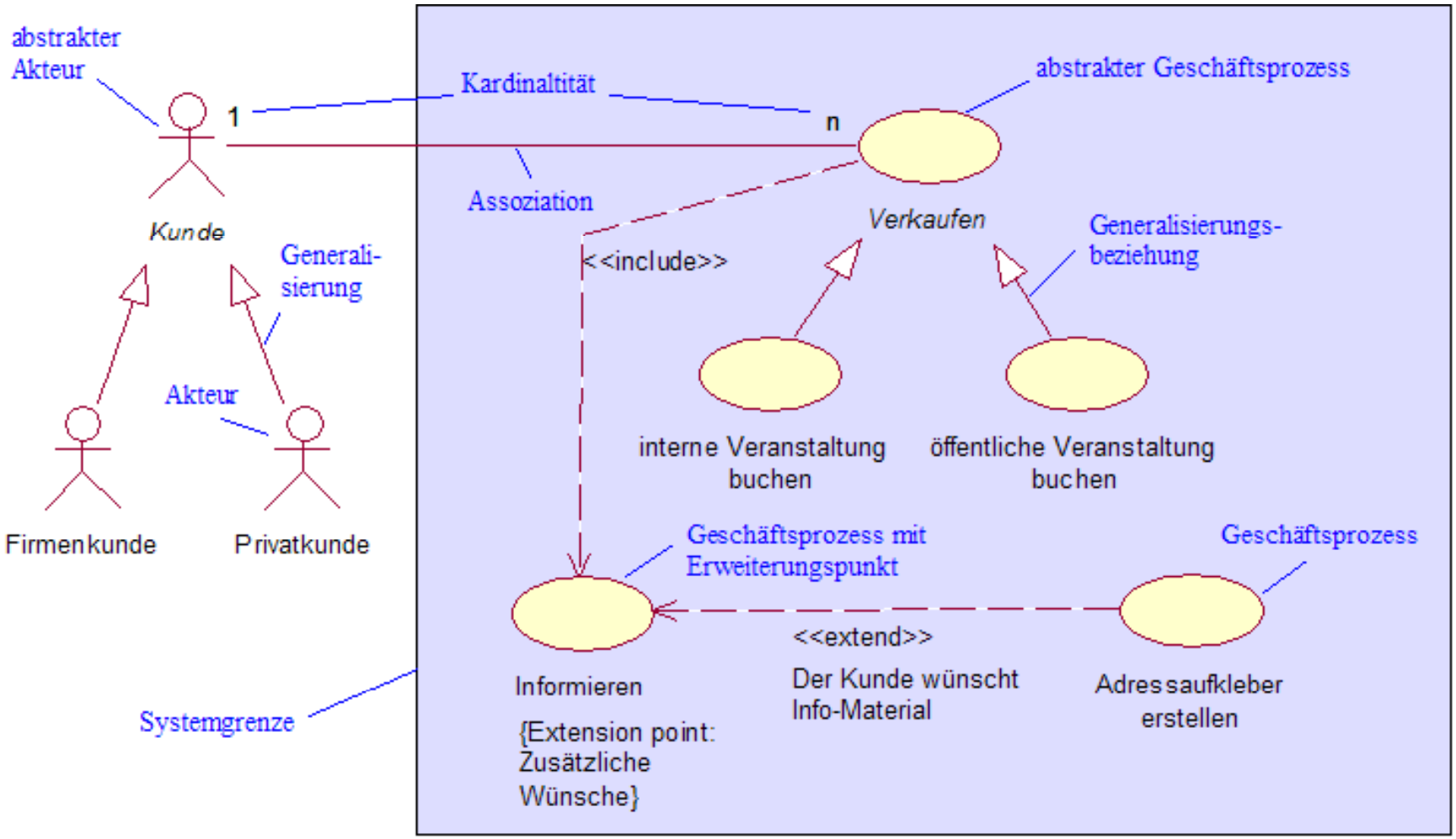
Generalisierungs-Beziehung

- übergeordneter Prozess vererbt sein Verhalten an die untergeordneten Prozesse
- abstrakte vs. konkrete Prozesse

➔ UML : siehe Programmierung und Programmiersprachen

Beispiel für ein Geschäftsprozessdiagramm

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Checkliste Geschäftsprozesse

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Ergebnisse

- Geschäftsprozessdiagramm
  - Alle Geschäftsprozesse und Akteure sind eingetragen
- Beschreibung der Geschäftsprozesse
  - Alle Geschäftsprozesse sind umgangssprachlich oder mittels Schablone beschrieben.

Konstruktive Schritte

1. **Akteure ermitteln**
   - Welche Personen führen diese Aufgaben zur Zeit durch und besitzen daher wichtige Kenntnisse über die durchzuführenden Arbeitsabläufe? Welche Rollen spielen diese Personen?
   - Welche Personen werden zukünftig diese Aufgaben durchführen und auf welche Vorkenntnisse muss die Benutzungsoberfläche abgestimmt werden? Welche Rollen spielen diese Personen?
   - Wo befindet sich die Schnittstelle des betrachteten Systems bzw. was gehört nicht mehr zu dem System?

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Checkliste Geschäftsprozesse

2. Geschäftsprozesse für die Standardverarbeitung ermitteln

- o Primäre und ggf. sekundäre Geschäftsprozesse betrachten
- o Welche Standardverarbeitung besitzen sie?

a) mittels Akteuren (z. B. Akteur „Veranstaltungsbetreuer“: Prozess „Veranstaltung betreuen“)

- ❖ Sind die Akteure Personen?
- ❖ Welche Arbeitsabläufe lösen sie aus?
- ❖ An welchen Arbeitsabläufen wirken sie mit?

b) mittels Ereignissen (Akteure sind externe Systeme) (z. B. Ereignis „Anmeldung liegt vor“)

- ❖ Erstellen einer Ereignisliste
- ❖ Für jedes Ereignis einen Geschäftsprozess identifizieren
- ❖ Externe und zeitliche Ereignisse (i. d. R. intern ausgelöst) unterscheiden

c) mittels Aufgabenbeschreibungen

- ❖ Was sind die Gesamtziele des Systems?
- ❖ Welches sind die zehn wichtigsten Aufgaben?
- ❖ Was ist das Ziel jeder Aufgabe?

3. Geschäftsprozesse für Sonderfälle formulieren

- o „Erweiterungen und Alternativen“ mittels Schablone erstellen
- o Aufbauend auf Standardfunktionalität mit extend die Sonderfälle formulieren, d.h. erweiterte Geschäftsprozesse beschreiben.

Checkliste Geschäftsprozesse

4. Aufteilen komplexer Geschäftsfälle

- Komplexe Schritte als eigene Sub-Prozesse spezifizieren (include)
- Komplexe Geschäftsprozesse (viele Sonderfälle) in mehrere Geschäftsprozesse zerlegen und Gemeinsamkeiten mit include modellieren
- Umfangreiche Erweiterungen als eigene Geschäftsprozesse spezifizieren und mittels extend an den Hauptprozess anbinden

5. Gemeinsamkeiten von Geschäftsprozessen ermitteln

- Auf redundanzfreie Beschreibung achten (include).

Analytische Schritte

6. »Gute« Beschreibung

- Verständlich für den Auftraggeber
- Extern wahrnehmbares Verhalten
- Fachliche Beschreibung des Arbeitsablaufs
- Standardfall vollständig und Sonderfälle separat
- Maximal eine Seite

7. Fehlerquellen

- Zu kleine und damit zu viele Geschäftsprozesse
- Zu frühe Betrachtung von Sonderfällen
- Zu detaillierte Beschreibung

Beispiel: Bearbeitung Schadensfall bei Versicherung

Geschäftsprozess: bearbeite Schadensfall

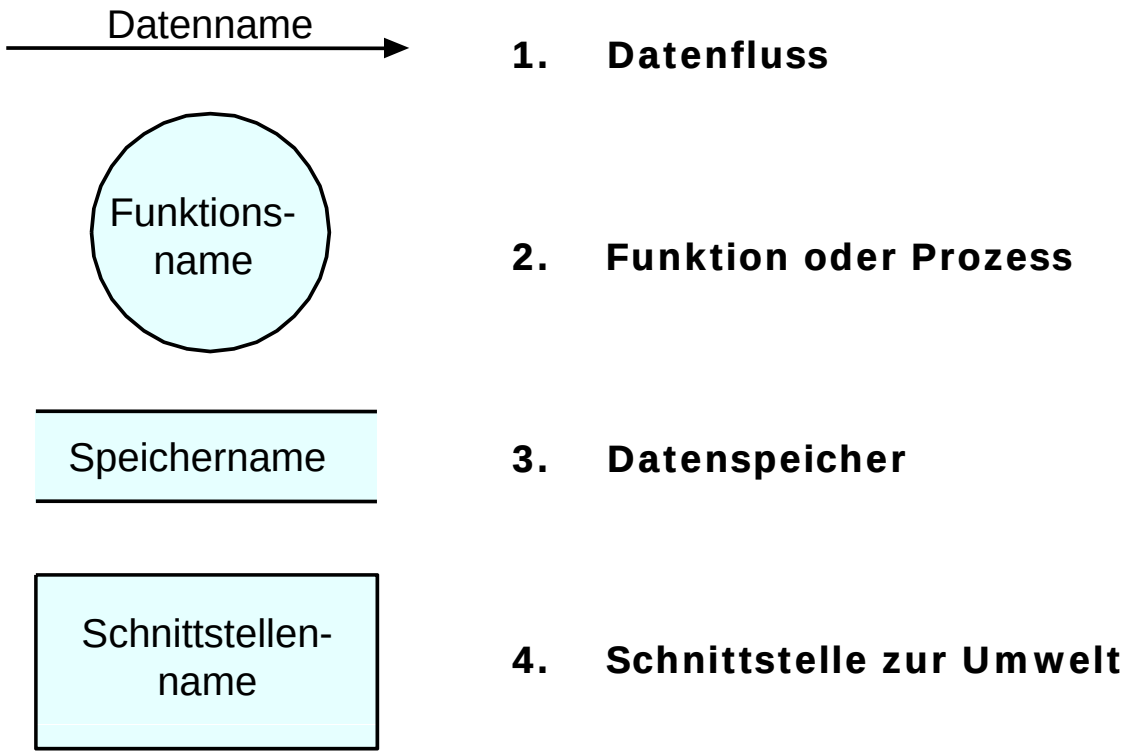
- **Ziel**: Bezahlung des Schadens durch die Versicherung
- **Kategorie**: primär
- **Vorbedingung**: -
- **Nachbedingung Erfolg**: Schaden ganz oder teilweise bezahlt
- **Nachbedingung Fehlschlag**: Forderung abgewiesen
- **Akteure**: Schadenssachbearbeiter
- **Auslösendes Ereignis**: Schadensersatzforderung der versicherten Person
- **Beschreibung**:
  - 1 Sachbearbeiter prüft die Forderung auf Vollständigkeit
  - 2 Sachbearbeiter prüft, ob gültige Police vorliegt
  - 3 Sachbearbeiter errechnet und überweist Betrag an Antragsteller
- **Erweiterungen**:
  - 1a vom Antragsteller vorliegende Daten sind nicht vollständig, Sachbearbeiter muss die Daten nachfordern
  - 2a Antragsteller besitzt keine gültige Police, Sachbearbeiter teilt ihm dies mit und schließt den Fall ab
  - 3a Schaden durch Police unvollständig gedeckt: Sachbearbeiter errechnet Teilbetrag und überweist diesen
- **Alternativen**: -

Datenflussdiagramm

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Ein **DFD** (*data flow diagram*) beschreibt die Wege von Daten bzw. Informationen zwischen Funktionen, Speichern und Schnittstellen und die Transformation der Daten bzw. Informationen durch Funktionen. DFDs stellen also die „Daten-Pipeline“ dar.

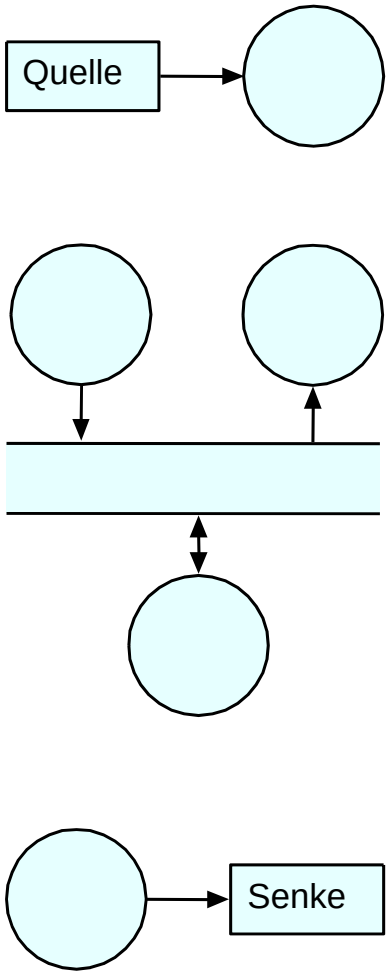
Notation nach / DeMarco 79 /



1. **Datenfluss**
2. **Funktion oder Prozess**
3. **Datenspeicher**
4. **Schnittstelle zur Umwelt**

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Datenflussdiagramm



Grundidee eines DFD besteht darin, dass man sich vorstellt, das zu entwickelnde System läuft bereits. Man macht sich keine Gedanken darüber, wie das System initialisiert und terminiert wird. Man konzentriert sich darauf, welche Informationen von wo nach wo durch das System fließen.

System hat **Schnittstellen** mit seiner Umwelt. Schnittstellen können Datenquellen und -senken sein

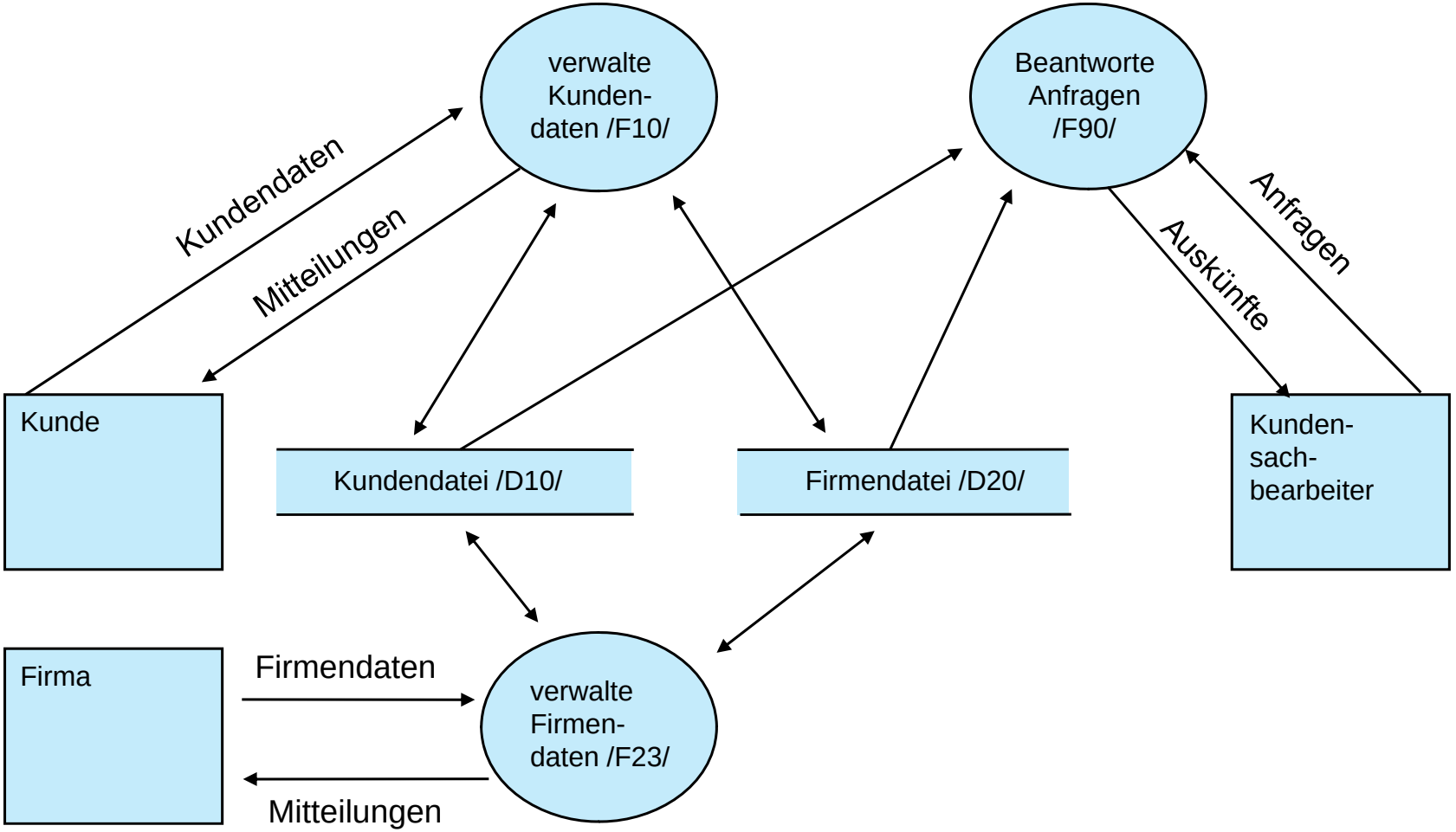
Umwelt besteht für das System aus Informations-quellen und Informationssenken.

Speicher sind Hilfsmittel zur Ablage von Informationen. Informationen können hineinfließen oder herausgelesen werden (siehe Pfeilrichtungen).

Funktionen transformieren den ankommenden Datenfluss in den abgehenden.

Datenflussdiagramm - Beispiel

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



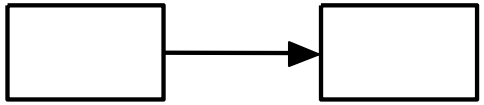
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Datenflussdiagramm - Syntaktische Regeln

Beim Zeichnen eines DFD sind folgende syntaktische Regeln einzuhalten

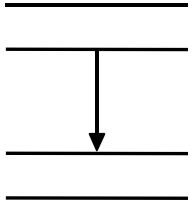
- Ein DFD enthält mindestens 1 Schnittstelle
- Jede Schnittstelle ist i. Allg. nur einmal vorhanden
  - Ausnahme: Wird das DFD unübersichtlich, dann kann eine Schnittstelle mehrfach gezeichnet werden
- Zwischen Schnittstellen gibt es keine Datenflüsse

Falsch:

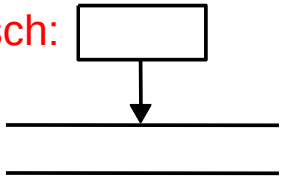


- Jeder Datenfluss hat einen Namen
  - Ausnahme: Datenflüsse, die zu Speichern führen und keinen Namen haben, beinhalten die gesamten gespeicherten Daten
- Zwischen Speichern dürfen keine direkten Datenflüsse gezeichnet werden

Falsch:



Falsch:



- Zwischen Schnittstellen und Speichern dürfen keine direkten Datenflüsse gezeichnet sein

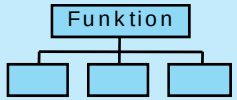
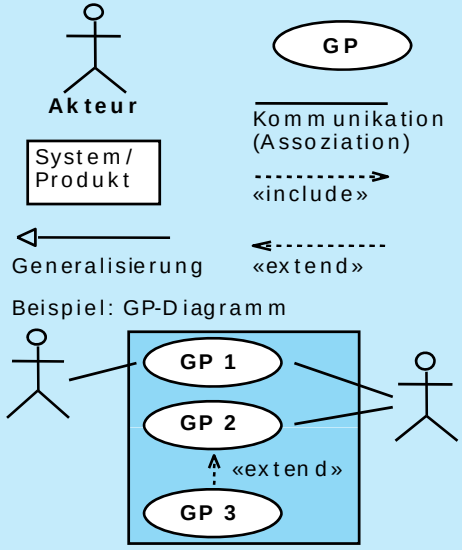
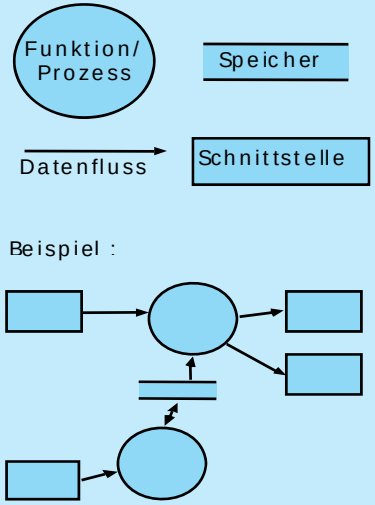
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Datenflussdiagramm - Semantische Regeln

- Das DFD beschreibt den Datenfluss, nicht den Kontrollfluss, daher enthält es weder Entscheidungen noch Schleifen
- Steht eine Schnittstelle für eine Vielzahl von beliebig vielen Instanzen, dann wird sie als eine Schnittstelle dargestellt, z. B. „Kunde“
- Wird das System durch eine eng begrenzte Anzahl von gleichartigen Schnittstellen begrenzt, die sich aber durch unterschiedliche Datenflüsse auszeichnen, dann ist eine getrennte Darstellung sinnvoll
- Eine Schnittstelle ist so zu wählen, dass sie die ursprüngliche Quelle oder Senke einer Information angibt
- Bei der Wahl einer Schnittstelle wird von der konkreten Ein- oder Ausgabe einer Information in das System vollständig abstrahiert (z. B. Drucker, Tastatur)
- Ein Datenflussname besteht aus einem Substantiv oder einem Adjektiv und einem Substantiv
    - Datenflussnamen enthalten niemals Verben
- Die Datenflussnamen sind so zu wählen, dass sie nicht nur die Daten, die fließen, beschreiben, sondern etwas darüber aussagen, was über die Daten bekannt ist (z. B. statt „Rechnungsnummer“ „gültige Rechnungsnummer“)
- Ein Funktions- bzw. Prozessname besteht aus einem einzigen Aktions-Verb gefolgt von einem einzigen konkreten Objekt oder einem konkreten Substantiv gefolgt von einem Aktions-Verb (z. B. „Rechnung erstellen“ oder „erstelle Rechnung“)
- Funktionsnamen repräsentieren Aktionen

Übersicht Basiskonzepte

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Basis-konzept	Funktionale Hierachie	Arbeitsablauf		Informationsfluss
Notation	Funktionsbaum	Geschäftsprozess (GP)		Datenflussdiagramm
»Muster« der Notation	Baumhierachie	Netz (UML)	Muster/Schablone	Netz
Elemente			Geschäftsprozess: Ziel: Kategorie: Vorbedingung: Nachbedingung Erfolg: Nachbedingung Fehlschlag: Akteure: Auslösendes Ereignis: Beschreibung: 1 2 Erweiterungen: 1a Alternativen: 1a	
Methodisches Vorgehen	top-down	Von den Akteuren ausgehend *(outside-in)*		Von den Schnittstellen ausgehend *(outside-in)*
Regeln	■ Zu einer Vaterfkt. nur fachlich eng zusammengehörige Fkt. ■ Pro Hierachieebene gleiches Abstraktionsniveau	■ GP-Name: Gerundium oder Substantiv-Verb oder Anfang-Ende ■ Zuerst Standardfall, dann Sonderfälle ■ Gemeinsames Verhalten: «include»-Beziehung ■ Erweitertes Verhalten: «extend»-Beziehung		■ Nur Datenfluss beschreiben, keinen Kontrollfluss ■ Schnittstelle ist Quelle/Senke von Informationen ■ Nur Funktionen/Prozesse greifen auf Schnittstellen und Speicher zu

Zusammenfassung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Die **funktionale Sicht** auf ein zu entwickelndes Produkt kann durch die hierarchische Gliederung von Funktionen, angeordnet in einem **Funktionsbaum**, beschrieben werden.

Durch **Geschäftsprozesse** können die Arbeitsabläufe von Akteuren mit dem Produkt auf einer hohen Abstraktionsebene beschrieben werden. Die Dokumentation erfolgt in **UML** durch **Geschäftsprozessdiagramme**. Zur Spezifikation einzelner Geschäftsprozesse kann eine **Geschäftsprozess-Schablone** eingesetzt werden.

Durch **DFD** kann der Informationsfluss zwischen Funktionen, Speicher und Schnittstellen zur **Umwelt** dargestellt werden.

Überblick LE 6

LE 5: Funktionale Sicht

LE 6: OO-Sicht

- Lernziele
- Objektorientierte Konzepte
- Zusammenfassung

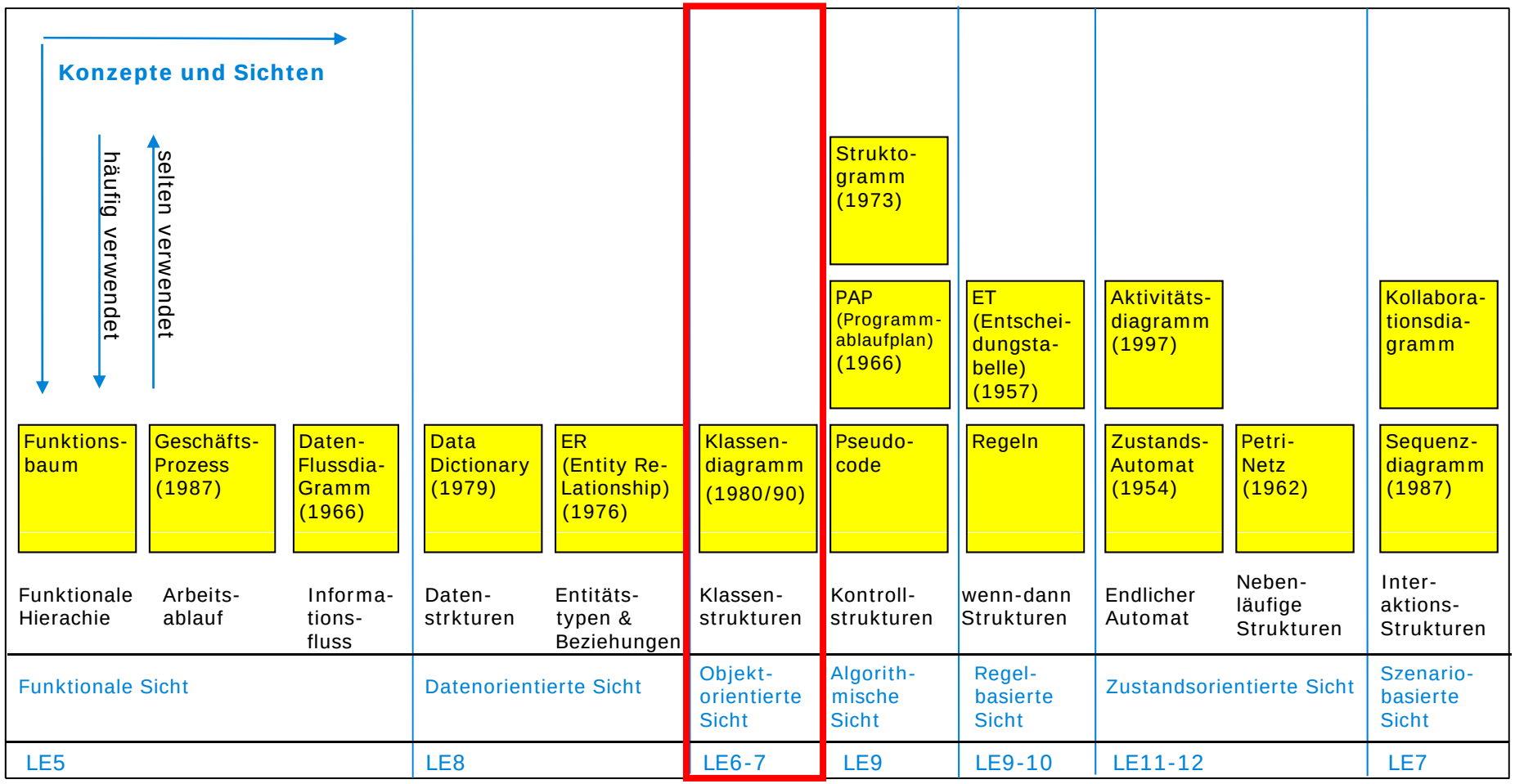
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Lernziele

1. Begriff „Objektorientierte Software-Entwicklung“ kennen
2. Begriffsdefinition an Beispielen: Klasse, Objekt, Attribut, Operation
3. Klassifikation von Operationen
4. Aufstellen von UML-Diagrammen für gegebene Problemstellung
5. CASE-Werkzeuge einsetzen können

Überblick

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Querverweis zu anderen Vorlesungen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Vorlesung Programmierung und Programmiersprachen

Die Vorlesung führt ein in grundlegende Konzepte der objektorientierten Programmierung und Modellierung am Beispiel von Java.

1. Grundlagen der objektorientierten Systemanalyse
2. Java - Grundbegriffe
3. Klassen, Objekte, Methoden
4. Abstraktionen, Generalisierungen, Schnittstellen
5. Vererbung und Polymorphie
6. Konzepte der Parallelverarbeitung

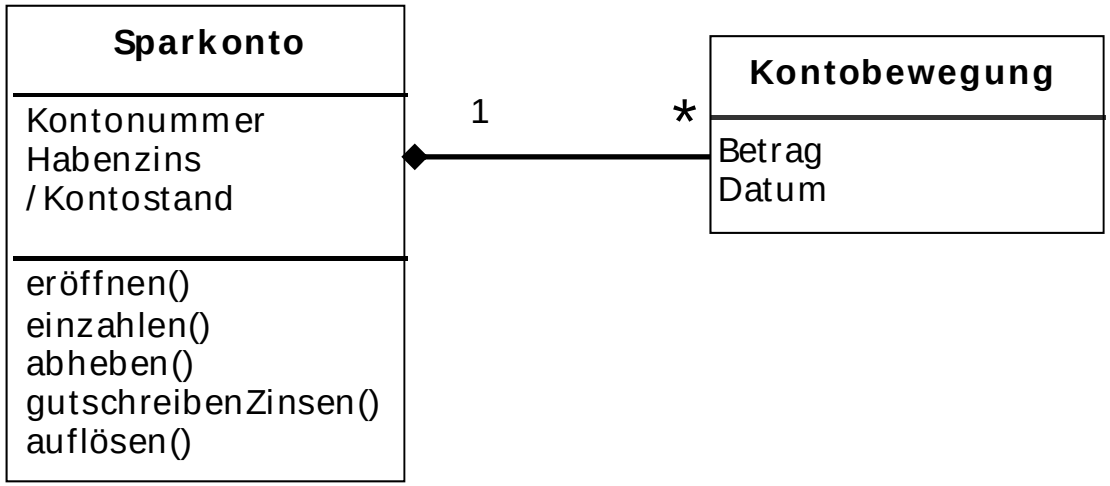
Wichtigste **OO-Konzepte** sind : Objekte, Klassen, Attribute, Operationen, Assoziationen, CRC-Karten, Vererbung, Pakete, Botschaften, Szenarios

Wir behandeln nur **CRC-Karten** und **Pakete** da die anderen Konzepte in der oben genannten Vorlesung schon ausführlich behandelt wurden.

CRC-Karten

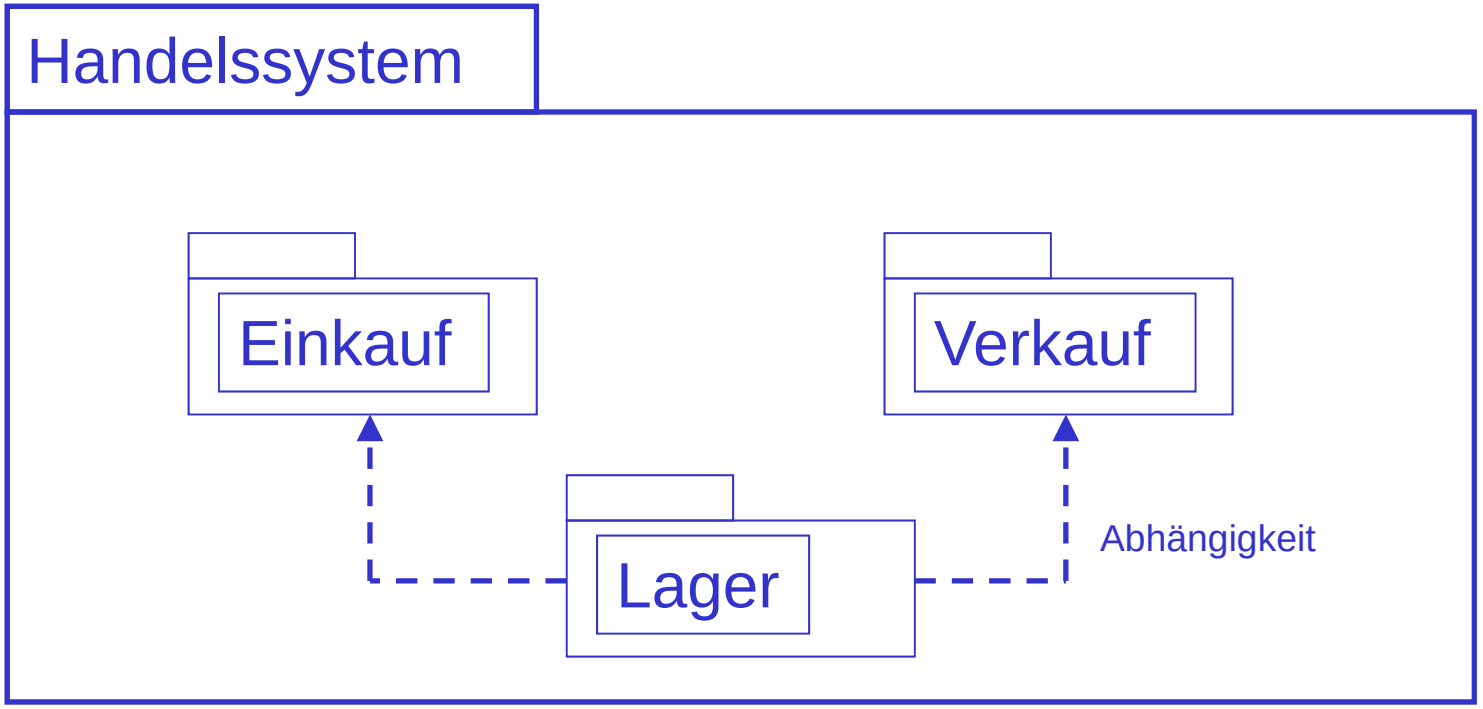
CRC-Karten (Class/Responsibility/Collaboration-Karten) sind Hilfsmittel zur OO-Modellierung in Form von Karteikarten. Oben auf der Karte wird der Name der Karte eingetragen. Restliche Karte wird in zwei Teile geteilt. Auf der einen Hälfte werden die **Verantwortlichkeiten** (responsibilities) der Karte notiert und auf der anderen die **Kollaborationen** (notwendige Klassen)

Class Sparkonto	
Responsibilities ■ verwaltet ein Sparkonto ■ delegiert Aufgaben an Kontobewegung	**Collaborations** ■ Kontobewegung



- **Pakete** sind **Strukturierungsmechanismus** und erlauben es, Komponenten zu einer größeren Einheit zusammenzufassen
- in UML zur Strukturierung beliebiger Modellelemente eingesetzt (Paketdiagramm)

Beispiel: Pakete eines Handelssystems



Änderung an „Lager“ erfordert evtl. Veränderungen an „Einkauf“ und „Verkauf“

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Methode zur Paketbildung

Kriterien: Das Paket soll eine logische Einheit bilden, d.h. es soll so strukturiert sein, dass es...

1. den Leser durch das Modell führt
2. einen Themenbereich enthält, der für sich allein betrachtet und verstanden werden kann
3. Klassen enthält, die logisch zusammengehören
   z.B. Artikel, Lieferant und Lager
4. für sich entworfen und evtl. implementiert werden kann, wobei eine wohldefinierte Schnittstelle zur Umgebung vorhanden ist.

Die **Schnittstelle** soll...

- Vererbungsstrukturen nur in vertikaler Richtung schneiden, d.h. zu jeder Unterklasse sollen alle Oberklassen in dem Paket enthalten sein
- keine Aggregation durchtrennen
- möglichst wenig Assoziationen enthalten

Die Kopplung zwischen den Klassen – innerhalb eines Pakets – soll möglichst groß und zwischen Paketen möglichst gering sein

- Jedes potenzielle Paket prüft man auf seine Kopplungseigenschaft.
- Verbesserungen durch Verschieben von Elementen zwischen den Paketen möglich

Zusammenfassung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Objektorientierte SW-Entwicklung basiert auf einer Reihe von Grundkonzepten. Hier werden verschiedene Diagrammarten benutzt, um unterschiedliche Aspekte darzustellen. Als grafische Notation wird UML verwendet.

OO-Konzepte sind gut geeignet für Datenkapselung

Für die Wartbarkeit des entstehenden Software-Systems ist es entscheidend, dass das Konzept der Vererbung sinnvoll eingesetzt wird

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag,Heidelberg, 2000.

- Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, Addison-Wesley, 1999
- Booch G., Object-Oriented Analysis and Design with Applications, 2. Auflage, The Benjamin/Cummings Publishing Company, Redwood City, 1994
- DeMarco T., Structured Analysis and System Specification, Yourdon Press, Englewood Cliffs, 1979
- Balzert Heide, Lehrbuch der Objektmodellierung – Analyse und Entwurf, Spektrum Akademischer Verlag, Heidelberg, 1999
- Jacobson I, Ericson M., Jacobson A., The Object Advantage Buisness Process Reengineering with Object Technology, Addison-Wesley, Workingham, 1994
- OMG Unified Modeling Language Specification, Version 1.3, June 1999, http://www.rational.com