

# 1 Use-Case-Modellierung

## 1.1 Wesentliche Aspekte

- Systemabgrenzung
- Akteure, d.h. Rollen (Situationen)
  - Benutzer
  - Systeme
- funktionale Zerlegung
  - von außen beobachtbare Interaktionen
  - keine Betrachtung der inneren Struktur / Abläufe

Wie schließt man auf Daten(strukturen)?

- anhand der Daten, die bei den Interaktionen benötigt / verwendet werden

Was kann man mit Use-Cases nicht modellieren?

- Abläufe

## 1.2 Beschreibungsform

### 1.2.1 Graphische Notation (UML)

- siehe Verwendung UMLet
- vor allem:
  - Abgrenzung
  - Akteure
  - Use-Cases (Ellipsen mit Text)
  - Beziehung Akteur - Use Case
- mit Bedacht / selten(er):
  - <<includes>> - Beziehung
    - Richtung: zum einzubeziehenden UC
    - der einzubeziehende UC wird **immer** mit verwendet
  - <<extends>> - Beziehung
    - Richtung: zum UC, der erweitert wird
    - der einzubeziehende UC wird **fallweise** mit verwendet, der Fall muss beschrieben werden (als Bedingung)
- Aufteilung in mehrere Diagramme möglich

### 1.2.2 Textbeschreibung

(siehe Vorlage in doc1)

Bezeichnung	konsistent zum UML-Diagramm
Ziel	<i>beschreiben Sie das Ziel des Anwendungsfalls</i>
Akteure	<i>geben Sie die Akteure an</i>
Auslösendes Ereignis	<i>warum wird der Anwendungsfall durchgeführt</i>

Bezeichnung	<i><b>konsistent zum UML-Diagramm</b></i>
Vorbedingung	<i>Systemzustand, der vor der Ausführung des Anwendungsfalls vorliegen muss</i>
Nachbedingung	<i>neuer Systemzustand, der nach der Ausführung des Anwendungsfalls vorliegt (keine Angabe, wenn es keine neuen Systemzustand gibt)</i>
Kategorie	<i>primär, sekundär oder optional</i>
Beschreibung	<i>beschreibender Text</i>

## 1.3 Prüfungen

- Konsistenz der Diagramme untereinander (wenn Sie mehrere UC-Diagramme verwenden)
- Konsistenz der Diagramme und der Textbeschreibung
  - insbesondere für jeden Anwendungsfall eine Textbeschreibung, für jede Textbeschreibung ein Use-Case
- Konsistenz der Textbeschreibungen
  - insbesondere hinsichtlich der Beschreibung der Vor- und Nachbedingungen
- Konsistenz zur FP-Analyse

## 2 Datenmodellierung mit Klassendiagrammen

### 2.1 Wesentliche Aspekte

- Auffinden der einzigartigen Klassen (existenzunabhängig)
- Auffinden der Beziehungen zwischen diesen Klassen

### 2.2 Beschreibungsform

#### 2.2.1 Graphische Notation (UML)

- siehe Verwendung UMLet
- vor allem:
  - Vererbung (Generalisierung - Spezialisierung)
  - Assoziationen
    - Aggregationen (existenzunabhängig)
    - Kompositionen (existenzabhängig)
    - Attribute
    - Rollen
    - Benennung in beide Richtungen
- Aufteilung in mehrere Diagramme möglich
- geben Sie in den Diagrammen die einzelnen Attribute und Methoden i.d.R. **nicht** an.

#### 2.2.2 Textbeschreibung

(siehe Vorlage in doc1)

## 2.3 Prüfungen

- Konsistenz der Diagramme untereinander (wenn Sie mehrere Klassendiagramme verwenden)
- Konsistenz der Diagramme und der Textbeschreibung
- Konsistenz der Textbeschreibungen

und

- Konsistenz zur UC-Modellierung: welche Daten werden wo benötigt?
- Konsistenz zur FP-Analyse