

# Systementwicklung

## Analyse

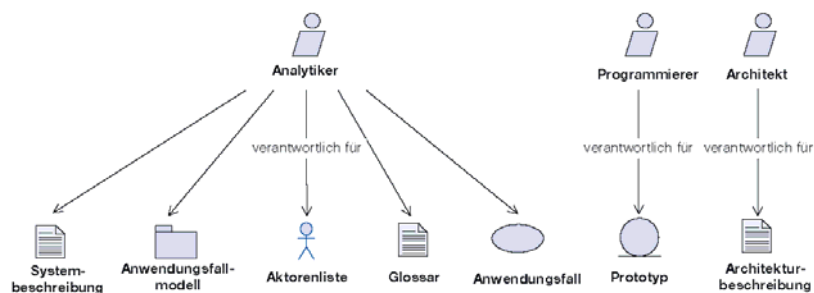
Uwe H. Suhl und Chris Bizer

SS 2008

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Ziele der Analyse

- Verstehen der Anforderungen durch das Projektteam.
- Verstehen des Anwendungsbereichs und dessen Umfeld.
- Erlernen der Fachsprache der AnwenderInnen.
- Herstellen eines Pflichtenhefts, das das gewonnene Wissen in geordneter und verständlicher Form repräsentiert.
- Produkte und Rollen im Arbeitsschritt Analyse:



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Ablauf der Veranstaltung

- 1. Gliederung und Inhalt des Pflichtenhefts**  
Anleitung für Sonnenschein - Übungsaufgabe 2
- 2. Modellieren mit UML**
  1. Use Case Diagramm
  2. Aktivitätsdiagramm
  3. Klassendiagramm
  4. Verteilungsdiagramm

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Gliederung Pflichtenheft

- 1. Systemidee**
- 2. Aktorenliste**
- 3. Anwendungsfälle**
  - UML-Anwendungsfalldiagramm für das Gesamtsystem
  - Tabellarische Beschreibung jedes Anwendungsfalls
  - UML Aktivitätsdiagramm je Anwendungsfall
- 4. Domainenmodell**
  - Objektklassen sowie deren Beziehungen und Eigenschaften
- 5. Architekturbeschreibung**
- 6. Sonstige Anforderungen und Regeln**
- 7. Glossar**
- 8. Schnittstellen**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Systemidee

### ■ Beschreibt in Textform die:

- Zielsetzung des Systems
  - Was leistet das System?
  - 5-10 wichtigste Leistungsmerkmale
  - Wie verbessert es die bisherige Situation?
- Abgrenzung des Systems
  - Was leistet das System nicht?
- Wichtigste Rahmenbedingungen
  - Hardware, Software, Organisation, Schnittstellen zu anderen Systemen

### ■ Auf das Wesentliche beschränken!

- Insgesamt höchstens 1 Seite.
- Stilistisch so gehalten, wie man das System auf einem Produktkarton beschreiben würde.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel: Systemidee

Flitz-Auto 1.0 ist die Software, die Sie in allen wichtigen Anwendungsfällen einer Autovermietung unterstützt: Sie können Kfz reservieren (inkl. Ändern und stornieren), vermieten (inkl. Vertragserstellung und Rücknahmeprotokoll) und die Vermietung abrechnen (Einzel-, Sammel- und Monatsrechnungen).

Selbstverständlich kann Flitz-Auto 1.0 auch die zu vermietenden Kfz, die vorhandenen Mietstationen und die Kunden mit allen wichtigen Stammdaten (Adressen, Bankverbindungen etc) verwalten. Eine Datenübernahme aus dem Altsystem ist möglich.

Weitere wichtige Eigenschaften von Flitz-Auto 1.0: Mandatenfähigkeit, Agenturbetrieb, Teilnahme am Single-Sign-On, Altdaten-Archivierung, verschiedene Auswertungen, Berichte und Statistiken (nicht im Internet), 24\*7-Betrieb. Eine Bedienung über Funktionscode ist nicht mehr nötig und nicht mehr möglich – Flitz-Auto 1.0 ist einfach bedienbar und mit integrierter Hilfe ausgestattet. Für alle Fälle kennt die Hilfe-Funktion jedoch die alten Funktionscodes und zeigt Ihnen dazu die neue Bedienung in Flitz-Auto.

Flitz-Auto 1.0 läuft im Netzwerk Ihres Unternehmens und bietet für Kunden und Agenturen auch einen Internetzugang. Systemvoraussetzungen sind Schnittstellen zum bestehenden Tarif- und Produktsystem TSC und zur Finanzbuchhaltung SAP-FI. Zusatzmodule zur Kfz-Überführung /-disposition, zur Schadenbearbeitung und zu Beschwerdemanagement sind ebenso wie Mehrwährungs- und Mehrsprachenfähigkeit für Flitz-Auto 2.0 in Planung.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Aktorenliste

- Beschreibt die Akteure, die mit dem System interagieren sowie deren Interessen.

- Beispiel:

Akteur	Beschreibung und Interessen	Ansprechpartner
Kunden	- Es gibt drei verschiedene Kundengruppen ... - Kunden wollen zu jeder Tageszeit mit möglichst geringem Aufwand Mietwagen reservieren.	
Call-Center Mitarbeiter	- Call-Center Mitarbeiter nehmen telefonisch Reservierungen an. - Wollen eine einfach und schnell zu bedienende System-Oberfläche. - Es bestehen Ängste, dass durch mehr Internet-Reservierungen die Arbeitsplätze im Call-Center gefährdet sind.	Frau Müller (Tel. -94353)
Administrator	Administratoren führen die Datensicherung durch und legen neue Benutzer an. Sie wollen ...	Herr Segen (Tel. -32543)

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Anwendungsfälle (Use-Cases)

- Ein Anwendungsfall beschreibt die Interaktionen zwischen einem (oder mehreren) Akteure und dem System bei der Bearbeitung einer bestimmten, abgegrenzten Aufgabe.

- Akteure

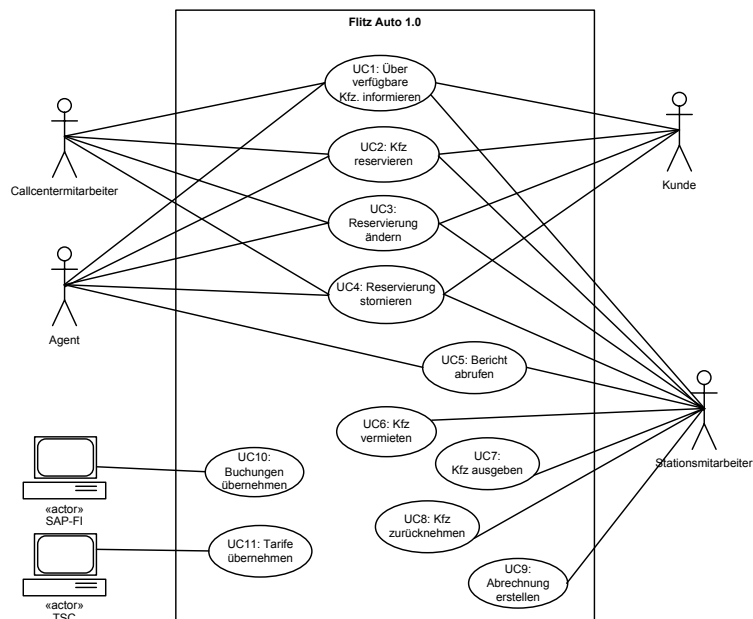
- Tauschen von außen Informationen mit dem System aus (Benutzer, andere Systeme, Geräte).
- Akteure werden durch die Rolle, die ein Benutzer gegenüber dem System einnimmt charakterisiert.

- Use Cases bilden die Basis für

- alle weiteren Artefakte im Systementwicklungs-Projekt
  - Architektur
  - Testfälle
  - Entwurf
  - Implementierung
- Aufwandsschätzung
- Vertrag mit AuftraggeberIn
- Abnahmekriterium

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel: Anwendungsfalldiagramm



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Anwendungsfallbeschreibung

Beschreibt einen Anwendungsfall in tabellarischer Form. Sie enthält:

1. Eine eindeutige Bezeichnung, welche in anderen Anwendungsfällen oder Dokumenten einfach referenziert werden kann, inkl Nummerierung.
2. Kurzbeschreibung des Anwendungsfalls (1-2 Sätze).
3. Die Angabe des oder der primären Akteure, welche diesen Anwendungsfall benutzen.
4. Auslöser der Anwendungsfalls
5. Vorbedingungen für die Ausführung des Anwendungsfalls.
6. Ergebnisse des Anwendungsfalls, welche sich ausschließlich auf Änderungen des Systemzustands und des Datenbestands beziehen.
7. Den genauen Ablauf des wichtigsten Erfolgsszenarios anhand der Anwender-System-Interaktion. Jeder Interaktionsschritt wird fortlaufend nummeriert.
8. Offene Fragen: Fragen, die mit den Anwendern in Bezug auf diesen Anwendungsfall noch geklärt werden müssen.
9. Bei größeren Projekten: Wer hat wann, was am Anwendungsfall geändert?

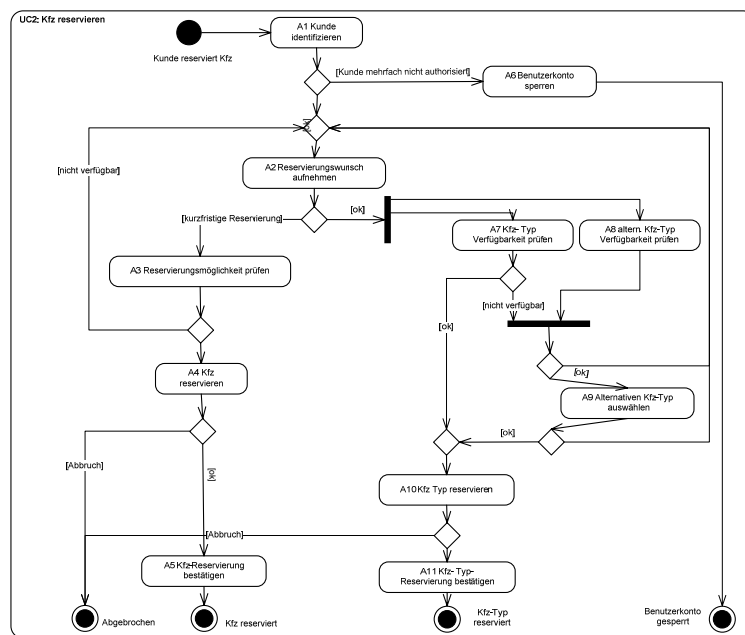
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel Anwendungsfallbeschreibung

UC2-1: Kfz reservieren (Call-Center)			
Kurzbeschreibung	Das Callcenter reserviert für einen Kunden für einen definierten Zeitraum ein Kfz.		
Akteure	Callcenter-Agent		
Auslöser	Ein Kunde ruft den Callcenter an, um ein Kfz zu reservieren.		
Vorbedingungen	keine		
Ergebnis	Reservierung gespeichert, Reservierungsbestätigung an den Kunden übermittelt		
Eingehende Informationen	Kundennummer oder Kundenname, Anrufername, Reservierungswunsch		
Essenzielle Schritte	<ol style="list-style-type: none"> <li>1. <b>Kunde identifizieren</b> Die Daten des Kunden werden überprüft, ggf. Neukunde anlegen.</li> <li>2. <b>Reservierungswunsch aufnehmen</b> Reservierungswünsche, z.B. Kfz-Typ, Zeitraum, Abhol- und Rückgabeort</li> <li>3. <b>Reservierungsmöglichkeit prüfen</b> System ermittelt, ob der Reservierungswunsch erfüllt werden kann.</li> <li>4. <b>Kfz reservieren</b> Reservierung wird im System gespeichert.</li> <li>5. <b>Reservierung bestätigen</b> Nach der Reservierung werden dem Kunden alle relevanten Informationen zugeschickt.</li> </ol>		
Offene Fragen	Darf der Abholort vom Rückgabeort abweichen?		
Änderungen	Datum	Mitarbeiter	Kommentar
	12.10.2006	Endres	Erste Version

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel UML Aktivitätsdiagramm



Übernommen aus Oestereich, 2006

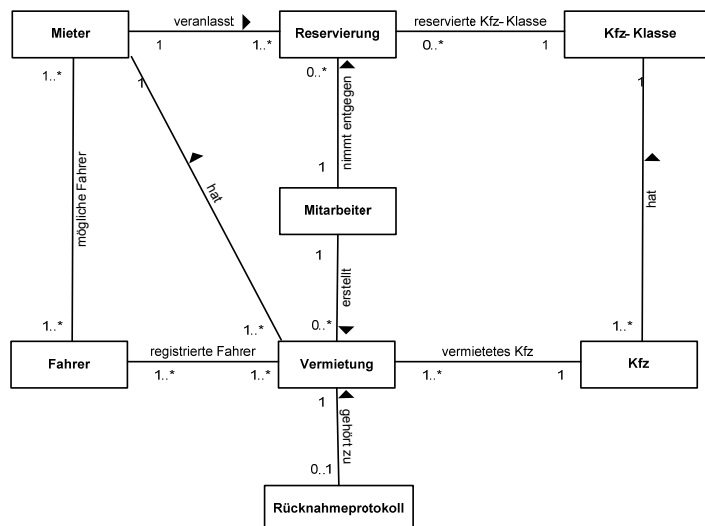
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Domänenmodell

- Das Domänenmodell identifiziert die wichtigsten fachlichen Gegenstände, die vom System repräsentiert werden sollen.
- Beispiele: Kunden, Mitarbeiter, Mietwagen, Vermietungen
- Wird als UML-Klassendiagramm dargestellt.
  - Klassen mit ihren Attributen (keine Operationen).
  - Assoziationen (Beziehungen) zwischen Klassen mit Namen/Rollen und Multiziplicitäten.
  - UML Klassendiagramm übernimmt die frühere Rolle eines Entity-Relationship-Diagramms.
- Domänenmodell wird mit Anwendungsfallbeschreibungen auf Konsistenz verglichen.
- Domänenmodell wird später in der Phase Entwurf mit implementierungs-technischen Details verfeinert.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel: Domainenmodell als UML Klassendiagramm

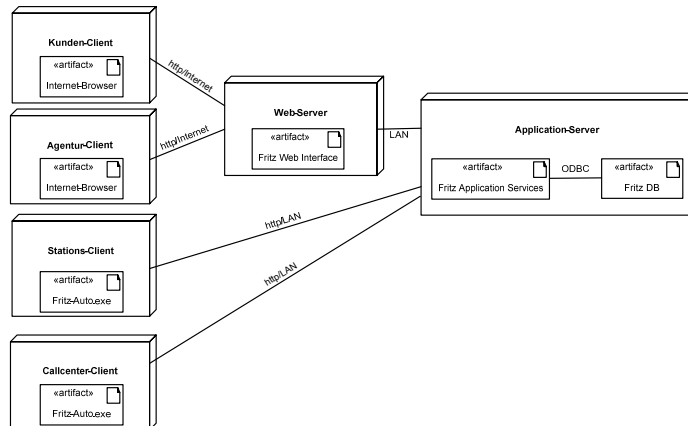


- Das UML Domänenmodell entspricht weitgehend dem Entity-Relationship Modell (ERD) mit Chen-Notation das in der Datenmodellierung benutzt wird

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Architekturbeschreibung

- Grobe Beschreibung der Systemarchitektur aus Anwendersicht.
- Besteht aus:
  - Text, der die einzelnen Komponenten des Systems beschreibt.
  - UML Verteilungsdiagramm zur Visualisierung der physikalischen Struktur des Systems



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Sonstige Anforderungen und Regeln

- Beschreibung aller außerhalb von Use Cases liegenden Anforderungen und Geschäftsregeln in einem zentralen Anforderungskatalog.

ID	Anforderung/Regel
R1	Höherstufung: Wenn ein Mietwagen der gebuchten Klasse nicht verfügbar ist, bekommt der Mieter einen Wagen einer höheren Klasse.
R2	Mindestmietzeit: Die Mindestmietzeit beträgt 1 Tag.
R3	Mindestalter des Fahrers: Ein Fahrer muss mindestens 21 Jahre alt sein.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)



## Glossar

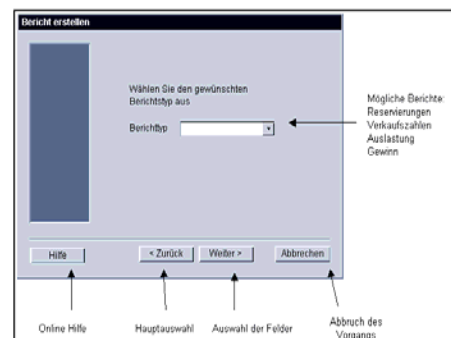
- Das Glossar definiert wichtige fachliche Begriffe
- Zweck des Glossars: Vermeidung von Missverständnissen
- Z.B. Definition der
  - fachlichen Klassen aus dem Klassendiagramm (z.B. Kunde)
  - Assoziationen und Rollen (Fahrer)
  - Wichtige fachliche Prozesswörter (reservieren, unverbindlich reservieren)

Begriff	Definition
Fahrer	Ein Mensch, der ein Kfz der Kfz-Vermietung fährt, fahren soll oder gefahren ist.
Kunde	Ein Kunde ist eine natürliche oder juristische Person, die die Dienstleistungen der Kfz-Vermietung nutzen kann.
reservieren	Reservieren ist der Vorgang, jemanden für einen gewissen Zeitraum die exklusive Option einzuräumen einen oder mehrere Gegenstände zu buchen. Eine Reservierung impliziert keinen rechtlich bindenden Vertrag.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Schnittstellen und Oberflächenprototyp

- Beschreibe zu jedem Use Case, die benötigten Schnittstellen
  - Ein- und ausgehende Daten
  - Eingabemasken (ggf. Oberflächenprototyp)
  - erzeugte Dokumente (Briefe, Berichte)
  - Daten-Schnittstellen zu externen Systemen (z.B. Webservice, EDI, XML)
- Oberflächenprototyp, visualisiert
  - Eingabemasken, Nutzerführung und Interaktion
  - noch keine Funktionalität
- Zukünftige BenutzerInnen können Anforderungen oft sehr viel besser anhand eines Prototypen formulieren.



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel Schnittstellenbeschreibung

Name	Kundensuche
Kurzbeschreibung	Mit diesem Dialog kann mit Hilfe von Suchbegriffen nach einem einzelnen Kunden gesucht werden.
Verwendung	Alle Mitarbeiter in den Vermiet-Stationen und im Callcenter, durchschnittlich 40-mal täglich pro Benutzer.
Eingabefelder	Verschiedene Suchbegriffe, z.B. Kundennummer, Name, Telefon, etc.
Anzeigefelder	Kundendaten (Name, Anschrift)
Verzweigungsmöglichkeiten	Kundenakte, Abbrechen, Kfz-Reservierung
Aktionen	Suche starten, Auswahl eines Kunden

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Checkliste Pflichtenheft

- Sind alle Use Cases bekannt, welche die Anwender durchführen möchten?
- Lässt sich jeder Use Case einer Gruppe von Anwendern zuordnen (Aktoren)?
- Ist für jeden Use Case klar, welche Daten dafür notwendig sind, und welche Daten dabei entstehen? Sind diese Daten im Domänenmodell enthalten?
- Ist das Anwendungsfalldiagramm übersichtlich und aussagekräftig?
- Ist die Benennung der Anwendungsfälle intuitiv verständlich?
- Ist die Beschreibung der Anwendungsfälle zur Erstellung eines ersten Prototypen und des Analysemodells ausreichend?
- Sind alle nicht funktionalen Anforderungen bekannt (Qualitätsanforderungen, Leistungsanforderungen, Sicherheitsanforderungen)?

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Kriterien für eine erfolgreiche Analyse

- **Richtige Ansprechpartner auf Kundenseite (zukünftige Anwender für Anforderungen, techn. Verantwortliche für techn. Infrastruktur, Management für Geschäftliches).**
- **Analysedokumentation berücksichtigt Sprache und Begriffe des Kunden und ist für diesen auch verständlich; alle Begriffe sind eindeutig definiert.**
- **Mehrere Iterationen inklusive Reviews unter intensiver Einbindung der Kundenvertreter.**
- **Vollständige und konsistente Dokumentation in Form eines Pflichtenhefts.**

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

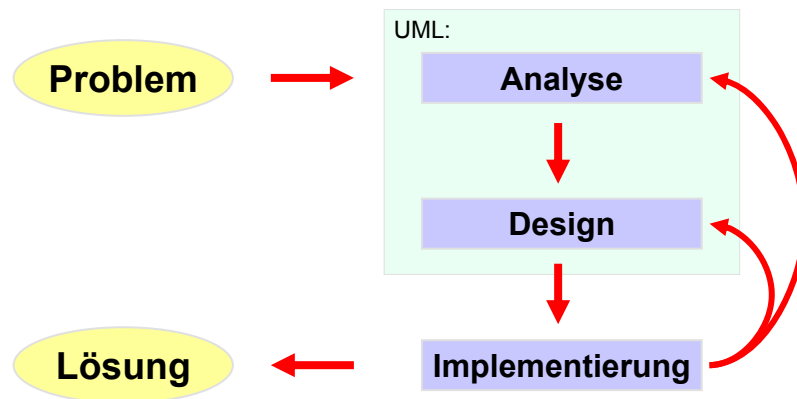
## Unified Modeling Language (UML)

- **Standardisierte Notation für die Analyse, den Entwurf und die Dokumentation von Softwaresystemen.**
- **Standardisiert durch die Object Management Group (OMG)**
  - 800 Mitglieder, inkl. IBM, HP, Microsoft, Oracle, DaimlerChrysler
- **Sehr breite Palette an UML Tools**
  - Z.B. Visio, UModel, Together
  - Liste <http://www.oose.de/umltools.htm>
- **Material und Links zu UML**
  - Chris Rupp et al.: UML 2 glasklar
  - Harald Störrle: UML 2 für Studenten
  - [http://de.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://de.wikipedia.org/wiki/Unified_Modeling_Language)

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

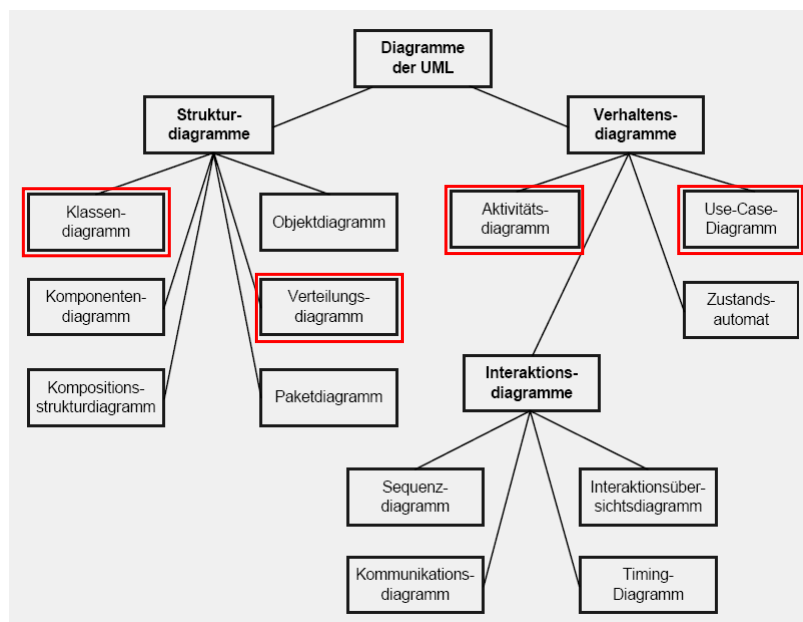
## Anwendung der UML

- UML unterstützt die der Implementierung vorgelagerten Arbeitsschritte Analyse und Design.
- UML wird zur Dokumentation der Implementierung verwendet.



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## UML Diagrammtypen

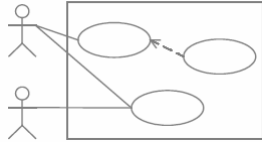


Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Wir verwenden für die Analyse

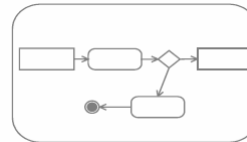
### Use Case Diagramm

Was leistet das System für seine Umwelt?



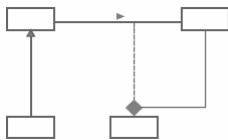
### Aktivitätsdiagramm

Wie laufen die Prozesse ab?



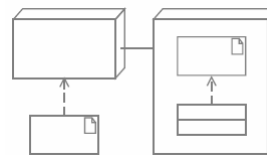
### Klassendiagramm

Aus welchen Klassen besteht das System?



### Verteilungsdiagramm

Wie verteilt sich das System physikalisch?



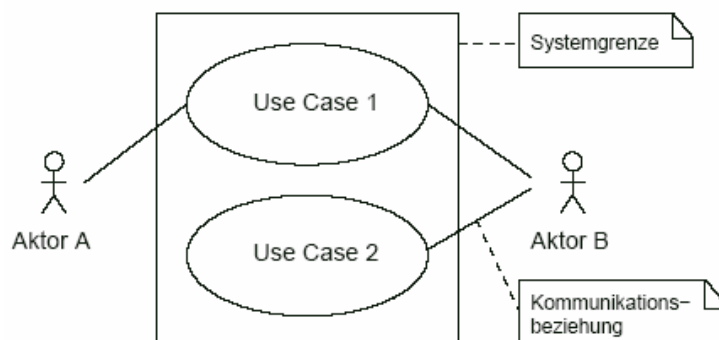
Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Use Case Diagramme

### ■ Fragestellung: Was soll mein System eigentlich leisten?

### ■ Dient der

- Darstellung der externen Nutzersicht auf das System.
- Zerlegung des Systems in handhabbare, logische Teile.



Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Elemente eines Use Case Diagramms

### ■ Das System

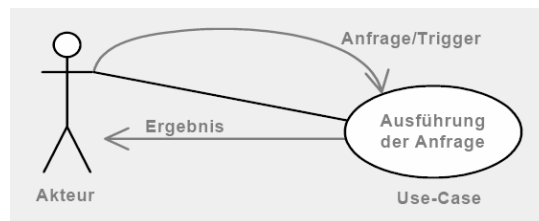
- Darstellung erfolgt durch ein Rechteck, das den Systemnamen enthält und dessen Kanten die Systemgrenzen darstellen.

### ■ Use Cases

- Darstellung als Ellipse, die den Namen und die Nummer des Use Cases enthält.
- Beschreibt eine Reihe von Aktionen, die nacheinander ein Verhalten formen.
- Wird immer von einem Akteur ausgelöst. Hat immer ein Ergebnis

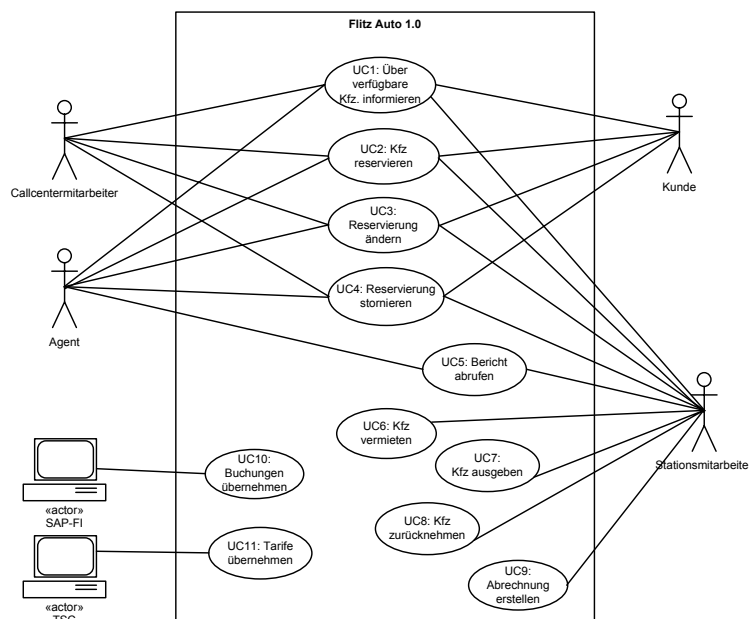
### ■ Akteure

- Darstellung als „Strichmännchen“
- Repräsentiert eine Rolle, die von außen mit dem System interagiert.



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

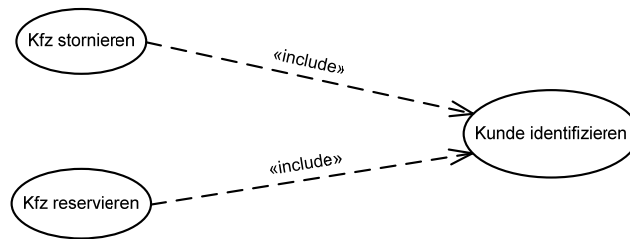
## Beispiel: Anwendungsfalldiagramm



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## include-Beziehung

- Zeigt, daß ein Use Case das Verhalten eines anderen importiert.
  - Ein Use Case darf durch mehrere andere Use Cases inkludiert werden.
  - Verhalten wird immer inkludiert – nicht optional
- Aufteilung des Diagramms in primäre und sekundäre Use Cases.



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Übung 1: Definition von Use Cases

- Entwerfen sie ein Use Case Diagramm für das Kassensystem in der Mensa (Gesamtsystem inkl. Kartenaufladung).
- Überlegen sie sich hierzu, wer die relevanten Akteure sind und wie sie mit dem System interagieren.

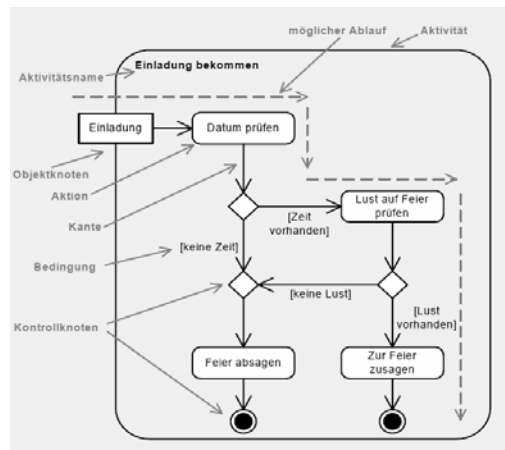
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## UML Aktivitätsdiagramm

■ Ein Aktivitätsdiagramm beschreibt einen Ablauf inklusive Nebenläufigkeiten und alternativen Entscheidungswegen.

■ Aktivitätsdiagramme bestehen aus:

- Mindestens einer Aktivität.
- Aktionen: Einzelschritte die durchlaufen werden.
- Objektknoten: Repräsentieren beteiligte Daten oder Gegenstände.
- Kontrollknoten zur Ablaufsteuerung
- Verbindenden Kanten: Stellen Kontroll- bzw. Objektflüsse dar.



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Grundelemente von Aktivitätsdiagrammen

### ■ Aktivität

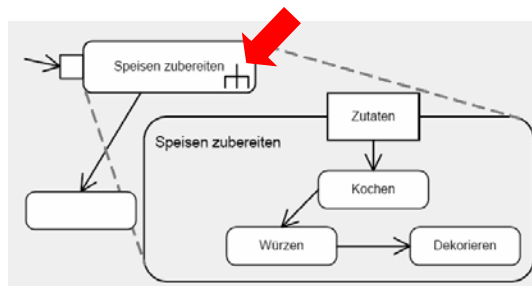
- Eine Aktivität ist ein zu beschreibender Ablauf.
- Darstellung durch einen Rahmen, der den Namen der Aktivität enthält.

### ■ Aktion

- Einzelschritt der Aktivität, z.b. Kundennummer zuteilen.
- Umgangssprachlich: Ablaufschritt

### ■ Aktionen mit Unteraktivität

- Aktionen können durch eine Unteraktivität näher beschrieben werden.
- Vorteil: Lesbarkeit und Übersichtlichkeit
- Kennzeichnung durch Harke rechts unten.

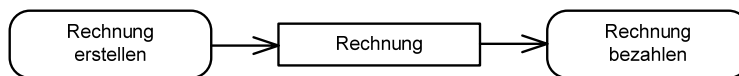


Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)



## Objektknoten

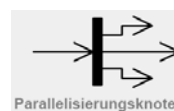
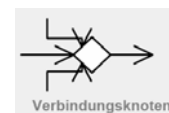
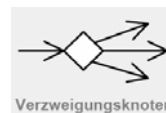
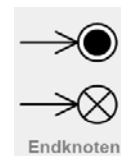
- Repräsentieren Objekte einer bestimmten Klasse.
- Ein Objektknoten ist das Ergebnis einer vorangegangenen Aktion.
- Ein Objektknoten ist die Eingabe für die folgende Aktion.
- Beispiele: Rechnung, Reservierung, Zwischenprodukt
- Darstellung von Objektknoten durch Rechtecke.
- Beispiel: Objektfluss zwischen zwei Aktionen



Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Kontrollknoten

- **Startknoten**
  - Startpunkt des Ablaufs bei Aktivierung.
- **Endknoten**
  - a) für Aktivitäten; beendet gesamte Aktivität.
  - b) für Kontrollflüsse; markiert Ablaufende.
- **Verzweigungsknoten**
  - spaltet Kante in mehrere Alternativen.
  - Die Bedingung der Verzweigung kann an den Kanten notiert werden.
- **Verbindungsknoten**
  - führt Kanten unsynchronisiert zusammen.
- **Parallelisierungsknoten**
  - ein eingehender Ablauf wird in mehrere parallele Abläufe geteilt.
- **Synchronisationsknoten**
  - vereint eingehende Abläufe zu einem gemeinsamen Ablauf.
  - Wartet, bis alle Abläufe eingegangen sind.

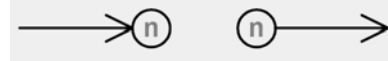


Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Sonstige Knoten

### ■ Sprungmarken

- Unterbrechung von Kanten um die Übersichtlichkeit zu erhöhen.



### ■ Ereignis

- Kann als Startknoten dienen



### ■ Signal

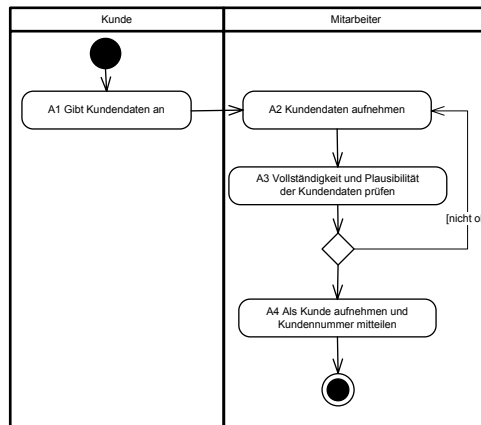
- Löst Ereignis aus



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Partitionen, Aktivitätsbereiche

- Dienen zur Unterteilung einer Aktivität in Bereiche mit gemeinsamen Eigenschaften
- z.B. Rolle, Abteilung, Verantwortlichkeit, Standort, Subsystem.
- Beispiel: Aktivitätsbereiche Kunde und Mitarbeiter



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Checkliste Aktivitätsdiagramm

- Existiert zu jedem Anwendungsfall ein Aktivitätsdiagramm?
- Wird jeder Anwendungsfallschritt aus der tabellarischen Anwendungsfallbeschreibung durch mindestens eine Aktivität im Aktivitätsdiagramm repräsentiert?
- Sind alle bekannten Ausnahmen und Varianten vollständig im Aktivitätsdiagramm dokumentiert?
- Sind alle Abbruchmöglichkeiten berücksichtigt?
- Sind alle Aktivitäten durchnummeriert?

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Übung 2: Aktivitätsdiagramm

- Zeichnen Sie ein Aktivitätsdiagramm für den Use Case „Essen kassieren“ aus Übung 1.
- Bitte beachten Sie, dass
  - manche Mensabesucher vergessen ihre Menakarte auf das Lesegerät aufzulegen und
  - dass der gespeicherte Betrag manchmal nicht für die Zahlung ausreicht.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## UML Klassendiagramm

### ■ Zentraler Diagrammtyp in UML

### ■ Dient der Darstellung der Struktur eines Systems

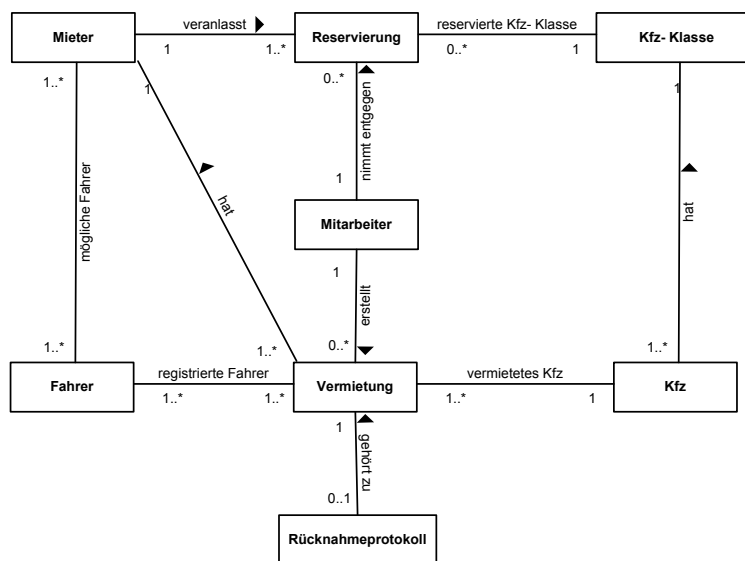
- dem Aufzeigen von statischen Eigenschaften und Beziehungen
- der Sammlung grundlegender Modellierungskonstrukte
- ist Antwort auf die Frage „Wie sind Daten und das Verhalten meines Systems strukturiert?“

### ■ Zwei grundlegende Einsatzszenarien

- konzeptuell-analytisch (Analysephase)
  - Ziel: Erfassung und Darstellung fachlicher Zusammenhänge.
  - Unwichtig: Technische Realisierungsdetails wie Attributdatentypen, Operationen, Schnittstellen.
- logisch (Entwurfsphase)
  - Ziel: Genaue technische Beschreibung des Systems
  - Dient als Implementierungsvorschrift bzw. als Grundlage für die automatische Quellcode-Generierung.

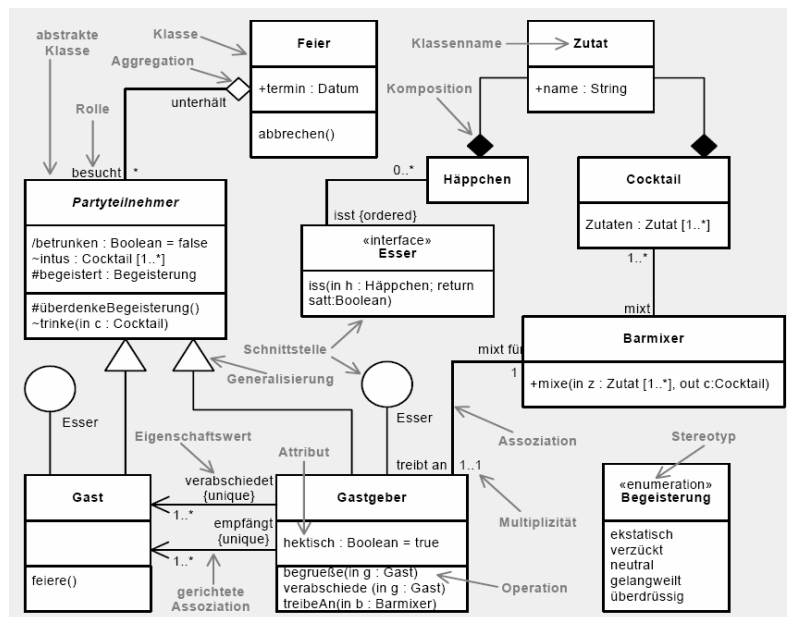
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel Analyse Klassendiagramm



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel: Entwurf Klassendiagramm



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Begriffe: Klassendiagramm

- **Klasse:** Eine Klasse ist die Definition der Attribute, Operationen und der Semantik für eine Menge von Objekten.
- **Objekt (Instanz):** Ausprägung einer oder mehrerer Klassen.
- **Schnittstelle:** Beschreibung des externen Verhaltens einer Klasse.  
(siehe Kapitel Entwurf)
- **Attribut:** Eigenschaften einer Klasse. Beschrieben durch Name, Datentyp etc.
- **Operationen:** Verhalten einer Klasse (siehe Kapitel Entwurf)
- **Assoziationen, mit den Sonderformen Aggregation und Komposition, beschreiben die Beziehungen zwischen Klassen.**
- **Generalisierungsbeziehungen:** Beschreibt Vererbungsbeziehungen zwischen Klassen.
- **Multiplizität:** Definiert das Intervall der erlaubten Kardinalitäten.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Klassen und Objekte

- Klassen Beschreiben eine Menge von Objekten mit gemeinsamen Attributen und Operationen.
- Klassen werden als Rechtecke dargestellt.
- Die Darstellung kann Attribute und Operationen enthalten.
- Objekte werden durch Unterstreichung des Klassennamens gekennzeichnet.

Klassenname

Klassenname

attribut

operation ()

### Klasse

Kunde
name: String adresse: String umsatz: Real
getName(): String umsatzErhoehen(n: Real)

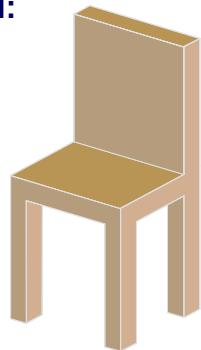
### Objekt

<u>:Kunde</u>
name = "Fritz Meier" adresse = "München" umsatz = 4590,30

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Attribute

- Eigenschaften einer Klasse.
- Attribute werden durch Name, Datentyp etc. beschrieben
- Für jedes Attribut wird Speicherplatz entweder in der Klassenausprägung (d.h. im Objekt) reserviert oder per Referenz (Pointer) gespeichert
- Beispiel:



Stuhl
+ typ : String + hoehe : double + breite : double + tiefe : double + hoeheSitzflaeche : double - gewicht : double - farbe : Farbe - besitzer : person

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Allgemeine Syntax der Attributdeklaration

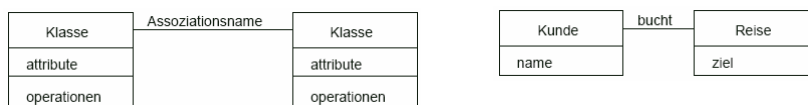


- **Sichtbarkeit** (siehe Kapitel Entwurf)
- **Name: Attributname, frei wählbar, klein geschrieben.**
- **Typ: Datentyp des Attributs. Standarddatentyp (Boolean, Integer, Real, String) oder Klassenname.**
- **Multiplizität: legt Unter- und Obergrenze der Anzahl der Ausprägungen fest.**
  - [1..\*], [2..4], [\*]. Nur notiert, wenn nicht [1].
  - Unterstützung von NULL-Werten wird durch Multiplizität [0..] ausgedrückt.
- **Beispiele**
  - name : String = 'Unbekannt'
  - vornamen : String [1..5]
  - gebDatum : Date
  - besitzer : Person

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Assoziationen

- **Eine Assoziation beschreibt eine Menge gleichartiger Beziehungen zwischen Objekten bestimmter Klassen.**



- **Beschrieben durch**
  - Assoziationsname, ggf. mit ► zur Kennzeichnung der Leserichtung
  - Assoziationsrollen
  - Multiplizitäten

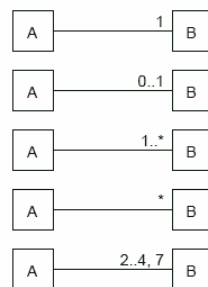
Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Multiplizitäten

- Geben an, wie viele Objekte einer Klasse mit wie vielen Objekten einer (meist anderen) Klasse in Beziehung stehen können.

*Notation:*

*Bedeutung:*



Jedes Objekt von A steht in Beziehung mit

genau einem Objekt von B

keinem oder einem Objekt von B

einem oder mehreren Objekten von B

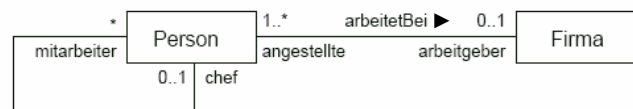
keinem oder einem oder mehreren Objekten von B

2 bis 4 oder 7 Objekten von B

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Assoziationsrollen

- Beschreiben, welche Funktionen die Objekte einer Klasse in einer Assoziation einnehmen.



Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

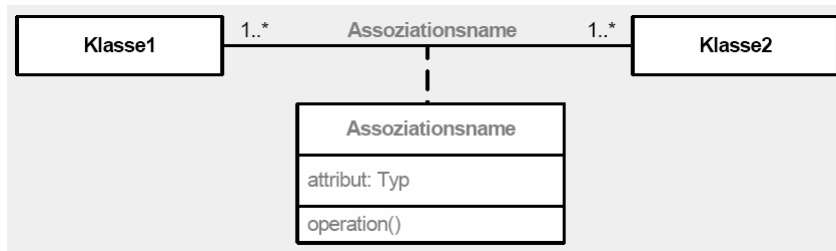


## Assoziationsklassen

- Eine Assoziationsklasse ist eine Assoziation die über eigene Attribute und ggf. Operationen verfügt.

- Darstellung

- Über gestrichelte Linie mit Assoziation verbunden.
- Klassenname entspricht Assoziationsnamen.



- Beispiel: Zur Assoziation Arbeitsverhältnis soll das von und bis Datum gespeichert werden.

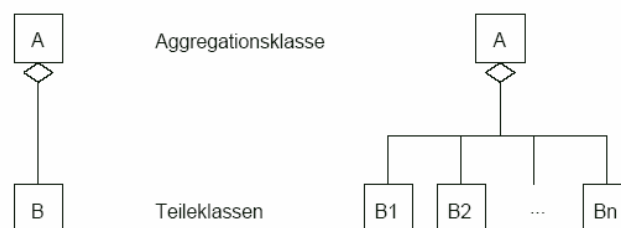
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Aggregation

- Spezielle Form der Assoziation, die eine "Gesamtheit-Teil"-Beziehung ausdrückt.

- Beispiele

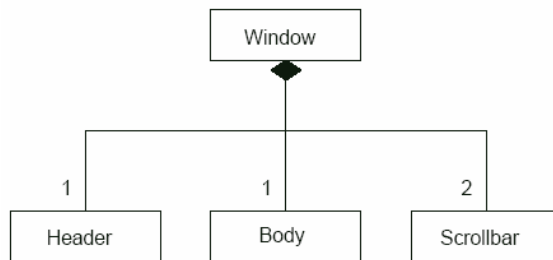
- ICE-Lok besitzt 6 Motoren,
- Stundenplan umfasst mehrere Vorlesungen
- Student ist Teil einer Arbeitsgruppe



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Komposition

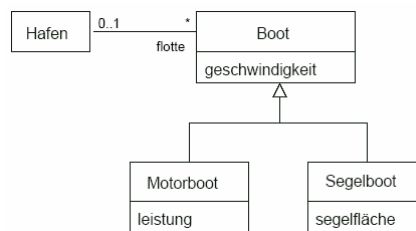
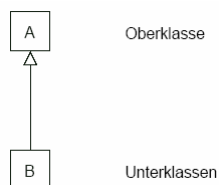
- **Spezielle Form der Assoziation, bei der das Teil existenzabhängig vom Ganzen ist.**
  - Einzelne Teile können ohne das Ganze nicht existieren.
  - Wenn man das Ganze löscht werden auch die Teile gelöscht.
- **Beispiel:**



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Vererbung

- **Relation zwischen einer "allgemeineren" Klasse (Ober- bzw. Superklasse) und einer "spezielleren" Klasse (Unter- bzw. Subklasse).**
- **Jedes Objekt der Unterklasse ist auch ein Objekt der Oberklasse.**
- **Unterklassen erben alle Attribute und Operationen von der Oberklasse.**
- **A ist eine Generalisierung von B. B ist eine Spezialisierung von A.**



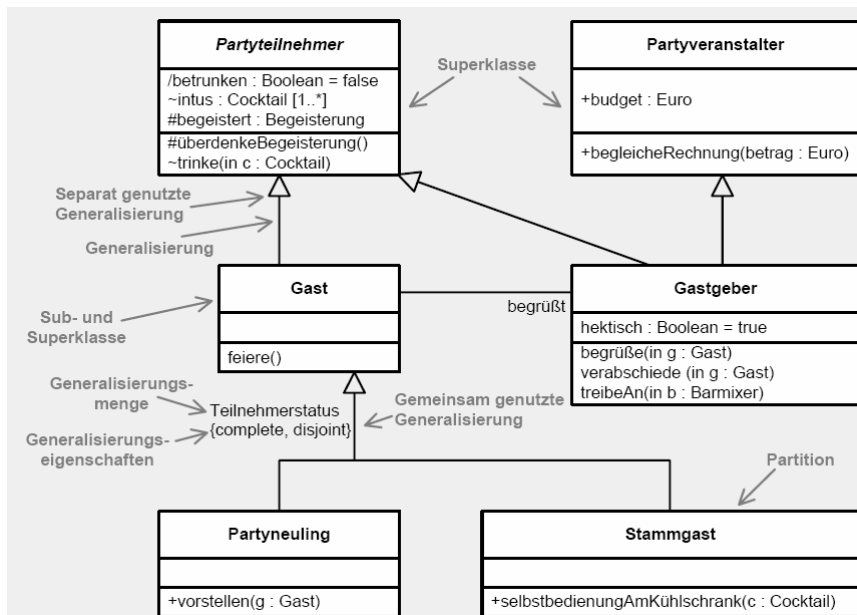
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Vorteile des Vererbungsprinzips

- Konzeptionelle Vereinfachung durch Zusammenfassen gemeinsamer Merkmale verwandter Klassen in einer Oberklasse (Generalisierung)
- Wiederverwendung bereits vorhandener Klassen durch Subklassenbildung (Spezialisierung)
- Vermeidung von Coderedundanzen.
- Erleichterung der Wiederverwendung des Codes.

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

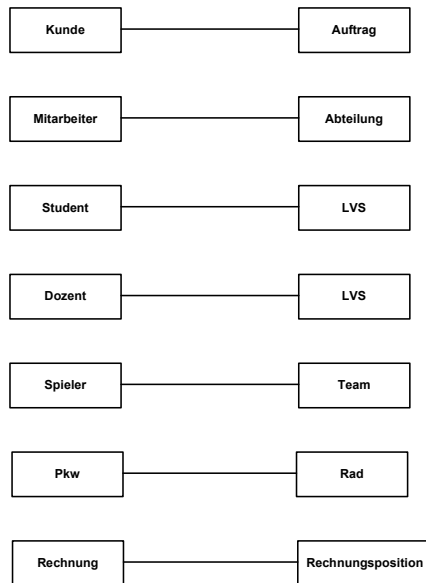
## Beispiel Vererbung



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Übung 3: Assoziationen

- Beschreiben Sie die Assoziationen möglichst genau.
- Annahmen:
  - Kunde wird man erst mit dem ersten Auftrag.
  - Eine Abteilung wird von einem Mitarbeiter geleitet.
  - Es gibt Gemeinschaftskredite.
  - Dozenten geben genau 2 LVS.
  - Spieler spielen in min. 0 und max. 2 Teams
  - Teams haben 2 bis 7 Spieler



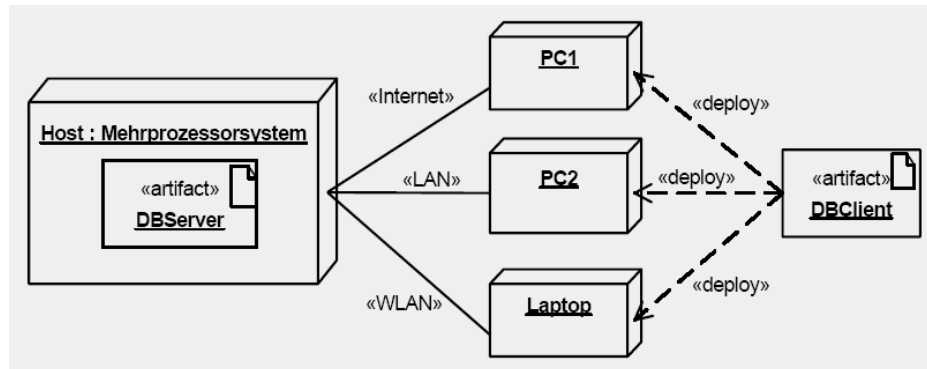
Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## UML Verteilungsdiagramm

- Verteilungsdiagramme dienen dem
  - Aufzeigen der Zuordnung von Softwarekomponenten auf Hardwarekomponenten.
  - Darstellung von Kommunikation und Abhängigkeit zwischen den Komponenten.
- Antwort auf Frage: Wie werden Systemkomponenten zur Laufzeit wohin verteilt?
- Anwendung im Projekt
  - Visualisierung von Systemtopologie
  - Client-Server-Systeme
- Notationselemente
  - Knoten
  - Kommunikationspfad
  - Verteilungsbeziehung

Freie Universität Berlin – Suhli, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Beispiel Verteilungsdiagramm



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)



## Knoten

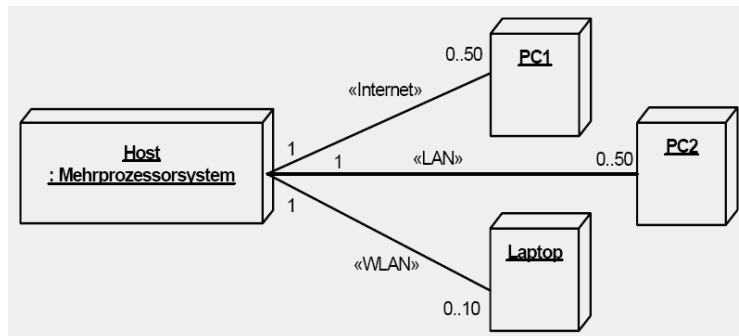
- **Repräsentiert Ressource, z.B. Rechner oder Sensor, die genutzt wird für**
  - **Installation**
  - **Bereitstellung**
  - **Ausführung von Artefakten**



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Kommunikationspfad

- Verbindet Knoten um Nachrichten auszutauschen.
- Darstellung wie Assoziation, da Kommunikationspfad eine Assoziation ist.
- Wird eingeteilt in
  - Ungerichteter Kommunikationspfad 
  - Gerichteter Kommunikationspfad 



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Verteilungsbeziehung

- Beziehung zwischen Artefakt und Knoten, auf dem es verteilt ist.
- Darstellung durch gestrichelten Pfeil und <<deploy>>



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS08 (Version vom 26.03.08)

## Verteilungsbeziehung

- Alternative Darstellung mittels <<deploy>> oder Inklusion in den Knoten.

