



Vienna University of Technology

Objektorientierte Modellierung

Anwendungsfalldiagramm



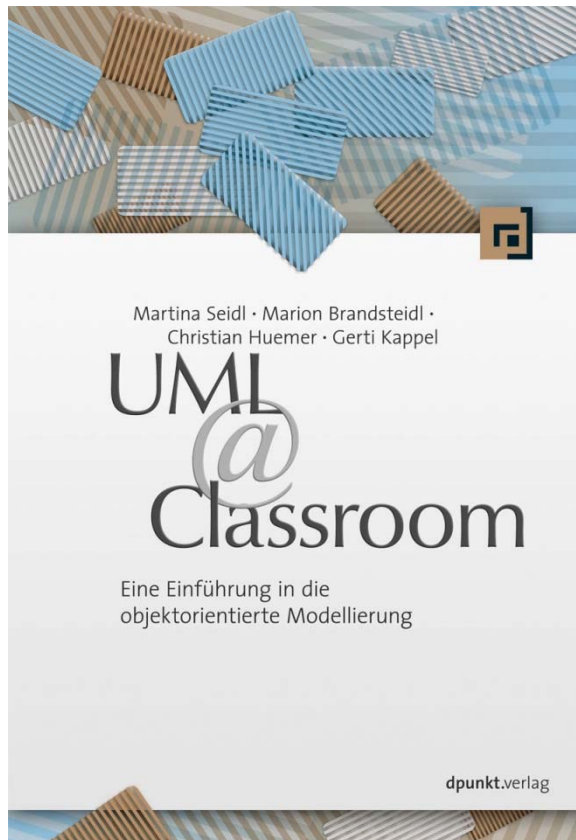
Business Informatics Group

Institute of Software Technology and Interactive Systems
Vienna University of Technology

Favoritenstraße 9-11/188-3, 1040 Vienna, Austria
phone: +43 (1) 58801-18804 (secretary), fax: +43 (1) 58801-18896
office@big.tuwien.ac.at, www.big.tuwien.ac.at

Literatur

- Die Vorlesung basiert auf folgendem Buch:



UML @ Classroom:

Eine Einführung in die objekt-orientierte Modellierung

Martina Seidl, Marion Brandsteidl,
Christian Huemer und Gerti Kappel

dpunkt.verlag

Juli 2012

ISBN 3898647765

- **Anwendungsfalldiagramm**
- **Strukturmodellierung**
- **Zustandsdiagramm**
- **Sequenzdiagramm**
- **Aktivitätsdiagramm**

Inhalt

- Einleitung
- Akteure
- Anwendungsfälle
- Beziehungen zwischen Anwendungsfällen und Akteuren
- Beziehungen zwischen Anwendungsfällen
- Beziehungen zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Beispiel: Studienabteilung
- Zusammenfassung



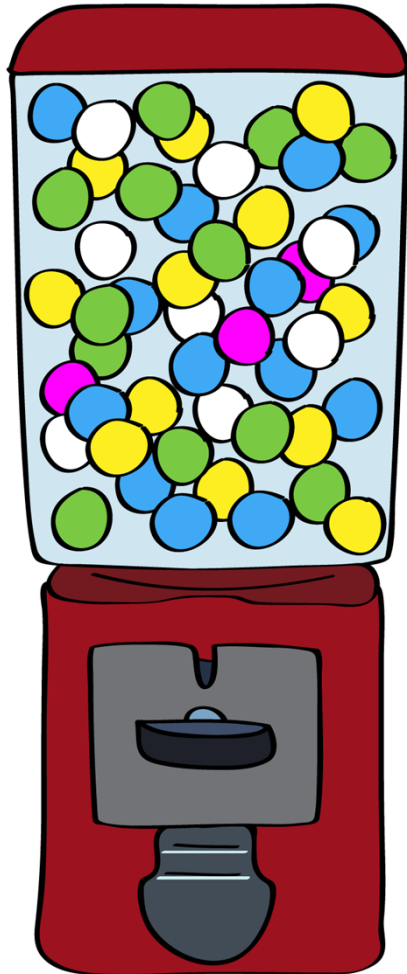
Einführung (1/2)

- Use Cases (= Anwendungsfälle) sind Ausgangspunkt vieler objekt-orientierter Entwicklungsmethoden.
- Zusätzlich dienen sie oft auch als Basiskonzept, das sich über den kompletten Analyse- und Designprozess hinweg spannt.
- Ausgangsfragen für den Einsatz von Anwendungsfällen:
 - Warum verwendet man Anwendungsfälle?
 - Wie sehen Anwendungsfälle aus?
 - Was macht man mit Anwendungsfällen, wenn man sie einmal definiert hat?
- Anwendungsfälle konzentrieren sich auf das fundamentale Problem bei der Entwicklung eines Systems, der Entwicklung einer Lösung für den Kunden bzw. Anwender, die der Kunde bzw. der Anwender auch gewünscht hat.

Einführung (2/2)

- Anwendungsfälle repräsentieren die Anforderungen der Kunden
- „Ein Anwendungsfall ist eine Sequenz von Transaktionen innerhalb eines Systems, deren Aufgabe es ist, einen für den einzelnen Akteur (Anwender) identifizierbaren Nutzen zu erzeugen.“ [Ivar Jacobson]
- Akteure interagieren mit dem System im Kontext der Anwendungsfälle
- Akteur:
 - Rolle, die jemand oder etwas einnimmt und die in Beziehung zum Geschäftsbereich steht, oder
 - Alles, das mit dem System interagiert
- Transaktionen innerhalb eines Systems implizieren, dass dem Akteur eine Reihe von Möglichkeiten geboten wird um mit dem System zu kommunizieren und dass durch sie ein messbarer Nutzen erzeugt wird.
- Ein messbarer Nutzen impliziert, dass die Ausführung einer Transaktion eine sichtbare, quantifizierbare und/oder qualifizierbare Auswirkung auf jene Dinge hat, die außerhalb des Systems liegen, im speziellen auf den Akteur.

Bsp.: Kaugummiautomat

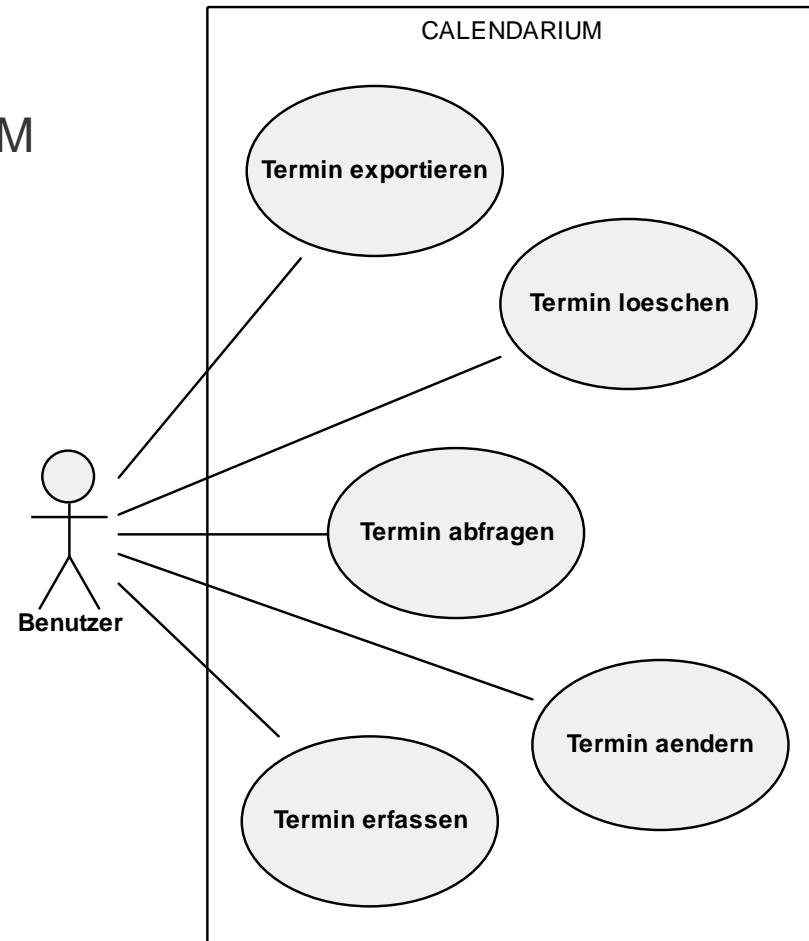


Use Case Kaugummi kaufen

- Standardablauf der Benutzerinteraktion
 - Münze einwerfen
 - Hebel drehen
 - Klappe öffnen
 - Kaugummi entnehmen

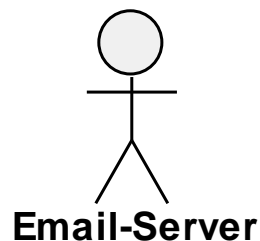
Bsp.: CALENDARIUM

- **System** (was wird beschrieben?)
 - der Onlinekalender CALENDARIUM
- **Akteur** (wer benutzt das System?)
 - Benutzer des Kalenders
- **Anwendungsfälle des Benutzers** (was machen die Akteure?)
 - Abfragen von Terminen
 - Exportieren von Terminen
 - Löschen von Terminen
 - Ändern von Terminen
 - Erfassen von Terminen

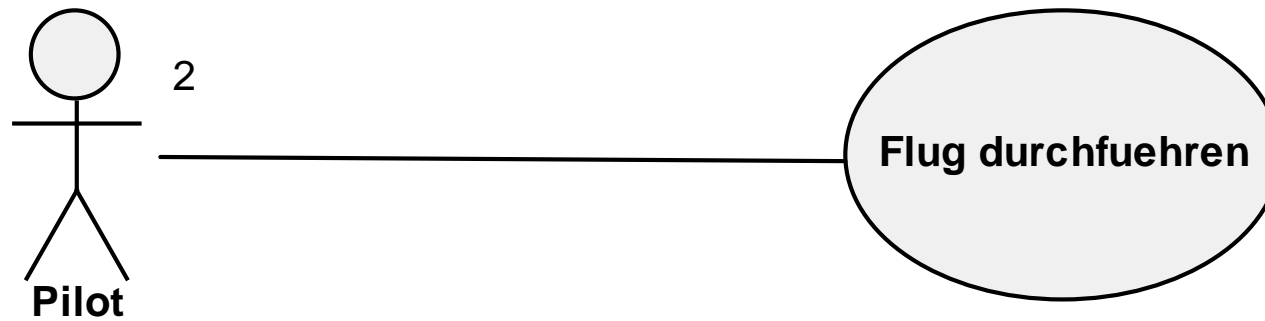


Akteur

- Akteure **interagieren** mit dem System...
 - indem sie das **System benutzen**,
d.h. die Ausführung von Anwendungsfällen initiieren
 - indem sie **vom System benutzt werden**,
d.h. Funktionalität zur Realisierung von Anwendungsfällen zur Verfügung stellen
- Akteur wird **durch Assoziationen mit Anwendungsfällen verbunden**,
d.h. er »kommuniziert« mit dem System
- Jeder Akteur muss mit **mindestens einem Anwendungsfall** kommunizieren
- Die Assoziation ist binär und kann **Multiplizitäten** aufweisen
- Notationsvarianten:



Beispiel für Multiplizitäten: Pilot



Akteur - Eigenschaften

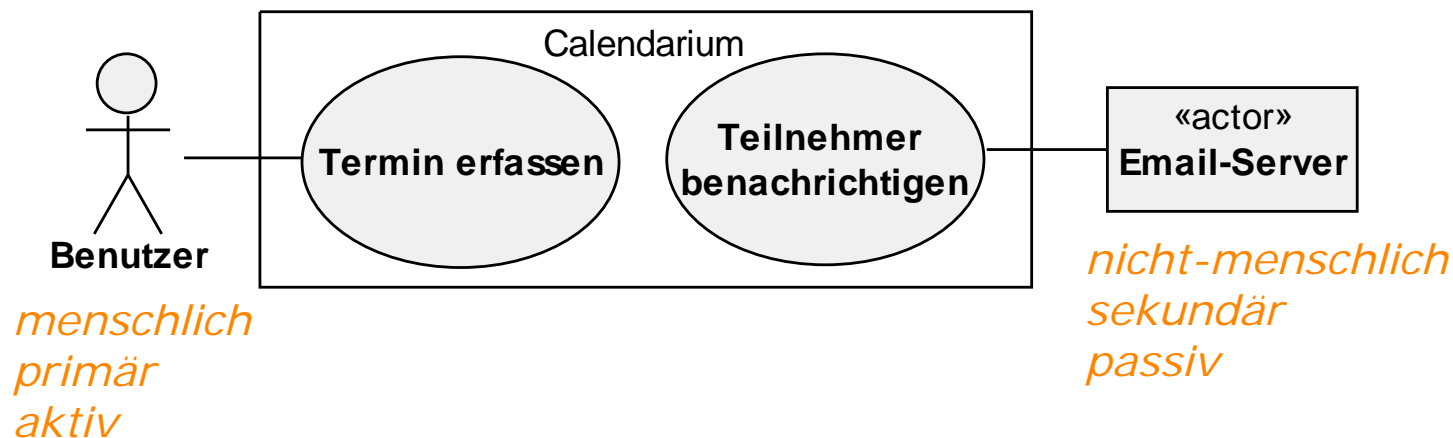
- Akteure repräsentieren **Rollen der Benutzer**
 - Konkrete Benutzer können gleichzeitig mehrere Rollen spielen, annehmen und ablegen
- Akteure befinden sich **klar außerhalb** der Systemgrenzen
- Üblicherweise werden Benutzerdaten auch innerhalb des Systems verwaltet. Diese werden als Objekte bzw. Klassen innerhalb des Systems modelliert.

- Beispiel: Kassier
 - Als Akteur am Kassenterminal
 - Die Rolle, in der die Person mit dem Kassensystem interagiert
 - Die Klasse Kassier umfasst Objekte, welche die Benutzerdaten beinhalten (Name, SozVersNr, ...)

Akteur - Klassifikation

- **Menschlich**
 - z.B. Anfänger, geübter Benutzer, Admin
- **Nicht-menschlich**
 - z.B. Fax-System, E-Mail-System
- **Primär:** Hauptnutznießer der Anwendung
- **Sekundär:** notwendig für das Funktionieren des Systems
- **Aktiv:** stößt selbst Anwendungsfälle an
- **Passiv:** stößt selbst keine Anwendungsfälle an

- **Beispiel:**



Anwendungsfall

- Anwendungsfälle (use cases) beschreiben das **Verhalten**, das von dem zu entwickelnden System **erwartet** wird
- Identifizierung durch Sammeln von Kundenwünschen und Analyse der textuellen Problemstellung
- Notationsvarianten
- Kurzbeschreibung als Notiz

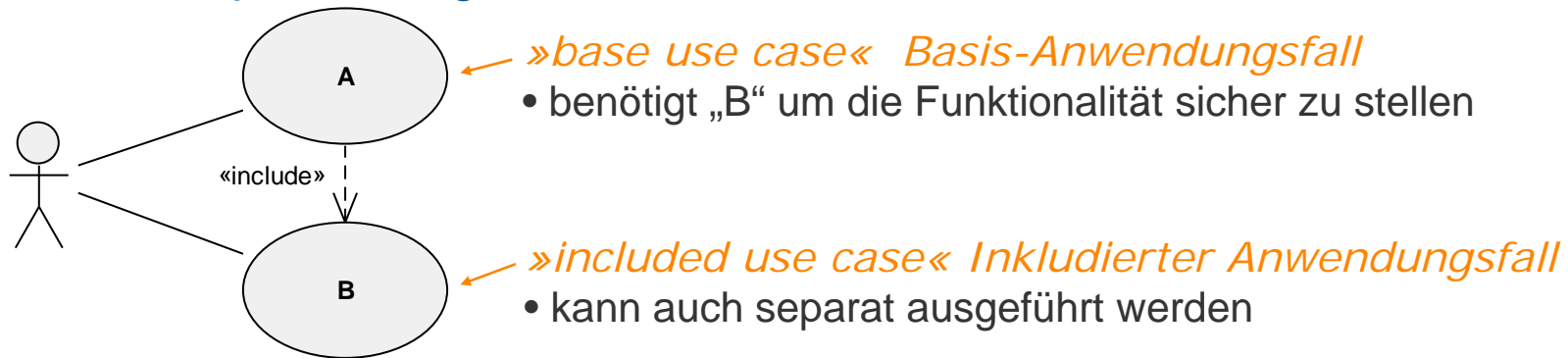


»Ein Termin kann für einen oder mehrere Teilnehmer von berechtigten Benutzern (müssen nicht notwendigerweise auch Teilnehmer sein) erfasst werden. Alle Teilnehmer müssen über diesen neuen Termin verständigt werden. Neue Termine müssen sofort in allen geöffneten, die jeweiligen Teilnehmer betreffenden Kalendern nachgezogen werden.«

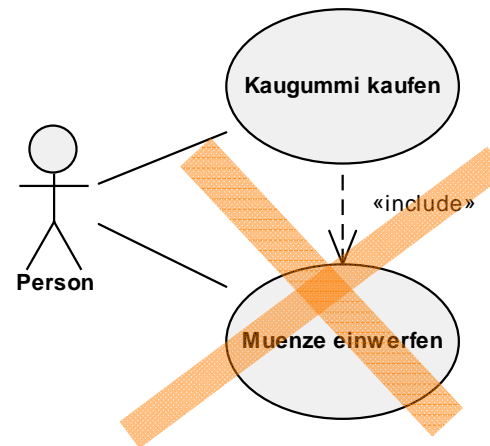
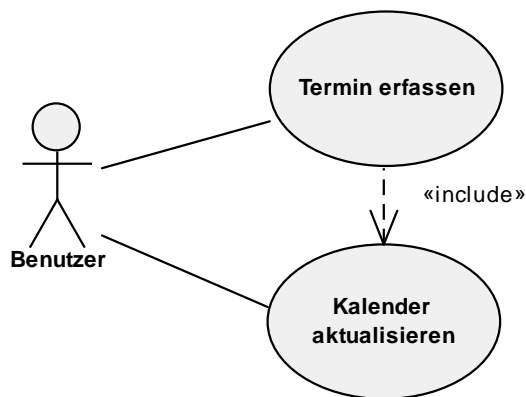


«include» - Beziehung

- Das **Verhalten** des benutzten Anwendungsfalls (inkludierter Anwendungsfall) wird in den benutzenden Anwendungsfall (Basis-Anwendungsfall) **eingebunden**
- B ist unbedingt notwendig, um die Funktionalität von A sicher zu stellen
- B kann separat ausgeführt werden

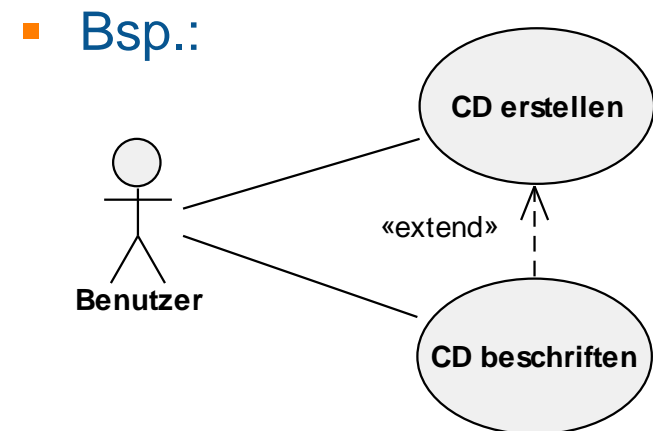
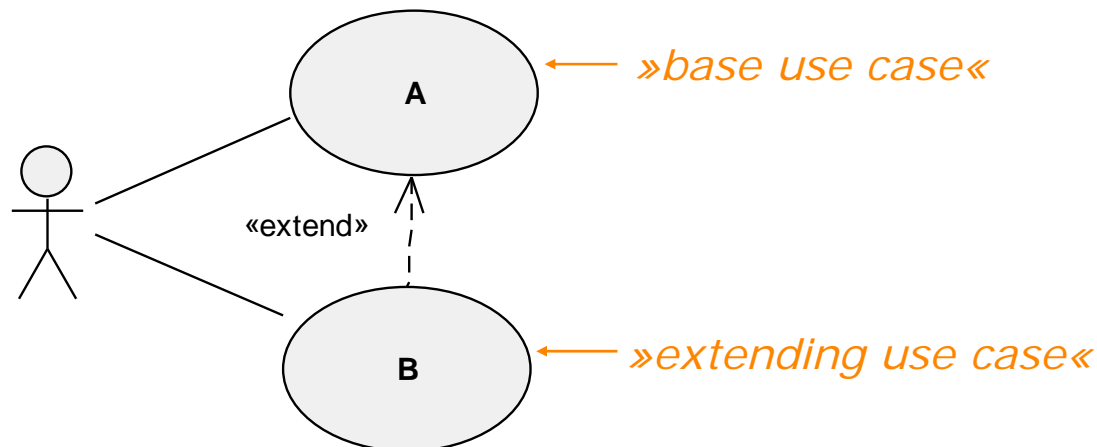


- Bsp.:



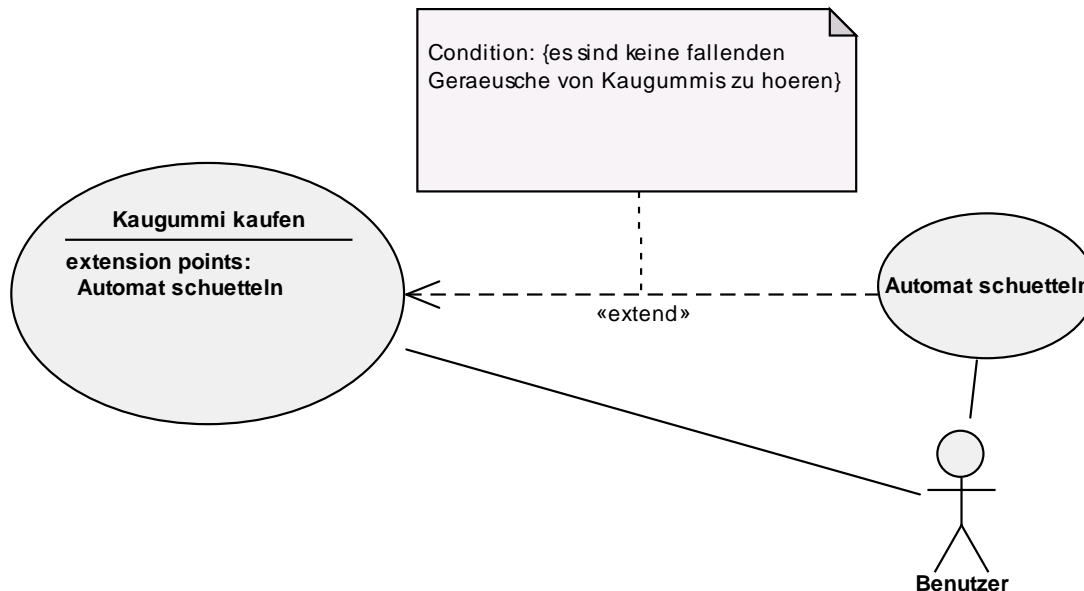
«extend» - Beziehung

- Das Verhalten von B **kann** in A **inkludiert werden**
 - Somit entscheidet A, ob B ausgeführt wird
- B kann von A aktiviert werden, muss aber nicht
- A bzw. B können auch separat ausgeführt werden
- Angabe des »Wo« durch Erweiterungsstellen in A
- Angabe des »Wann« durch Bedingung in A bzw. als Teil der «extend»-Beziehung



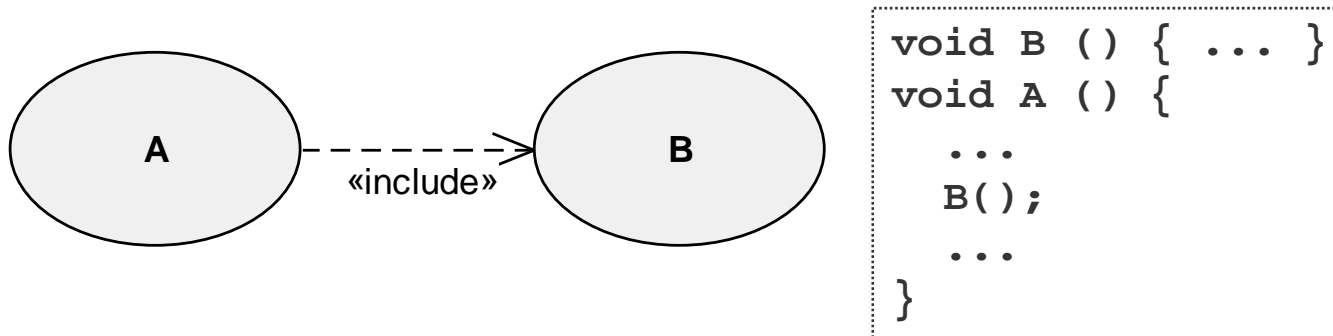
«extend» - Beziehung: Erweiterungsstellen

- Mehrere Erweiterungsstellen (extension points) je Anwendungsfall möglich
- Namen von Erweiterungsstellen
 - müssen eindeutig sein
 - müssen nicht mit den Namen der erweiternden Anwendungsfälle übereinstimmen
- Beispiel: Kaugummiautomat

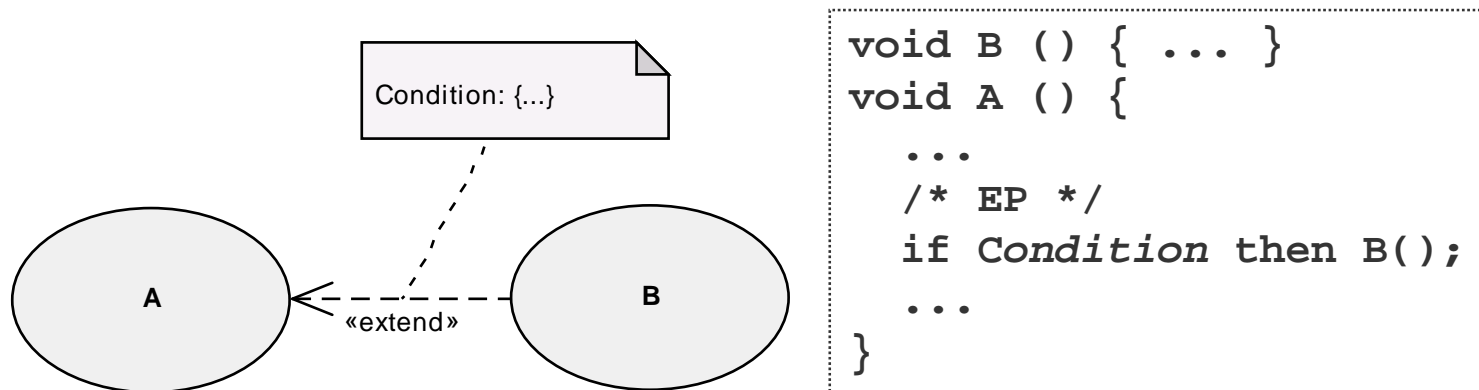


Analogien zu Programmiersprachen

- `<<include>>` entspricht Unterprogrammaufruf bzw. Makroexpansion



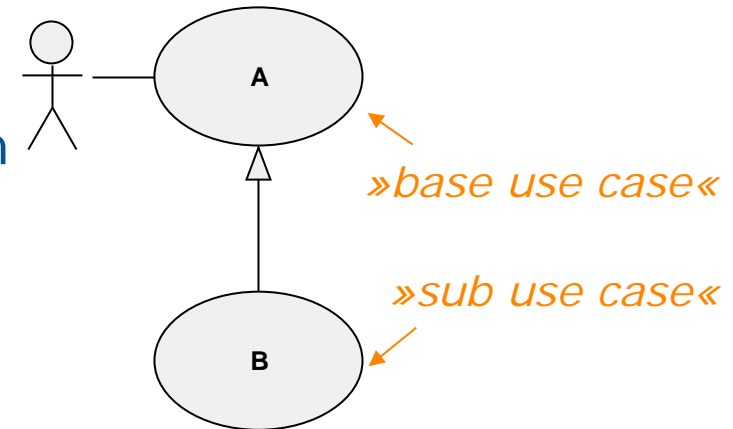
- `<<extend>>` entspricht bedingtem Unterprogrammaufruf



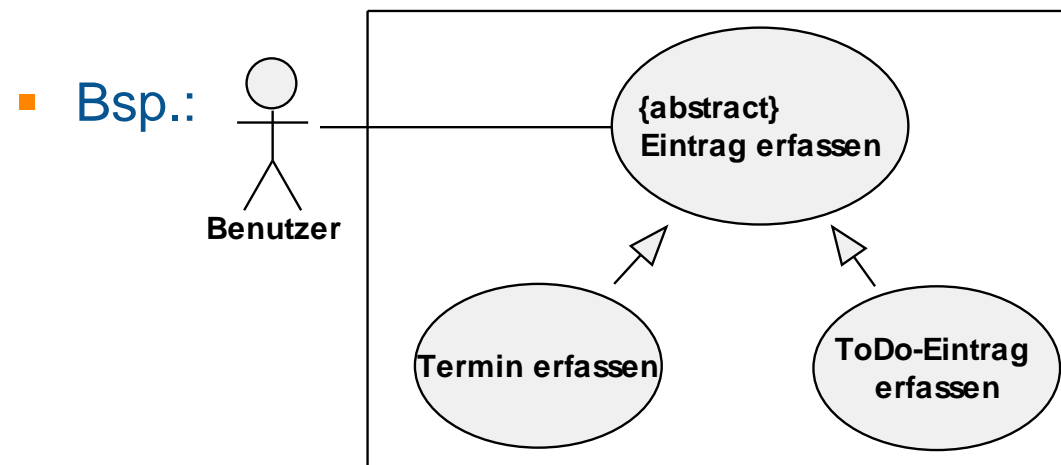
- **Generalisierung** von Anwendungsfällen entspricht etwa dem `super`-Konstrukt von Java

Generalisierung bei Anwendungsfällen

- B erbt das Verhalten von A und kann dieses überschreiben oder ergänzen
- B erbt alle Beziehungen von A
- B benötigt A (übernimmt Grundfunktionalität von A)
- B entscheidet, was von A ausgeführt bzw. geändert wird

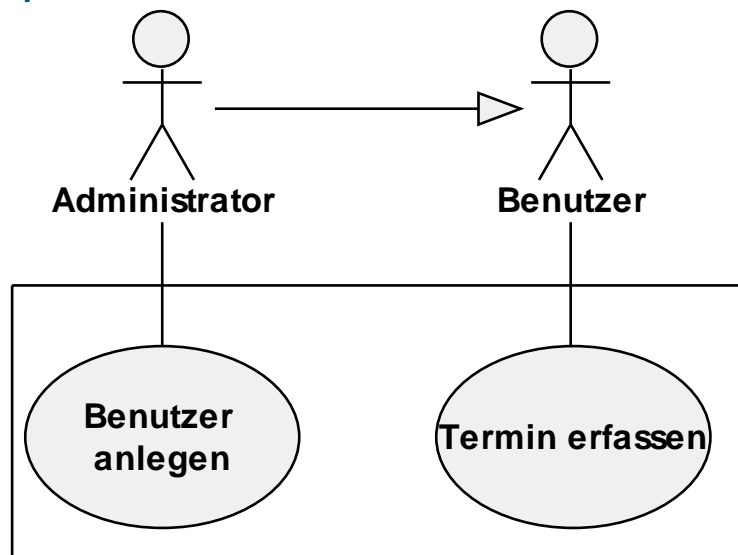
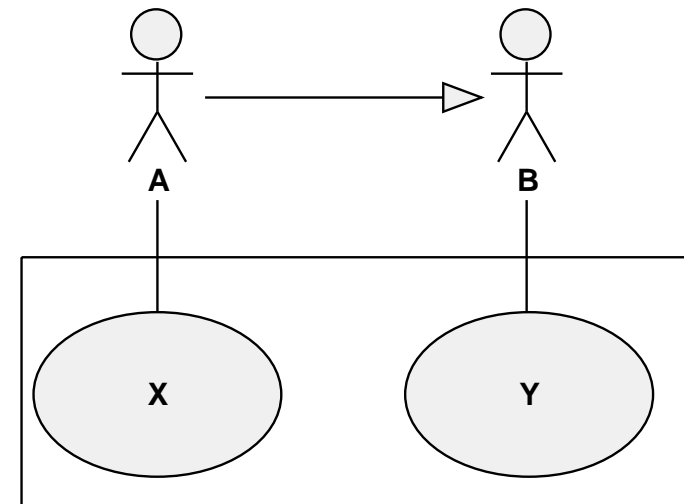


- Modellierung abstrakter Anwendungsfälle möglich: {abstract} abstrakte Anwendungsfälle sind nicht ausführbar!



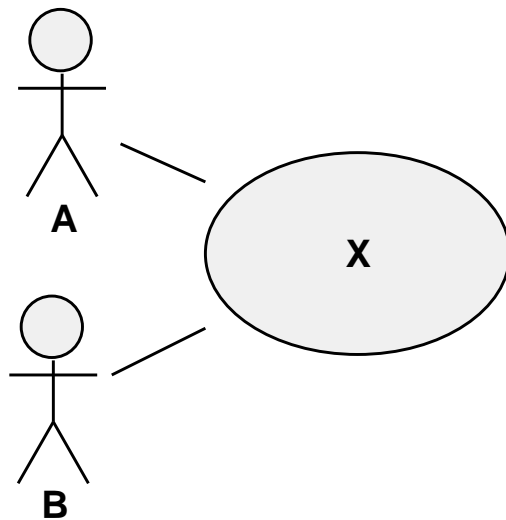
Generalisierung bei Akteuren (1/2)

- Akteur A erbt von Akteur B
- A kann mit den Anwendungsfällen X und Y kommunizieren
- B kann nur mit Y kommunizieren
- Mehrfachvererbung ist erlaubt
- Bsp.:

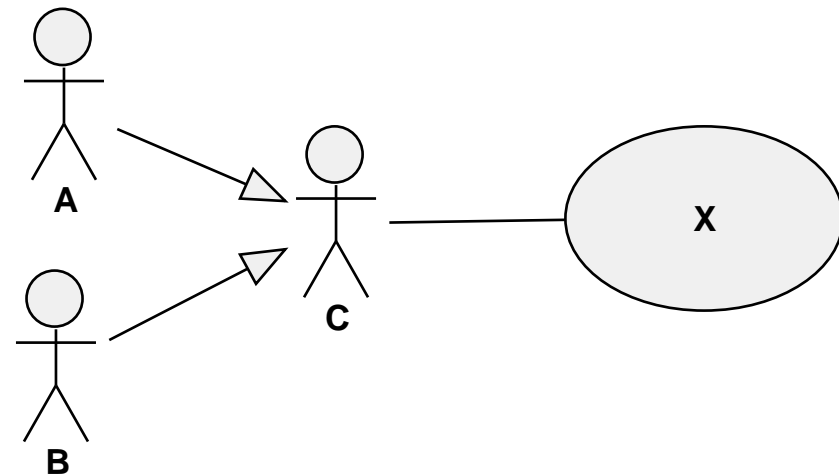


Generalisierung bei Akteuren (2/2)

- Unterscheidung, ob mehrere Akteure gemeinsam mit einem Anwendungsfall kommunizieren können oder müssen.



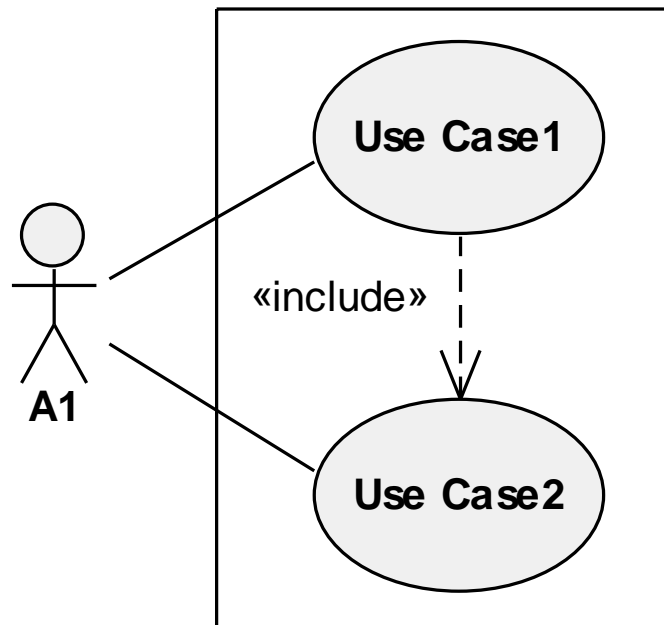
A und B kommunizieren mit X



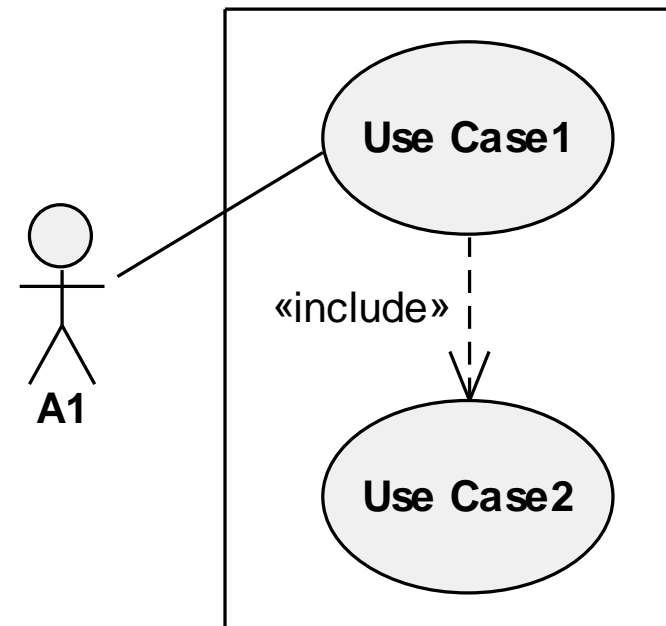
A oder B kommuniziert mit X

<<include>> — Standard vs. Best Practice

Standard

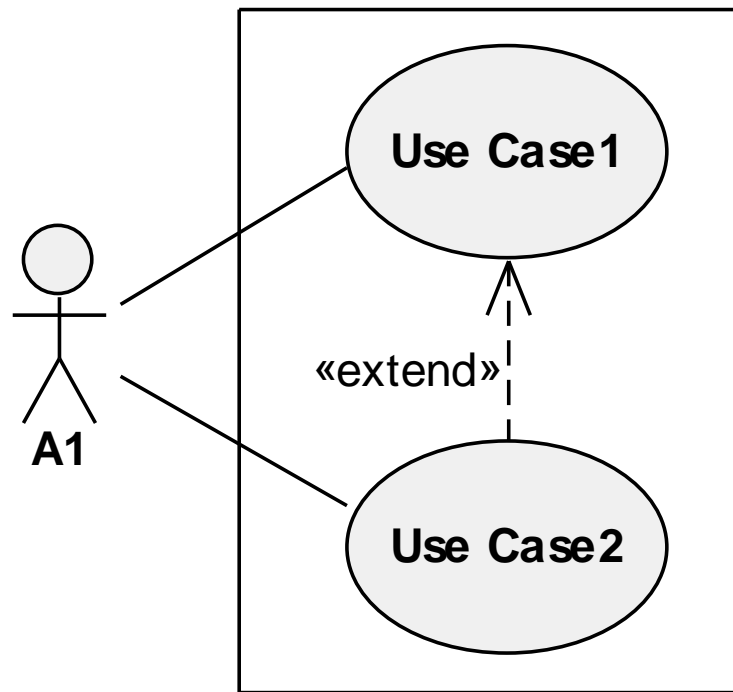


Best Practice

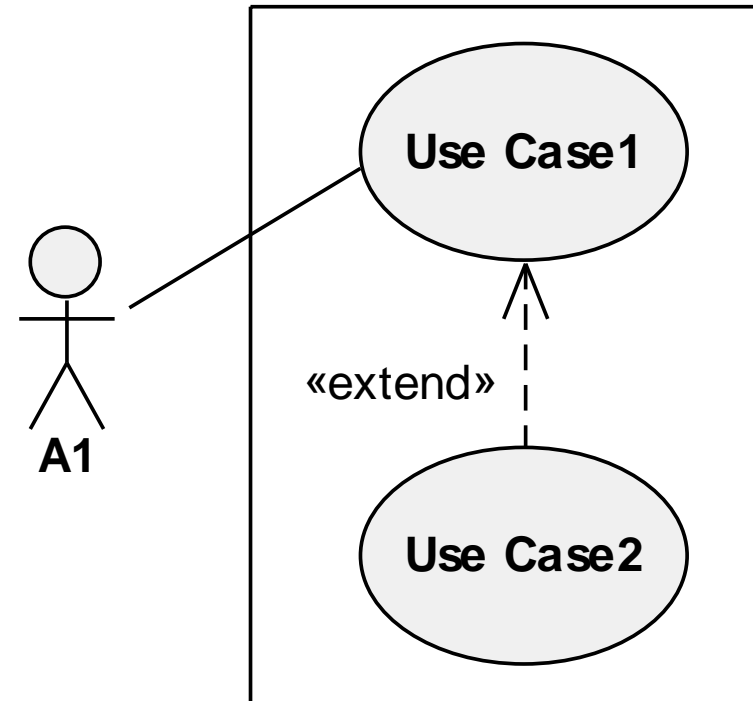


<<extend>> — Standard vs. Best Practice

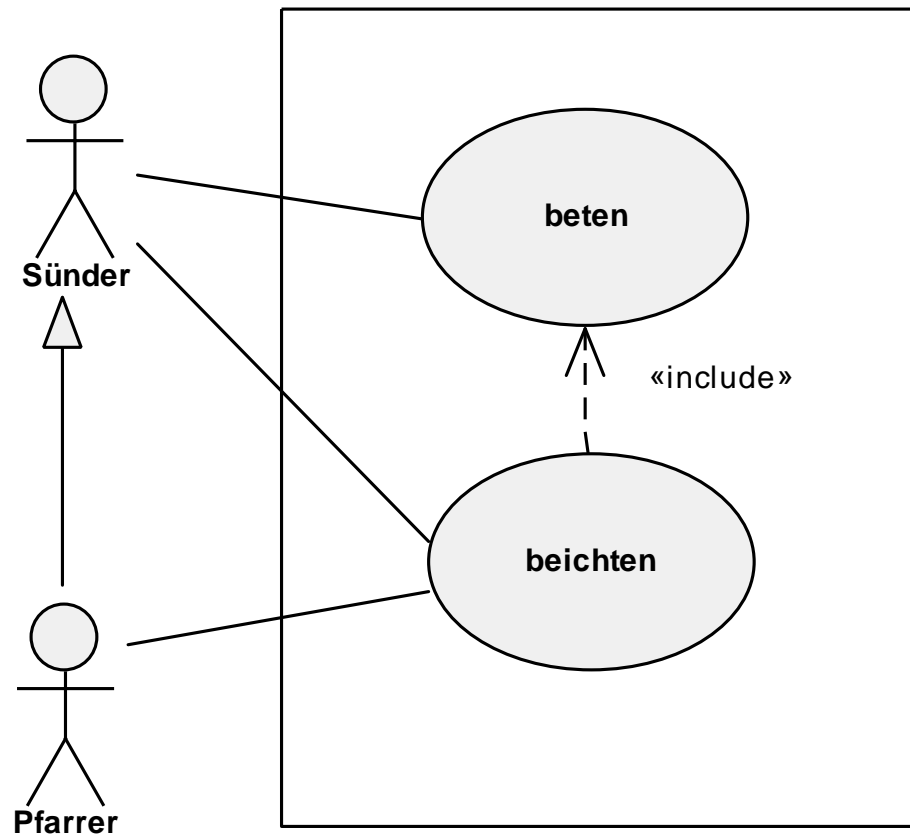
Standard



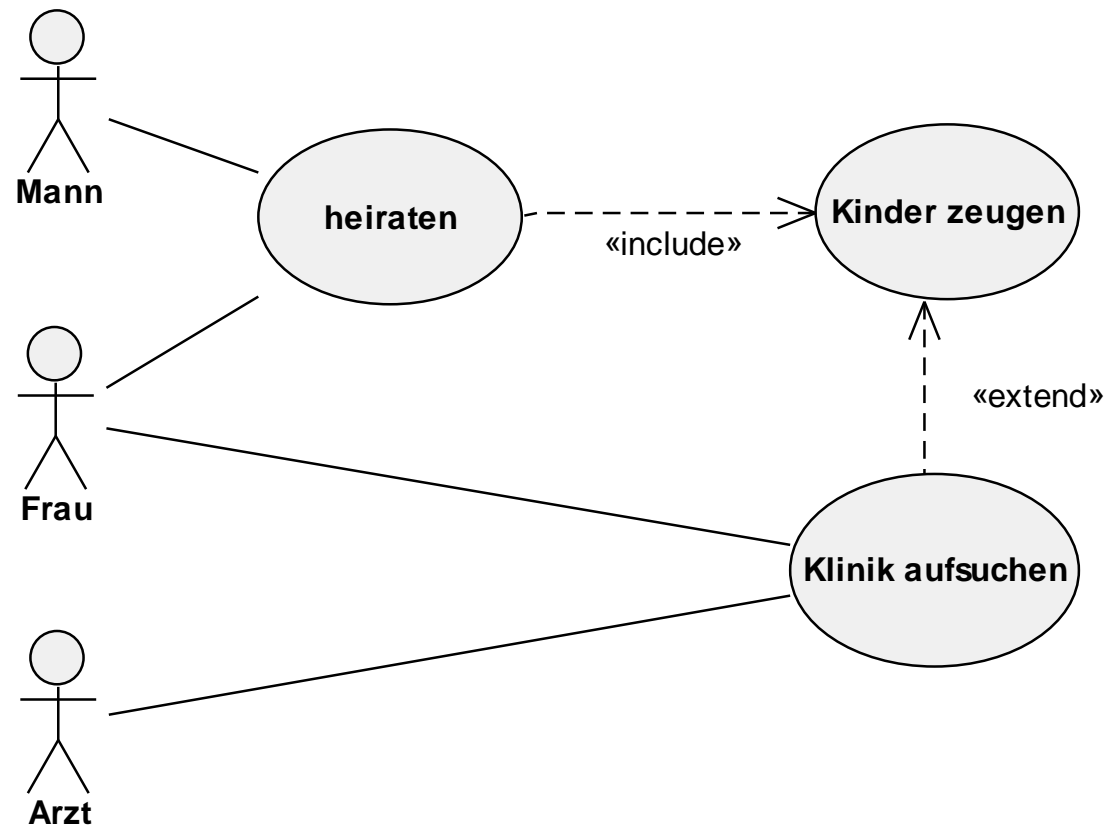
Best Practice



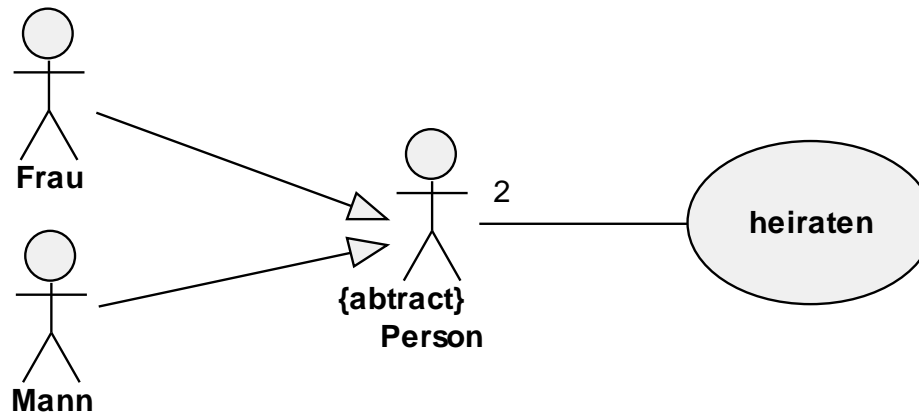
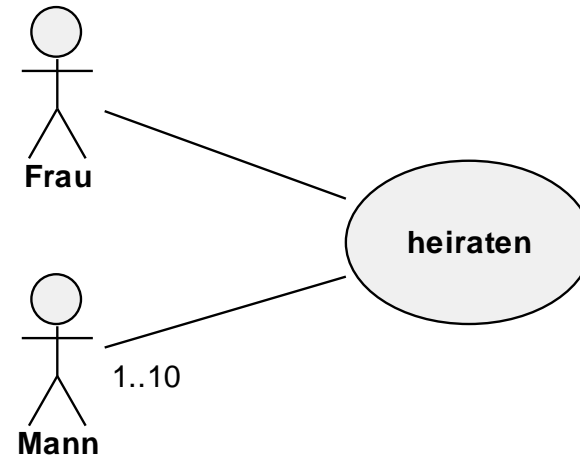
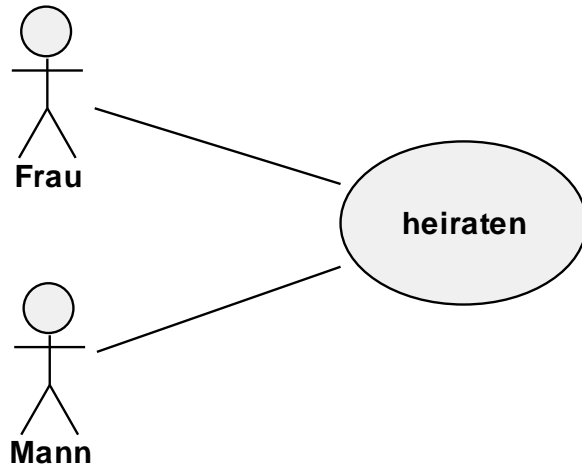
Beispiel: Beichten



Beispiel: Heiraten und Kinder zeugen



Beispiel: 3 Arten von Heiraten



Identifikation von Akteuren

- Wer **benutzt** die wesentlichen Anwendungsfälle?
- Wer braucht Systemunterstützung für die **tägliche Arbeit**?
- Wer ist für die **Systemadministration** zuständig?
- Mit welchen externen **Geräten / (Software-) Systemen** muss das System kommunizieren können?
- Wer oder was interessiert sich für die **Ergebnisse** des Systems?

Identifikation von Anwendungsfällen

- Nach der Identifikation der Akteure
- Trigger für Anwendungsfälle suchen
 - Trigger = Ereignisse, die eintreten müssen, damit das System veranlasst wird ein Ergebnis zu produzieren.
 - Der Aufruf des Systems erfolgt oft durch einen Akteur, der damit Akteur des Anwendungsfalles wird.
 - Es werden folgende Trigger unterschieden: interne, externe und zeitliche Trigger.

Anwendungsfallbeschreibung

- **Strukturierter Text**

- Name
- Kurzbeschreibung
- Vorbedingung: Voraussetzung für erfolgreiche Ausführung
- Nachbedingung: Systemzustand nach erfolgreicher Ausführung
- Fehlersituationen: nur problembereichsrelevante Fehler
- Systemzustand im Fehlerfall
- Akteure, die mit dem Anwendungsfall kommunizieren
- Trigger: auslösende Ereignisse für den Anwendungsfall
- Standardablauf: einzelne Schritte / andere Anwendungsfälle
- Alternativabläufe: Abweichungen vom Standardablauf

[nach A. Cockburn: Goals and Use Cases. Journal of Object-Oriented Programming, Sept. 1997]

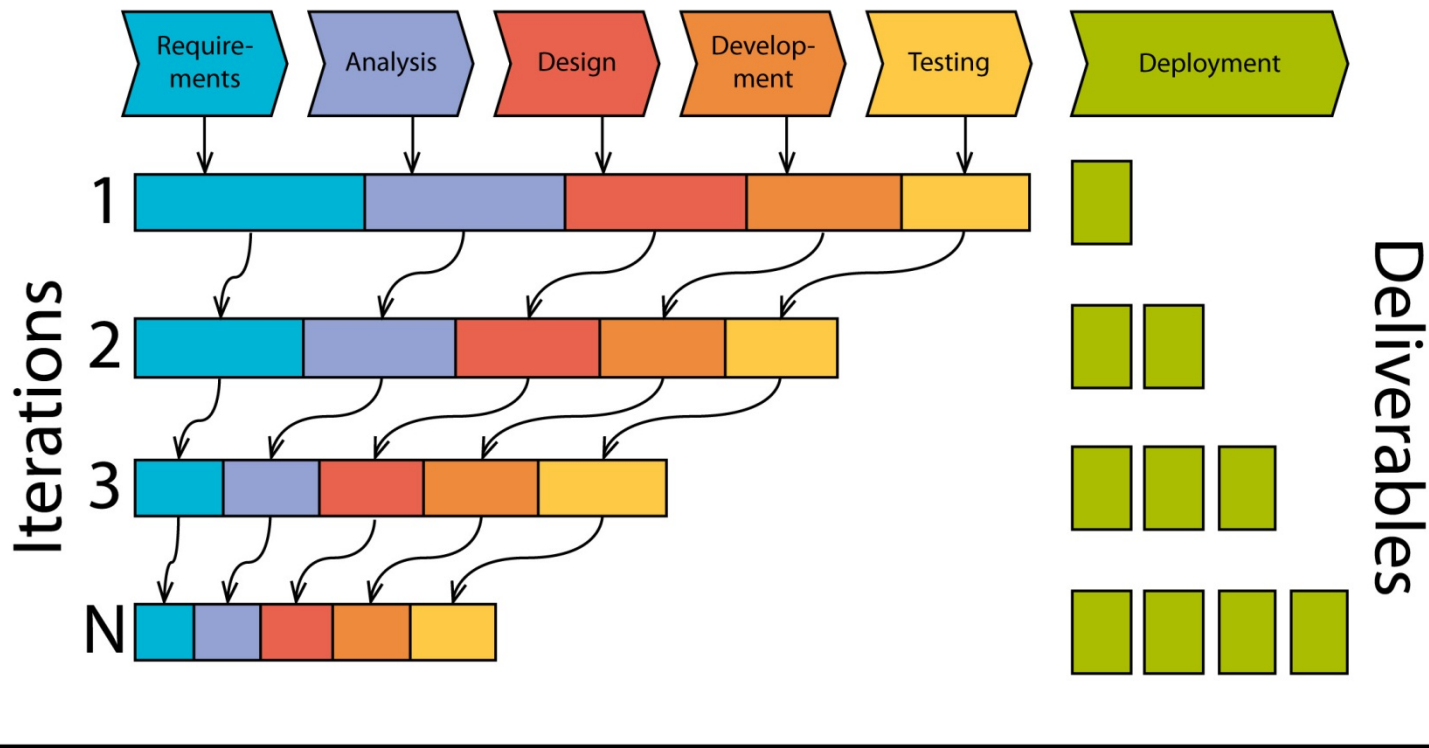
- **Beispiel folgt später in den Folien**



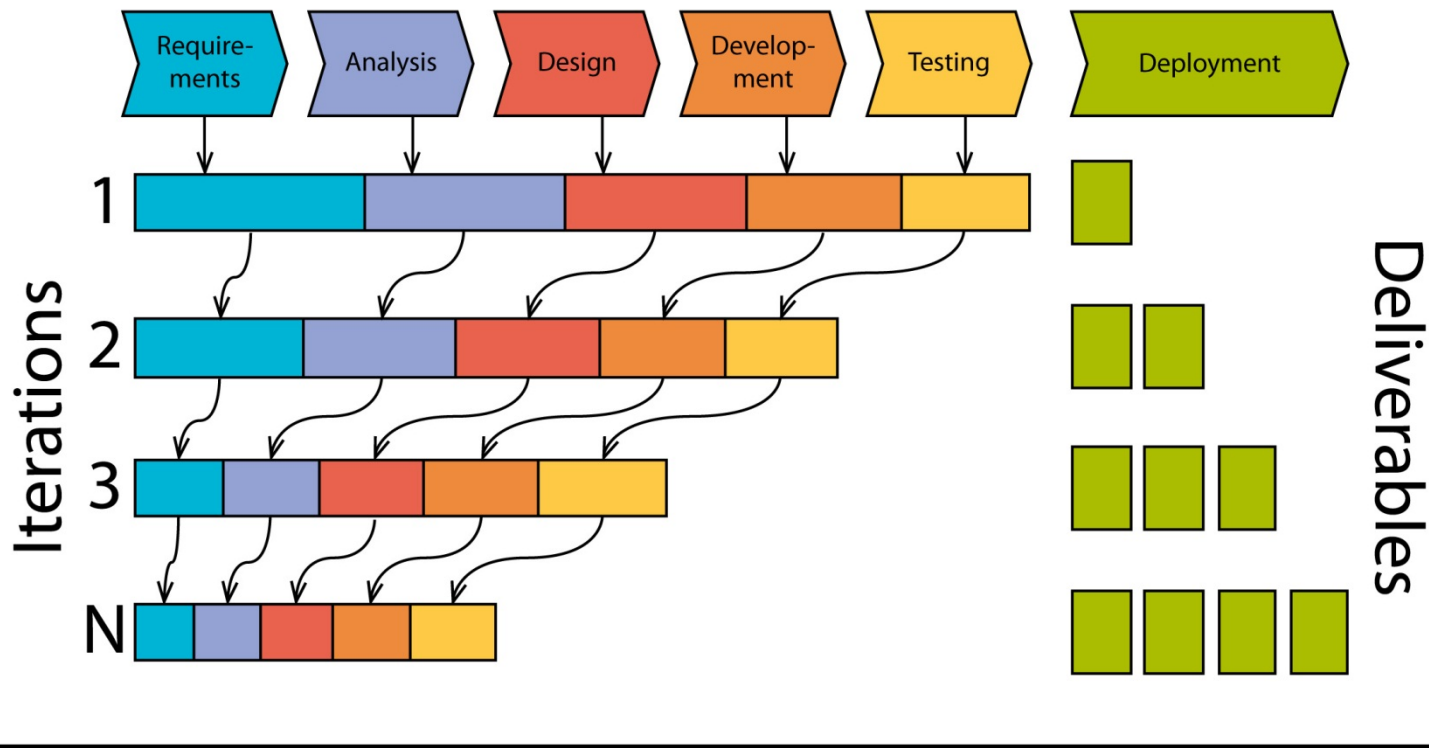
Regeln zur Anwendungsfallmodellierung

- Die wichtigsten funktionalen Anforderungen müssen in den Anwendungsfällen festgehalten werden.
- Ein Anwendungsfall....
 - beschreibt eine Transaktion für die der Auftraggeber bezahlt.
 - beschreibt einen typischen Fall, ein System zu verwenden und nicht mehr.
 - ist wie ein Theaterstück. Die Anwendungsfallbeschreibung enthält die Choreographie.
 - hat eine Einleitung, einen Hauptteil und einen Schluss.
 - soll so einfach wie möglich beschrieben sein.
 - muss präzise definiert sein.
 - ist dann fertig beschrieben, wenn der Kunde, die Anwender und die Softwareentwickler ihn akzeptieren.
 - stellt die Grundlage für einen Systemtest dar.
 - sollte mit maximal zwei Seiten beschrieben werden.

Use Cases

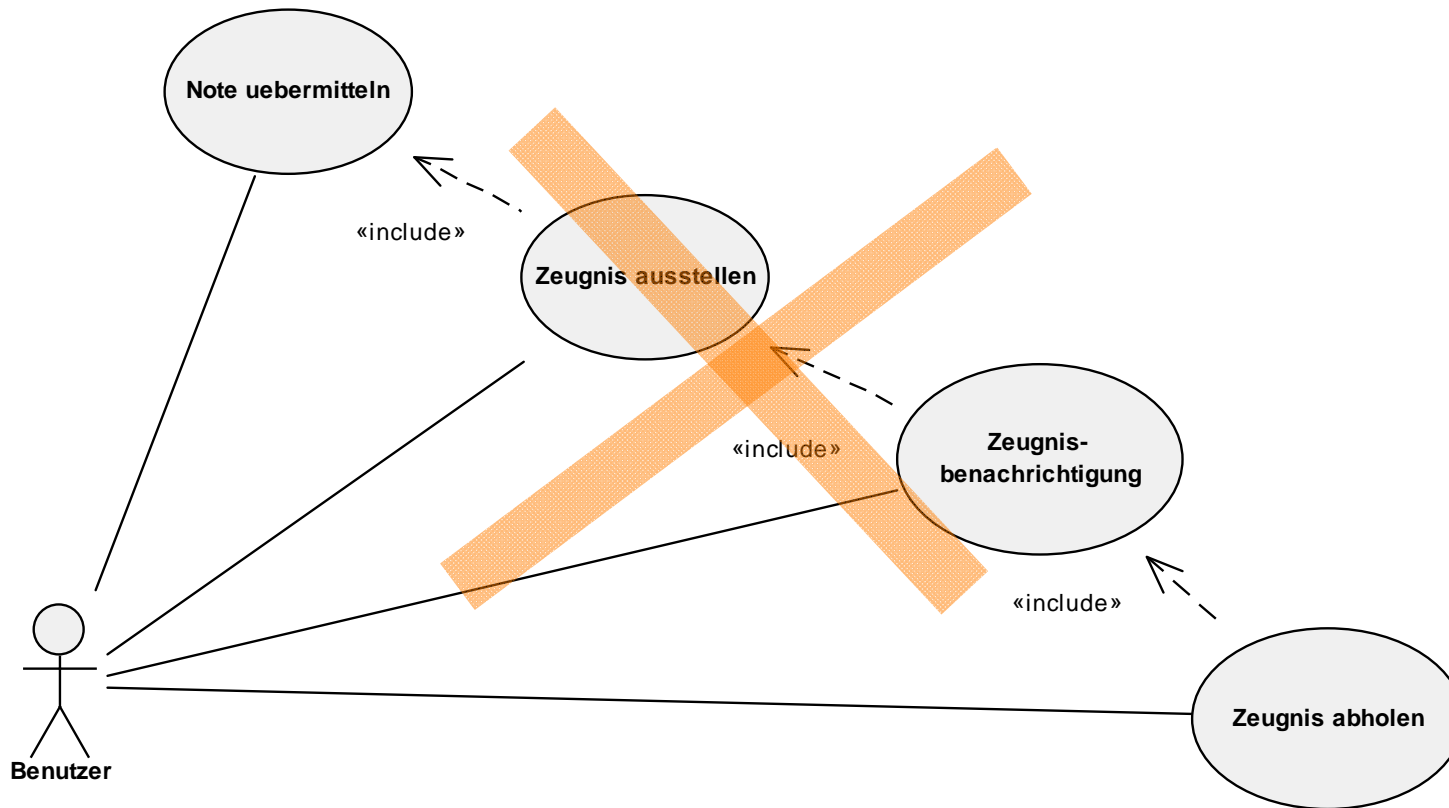


Use Cases



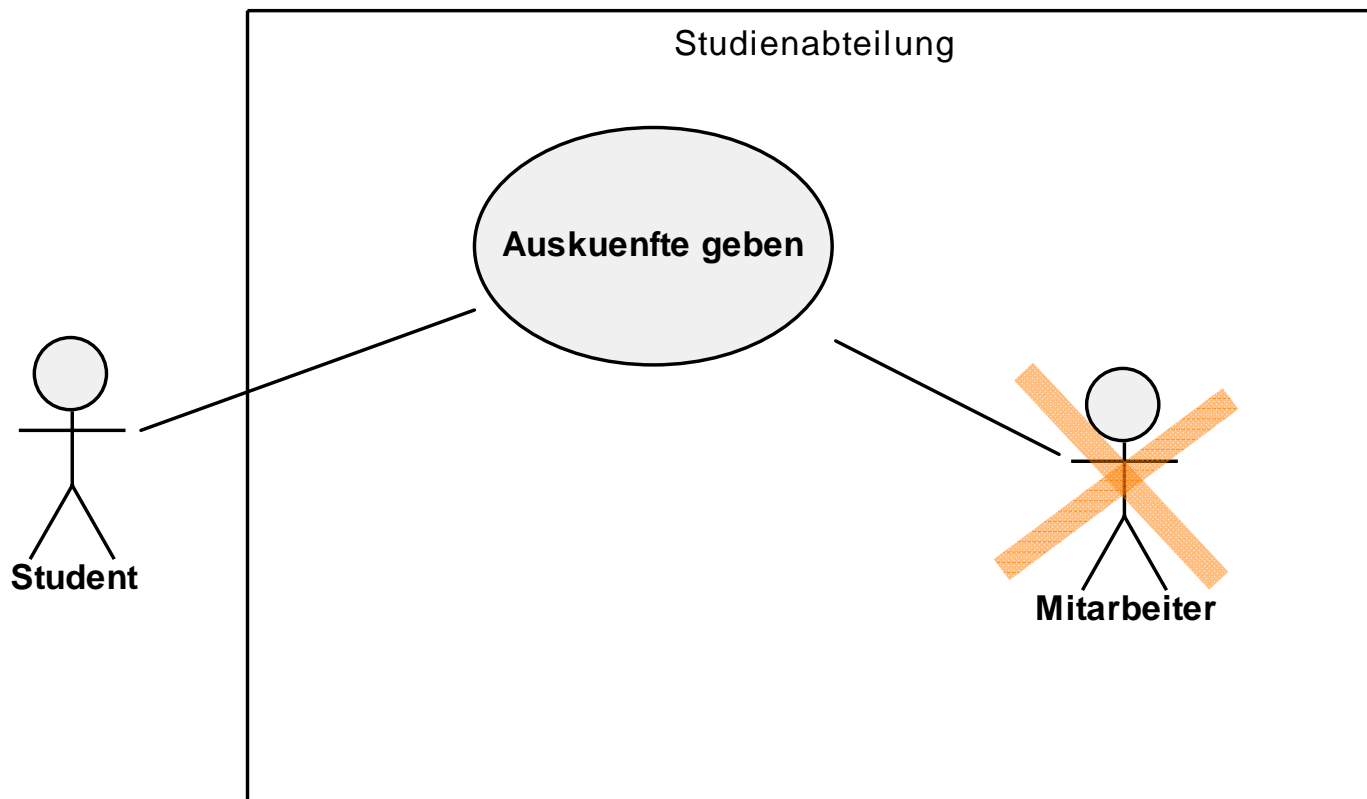
Typische Modellierungsfehler (1/6)

- Anwendungsfalldiagramme modellieren **keine Abläufe!**

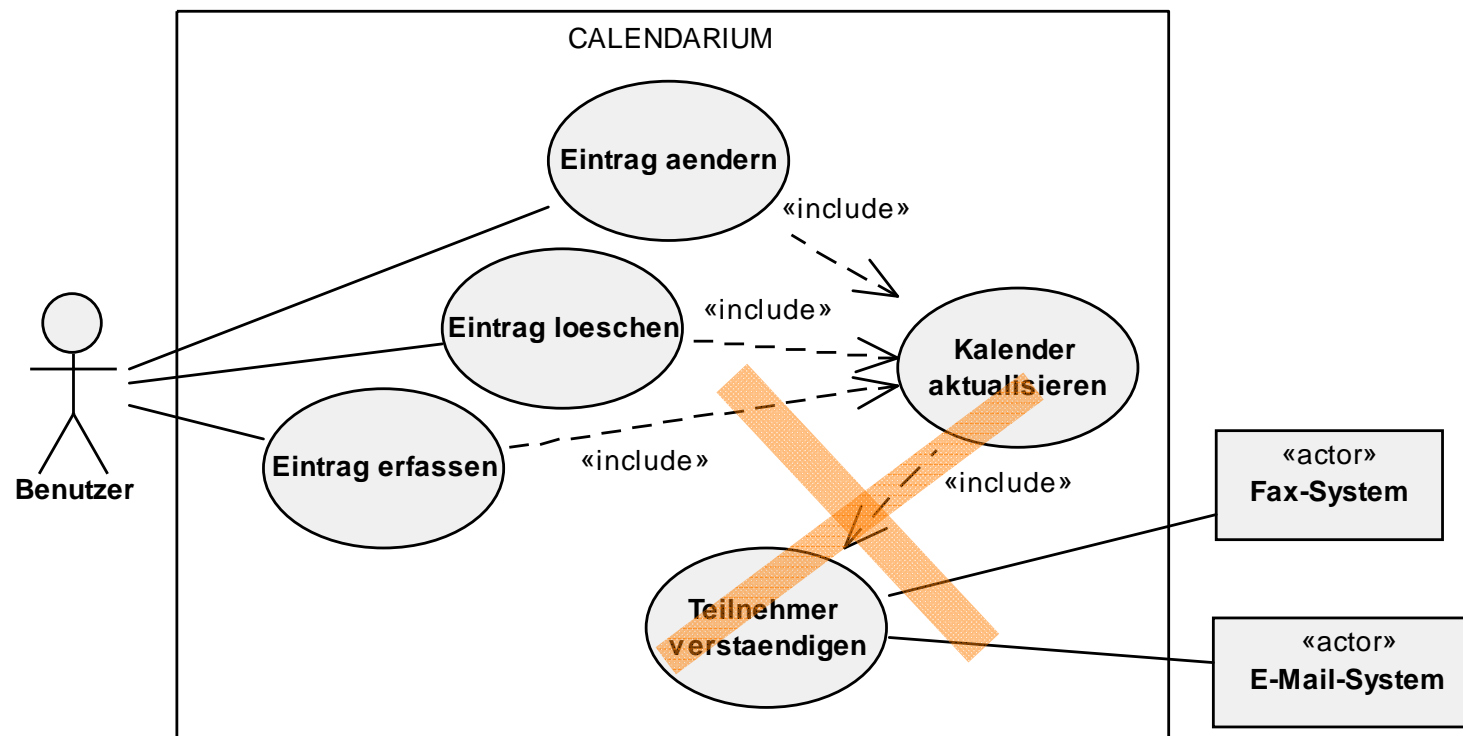


Typische Modellierungsfehler (2/6)

- Akteure sind immer außerhalb der Systemgrenzen!



Typische Modellierungsfehler (3/6)

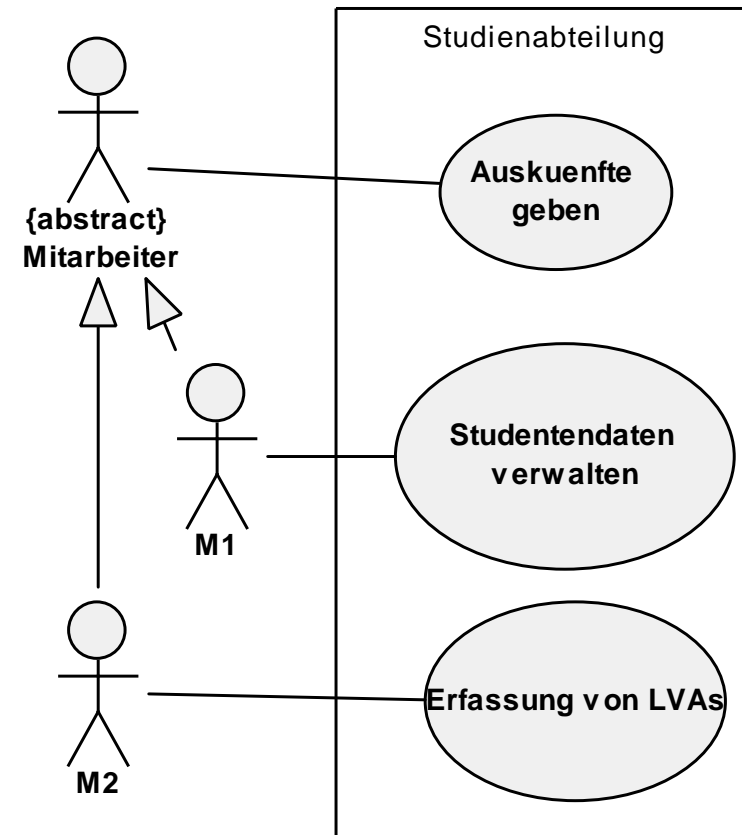
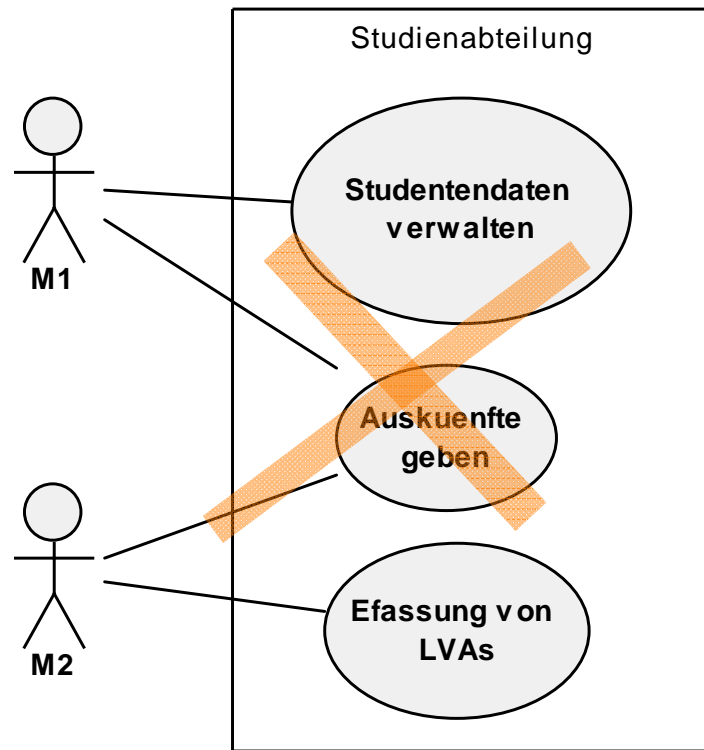


Obwohl der *Ablauf* so ist:

- 1) Eintrag erfassen
- 2) Kalender aktualisieren
- 3) Teilnehmer verständigen

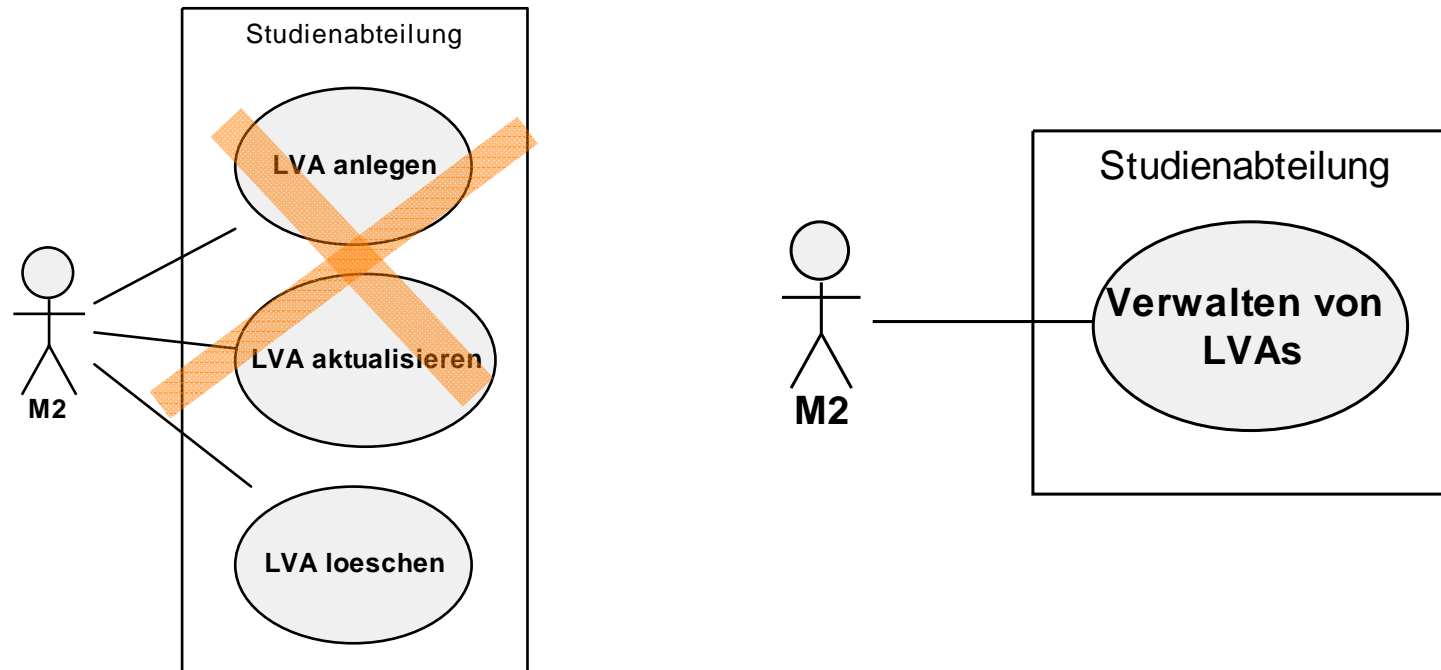
ist dieses Modell **falsch**, denn Teilnehmer verständigen ist nicht *Teil* von Kalender aktualisieren.

Typische Modellierungsfehler (4/6)



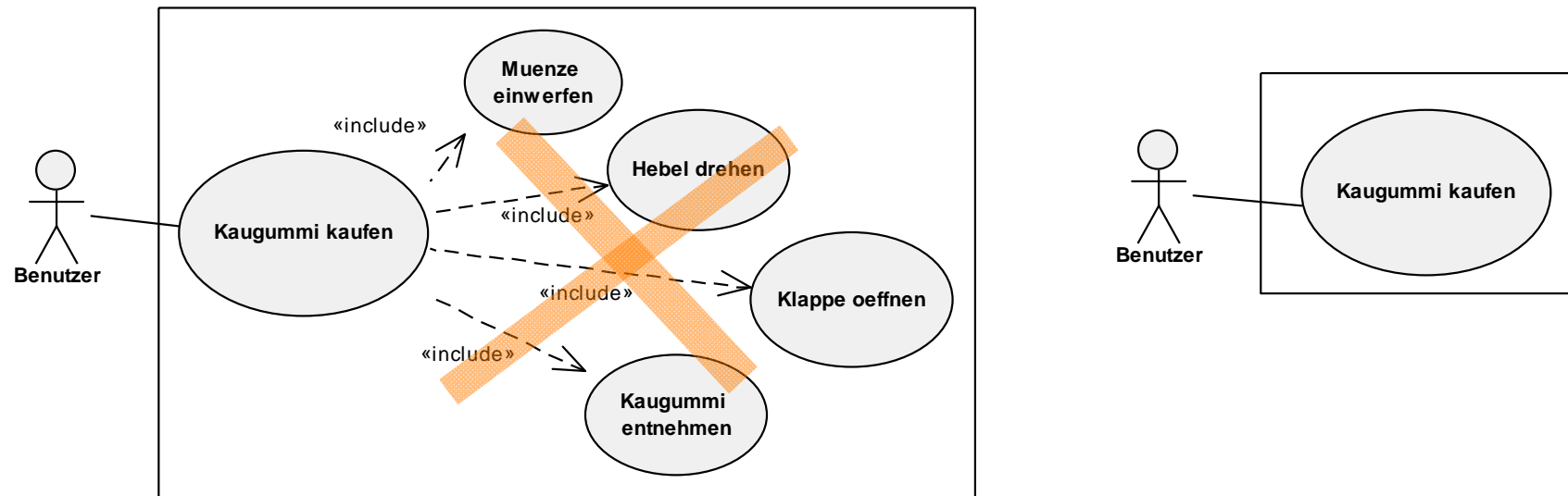
- Beim Anwendungsfall *Auskünfte geben* sind M1 **oder** M2 involviert

Typische Modellierungsfehler (5/6)



- Viele kleine Anwendungsfälle werden zu einem Anwendungsfall zusammengefasst, der die unterschiedlichen Möglichkeiten enthält

Typische Modellierungsfehler (6/6)



- Die einzelnen Schritte werden immer gemeinsam in einer vordefinierten Reihenfolge ausgeführt, daher handelt es sich nur um **einen** Anwendungsfall

Bsp.: Die Studienabteilung

- **Ziel:** vereinfachte Darstellung des Systems "Studienabteilung" einer Universität

1. Verbale Beschreibung

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. Studenten können hier immatrikulieren und inskribieren, sowie sich aber auch wieder vom Studium abmelden.
- Studenten erhalten hier Zeugnisse. Zeugnisrelevante Daten werden durch Lehrende an die Studienabteilung übermittelt. Die Studenten werden dann automatisch durch das Benachrichtigungssystem informiert.
- Es wird zwischen 2 Arten von Mitarbeitern unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studentendaten befassen (M1) und b) jene, die alle restlichen Aufgaben erfüllen (M2), wobei aber alle Mitarbeiter (M1 und M2) Auskünfte geben (per Telefon oder per Mail).
- M2-Mitarbeiter stellen Zeugnisse aus, sobald der/die Studierende diese abholt. Weiters werden von M2-Mitarbeitern Lehrveranstaltungen erfasst. Bei der Erfassung einer Lehrveranstaltung kann gleich ein Hörsaal reserviert werden.

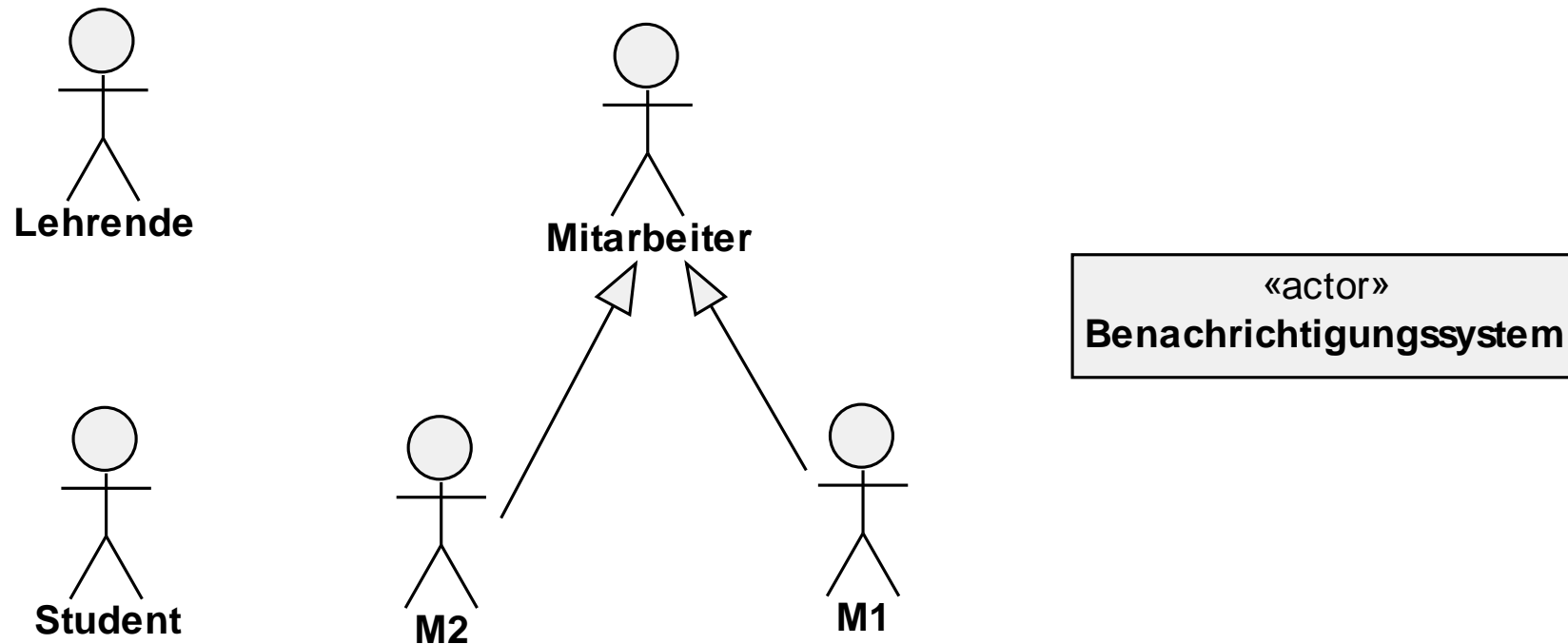
Bsp.: Akteure (1/2)

Wer sind die Benutzer?

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. **Studenten** können hier immatrikulieren und inskribieren, sowie sich aber auch wieder vom Studium abmelden.
- Studenten erhalten hier Zeugnisse. Zeugnisrelevante Daten werden durch **Lehrende** an die Studienabteilung übermittelt. Die Studenten werden dann automatisch durch das **Benachrichtigungssystem** informiert.
- Es wird zwischen 2 Arten von **Mitarbeitern** unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studentendaten befassen (**M1**) und b) jene, die alle restlichen Aufgaben erfüllen (**M2**), wobei aber alle Mitarbeiter (M1 und M2) Auskünfte geben (per Telefon oder per Mail).
- M2-Mitarbeiter stellen Zeugnisse aus, sobald der/die Studierende diese abholt. Weiters werden von M2-Mitarbeitern Lehrveranstaltungen erfasst. Bei der Erfassung einer Lehrveranstaltung kann gleich ein Hörsaal reserviert werden.

Bsp.: Akteure (2/2)

- Wer sind die Benutzer?



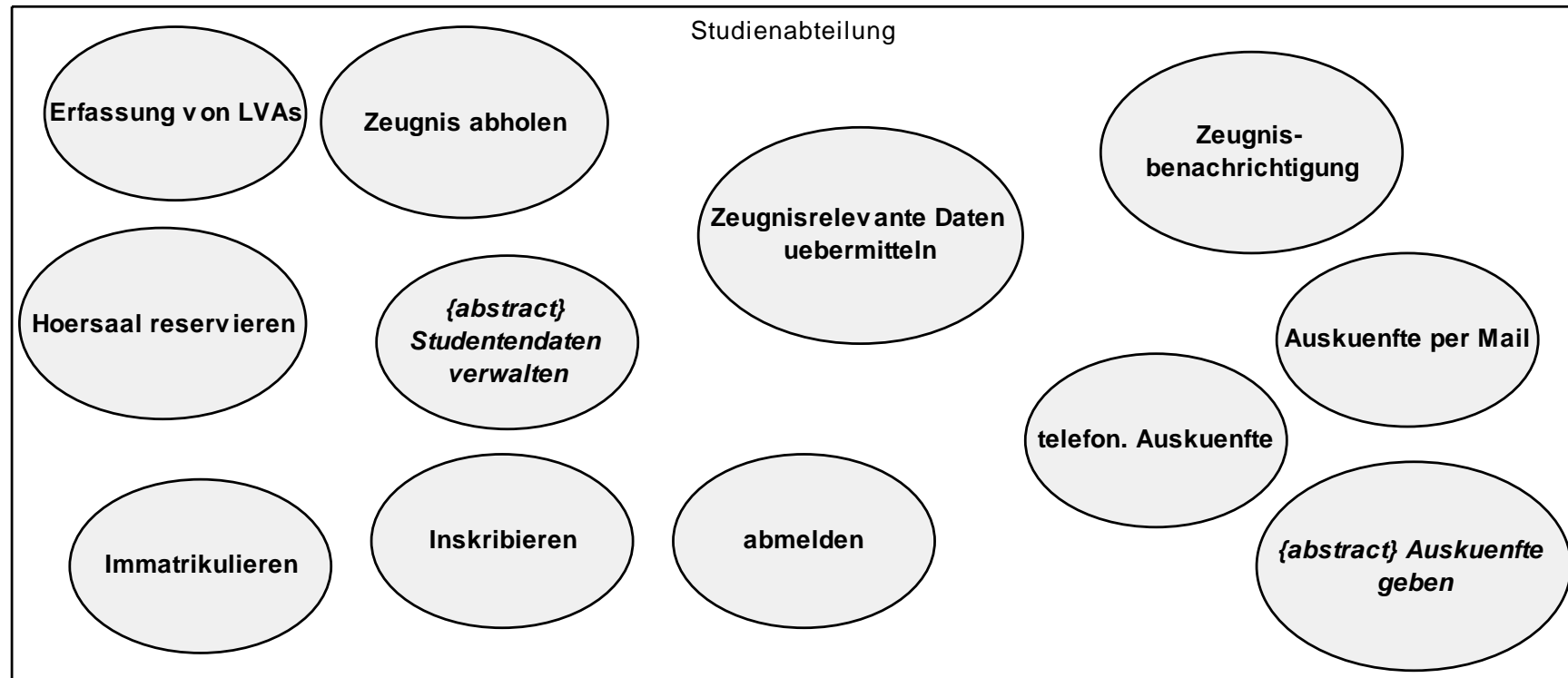
Bsp.: Anwendungsfälle (1/2)

Welche Funktionen bietet die Studienabteilung?

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. Studenten können hier **immatrikulieren** und **inskribieren**, sowie sich aber auch wieder vom Studium **abmelden**.
- Studenten **erhalten hier Zeugnisse**. Zeugnisrelevante Daten werden durch Lehrende an die **Studienabteilung übermittelt**. Die Studenten werden dann automatisch durch das Benachrichtigungssystem **informiert**.
- Es wird zwischen 2 Arten von Mitarbeitern unterschieden: a) solche, die sich ausschließlich mit der **Verwaltung von Studentendaten** befassen (M1) und b) jene, die alle restlichen Aufgaben erfüllen (M2), wobei aber alle Mitarbeiter (M1 und M2) **Auskünfte geben (per Telefon oder per Mail)**.
- M2-Mitarbeiter **stellen Zeugnisse aus**, sobald der/die Studierende diese abholt. Weiters werden von M2-Mitarbeitern **Lehrveranstaltungen erfasst**. Bei der Erfassung einer Lehrveranstaltung kann gleich ein **Hörsaal reserviert** werden.

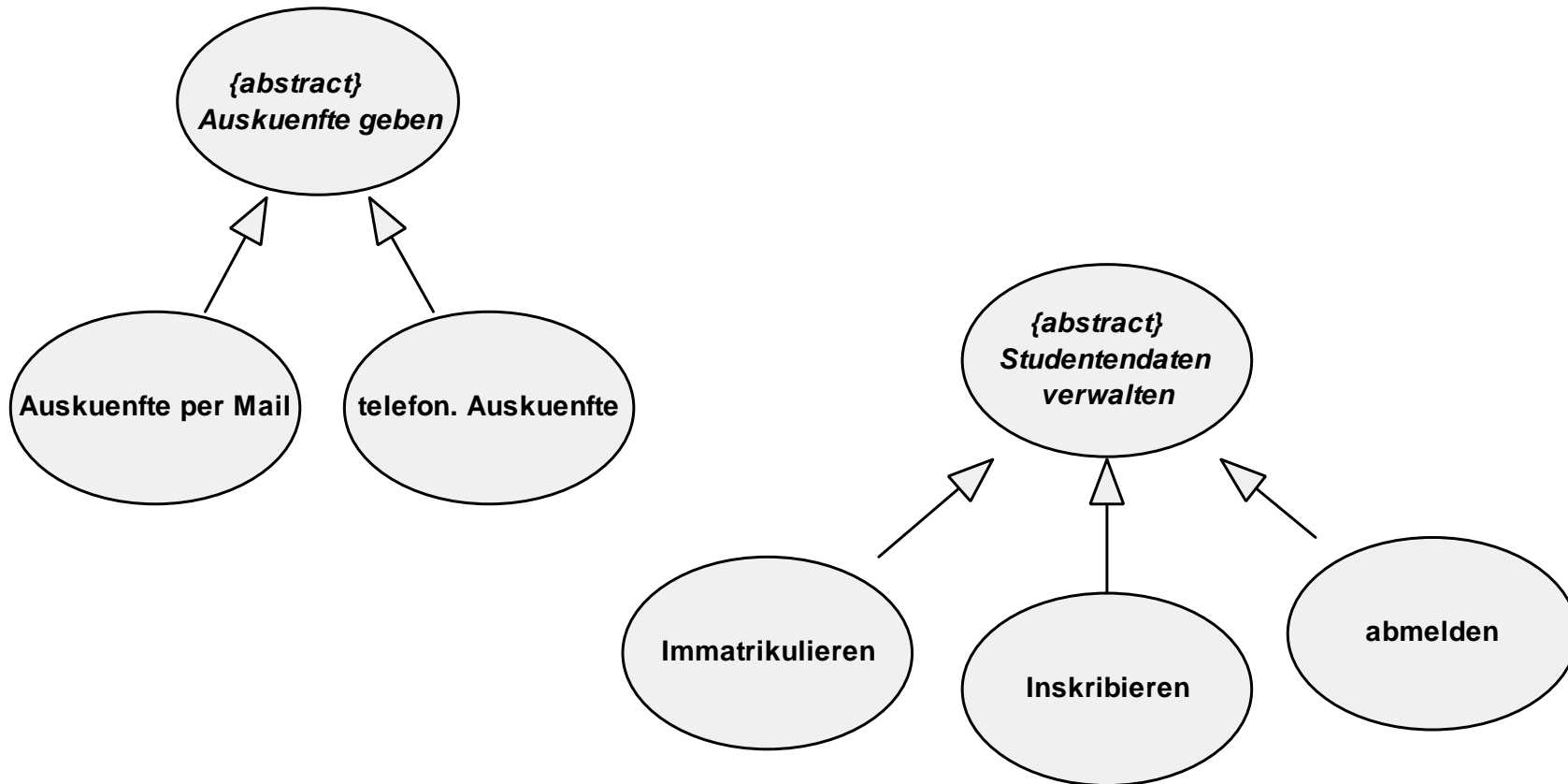
Bsp.: Anwendungsfälle (2/2)

- Welche Funktionen bietet die Studienabteilung?



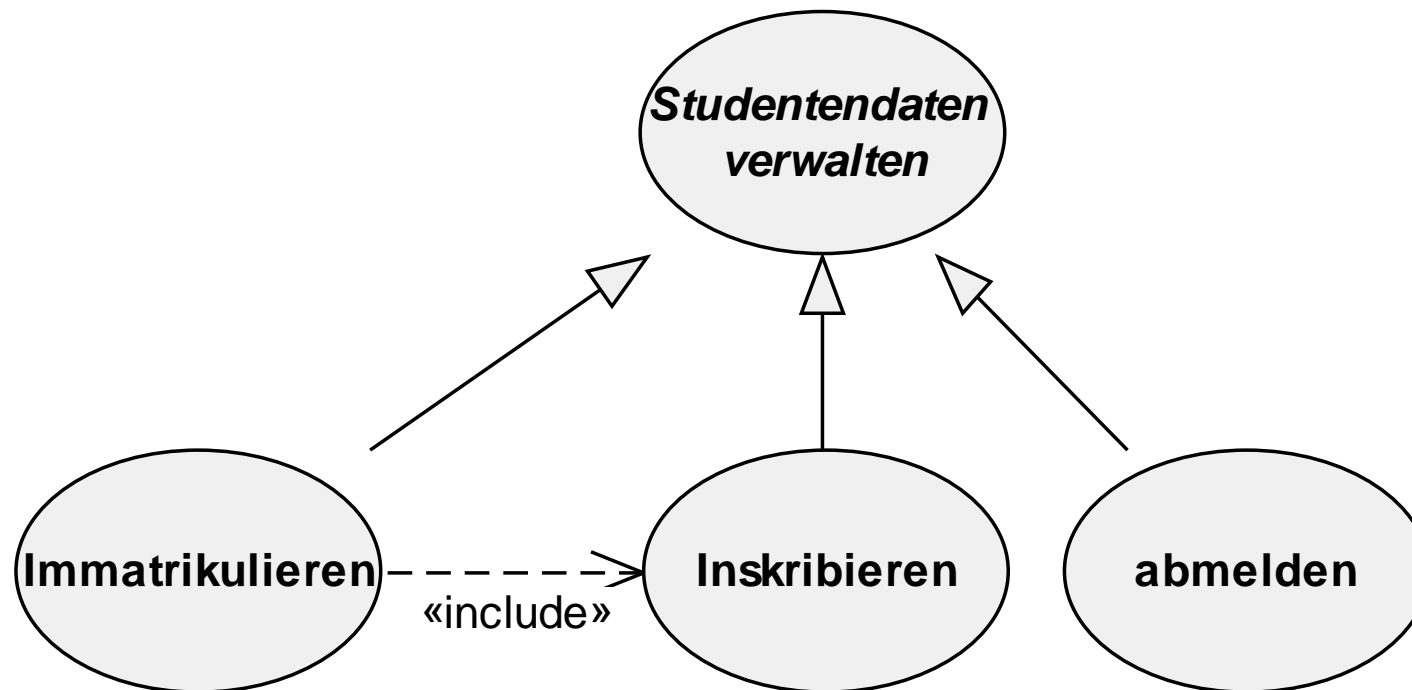
Bsp.: Generalisierungen

- Welche Anwendungsfälle können zusammengefasst werden?



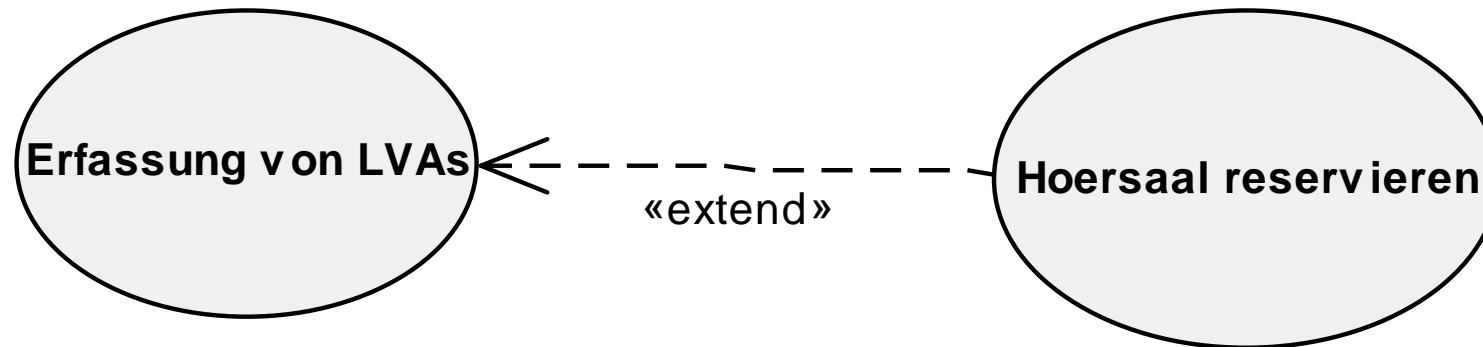
Bsp.: <<include>>

- Welche Anwendungsfälle sind Bestandteil eines anderen Anwendungsfalls?



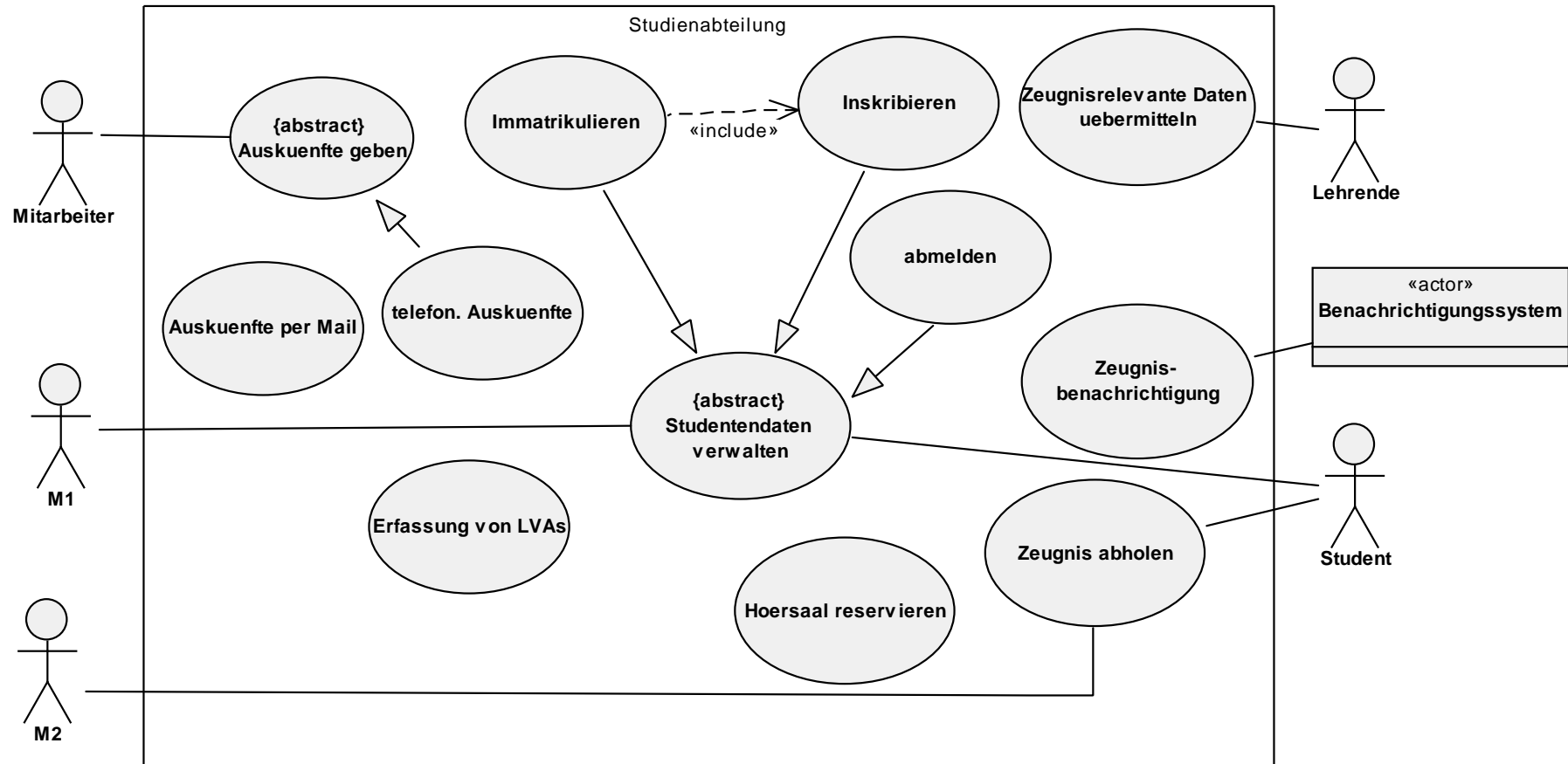
Bsp.: <<extend>>

- Welche Anwendungsfälle sind optional Bestandteile eines anderen?



Bsp.: Assoziationen

- Welche Akteure benutzen welche Anwendungsfälle?



Bsp.: Beschreibung (1/2)

- **Name:** Zeugnis abholen
- **Kurzbeschreibung:** Wenn ein Student eine Prüfung abgelegt hat, kann er sich sein Zeugnis in der Studienabteilung abholen.
- **Vorbedingung:** Student hat LVA absolviert
- **Nachbedingung:** Zeugnis befindet sich in ausgedruckter Form beim Studenten
- **Fehlersituationen:** keine zeugnisrelevanten Daten übermittelt; das Zeugnis wurde bereits ausgestellt, der Studierende will kein kostenpflichtiges Duplikat
- **Systemzustand im Fehlerfall:** Student hat kein Zeugnis
- **Akteure, die mit dem Anwendungsfall kommunizieren:** Student, M2
- **Trigger:** Wunsch nach Zeugnis

Bsp.: Beschreibung (2/2)

- **Standardablauf:**

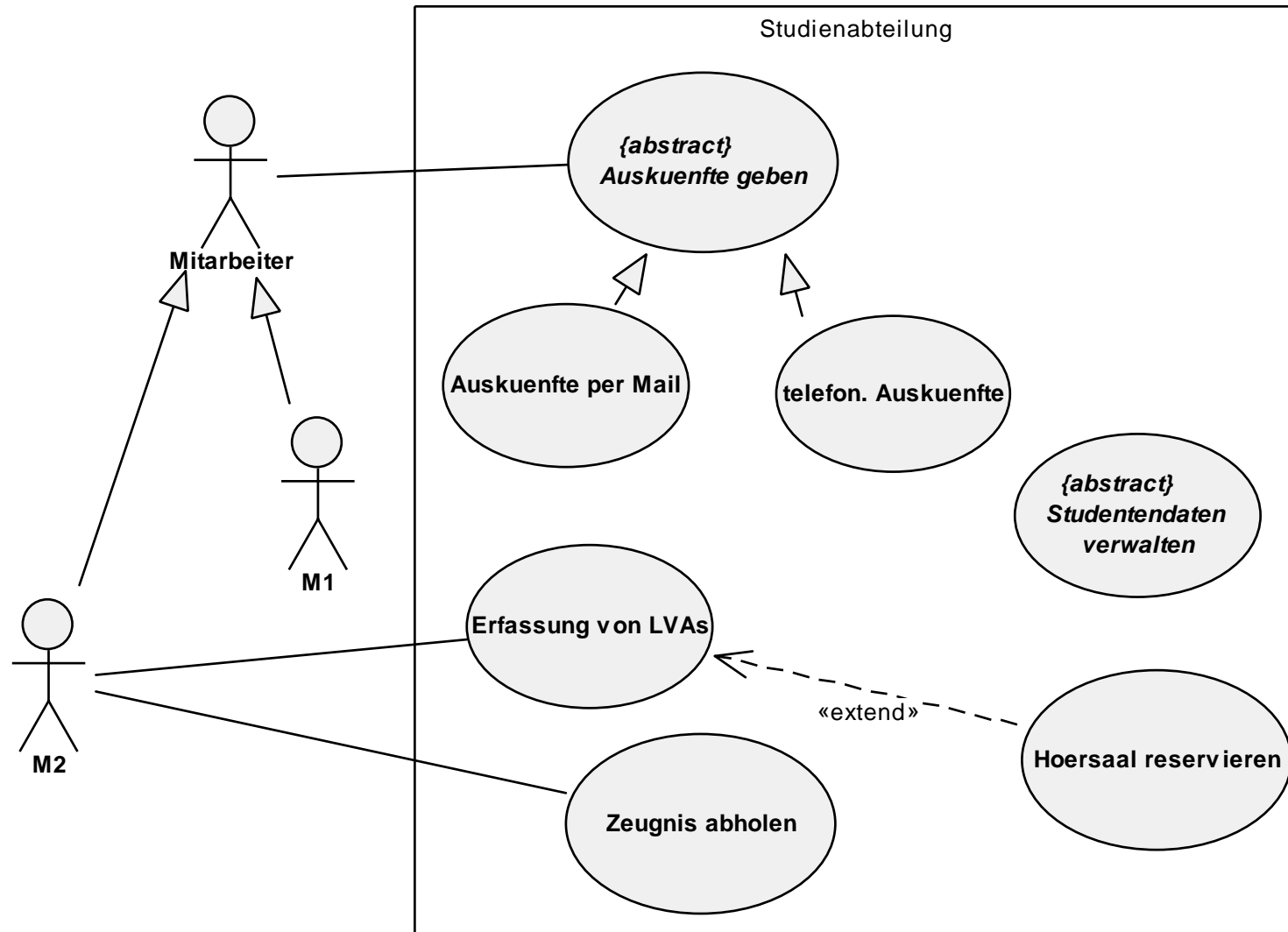
- (1) Student geht in Studienabteilung.
- (2) Student weist sich aus.
- (3) Mitarbeiter überprüft, ob das Zeugnis vorhanden ist.
- (4) Mitarbeiter überprüft, ob das Zeugnis bereits ausgedruckt wurde.
- (5) Mitarbeiter druckt das Zeugnis aus.
- (6) Student geht mit dem Zeugnis wieder.

- **Alternativabläufe:**

- (5'') Das Zeugnis wurde schon einmal ausgestellt – Mitarbeiter teilt Student mit, dass es gegen Kostenersatz noch einmal ausgestellt werden kann.
- (6'') Student geht in die Quästur und bezahlt für das Zeugnis.
- (7'') Student bekommt einen Beleg.
- (8'') Student geht mit dem Beleg in die Studienabteilung und ein Mitarbeiter druckt das Zeugnis aus.
- (9'') Student geht mit dem Zeugnis wieder.

Bsp.: Modellierungstipps (1/2)

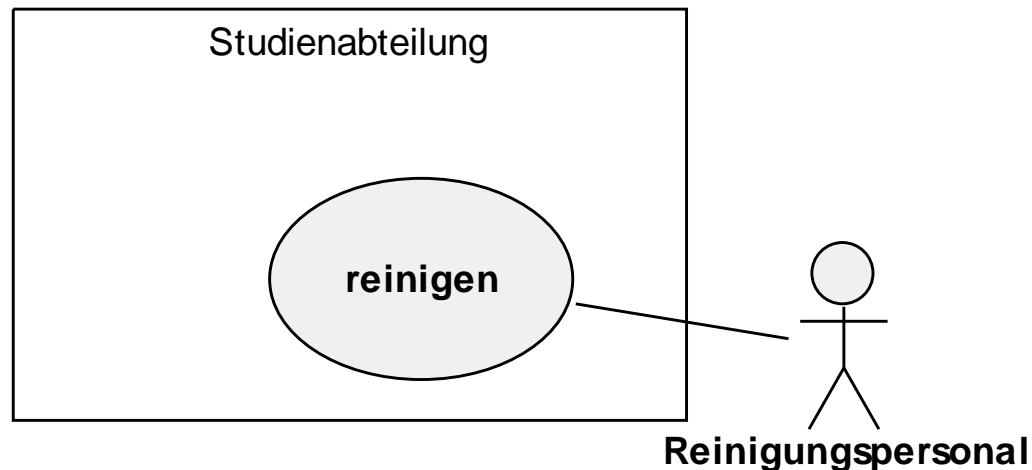
- Die Anwendungsfälle sollten auf der gleichen Abstraktionsebene sein.



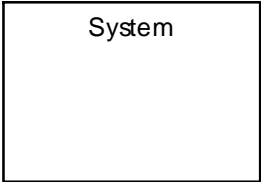
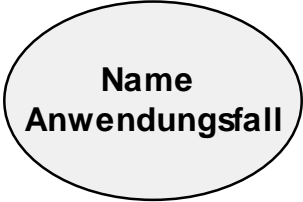

Bsp.: Modellierungstipps (2/2)

- Es sollen nur Anwendungsfälle und Akteure inkludiert werden, die relevant sind.
- Natürlich gibt es noch andere Personen, die etwas mit der Studienabteilung zu tun haben, die aber in diesem Kontext der Modellierung vernachlässigbar sind.




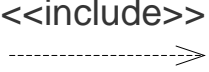
- Beispiel:



Anwendungsfalldiagramm - Elemente (1/2)

Name	Syntax	Beschreibung
Systemgrenze		Grenze zw. dem eigentlichen System und den Benutzern des Systems
Anwendungsfall		vom System erwartetes Verhalten
Akteur		Rolle der Systembenutzer

Anwendungsfalldiagramm - Elemente (2/2)

Name	Syntax	Beschreibung
Assoziation		Beziehung zwischen Anwendungsfällen und Akteuren
Generalisierung		Vererbungsbeziehung von Anwendungsfällen und Akteuren
extend		<i>A extends B</i> : opt. Verwenden von Anwendungsfall A durch Anwendungsfall B
include		<i>A includes B</i> : notw. Verwenden von Anwendungsfall B durch Anwendungsfall A

Zusammenfassung

- Sie haben diese Lektion verstanden, wenn Sie wissen ...
- dass mit dem Anwendungsfalldiagramm das Verhalten eines Systems aus der Sicht der Benutzer beschrieben wird.
- dass an einem Anwendungsfalldiagramm Akteure und Anwendungsfälle beteiligt sind.
- dass bei einem Anwendungsfalldiagramm die Grenzen des Systems klar definiert sein müssen und sich die Akteure immer außerhalb des Systems befinden.
- wie das Zusammenspiel zwischen Akteuren und Anwendungsfällen aussieht.
- dass es sowohl für Anwendungsfälle als auch für Akteure Generalisierungsbeziehungen gibt.
- worin der Unterschied zwischen `<<include>>` und `<<extend>>` besteht.

