

Software Engineering

SS 2005

Prof. Dr. Barbara Paech, Jürgen Rückert



Institut für Informatik
Im Neuenheimer Feld 326
69120 Heidelberg
<http://www-swe.informatik.uni-heidelberg.de>
paech@informatik.uni-heidelberg.de



RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



2. Beschreibungstechniken

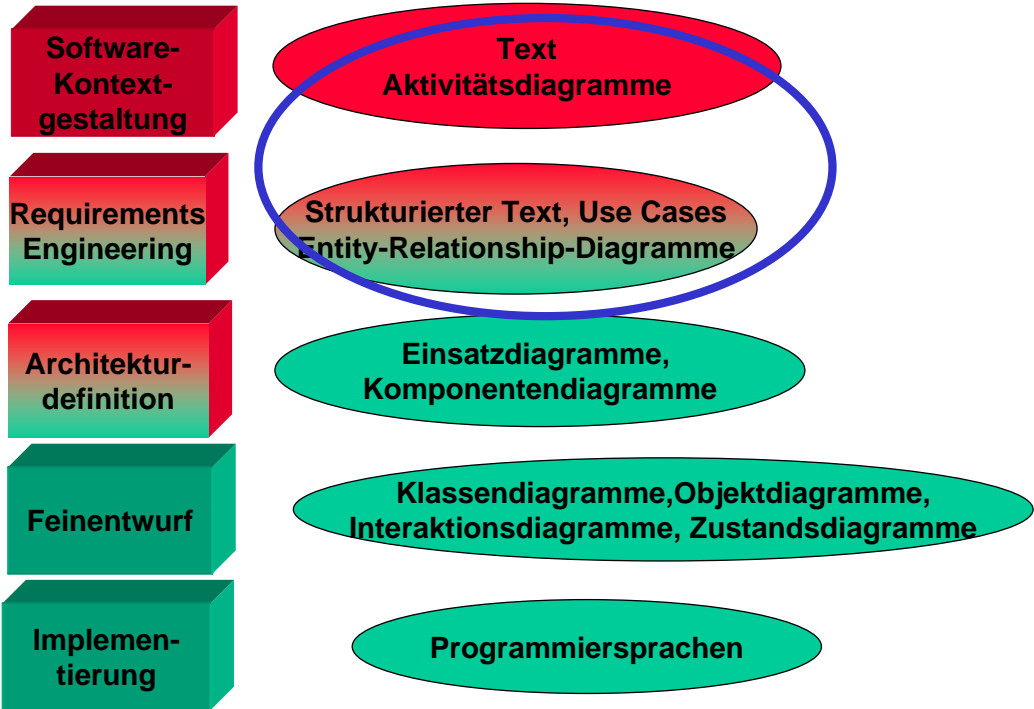
- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



1.3.5. Beschreibungstechniken

2. Beschreibungstechniken

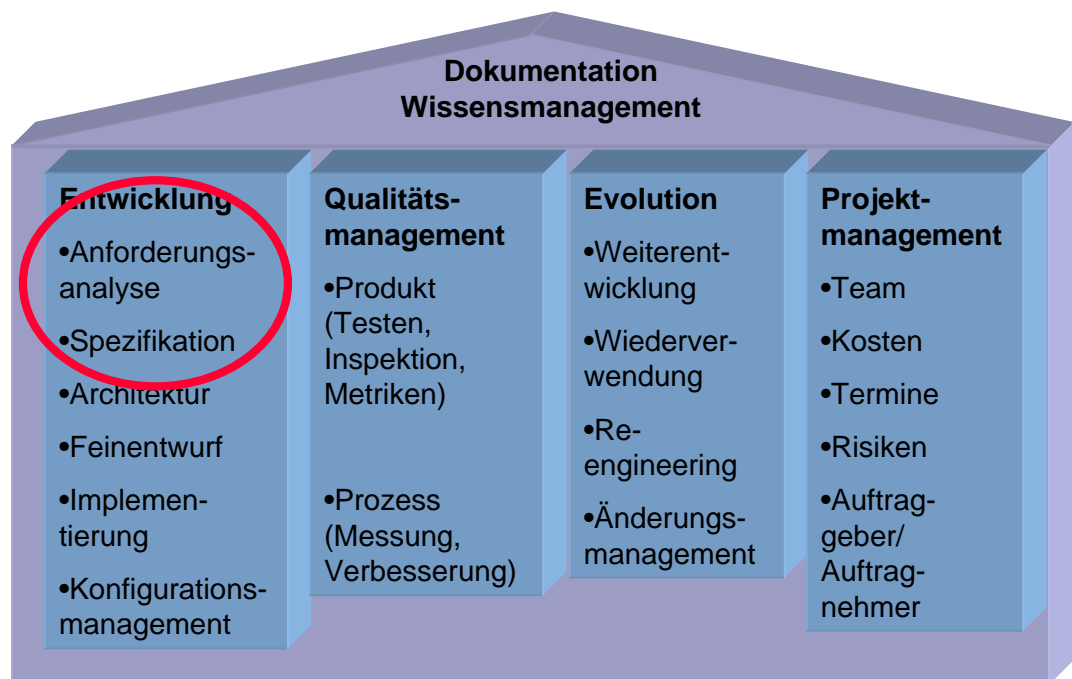
- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



1.3.1. Aufgabenbereiche des Engineering

2. Beschreibungstechniken

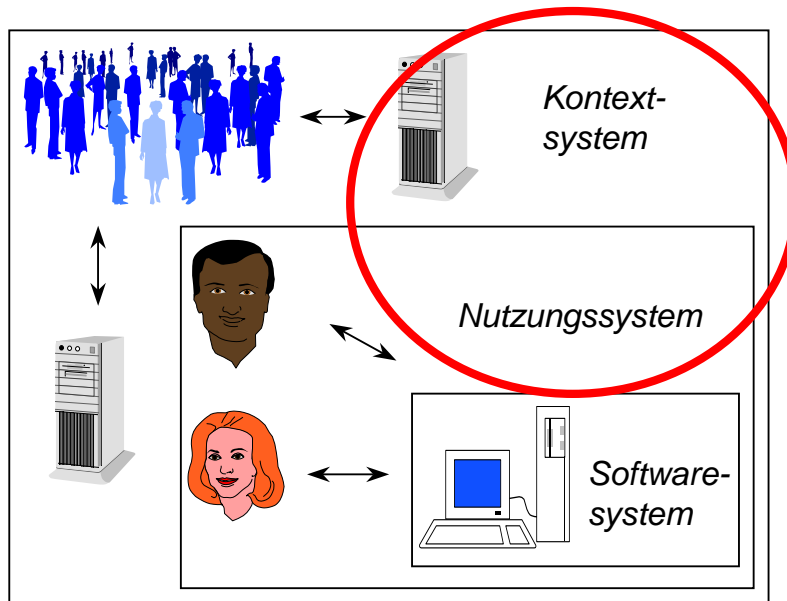
- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



1.3.2. Gestaltungsbereiche der SW-Entwicklung

2. Beschreibungstechniken

- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

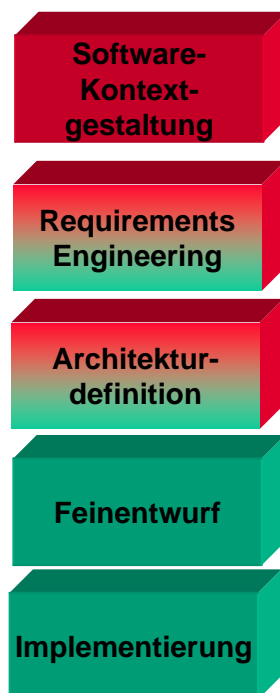


[Paech2000]

1.3.1. Gestaltungsentscheidungen

2. Beschreibungstechniken

- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



1.2.2. SWE ist schwierig!

2. Beschreibungstechniken

- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- ▶ 2.5. Nutzungsbeschreibung

- ◆ Nur 16 % der Projekte erfolgreich, 31 % ohne Ergebnis, 53 % nicht im Plan
- ◆ Zeitüberschreitung bis zu 222%, Kostenüberschreitung bis zu 189 %
- ◆ **45 % der Funktionalität nie benutzt**
- ◆ Was geht schief ?
 - **Unvollständige Anforderungen** 13.1%
 - **Kunden nicht ausreichend einbezogen** 12.4 %
 - Mittel nicht ausreichend 10.6 %
 - **Unrealistische Erwartungen** 9.9 %
 - mangelnde Unterstützung durch Management 9.3 %
 - **Änderungen in den Anforderungen** 8.7 %
 - mangelnde Planung 8.1 %

Quelle: Standish Group 1995

2.5. Nutzungsbeschreibung

2. Beschreibungstechniken

- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- ▶ 2.5. Nutzungsbeschreibung

- ◆ 2.5.1. Übersicht
- ◆ 2.5.2. Aufgaben und Domänenebene: Rollen, Aufgaben, Daten
- ◆ 2.5.3. Interaktionsebene: Use Cases
- ◆ 2.5.4. Interaktionsebene: Systemfunktionen und UI-Struktur
- ◆ 2.5.5. Systemebene: GUI

2.5.1. Nutzungsbeschreibung

- ◆ Welche Entscheidungen müssen bei der Festlegung der Systemnutzung getroffen werden?

2. Beschreibungs-
techniken

2.1. Modellierung

2.2. Objekt-
orientierung

2.3. Klassen-
diagramme

2.4. Zustands-
diagramme

2.5. Nutzungs-
beschreibung

2.5.1. Aufgabenorientierte Anforderungsspezifikation

2. Beschreibungs-
techniken

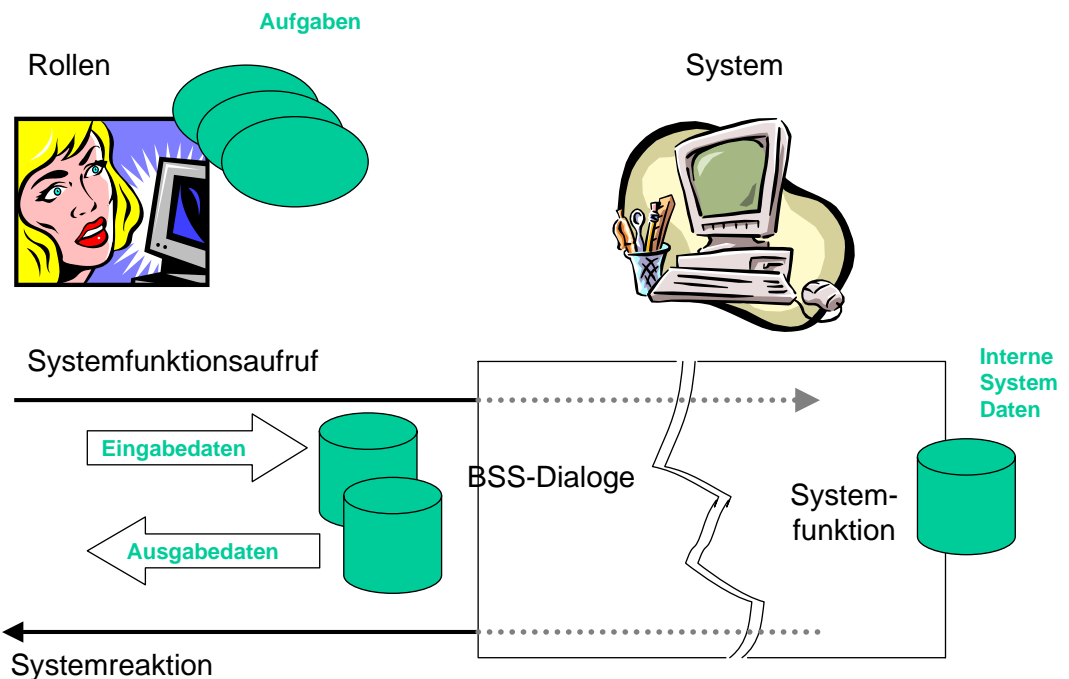
2.1. Modellierung

2.2. Objekt-
orientierung

2.3. Klassen-
diagramme

2.4. Zustands-
diagramme

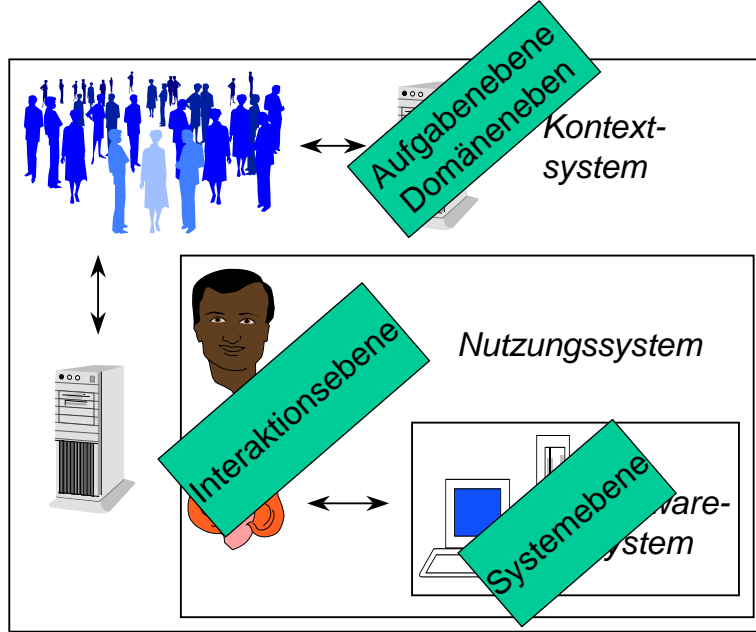
2.5. Nutzungs-
beschreibung



1.3.2. Gestaltungsbereiche der SW-Entwicklung

2. Beschreibungstechniken

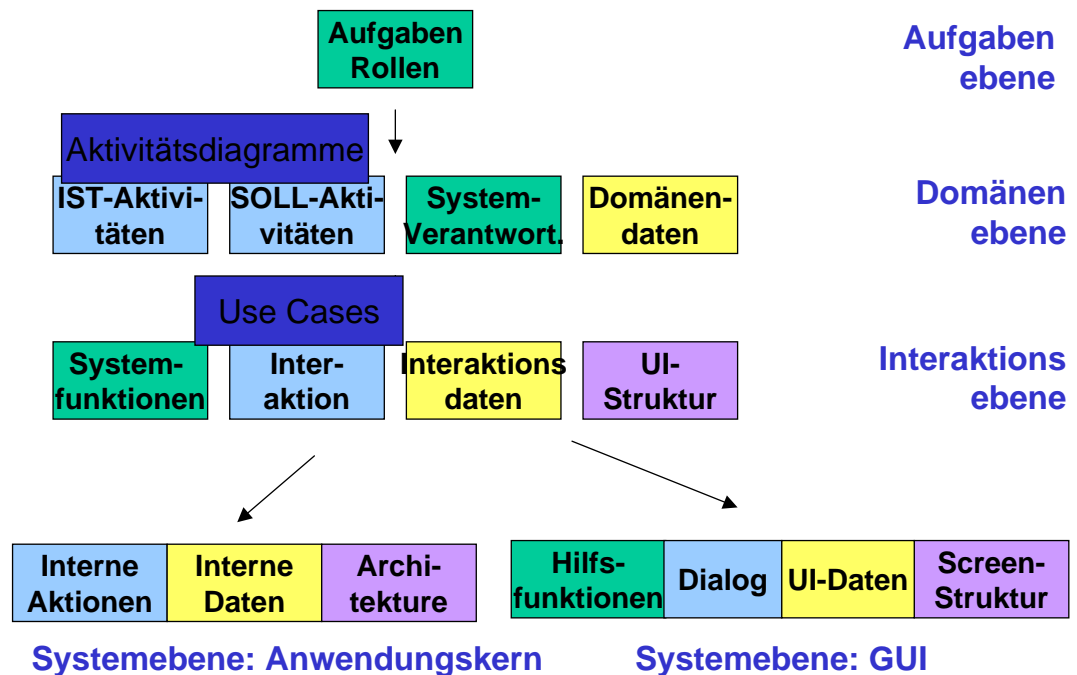
- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



2.5.1. Beschreibungsebenen

2. Beschreibungstechniken

- 2.1. Modellierung
- 2.2. Objektorientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ Zuerst Fokus auf **Geschäftsprozess- und Arbeitsplatzgestaltung**
 - Welche **Rollen** soll das System unterstützen?
 - Welche **Aufgaben** haben diese Rollen?
 - **IST**: Aus welchen **Aktivitäten** bestehen diese Rollen?
 - **Domänendaten**: Welche Informationen sind bei diesen Aktivitäten wichtig?

Abstrahiert vom Softwaresystem!

- ◆ Dann Fokus auf **IT-Innovation**
 - **SOLL**: Wie **ändern** sich die Aufgaben und Aktivitäten durch das System?
 - **SOLL**: Welche Aktivitäten soll das System unterstützen (**Systemverantwortlichkeiten**)?

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ Eine **Benutzerrolle** ist eine abstrakte Zusammenfassung von **Bedürfnissen, Interessen, Erwartungen, Verhalten und Verantwortlichkeiten**, die charakteristisch ist für eine Menge von zukünftigen Systembenutzern [nach Constantine/Lockwood99].
- ◆ Ein **Benutzerprofil** beschreibt **das Wissen und die Fähigkeiten** von typischen Benutzern.
- ◆ Kann erhoben werden durch
 - Befragung der Benutzer
 - Befragung von Surrogat-Nutzern (Marketing, Vertrieb, Hot-line, Trainer)
 - Untersuchung von Dokumenten im Geschäftsprozess

2.5.2. Wie kann ich Benutzer beschreiben?

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

◆ Rollenbeschreibung

- Aufgaben
- Erfolgskriterien
- Kommunikationspartner
- Innovationsgrad

◆ Benutzerprofil

Wissen/Erfahrungen/Fähigkeiten

- bzgl. Aufgaben
- bzgl. Softwaresystem

2.5.2. Beispiel: Schalterangestellte in Uni-Bibliothek

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

◆ Rollenbeschreibung

- **Aufgaben:** bzgl. Leser: Beratung, Anmeldung, Abmeldung; bzgl. Bücher: Ausgabe, Rücknahme
- **Erfolgskriterium:** Leserzufriedenheit, Buchbestand aktuell
- **Kommunikationspartner:** Leser, Bibliothekare
- **Innovationsgrad:** gering

◆ Benutzerprofil

- **Vorwissen Bibliotheksaufgaben:**
 - Bücher: muss für Beratung ausreichend sein
 - Bibliotheksabläufe: oft gering, da studentische Hilfskraft
- **Vorwissen Software:**
 - oft gering, da meist GeisteswissenschaftlerIn

2.5.2. Wie kann ich Aufgaben beschreiben?

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

◆ Arbeitsaufgabe

➤ Vorgabe zum Handeln, zur Erreichung eines bestimmten Ziels, unter bestimmten Voraussetzungen, mit bestimmten Vorgehensweisen

- ◆ Eine Aufgabenbeschreibung ist eine abstrakte Zusammenfassung davon, **warum** die Aufgabe ausgeführt wird (Ziele, Ursachen, Priorität), **wie** sie ausgeführt wird und **welche Abhängigkeiten** das zur Umgebung hat (Vorbedingung, Input, Output, Ressourcen, Gestaltungsmöglichkeiten)
- ◆ Kann erhoben werden durch
 - Befragung der Benutzer
 - Befragung von Surrogat-Nutzern (Marketing, Vertrieb, Hot-line, Trainer)
 - Untersuchung von Dokumenten im Geschäftsprozess
 - Beobachtung

Folie 17

2.5.2. Gute Aufgaben: Humane Arbeit

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung



Quelle: Eberhard Ulich: „Arbeitspsychologie“, Schäffer-Poeschel Verlag, Stuttgart, 1994

Folie 18

2.5.2. Wie kann ich Aufgaben beschreiben?

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

◆ Aufgabenbewertung

- Ziele
- Eingriffsmöglichkeiten
- Ursachen
- Priorität

◆ Aufgabendurchführung

- Durchführungsprofil (Häufigkeit, Kontinuität, Komplexität)
- Ausgangssituation (Vorbedingung)
- Info-In
- Info-Out
- Ressourcen (Arbeitsmittel, beteiligte Rollen)

2.5.2. Beispiel: Leseranmeldung

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ **Ziel:** Leser ist zur Nutzung der Bibliothek berechtigt
- ◆ **Eingriffsmöglichkeiten:** Berechtigung des Lesers prüfen
- ◆ **Ursachen:** Leser möchte Bibliothek nutzen
- ◆ **Priorität:** hoch
- ◆ **Durchführungsprofil:** häufig, Unterbrechungen nicht möglich, geringe Komplexität
- ◆ **Ausgangssituation:** Leser noch nicht berechtigt
- ◆ **Info-In:** Leserdaten
- ◆ **Info-Out:** Ausweis
- ◆ **Ressourcen:** Schalterangestellte, Leserkartei

2.5.2. IST- und SOLL-Aktivitäten

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- ▶ 2.5. Nutzungs-beschreibung

- ◆ Werden wie Aufgaben beschrieben
- ◆ Verfeinern die Aufgaben in Teilaufgaben (= Aktivitäten), wenn nötig (d.h. wenn Aufgabe komplex)
- ◆ Folgen von Aktivitäten können durch UML-Aktivitätsdiagramme beschrieben werden

2.5.2. Systemverantwortlichkeiten

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- ▶ 2.5. Nutzungs-beschreibung

- ◆ Diejenigen Aktivitäten, die das Softwaresystem unterstützen soll, werden Systemverantwortlichkeiten genannt

2.5.2. Wie kann ich Domänendaten beschreiben?

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

- ◆ **Domänendaten** beschreiben Entitäten (Dinge und Konzepte), die im Kontext wichtig sind
- ◆ Werden durch **Entity-Relationship-Diagramme** beschrieben
- ◆ Oft auch **Glossar** ausreichend
- ◆ Erklären die in den UC verwendeten Begriffe

2.5.2. ER-Diagramm

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

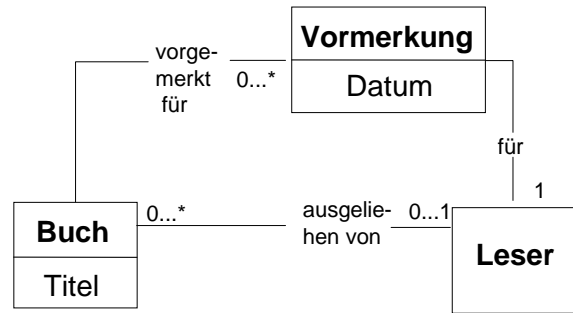
- ◆ Ursprünglich nur für Datenbankentwicklung (Peter Chen, 1976)
- ◆ Hilft die **grundlegenden Entitäten der realen Welt** und **deren Beziehungen** zu verstehen
- ◆ Heutzutage kann man ERD als eine **einfache Form der Klassendiagramme** ansehen, das man verwendet, wenn man **keine Operationen** und **nur einfache Beziehungen** (d.h. keine Aggregation, Vererbung, etc.) modellieren will

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

- Ein Buch hat einen Titel. Es kann höchstens von einem Leser ausgeliehen werden. Jeder Leser kann mehrere Bücher ausleihen.

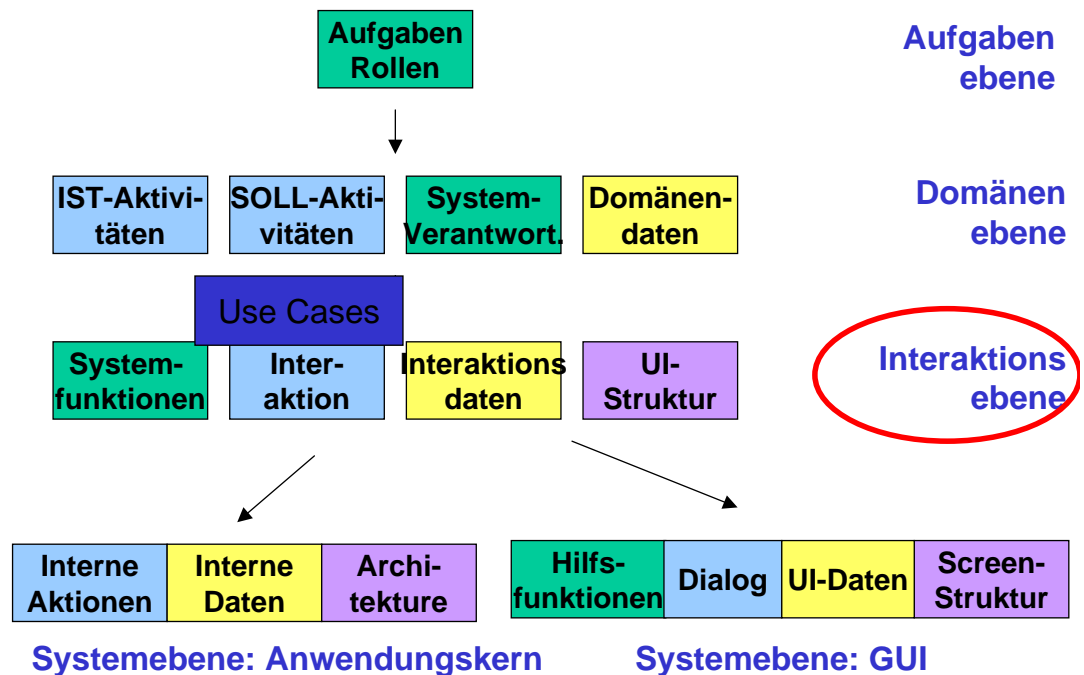
- Eine Vormerkung hält den Reservierungswunsch eines Lesers für ein Buch fest. Dabei wird auch das Datum, an dem die Vormerkung erfolgte festgehalten.



2.5.1. Beschreibungsebenen

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



2.5.3. Fragen der Interaktionsebene

- ◆ Fokus auf der Gestaltung der Grenze/Schnittstelle zwischen Mensch und Maschine
 - Welche Funktionen bietet das System an?
 - Funktionsbeschreibung
 - Wie interagiert / kommuniziert der Benutzer mit dem System?
 - Use Cases (Anwendungsfälle)
 - Welche Daten tauschen Benutzer und System aus?
 - Verfeinertes Datenmodell
 - In welchen Zusammenhängen kann der Benutzer welche Funktionen aufrufen? Wie gliedert sich die Gesamtfunktionalität in Teilbereiche?
 - User Interface Struktur (UI-Struktur)
 - Achtung: Nicht GUI-Struktur, d.h. konkretes Layout noch nicht wichtig

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

2.5.3. Klärung Use Case

- ◆ Ein Use Case fokussiert auf die
 - Interaktion zwischen Benutzer und Software
 - bei der Ausführung einer (Folge von) Systemfunktion(en)
 - zur Erreichung eines bestimmten Ziels
- ◆ Beschreibung von Interaktionsfolgen
 - gut geeignet für Kommunikation mit Benutzer

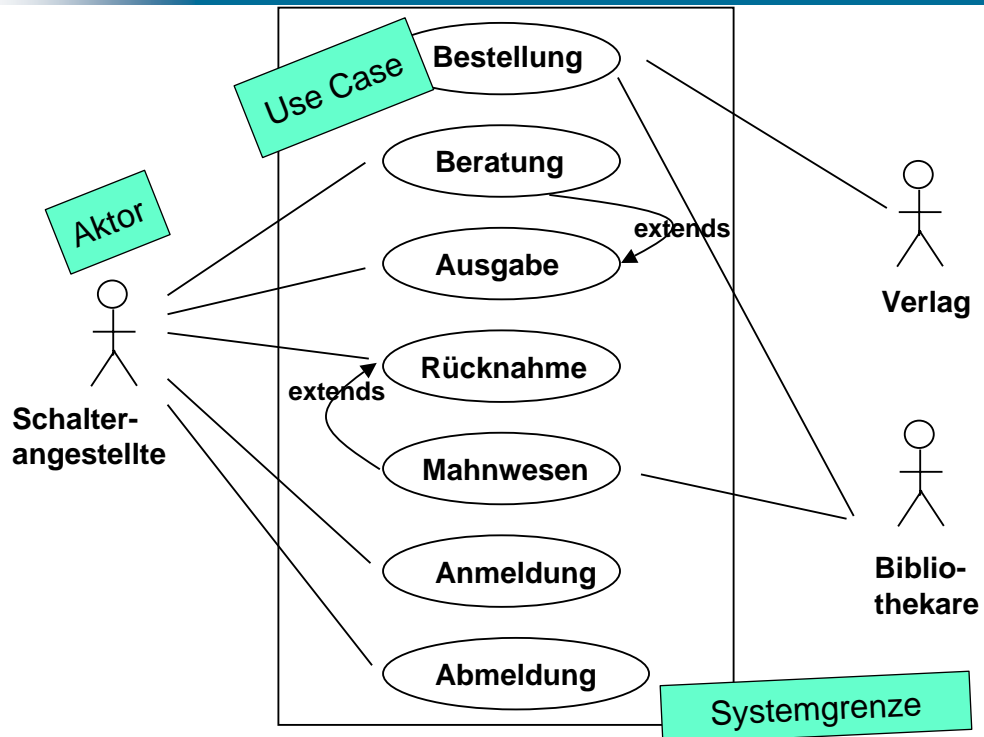
2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

2.5.3. Use Case Diagramm

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



2.5.3. Systemverantwortlichkeitsübersicht

2. Beschreibungs-
techniken

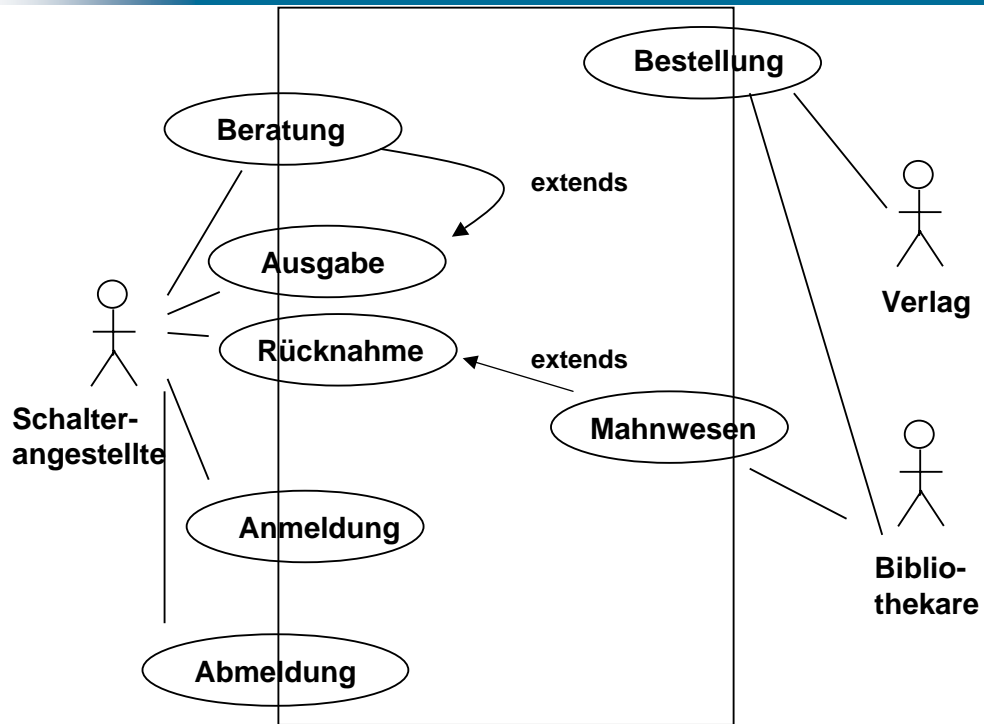
- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

- ◆ Wir verwenden das UC Diagramm, um eine Übersicht über Systemverantwortlichkeiten zu geben
- ◆ Dabei werden die „Bubbles“ **AUF** der Systemgrenze gezeichnet, um deutlich zu machen, dass diese Aufgaben von Aktor und System gemeinsam durchgeführt werden
- ◆ Bubbles **IM** System beschreiben dann Systemfunktionen

2.5.3. Systemverantwortlichkeitsübersicht

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



Folie 31

2.5.3. Klärung: Akteur

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

- ◆ befindet sich **außerhalb** der Software
- ◆ interagiert mit dem System (aktiv oder passiv)
- ◆ ein Akteur repräsentiert einen Menschen, der eine **Rolle** inne hat, oder er repräsentiert ein externes System, z.B.:
 - Richtig: Systemadministrator
 - **Falsch: Heinz Müller (was ist seine Rolle?)**
 - Richtig: Datenbank
- ◆ Akteure können im Use Case unterschiedliche Funktionen haben
 - Initiator
 - Externer Server
 - Empfänger
 - Zwischenstufe für Interaktion zwischen System und anderem Akteur

Folie 32

2.5.3. Beschreibung eines Use Cases

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

Name	Kurzbezeichnung des Use Cases
Aktor	Welcher Aktor löst den Use Case aus?
Ziel	Was soll durch den Use Case erreicht werden?
Beschreibung	Was soll normalerweise passieren?
Ausnahmefälle	Welche Ausnahmen gibt es? Was soll dann passieren?
Regeln	Komplexe funktionale oder kausale Zusammenhänge
Qualitätsanforderungen	Welche übergreifenden Eigenschaften sind wichtig?
Daten, Funktionen	Welche Daten und Funktionen werden verwendet ?
Vorbedingungen	Zustand von System und Umgebung aus Sicht des Aktors <i>bevor</i> der Use Case eintritt
Nachbedingung	Zustand des Systems aus Sicht des Aktors <i>nachdem</i> der Use Case erfolgreich beendet ist

Folie 33

2.5.3. Kurzes Beispiel: Use-Case Leser anmelden

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ **Aktor:** BibliotheksangestellteR
- ◆ **Ziel:** Bibliotheksbenutzerdaten sind im System aufgenommen, LeserIn hat Ausweis.
- ◆ **Vorbedingungen:** Keine.
- ◆ **Beschreibung des Ablaufs:** Aktor prüft Berechtigung und ruft dann Leseranmeldung auf. Das System erzeugt eine neue Lesernummer und ermöglicht Aufnahme der Leserdaten. Der Aktor gibt Name, Adresse und Semesterdaten der LeserIn ein. Das System druckt einen Ausweis mit der Nummer und den Daten. Der Aktor gibt den Ausweis an die LeserIn aus.
- ◆ **Regeln:** Name der LeserIn muss eindeutig sein.
- ◆ **Ausnahmefälle:** LeserIn nicht berechtigt .
- ◆ **Qualitätsanforderungen:** strenge Überprüfung der Eingabedaten.
- ◆ **Daten:** Leserdaten. **Funktion:** Leseranmeldung.
- ◆ **Nachbedingungen:** Zu jeder LeserIn ein Datensatz; Nummer und Name eindeutig.

Folie 34

2.5.3. Langes Beispiel: Use Case Sitzplatzzuweisung

- ◆ **Aktor:** PassagierIn, evtl. SchalterangestellteR.
- ◆ **Ziel:** PassagierIn erhält Sitzplatz für gebuchten Flug.
- ◆ **Vorbedingung:** PassagierIn hat eine gültige Flugbuchung der Fluglinie.
- ◆ **Beschreibung:**

A1: EinE PassagierIn verlangt einen Sitzplatz für den Flug XY123. Dies kann Teil des Eincheckens oder eine Vorreservierung sein.

S1: Das System (evtl. mithilfe der/des Schalterangestellten) fragt nach dem Flugdatum, der Flugnummer, dem Flughafen und dem Namen.

A2: Der/Die PassagierIn gibt diese Informationen an. Statt des Namens kann er/sie auch die Vielfliegernummer angeben.

[Ausnahmefall: Es ist zu früh, um Sitzplatzzuweisungen durchzuführen.]

S2.1: Das System findet eine Flugbuchung. **[Ausnahmefall: Keine Buchung gefunden.]** Falls der Flugbuchung bereits ein Sitzplatz zugewiesen wurde, wird diese dem/der PassagierIn ausgehändigt und die Möglichkeit gegeben, den Platz zu ändern.

2.5.3. Use Case Sitzplatzzuweisung (cont.)

S2.2: Falls keine Sitzplatzzuweisung existiert oder der/die Passagier/ in diese ändern möchte, erfragt das System bevorzugte Plätze: (1) Fenster oder Gang, (2) Nichtraucher oder Raucher.

A3: Akteur gibt die Daten an.

S3: Das System nutzt diese Informationen und die Vielfliegernummer (falls angegeben), um einen passenden Sitzplatz zuzuweisen, abhängig von einer früheren Sitzplatzzuweisung und der Geschäftspolitik der Fluggesellschaft. **Falls nötig**, fragt das System zusätzlich nach, ob der/die PassagierIn auch folgendes akzeptieren würde: (1) einen Sitzplatz mit Blick auf die Tragfläche, (2) einen Sitz am Notausgang?

A4: Akteur gibt die Daten an.

S5: Das System schlägt eine Sitzplatzzuweisung vor. Falls nicht alle Wünsche des/der Passagiers/in berücksichtigt werden können, versucht das System so viele wie möglich zu erfüllen.

[Ausnahmefall: Keine Sitzplatzzuweisung möglich.]

A6: Der/Die PassagierIn akzeptiert den Platz oder bittet um Änderungen.....

2.5.3. Use Case Sitzplatzzuweisung (cont.)

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

◆ Ausnahmefälle:

A2.Zu früh: Falls das aktuelle Datum zu früh ist, teilt das System dem/der PassagierIn mit, ab wann Sitzplatzzuweisungen möglich sind.

S2.1.Keine Buchung gefunden: Falls keine Flugbuchung gefunden wurde, prüft das System die vorhandenen Daten gegen die Buchung nach teilweisen Übereinstimmungen. Falls auch jetzt keine Buchung gefunden werden kann, empfiehlt das System dem/der PassagierIn, eine Buchung durchführen zu lassen.

S5.Keine Sitzplatzzuweisung möglich: Falls die Zuweisung eines Sitzplatzes unmöglich ist unter Berücksichtigung von freien Sitzplätzen, Anzahl der Tage bis zu Abflug, Flugklasse und Vielfliegerstatus, empfiehlt das System dem/der PassagierIn, eine Sitzplatzzuweisung zum Zeitpunkt des Check-Ins durchführen zu lassen. Falls dies bereits der Check-In ist, setzt das System den/die PassagierIn auf die Standby-Liste in der Reihenfolge des Erscheinens.

Folie 37

2.5.3. Use Case Sitzplatzzuweisung (cont.)

2. Beschreibungs-
techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

Regeln:

<Definition für frühesten Zeitpunkt der Reservierung, Definition für Sitzplatzzuweisung>.

Qualitätsanforderungen:

starke Benutzerführung, da Laien als Benutzer.

Daten:

Passagierdaten, Fludaten, Flugbuchung, Sitzplatzzuweisung, Standby-Liste.

Funktionen:

Suche einer Flugbuchung, Durchführen einer Sitzplatzzuweisung.

Nachbedingungen:

PassagierIn hat eine Sitzplatzzuweisung. Diese ist im System gespeichert.

Folie 38

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ UC werden in der Literatur in verschiedenen Detaillierungsgraden beschrieben
 - Kundenanforderungen
 - Softwarespezifikation
 - Benutzungsschnittstelle

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ *Der Actor erhält vom Bibliotheksbenutzer ein Buch und ruft daraufhin die Rückgabefunktion auf. Er gibt Buchdaten ein und das System aktualisiert diese entsprechend. Falls das Buch vorgemerkt war, benachrichtigt das System den entsprechenden Bibliotheksbenutzer, Der Actor teilt die erfolgreiche Rückgabe dem Bibliotheksbenutzer mit.*
- ◆ Abstrahiert weitgehend von Ausnahmen
- ◆ Entscheidungsebene Arbeitsgestaltung: Alternative wäre z.B. direkte Rückgabe des Bibliotheksbenutzers mithilfe eines Lesegerätes.
- ◆ Sehr abstrakt: => Kann besser auf Ebene der Systemverantwortlichkeiten beschrieben werden. (Aufgabenebene)

2.5.3. Beispiel : Systemanforderungsebene

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ *Der Aktor ruft die Rückgabefunktion auf. Er gibt Buchdaten ein. **Das System ermittelt aus dem Ausleiheintrag die Benutzerdaten und löscht den Ausleiheintrag. [Ausnahme: kein Ausleiheintrag beim Buch]. Es löscht auch den Ausleiheintrag beim Leser [Ausnahme: kein Ausleiheintrag beim Leser]. Das System ermittelt auch eine evtl. Vormerkung des Buches und falls das Buch vorgemerkt war, benachrichtigt das System den entsprechenden Bibliotheksbenutzer. Das System teilt die erfolgreiche Bearbeitung dem Aktor mit.***
- ◆ **Ausnahmen** aufgrund der **internen Verarbeitung**
- ◆ **Entscheidung Performanz:** z.B. Leseraktualisierung und Vormerkungsbenachrichtigung können parallel geschehen.
- ◆ **Passt am besten zu UI-Struktur-Granularität (Interaktionsebene)**
- ◆ Beachte: Detail der **Internen Aktionen in Systemfunktionsbeschreibung** definieren!

Folie 41

2.5.3. Beispiel: Benutzungsschnittstellen-Ebene

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ *Das System zeigt eine **Eingabemaske für Buchdaten. Der Aktor gibt Buchtitel oder Buchnummer ein. [Ausnahme: Ungültige Buchnummer, keine eindeutige Identifikation]. Das System zeigt daraufhin den vollständigen Buchdatensatz an. Der Aktor wählt die Rückgabefunktion aus. Das System aktualisiert Buch- und Benutzerdaten und zeigt beide Sichten sowie eine Bestätigungsabfrage an [Ausnahme: Buch hat keinen Ausleiheintrag]***
- ◆ **Ausnahmen** aufgrund Benutzereingaben oder Systemausgaben
- ◆ **Entscheidung Paradigma:** Alternative, z.B. zuerst Funktionsauswahl dann Objektauswahl
- ◆ **Entscheidung Ergonomie:** Bestätigungsabfrage
- ◆ **Sehr detailliert: =>Kann besser bei Dialogen beschrieben werden (GUI-Ebene)**

Folie 42

2.5.3. Interaktionsbeschreibung durch UC

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-
beschreibung

Use Case Template	Kundenanforderungen	Softwarespezifikation	Benutzungsschnittstellenbeschreibung
Aktor	Rolle	Rolle	Rolle
Ziel	Aufgabe der Rolle	Interne Aufgliederung einer komplexen Systemfunktion	Einzelheiten der Interaktion zwischen System und Aktor
Beschreibung	Informationsfluss zwischen Aktor, System u. anderen Aktoren	Eingaben zu Systemfunktionen, deren Verarbeitung durch Änderungen auf Systemdaten u. Ausgaben	Benutzungsschnittstellenkommandos, Systemausgaben auf dem Bildschirm
Ausnahmefälle	Nur sehr wichtige, die Arbeitsgestaltung beeinflussen	Bzgl. der internen Verarbeitung	Bzgl. Benutzereingaben, Systemausgaben
Regeln	Zusammenarbeit mit anderen Aktoren	Geschäftsregeln, die bei der Abarbeitung der Systemfunktionen zu beachten sind	Formatvorgaben, Gültigkeitsbereich für Ein/Ausgaben
Qualitätsanforderungen	Für die Arbeit (z.B. Humane Arbeit)	Für die interne Durchführung der Systemfunktion (z.B. Volumen, Performanz)	Für die Interaktion (z.B. Selbstbeschreibungsfähigkeit)
Daten	Informationen, Produkte	Systemdaten	Sichten
Funktionen	Systemfunktionen	Interne Teilschritte	Interaktionsteilschritte
Vorbedingung	Bzgl. Aufgabe	Bzgl. Systemzustand	Bzgl. Systemzustand, Ein/Ausgabegeräte
Nachbedingung	Ergebnis der Aufgabe	Systemzustand	Bzgl. Systemzustand, Ein/Ausgabegeräte

Folie 43

2.5.3. Graphische Darstellung von Use Cases

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-
beschreibung

- ◆ **Datenflussdiagramme (UML: Activity Diagram)**
 - Datenflussdiagramme beschreiben exemplarisch Datenfluss zwischen System und externen Aktoren
 - gut geeignet zur Beschreibung der Arbeitsteilung
- ◆ **Kontrollflussdiagramme (UML: Interaction Diagram)**
 - beschreiben Kontrollfluss zwischen Aktivitäten
 - gut geeignet zur Beschreibung der Datenänderungen
- ◆ **Problem: Mit Modellen lassen sich meist nur typische Abläufe beschreiben, nicht alle möglichen Abläufe**

Folie 44

2.5.3. Optimierung mit “Extends” und “Includes”

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ Die **“Includes”-Beziehung** wird verwendet, um **sich wiederholende Teile** von Interaktionsabläufen in gesonderten Use Cases zu kapseln. Dadurch werden Wiederholungen in der Beschreibung vermieden. Der gekapselte Teil wird immer an definierten Stellen im Use Case ausgeführt.
- ◆ Die **“Extends”-Beziehung** wird verwendet, um die Behandlung von umfangreichen Sonderfällen (Ausnahmen, optionales Verhalten) in einem gesonderten Use Case zu beschreiben. Die Ausführung der beiden Use Cases ist **miteinander verwoben**.
- ◆ Beide Beziehungen dienen zur Optimierung der Lesbarkeit und Effizienz der Beschreibung von Use Cases, **nachdem** die Use Cases detailliert ausgearbeitet wurden.

2.5.3. Klärung: Use Case und Szenario

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ Ein **Szenario**
 - ist ein spezifischer Interaktionsablauf.
- ◆ Ein Use Case
 - ist eine abstrakte Beschreibung einer **endlichen Menge** von Szenarien.
- ◆ Eine Auswahl von typischen Szenarien kann **Vorlage** für die Erstellung eines Use Case sein.
- ◆ Ein Use Case kann durch die **nachträgliche** Erstellung von Szenarien validiert werden.

2.5.3. Probleme mit Use Cases

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

- ◆ Wissenserwerb vs. Spezifikation
 - **Abdeckung** des Gesamtverhaltens durch Use Cases
- ◆ **Granularität und Inhalt**
 - unzulässige Vereinfachung
 - keine klare Trennung von
 - Aktor-System
 - Typ-Instanz
 - Normalfall-Ausnahme
 - Kundenanforderungen, Systemfunktionsbeschreibung, Benutzungsschnittstellengestaltung
 - IST-SOLL
- ◆ **Pflege** (Verwaltung, Änderbarkeit)

2.5.3. Verwendung von Use Cases

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

- ◆ **Beschreibung der funktionalen Anforderungen**
 - verständlich für die Benutzer
 - aus Sicht der Benutzer
 - guter Übergang zur Benutzungsschnittstellengestaltung
 - auch als Einheit zur Projektplanung
- ◆ **Als Mittel zur Anforderungsermittlung**
 - Modellcharakter
 - Detail, Vollständigkeit, Management (Änderbarkeit) **nicht** so wichtig
- ◆ **Als Mittel zur Anforderungsspezifikation**
 - Vorgabe für Entwurf
 - Detail, Vollständigkeit, Management (Änderbarkeit) **sehr** wichtig

2.5.3. Strukturierung der Interaktion

- ◆ Wenn es viele UCs gibt, müssen die Daten und Funktionen in verschiedene zusammenhängende Bereiche gebündelt werden => **UI-Struktur**
- ◆ UI-Struktur gruppiert zusammengehörige Funktionen und Daten in **Arbeitsbereiche (auch Workspaces, Virtual Windows)**
- ◆ Abstrahiert vom konkreten Layout und Bildschirmteilung, stellt logische Sicht der Benutzer auf die Interaktionsstruktur dar
- ◆ Entsteht **parallel** zu den UC

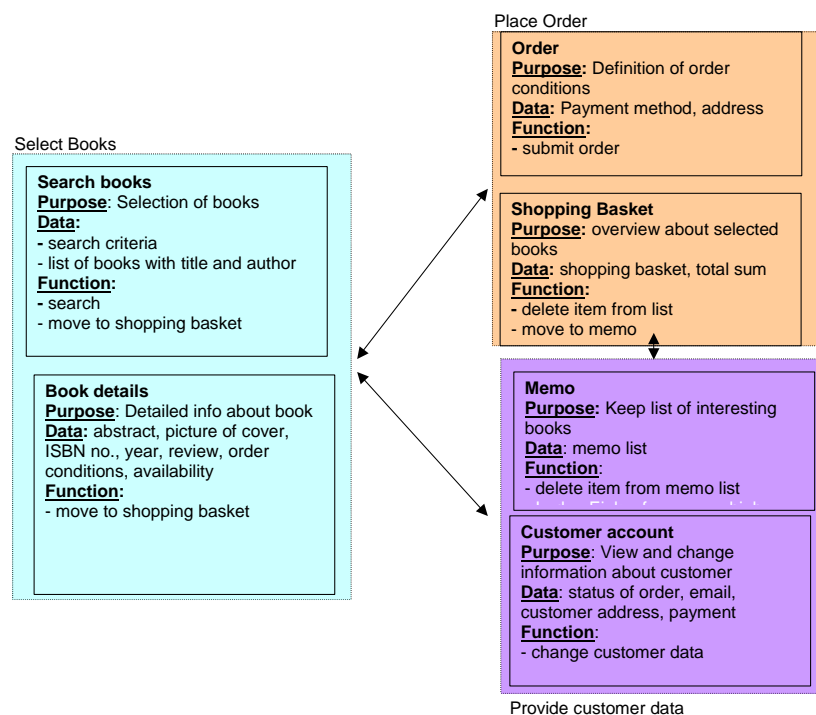
2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung

2.5.4. Beispiel UI-Struktur für Web-Bookstore

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassendiagramme
- 2.4. Zustandsdiagramme
- 2.5. Nutzungsbeschreibung



2.5.4. Systemfunktionen und Interaktionsdaten

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

- ◆ Sind die in den UC genannten Systemfunktionen komplex, d.h. viele Daten oder mehrere Teilschritte oder komplexe Berechnungsregel, dann sollte eine **eigenständige Systemfunktionsbeschreibung** erstellt werden.
- ◆ **Interaktionsdaten** verfeinern die Domänendaten um die in UC- und Systemfunktionsbeschreibung gewonnenen Informationen. Werden durch ein verfeinertes ER-Diagramm dargestellt.

Folie 51

2.5.4. Wie beschreibe ich eine Systemfunktion?

2. Beschreibungs- techniken

- 2.1. Modellierung
- 2.2. Objekt-orientierung
- 2.3. Klassen-diagramme
- 2.4. Zustands-diagramme
- 2.5. Nutzungs-beschreibung

Name	Kurzbezeichnung der Funktion
Eingangsdaten	Welche Daten verarbeitet die Funktion?
Ausgangsdaten	Welche Daten erzeugt oder verändert die Funktion?
Beschreibung	Wie soll das Ergebnis normalerweise berechnet werden?
Ausnahmefälle	Welche Ausnahmen gibt es? Was soll dann passieren?
Regeln	Komplexe funktionale oder kausale Zusammenhänge bei der Berechnung
Qualitätsanforderungen	Welche übergreifenden Eigenschaften sind wichtig?
Vorbedingungen	Zustand von System und Umgebung aus Sicht des Aktors <i>bevor</i> die Funktion ausgeführt wird
Nachbedingung	Zustand des Systems aus Sicht des Aktors <i>nachdem</i> die Funktion erfolgreich beendet

Folie 52

2.5. Zusammenfassung Aufgaben und Use Cases

- ◆ Rollen- und Aufgabenbeschreibung sowie Datenmodellierung hilft den **Nutzungskontext** (das Kontextsystem) zu **verstehen**.
- ◆ Früher Fokus auf Technik-Gestaltung sollte vermieden werden.
- ◆ Verwendung von Use Cases zur Anforderungsermittlung oder -spezifikation
- ◆ Beschreibung anhand eines **Formblatts**
- ◆ Bei der Erstellung stehen **Aktoren und deren Aufgaben** im Vordergrund
- ◆ Überblick durch Use Case Diagramm

2.5. Zusammenfassung Interaktion

- ◆ Die Entscheidungen auf der Interaktionsebene sind oft schwierig, da hier Mensch und Maschine „aufeinander treffen“.
- ◆ Hier wird sichergestellt, dass **die richtige Funktionalität benutzungsfreundlich** angeboten wird.
- ◆ UI-Struktur ermöglicht **frühe (und billige!)** Diskussion mit Benutzern.
- ◆ Bei Use Case Erstellung ohne UI-Struktur ist es schwierig, die geeignete Abstraktionsebene zu finden.

2. Beschreibungs- techniken

2.1. Modellierung

2.2. Objekt- orientierung

2.3. Klassen- diagramme

2.4. Zustands- diagramme

2.5. Nutzungs- beschreibung

- ◆ B. Paech, Aufgabenorientierte Softwareentwicklung, Integrierte Gestaltung von Unternehmen, Arbeit und Software, Springer Verlag 2000
- ◆ Paech, B., Kohler, K., Task driven requirements in object-oriented development, in Leite, J., Doorn, J.,(eds.) *Perspectives on Requirements Engineering*, Kluwer Academic Publishers, 2003
- ◆ A. Cockburn, Writing Use Cases, Addison-Wesley 2001
- ◆ S. Lauesen, User Interface Design - A software engineering perspective, Addison-Wesley, 2005
- ◆ L. Constantine and L. Lookwood: Software For Use, ACM Press, 1999