

Objektorientierte Modellierung

Prof. Dr. Roland Dietrich

2. Anforderungsanalyse mit UML

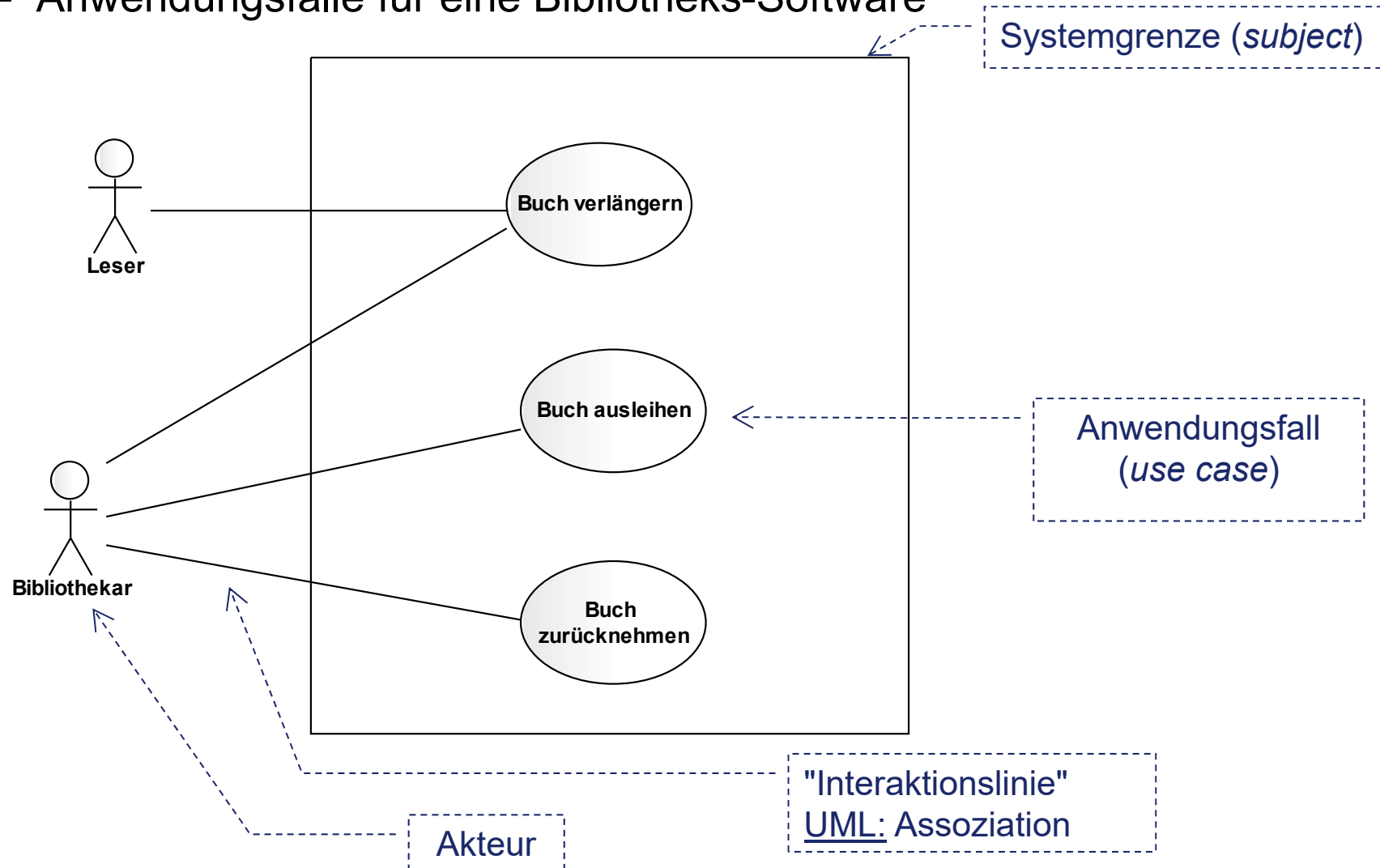
- **Anwendungsfälle
(Use Cases)**
- **Methodik und Qualitätskriterien**

1. Objektorientierte Softwareentwicklung ✓
2. Anforderungsanalyse mit UML
 - Anwendungsfalldiagramme
3. Statische Modellierung mit UML
 - Klassendiagramme
Objekte und Klassen, Assoziationen, Vererbung
 - Paketdiagramme
4. Der Analyseprozess und Analysemuster
2. Dynamische Modellierung mit UML
 - Interaktionsdiagramme (Sequenz- und Kollaborationsdiagramme)
 - Aktivitätsdiagramme
 - Zustandsautomaten
5. Entwurf mit UML
6. Implementierung in C++

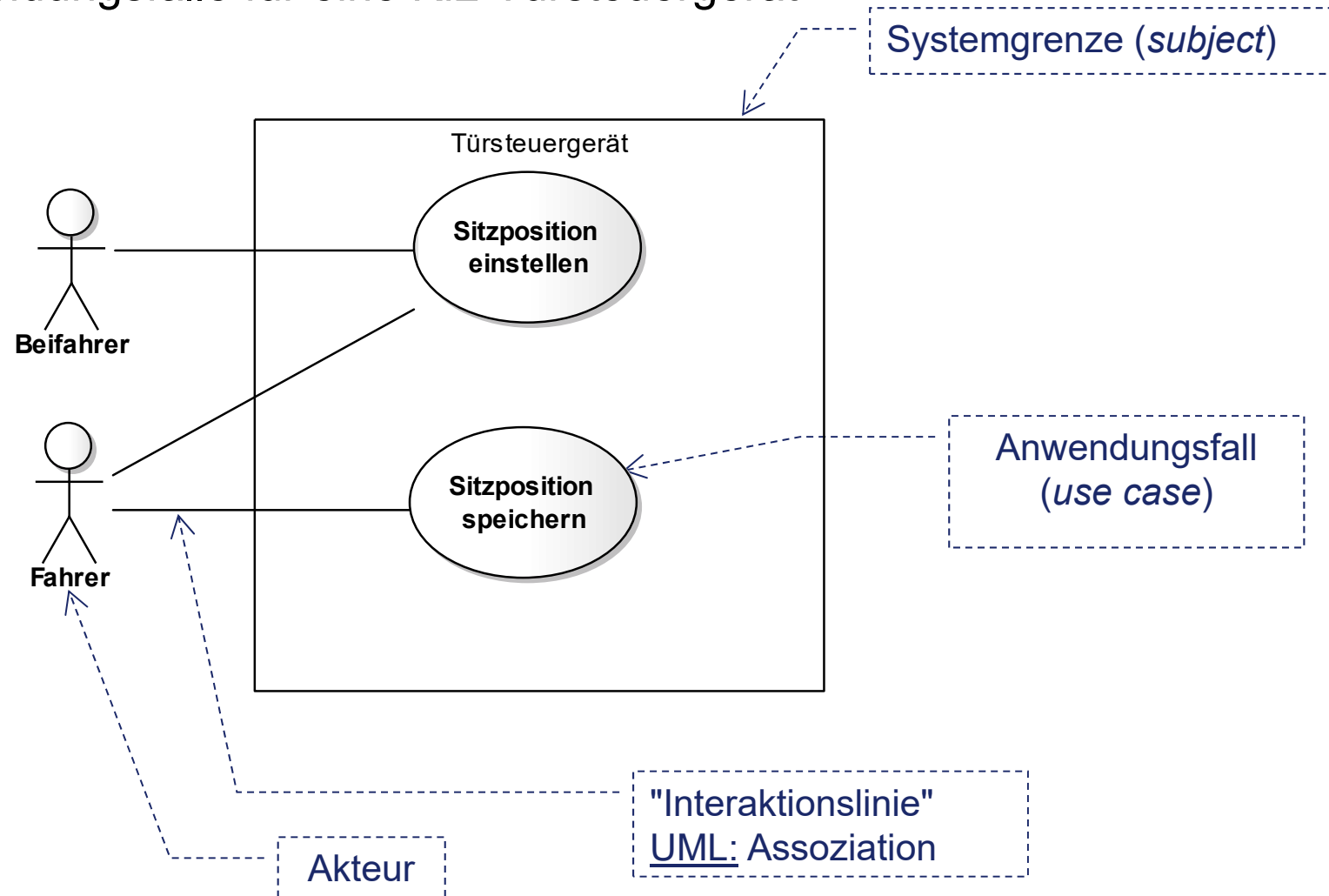
- Ziel der Softwareerstellung: Unterstützung von **Geschäftsprozessen** in **Unternehmen**
 - Sequenz von **unternehmensinternen** Aktivitäten
 - Ausgelöst von an den Prozessen beteiligten Personen (z.B. Kunden)
 - Werden durchgeführt, um ein bestimmtes Ziel zu erreichen
- Aufgabe der (Anforderungs-) Analyse
 - Festlegen, welche Aufgaben durch Software abgedeckt werden oder ob auch organisatorische Schritte enthalten sind
 - Welche Geschäftsprozesse sollen wie durch die Software unterstützt werden?
 - **Anwendungsfälle**
- Analog: Technische Prozesse
 - Sequenz von Steuerungs-Aktivitäten für Technische Geräte
 - Ausgelöst von „Signalen“ aus der Umgebung

- Werden zur Beschreibung der **funktionalen Systemanforderungen** eingesetzt
 - *Was soll mein System leisten?* → **Anwendungsfall** (Use Case)
 - *Wer ist an dieser Leistung beteiligt?* → **Akteur** (Actor)
 - beschreiben eine "Black-Box"-Sicht auf das System von außen
 - Anwendungsfälle werden in einem **System** (Systemgrenze, *Subject*) zusammengefasst
 - Die Akteure befinden sich außerhalb des Systems
- Anwendungsfalldiagramm
 - Zeigt Anwendungsfälle, beteiligte Akteure und (optional) das System
 - Zeigt Beziehungen
 - Anwendungsfall \leftrightarrow Anwendungsfall
 - Anwendungsfall \leftrightarrow Akteur
 - Akteur \leftrightarrow Akteur
 - Diagrammkopf: **uc** <Diagrammname>

- Beispiel
 - Anwendungsfälle für eine Bibliotheks-Software



- Beispiel
 - Anwendungsfälle für eine Kfz-Türsteuergerät



- Definition [UML]

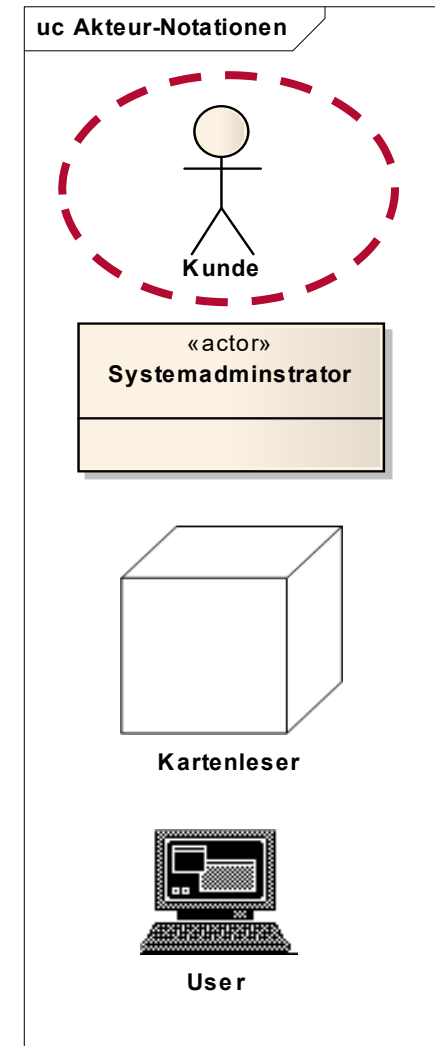
*An **Actor** models a type of role played by an entity that interacts with the subjects of its associated UseCases (e.g., by exchanging signals and data). Actors may represent roles played by human users, external hardware, or other systems.*

- Ein Akteur ist eine **Rolle**, die ein Benutzer des Systems spielt
- Hat einen Einfluss auf das System
- Ist häufig eine Person
- Kann auch eine Organisationseinheit oder ein anderes System sein
- Befindet sich immer **außerhalb** des Systems und interagiert mit dem System

- Beispiele:

- Der Leser, der Bücher ausleiht (nicht „Student“)
- Der „Kundensachbearbeiter“ (nicht der „Mitarbeiter“)
- Der „Beifahrer“, der „Fahrer“ (nicht der „Passagier“)

- Merkmale des Modellelements „Akteur“
 - Akteure haben einen Namen
 - Können über Assoziationen ("Interaktionslinien") verbunden sein mit
 - Anwendungsfällen
→ Der Akteur interagiert mit dem Anwendungsfall
 - *Komponenten*
 - *Klassen*
- Notation
 - Normalfall:
 - „Strichmännchen“
 - Alternativen
 - Klasse mit Stereotyp <<Actor>>
 - Würfel (insbes. bei Fremdsystemen)
 - frei wählbares Symbol



- Definition [UML]

*A **use case** ...specifies a set of behaviors performed by that subject, which yields an observable result that is of value for Actors or other stakeholders of the subject.*

- Ein **Anwendungsfall (use case)** besteht aus mehreren zusammenhängenden Aktivitäten, die von einem System durchgeführt werden, um ein Ziel zu erreichen bzw. ein messbares Ergebnis zu produzieren
- Ein Anwendungsfall wird stets durch einen Akteur ausgelöst

- Beispiele:

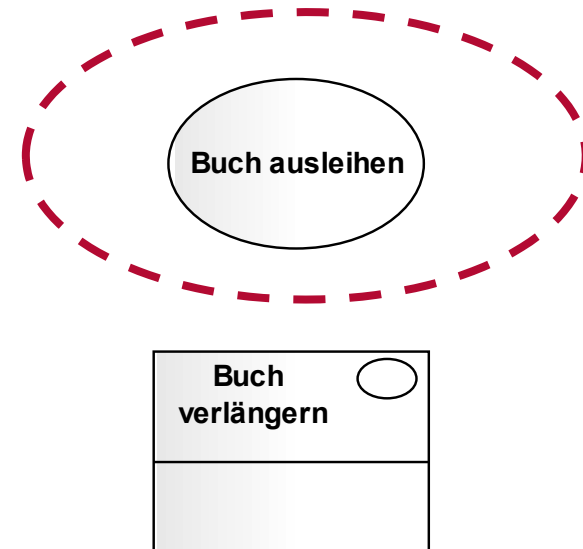
Anwendungsfälle

eine online-Überweisung durchführen
einen Wareneingang bearbeiten
ein Buch ausleihen
ein Buch verlängern
Fenster öffnen

Akteure

Bankkunde
Lagerarbeiter
Bibliothekar(in)
Leser, Bibliothekar(in)
Fahrer

- Merkmale des Modellelements „Anwendungsfall“
 - Ein Anwendungsfall **muss** einen **Namen** haben
 - Ein Anwendungsfall kann einem **System** (*subject*) zugeordnet sein
 - Ein Anwendungsfall kann an binären **Assoziationen** mit Aktueren beteiligt sein
 - Der Akteur interagiert mit dem Anwendungsfall
 - Kann an *includes*-, *extends*- und Generalisierungs-Beziehungen beteiligt sein
 - Kann Erweiterungspunkte besitzen
- Notation
 - Normalfall:
 - Ellipse
 - Alternative:
 - Klasse mit Use-Case-Symbol



- Definition [UML]:

*A **subject** of a use case could be a system or any other element that may have behavior....*

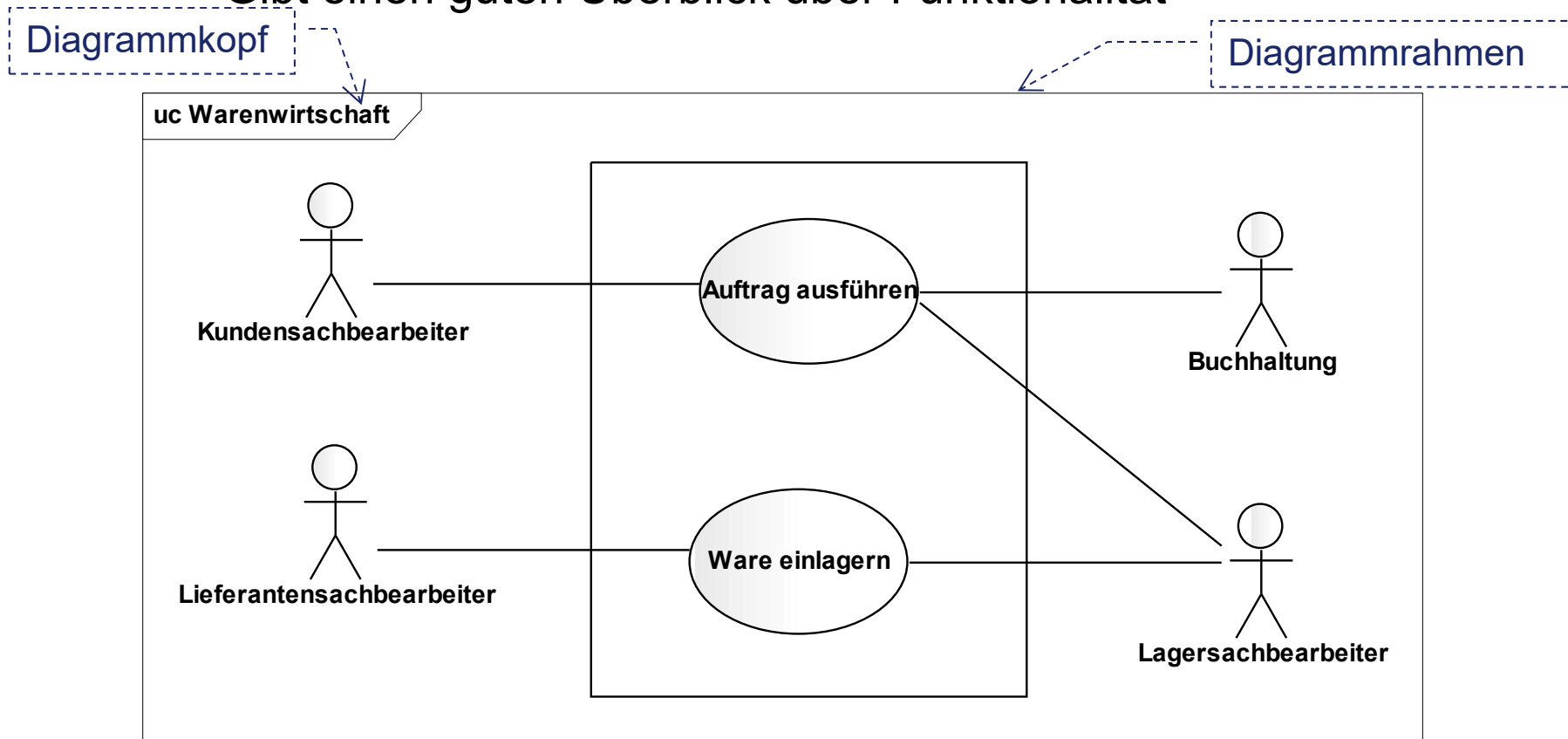
Each use case specifies a unit of useful functionality that the subject provides to its users.

- Das **System** realisiert die Anwendungsfälle, die ihm zugeordnet sind
- Anwendungsfälle **müssen nicht** einem System zugeordnet werden!

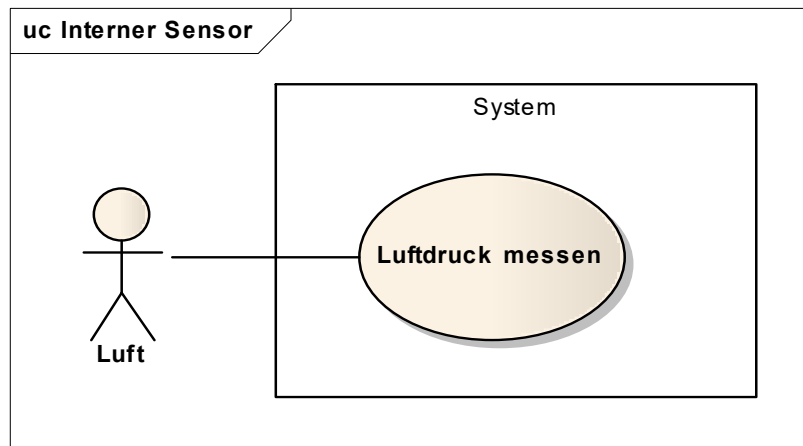
- Anmerkungen:

- Das *subject* eines Anwendungsfalls ist laut UML-Spezifikation ein **Classifier**
 - Use Cases können auch anderen Modellelementen zugeordnet werden
Klassen, Interfaces, Packages, Komponenten,...
- Die "Systemgrenze" in UML-Diagrammen abstrahiert von der konkreten Realisierung
 - ist während der Anforderungsanalyse sinnvoll

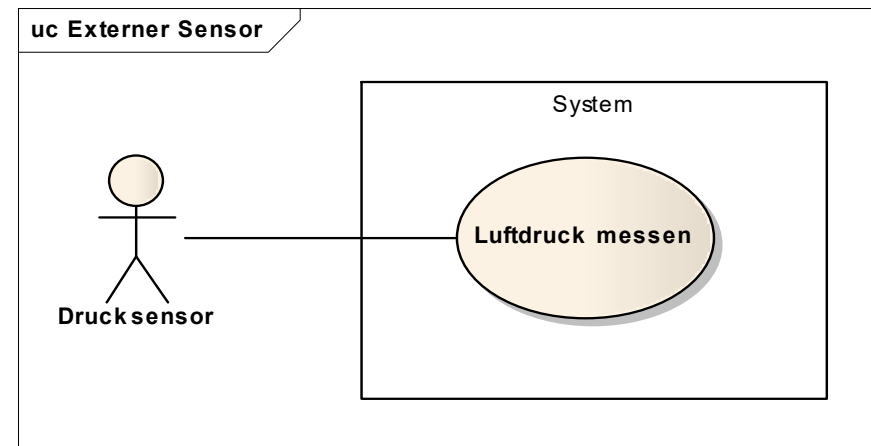
- Anwendungsfalldiagramm (*use case diagram*)
 - Modelliert Akteure, Use-Cases und die Systemgrenze (optional)
 - Modelliert Beziehungen zwischen Use-Cases
 - Gibt einen guten Überblick über Funktionalität



- Anwendungsfalldiagramme klären die Systemgrenzen
 - Akteure stehen **außerhalb des Systems**
 - Beispiel:
 - Gehört ein Sensor zu einem System, oder steht er außerhalb?

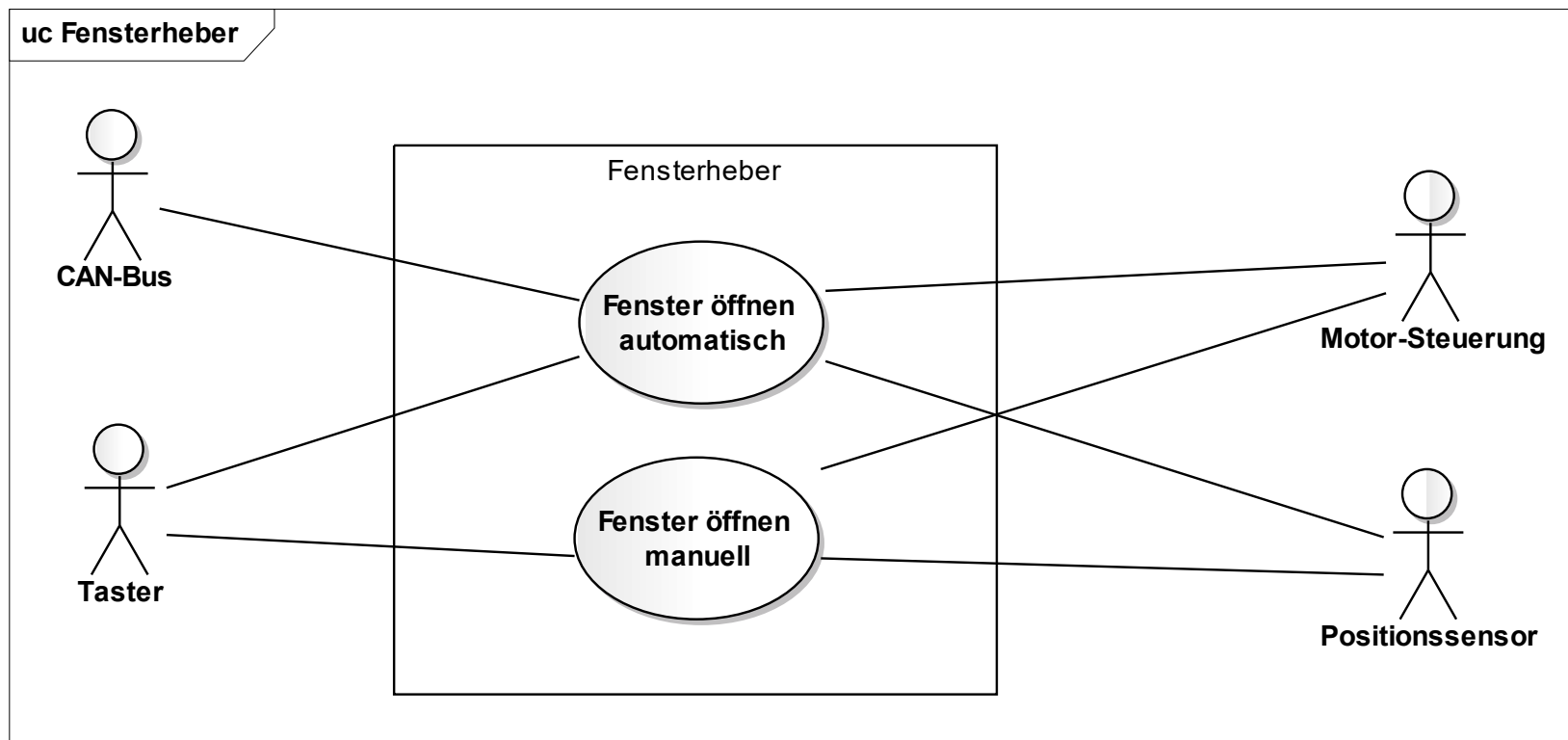


Zum System gehört ein Druckmesser
→ Akteur ist die Luft, die gemessen wird und Eingaben an den Druckmesser liefert.



Der Druckmesser gehört nicht zum System
→ Er ist Akteur und liefert die "Eingaben" an das System

- Anwendungsfalldiagramme klären die Systemgrenzen
 - Teilsystem „Fensterheber“ des Türsteuergeräts
 - Hier wird nur die Steuerungssoftware als System betrachtet
 - Die technischen Einrichtungen gehören nicht zum zu entwickelnden System



- Anwendungsfall-Beschreibungen
 - Textuell mit Strukturvorgaben (**Anwendungsfallschablone**)
 - Typische Elemente einer textuellen Anwendungsfallbeschreibung
 - Name
 - Ziel
 - beteiligte Akteure
 - Vorbedingungen, Nachbedingungen (Erfolg/Fehlschlag)
 - Beschreibung des Ablaufs
 - Grafisch durch andere Diagramme
 - Aktivitätsdiagramme
 - Sequenzdiagramme
 - Zustandsautomaten
 - Typisch
 - Es werden einzelne **Szenarien** für einen Anwendungsfall beschrieben
 - Abläufe, die unter ganz bestimmten Bedingungen eintreten
 - » Zunächst wird das "Normalverhalten" beschrieben
 - » Zusätzlich können "Sonderfälle" (alternative Abläufe, Fehlerfälle, Erweiterungen) beschrieben werden

- **Anwendungsfallschablone** (*use case template*) [Balzert 05]
 - **Anwendungsfall:** Name des Anwendungsfalls, evtl. Nr.
 - **Ziel:** globale Zielsetzung bei erfolgreicher Ausführung des Anwendungsfalls
 - **Kategorie:**
 - primär (notwendig, häufig benötigt)
 - sekundär (notwendig, selten benötigt)
 - optional (nützlich, nicht unbedingt notwendig)
 - **Vorbedingung:** erwarteter Zustand, bevor der Anwendungsfall beginnt
 - **Nachbedingung Erfolg:** erwarteter Zustand nach „erfolgreicher“ Ausführung des Anwendungsfalls
 - **Nachbedingung Fehlschlag:** erwarteter Zustand nach „fehlgeschlagener“ Ausführung des Anwendungsfalls
 - **Akteure**

- **Anwendungsfallschablone (Forts.)**
 - **Auslösendes Ereignis**
 - **Beschreibung:** Hier wird der Standardablauf beschrieben
 - 1 Erste Aktion
 - 2 Zweite Aktion
 - **Erweiterungen:**
 - 1a Erweiterung des Funktionsumfangs der ersten Aktion
 - 2a Erweiterung des Funktionsumfangs der zweiten Aktion
 - **Alternativen:**
 - 1a erste alternative Ausführung der ersten Aktion
 - 1b zweite alternative Ausführung der ersten Aktion
 - 2a erste alternative Ausführung der zweiten Aktion

- **Beispiel:**

AF1: Auftrag ausführen

- **Ziel:** Ware an Kunden geliefert
- **Vorbedingung:** -
- **Nachbedingung Erfolg:** Ware ausgeliefert (auch Teillieferungen), Rechnungskopie bei Buchhaltung
- **Nachbedingung Fehlschlag:** Mitteilung an Kunden, dass nichts lieferbar
- **Akteure:** Kundensachbearbeiter, Lagersachbearbeiter, Buchhaltung
- **Auslösendes Ereignis:** Bestellung des Kunden geht ein

- **Beispiel (Forts.):**
 - **Beschreibung:**
 - 1 Kundendaten abrufen
 - 2 Lieferbarkeit prüfen
 - 3 Rechnung erstellen
 - 4 Auftrag vom Lager ausführen lassen
 - 5 Rechnungskopie an Buchhaltung geben
 - **Erweiterung:**
 - 1a Kundendaten aktualisieren
 - **Alternativen:**
 - 1a Neukunden erfassen
 - 3a Rechnung mit Nachnahme
 - 3b Rechnung mit Bankeinzug erstellen
 - 3c Meldung an Kunden, dass Ware nicht lieferbar
(→ 4 und 5 entfallen)

• Beispiel

Anwendungsfall: Sensor kalibrieren

Ziel: Drucksensor ist kalibriert;
das System zeigt dies durch
grüne LED an.

Akteure: Systemadministrator (primär),
Drucksensor (sekundär)

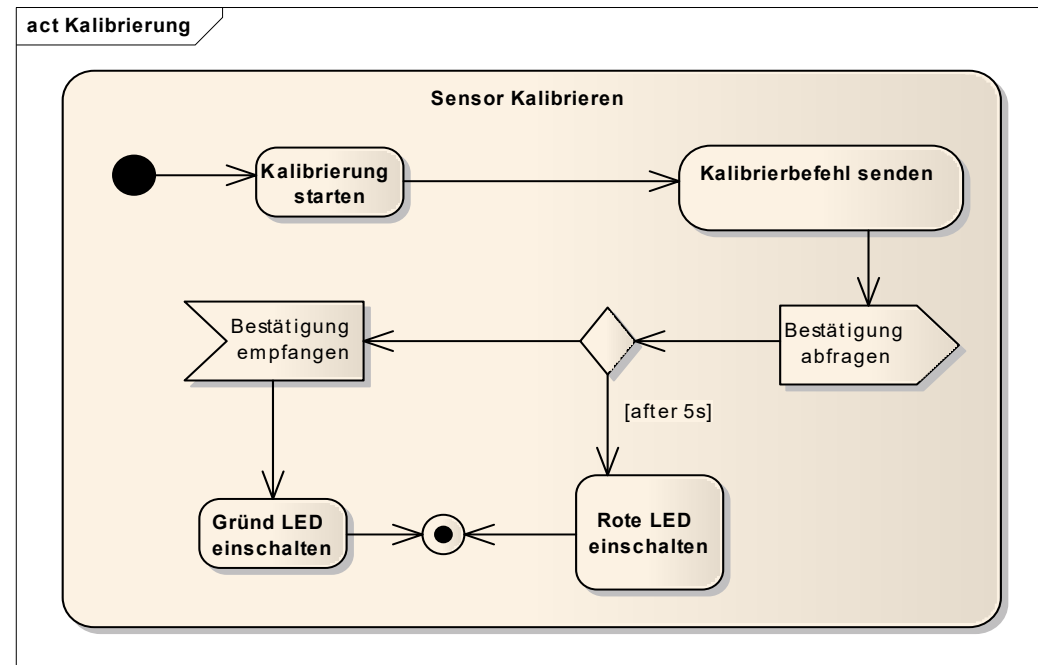
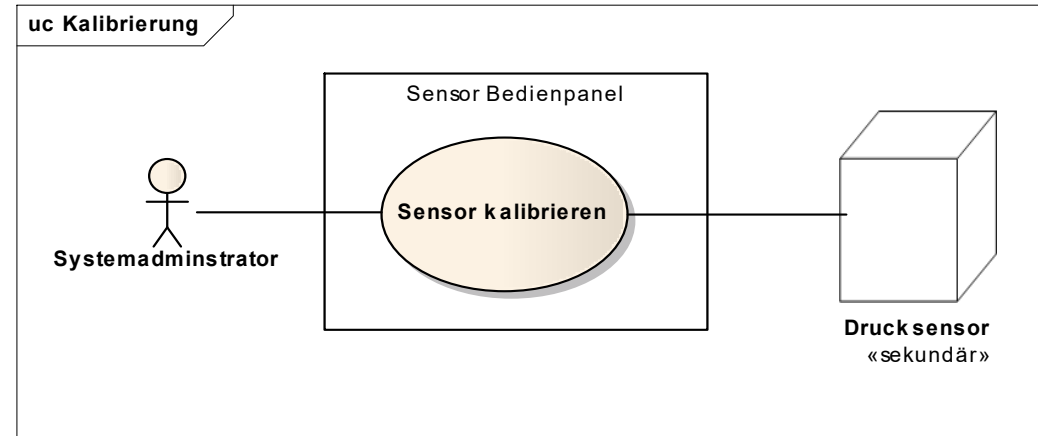
Vorbedingung:
Drucksensor aktiv

Beschreibung:

1. Systemadmin startet Kalibrierung
2. System sendet Kalibrierbefehl an den Drucksensor
3. Innerhalb 5 sec bestätigt der Drucksensor die Kalibrierung
4. Grüne LED wird eingeschaltet

Alternativen:

- 3a. 5 sec sind um und der Drucksensor hat nicht geantwortet
→ Rote LED einschalten



Aktivitätsdiagramm zur Beschreibung des Anwendungsfalls

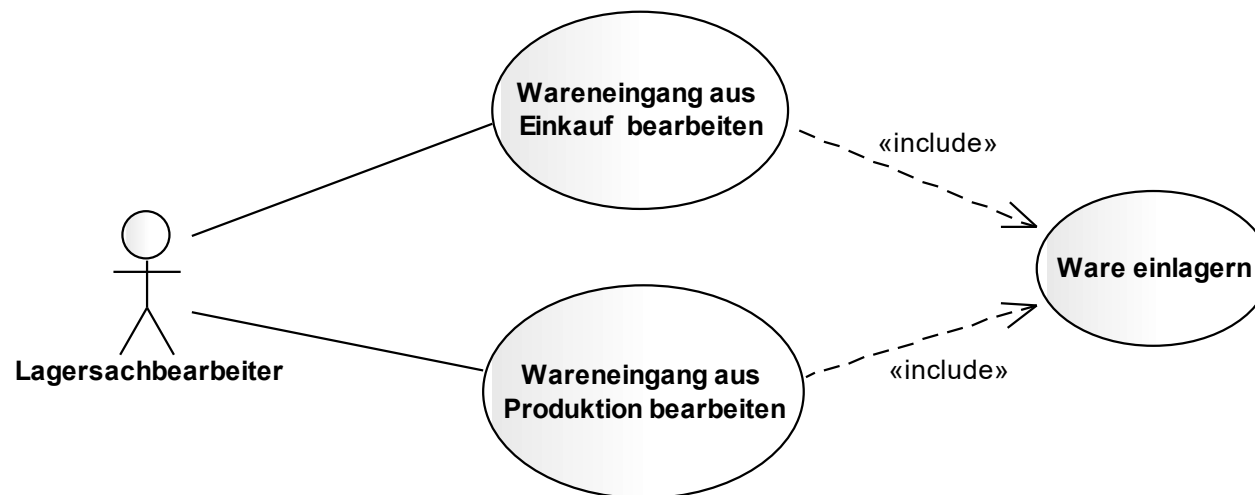
- Aufgabe:
 - Für eine Stadtbibliothek soll ein Softwaresystem entwickelt werden. Analysieren Sie die typischen Geschäftsprozesse zur Ausleihe und Verwaltung von Büchern und erstellen Sie ein Anwendungsfalldiagramm.
 - Beschreiben Sie die Anwendungsfälle nach der Anwendungsfallschablone

- Definition [UML]:

***Include** is a directed relationship between two use cases, indicating that the behavior of the included use case is inserted into the behavior of the including use case.*

- Anwendung:

- Beschreibung von gemeinsamem Verhalten, das in mehreren Anwendungsfällen vorkommt („*Unteranwendungsfall-Aufruf*“)
 - Der „Unteranwendungsfall“ muss nur einmal beschrieben werden



Beachte:

Der Akteur löst den Anwendungsfall „Ware einlagern“ hier nicht mehr direkt aus!

Die Beziehung ist in Pfeilrichtung zu lesen:

„Wareneingang aus Einkauf bearbeiten“ beinhaltet „Ware einlagern“

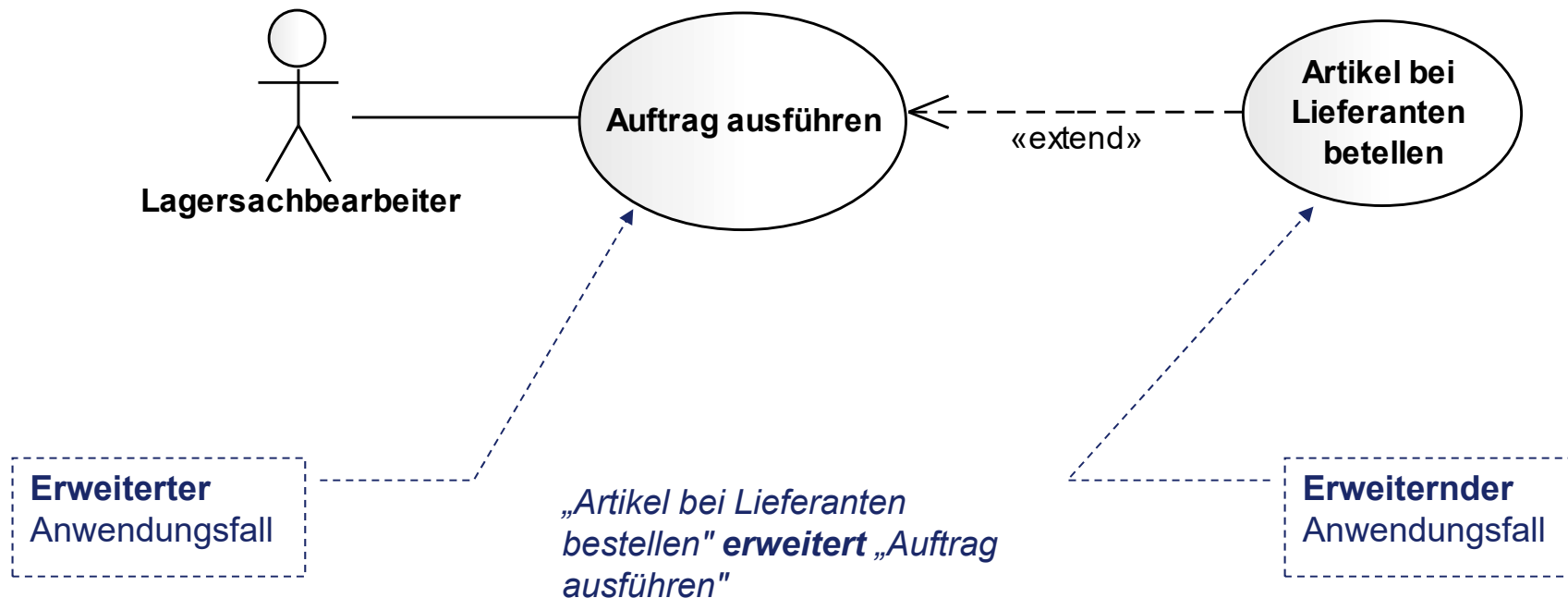
- Definition [UML]:

***Extend** is a relationship from an **extending use case** to an **extended use case** that specifies how and when the behavior defined in the extending use case can be inserted into the behavior defined in the extended use case.*

- Merkmale:

- Ein **erweiterter** Anwendungsfall
- Ein **erweiternder** Anwendungsfall
- Beliebige viele **Erweiterungspunkte**
 - müssen Erweiterungspunkte des erweiterten Anwendungsfalls sein
 - jeder Verhaltens-Abschnitt des erweiternden Anwendungsfalls ist einem Erweiterungspunkt zugeordnet.
- Optional eine **Bedingung**
 - muss erfüllt sein, dass bei Erreichen des ersten Erweiterungspunkts die Erweiterung ausgeführt wird

- Anwendung:
 - Beschreibung von Ausnahmesituationen und Sonderfällen im Ablauf von Anwendungsfällen
- Notation



Beziehungen: Extends

- Extends-Beziehung ermöglicht "exemplarische" Modellierung
 - Zunächst "Sonnenscheinszenarien" beschreiben
 - Erst später Ausnahmen und Fehlerfälle hinzufügen
- Extends-Beziehungen ermöglichen, Komplexität zu verbergen
- Erweiterte Anwendungsfälle müssen in sich vollständig sein
 - sie können unabhängig vom erweiternden ablaufen
- Erweiternde Anwendungsfälle müssen **nicht** alleine ablaufen können
 - sie sind meist nicht mit einem Akteur verbunden

- Definition [UML]:

*An **extension point** identifies a point in the behavior of a use case where that behavior can be extended by an extend relationship.*

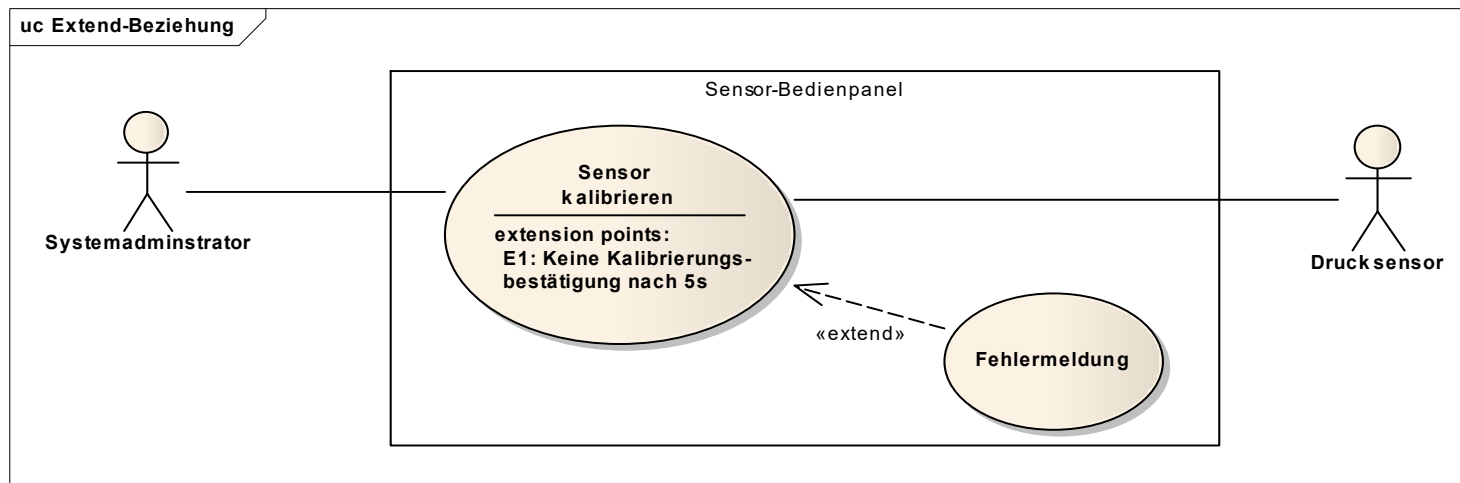
Each extension point has a unique name within a use case.

- Merkmale eines **Erweiterungspunkts**

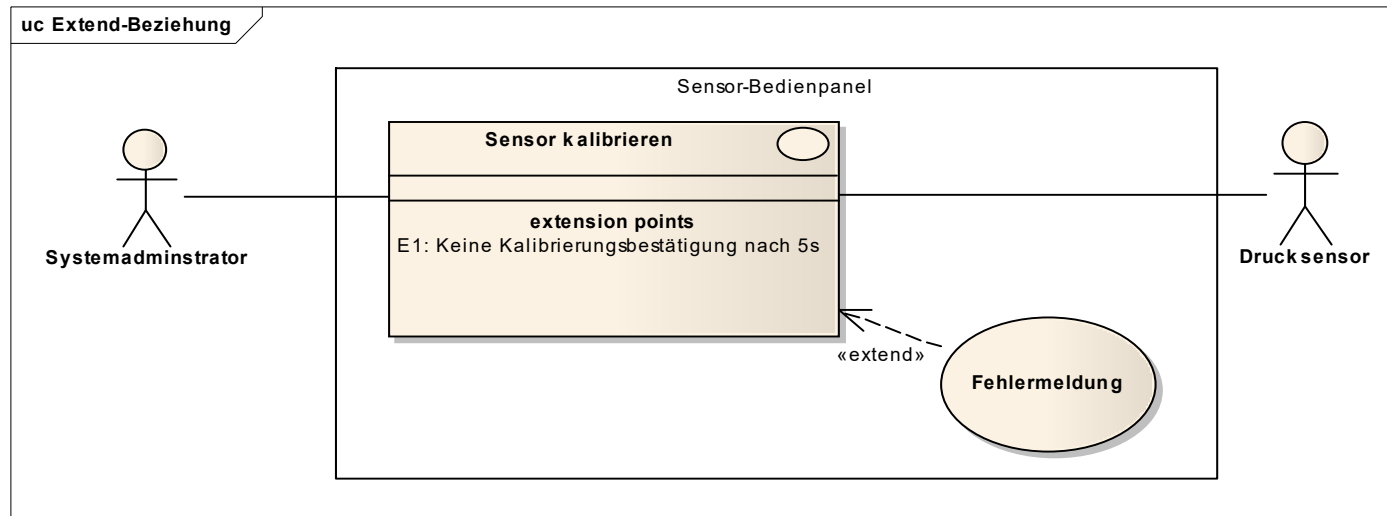
- Name + Beschreibung
- Keine Vorgaben für die Beschreibung

- Notation:

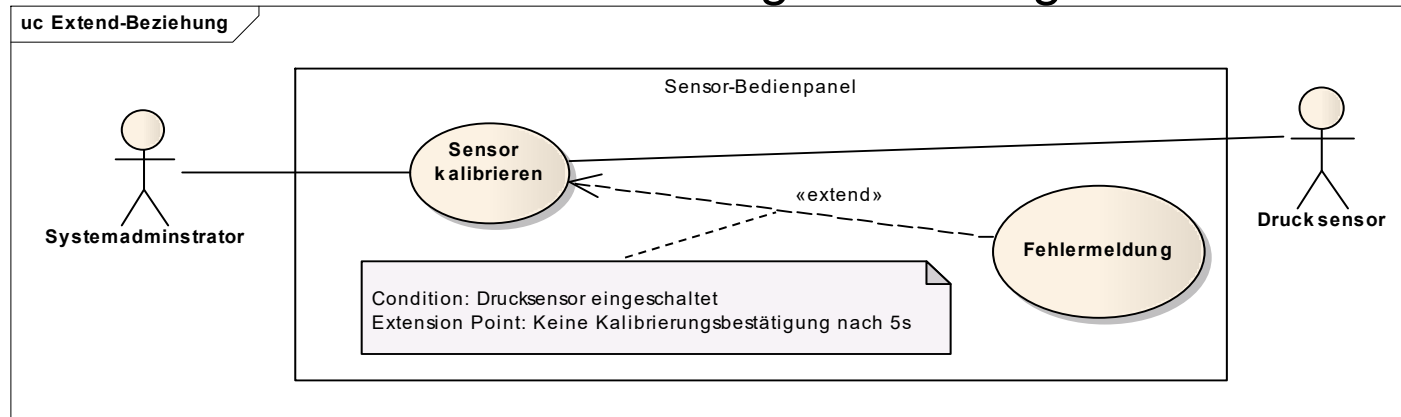
- In einer "Abteilung" (Compartment) des Anwendungsfalls



- Notation
 - in einer Abteilung in der Klassendarstellung

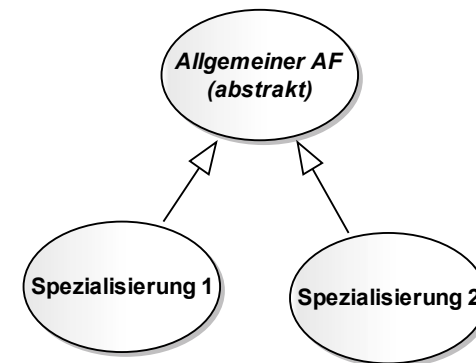


- in einer Notiz an der Erweiterungsbeziehung

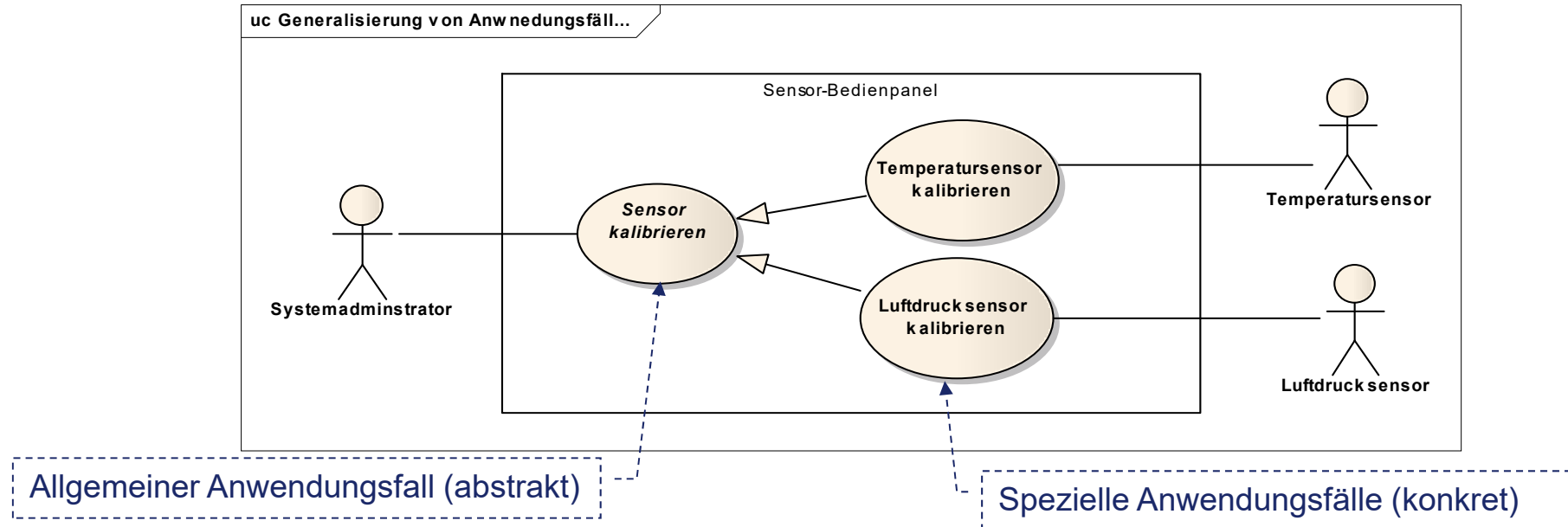


- **Generalisierung von Anwendungsfällen**

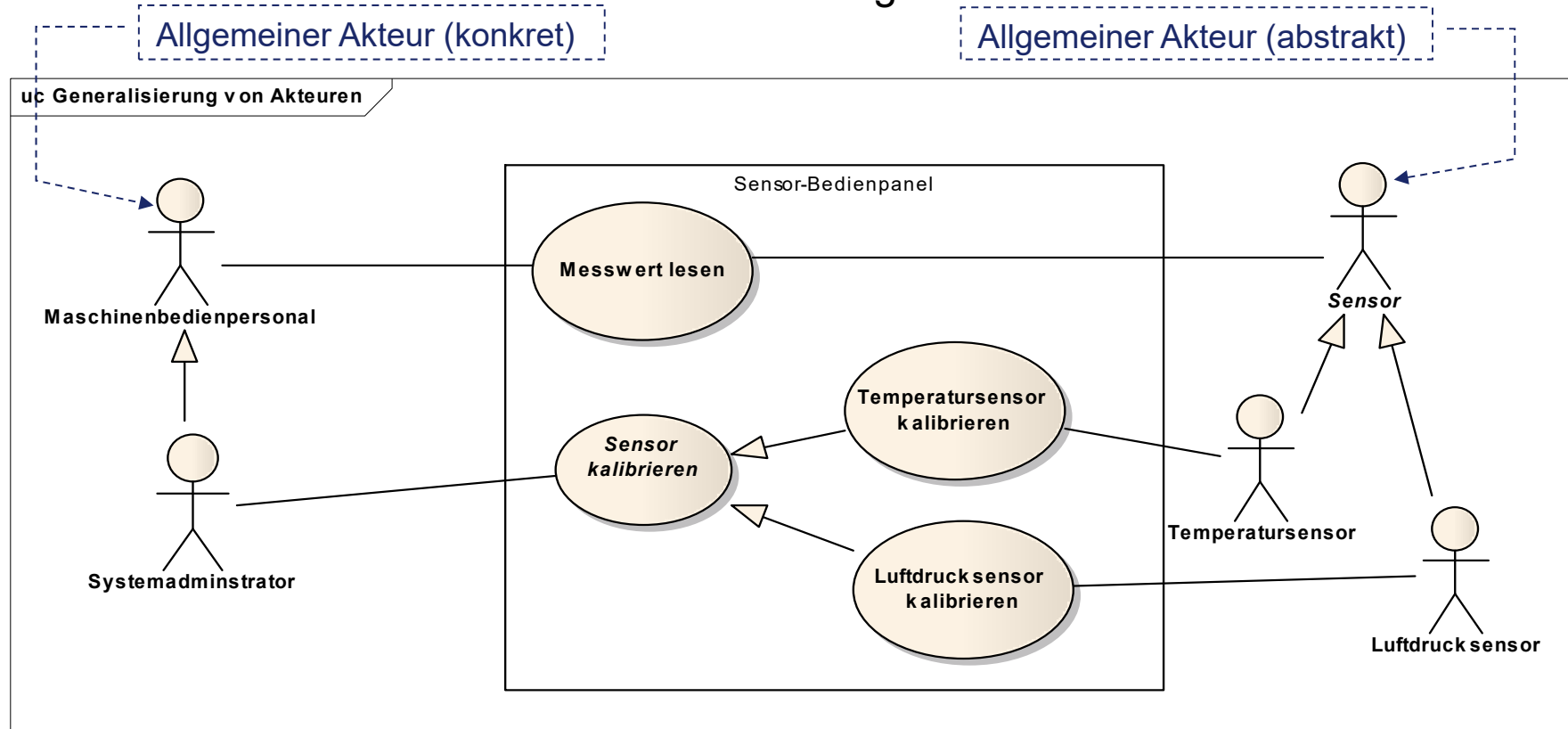
- "Ein Anwendungsfall ist ein Spezialfall eines anderen"
- Der allgemeine Anwendungsfall kann **abstrakt** sein (vgl. Klassendiagramme!)
 - Den allgemeine AF gibt es in der Realität so nicht
 - Der allgemeine AF wird stets als einer seiner Spezialisierungen ausgeführt
- Die Bedeutung von "Vererbung" ist nicht spezifiziert
 - **Nicht:** ein Anwendungsfall "erbt" das Verhalten eines anderen d.h. er läuft in Teilen genauso ab, wie der Allgemeinere
 - Das ist *includes*!
 - **Aber:** Die Akteure des allgemeinen AF sind auch Akteure des speziellen
- Notation
 - Abstrakter AF:
 - Namen wird *kursiv* geschrieben!



- Generalisierung von Anwendungsfällen
 - Beispiel
 - Sensor kalibrieren heißt **entweder** Temperatursensor kalibrieren **oder** Luftdrucksensor kalibrieren
 - » Generalisierungen beschreiben **oder**-Beziehungen!
 - Akteure beim AF *Temperatursensor kalibrieren*:
 - » Temperatursensor und Systemadministrator
 - Akteure beim AF *Luftdrucksensor kalibrieren*:
 - » Luftdrucksensor und Systemadministrator



- Generalisierung von Akteuren
 - Der spezielle Akteur erbt alle Eigenschaften des allgemeinen
 - Insbesondere Interaktionsbeziehungen

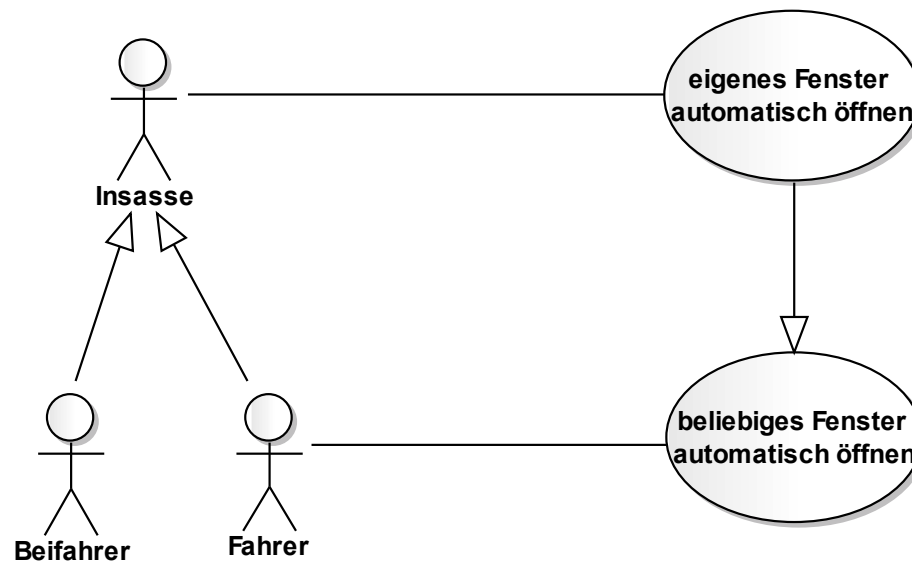


Der Systemadministrator kann Sensoren kalibrieren **und** Messwerte lesen

Der Temperatur- **und** der Luftdrucksensor interagieren mit dem AF "Messwerte lesen"

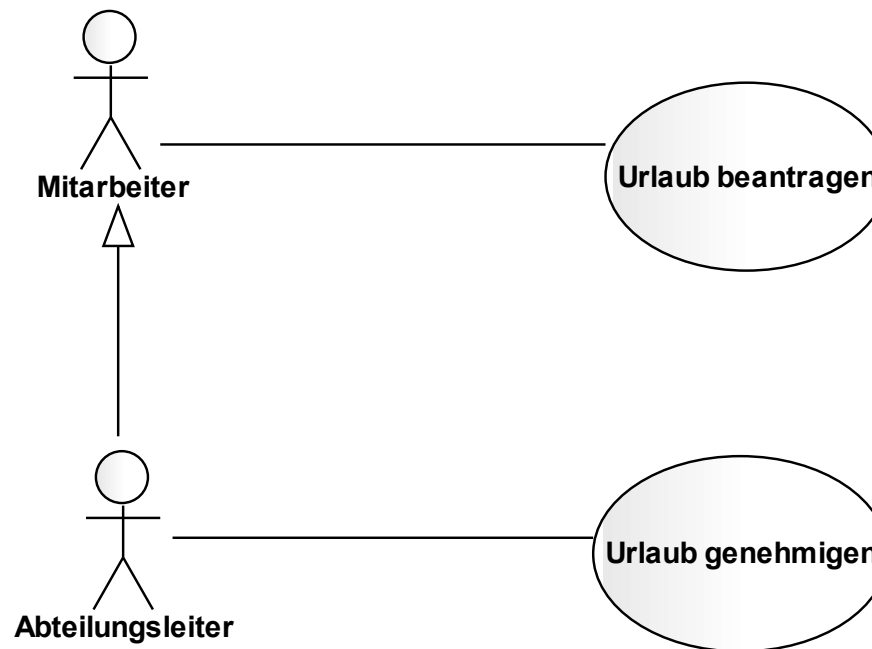
- **Beispiel**

- Alle Fahrzeug-Insassen können ihr eigenes Fenster öffnen
- Der Fahrer kann alle Fenster öffnen (insbesondere sein eigenes!)



- **Beispiel**

- Alle Mitarbeiter können Urlaub beantragen (auch der Abteilungsleiter)
- Nur der Abteilungsleiter kann Urlaub genehmigen



- Akteure ermitteln
 - Welche Personen sollen Aufgaben mit einem geplanten Softwaresystem durchführen?
 - Wer gibt Daten in das System ein?
Wer erhält die Ausgabedaten des Systems?
 - Welche Schnittstellen besitzt das System?
 - Benutzerschnittstelle
 - Andere Systeme
 - Gibt es zeitliche Ereignisse, auf die das System reagieren muss?
 - Evtl. Akteur „Zeit“ einführen, wenn das System automatisch reagiert
 - Oft muss ein anderer Akteur die Zeit überwachen und bei Eintreten des zeitlichen Ereignisses eine Anwendungsfall auslösen
 - Diesen Akteur modellieren

- Von Akteuren zu Anwendungsfällen
 - Welche Arbeitsabläufe werden von **Akteuren** mit Hilfe des Systems ausgeführt und liefern ihm ein Ergebnis?
 - Ergibt für jeden in sich abgeschlossenen Arbeitsablauf einen Anwendungsfall
 - Beispiel: Seminarorganisation

Eine Firma muss Weiterbildungsseminare organisieren, d.h. die Kunden, das Angebot (Seminarthemen, Seminare, Referenten) sowie Anmeldungen von Kunden zu Seminaren verwalten.

 - Ein **Angebotssachbearbeiter** muss Referenten registrieren sowie Seminarthemen, die von den Referenten gehalten werden. Zu den Seminarthemen können dann Seminare an best. Terminen stattfinden
 - AF Referent registrieren
 - AF Seminarthema registrieren
 - AF Seminar registrieren
 - Ein **Kundensachbearbeiter** muss neue Kunden registrieren sowie Anmeldungen entgegennehmen und verarbeiten
 - AF Kunde registrieren
 - AF Anmeldung bearbeiten

- Von Ereignissen zu Anwendungsfällen
 - Auf welche **Ereignisse** muss das Softwaresystem durch geeignete Arbeitsabläufe reagieren?
 - externe Ereignisse
 - zeitliche Ereignisse
 - Ereignislisten erstellen
 - für jedes Ereignis einen Anwendungsfall identifizieren
 - Beispiel: Seminarorganisation
 - Seminaranmeldung trifft ein (externes Ereignis)
 - AF Anmeldung bearbeiten
 - Teilnehmer sagt ab (externes Ereignis)
 - AF Stornierung bearbeiten
 - Noch zwei Wochen bis zum Seminarbeginn (zeitliches Ereignis)
 - AF Rechnungen erstellen

Anmerkung: Die Ereignisse (auch zeitliche) werden hier von einem Akteur (z.B. dem Kundensachbearbeiter) in der realen Welt wahrgenommen. Die Akteure lösen dann zu gegebener Zeit den zugehörigen Anwendungsfall aus.

- **Beispiel: Bibliothek**
 - Ereignisse (externe und zeitliche) und Anwendungsfälle, die sie auslösen

| <u>Art</u> | <u>Ereignis</u> | <u>Anwendungsfall</u> |
|------------|--------------------------------|-------------------------|
| ext. | neuer Leser | registrieren des Lesers |
| ext. | Leser leiht Buch aus | ausleihen von Büchern |
| ext. | Leser gibt Buch zurück | zurückgeben von Büchern |
| zeitl. | Rückgabedatum überschritten | mahnen des Lesers |
| ext. | Leser macht Vorbestellung | eintragen Vorbestellung |
| zeitl. | Abholfrist abgelaufen | prüfen der Abholfristen |
| ext. | Buch trifft ein | erfassen neuer Bücher |

- Anwendungsfallbeschreibungen mit der AF-Schablone
 - Zuerst **Standardverarbeitung** beschreiben ohne Sonderfälle zu betrachten
 - Verarbeitung, die in der Mehrzahl der Fälle auszuführen ist
 - Verarbeitung die „im Idealfall“ auszuführen ist
 - Verarbeitung die „im einfachsten Fall“ auszuführen ist
 - Danach die **Sonderfälle** beschreiben
 - Optionale Teile eines Anwendungsfalls (Erweiterungen)
 - Alternative Abläufe eines Anwendungsfalls
 - Abläufe, die nur selten stattfinden
- Vereinfachung Komplexer Anwendungsfälle
 - Komplexe Abläufe, die in mehreren Anwendungsfällen vorkommen, separat beschreiben und mit <<include>> einbinden
 - Umfangreiche Erweiterungen als eigenen Anwendungsfall definieren und mit <<extend>> einbinden.

- **Beispiel:** *Versicherung*

AF: *bearbeite Schadensfall*

Ziel: Bezahlung des Schadens durch die Versicherung

Kategorie: primär

Nachbedingung Erfolg: Schaden ganz oder teilweise bezahlt

Nachbedingung Fehlschlag: Forderung abgewiesen

Akteure: Schadenssachbearbeiter

Auslösendes Ereignis: Antragsteller meldet Schadensfall

Beschreibung:

- 1 Sachbearbeiter prüft die Forderung auf Vollständigkeit
- 2 Sachbearbeiter prüft, ob eine gültige Police vorliegt
- 3 Sachbearbeiter prüft alle Details der Police
- 4 Sachbearbeiter errechnet den Betrag und überweist ihn an den Antragsteller

- Beispiel: *Versicherung* (Forts.)

Erweiterungen:

1a Die vorliegenden Daten vom Antragsteller sind nicht vollständig:

→ Sachbearbeiter muss Daten nachfordern

2a Der Antragsteller besitzt keine gültige Police

→ Sachbearbeiter schließt den Fall ab

Alternativen:

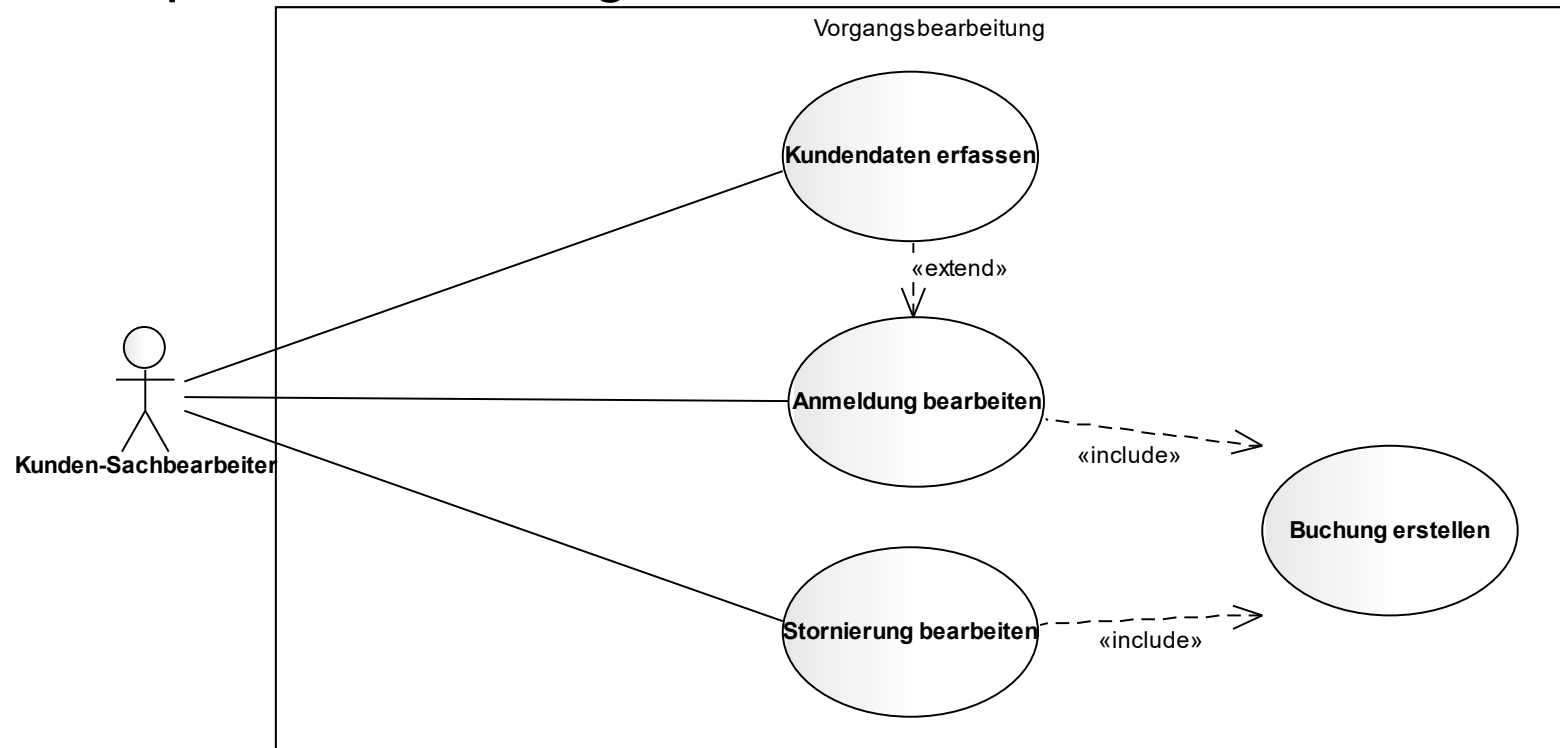
4a Der Schaden wird durch die Police nicht abgedeckt

→ Sachbearbeiter teilt dies dem Versicherungsnehmer mit und schließt den Fall ab

4b Der Schaden wird durch die Police nur unvollständig abgedeckt

→ Sachbearbeiter verhandelt mit dem Antragsteller, bis zu welchem Grad der Schaden bezahlt wird und überweist einen Teilbetrag

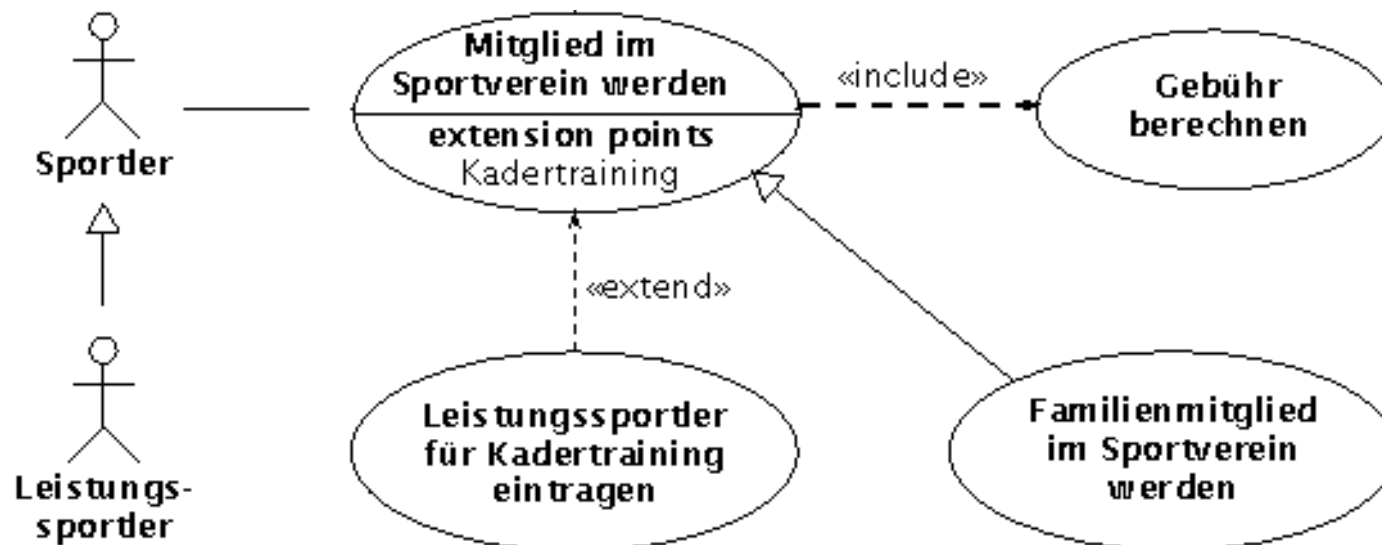
- Beispiel: *Seminarorganisation*



- Sowohl beim Anmelden zu einem Seminar als auch beim Stornieren eines Seminars (wenn es z.B. eine Warteliste gibt) kann eine Buchung erstellt werden.
- Nur wenn beim bearbeiten einer Anmeldung der Kunde noch nicht registriert ist, wird er neu registriert
- Der Anwendungsfall „Kunde registrieren“ kann auch unabhängig von einer Anmeldung stattfinden

- Generalisierung
 - Generalisierung ist immer eine „**is-a**“-Beziehung
 - Der spezielle Aktuer **ist** auch **ein** allgemeiner Akteur
 - Der spezielle Anwendungsfall **ist** auch **ein** allgemeiner Anwendungsfall
 - Generalisierung von Akteuren
 - Sind mehrere Akteure mit dem gleichen Anwendungsfall verbunden?
 - Generalisierung von Anwendungsfällen
 - Können „ähnliche“ Anwendungsfälle verallgemeinert werden durch einen Basis-Anwendungsfall?
 - Handelt es sich um einen abstrakten Basis-Anwendungsfall?
 - Ein abstrakter Anwendungsfall kann nicht ausgeführt werden
 - Nur die Spezialisierungen sind echte Anwendungsfälle
 - Vorsicht: nicht mit *extends* oder *include* verwechseln!
 - Spezialisierte Anwendungsfälle sind vollständige Anwendungsfälle
 - *extend*- und *include*-Anwendungsfälle im Allgemeinen nicht
 - Vorsicht: es wird kein Verhalten „vererbt“

- Beispiel:
 - Generalisierung von Anwendungsfällen
 - Generalisierung von Akteuren
 - Generalisierung vs. *include*



Quelle: [Balzert 05], Abb. 4.2-1

- „Gute“ Beschreibungen
 - Der Auftraggeber soll sie lesen und verstehen können
 - Beschreibung der Kommunikation der Akteure mit dem System
 - Nicht interne Systemstrukturen und Algorithmen
 - Überschaubare Größe: maximal ca. 1 Seite
- Anwendungsfall-Namen
 - Möglichst aussagekräftige und präzise Benennung der Anwendungsfälle
 - Name soll ein Verb enthalten, evtl. durch ein Substantiv ergänzt
 - **Was** wird **womit** gemacht?
 - z.B.: Buch ausleihen, Konto eröffnen
- Akteur-Namen
 - Muss die **Rolle** eines Akteurs gegenüber dem System ausdrücken!
 - Substantiv im Singular

- Konsistenz Akteure und Anwendungsfälle
 - Jeder Akteur soll mit mindestens einem Anwendungsfall interagieren
 - Jeder Anwendungsfall soll mit mindestens einem Akteur interagieren
 - Ausnahmen:
 - Anwendungsfälle, die über *include* oder *extend* angebunden sind
 - Spezielle Anwendungsfälle, deren Generalisierung mit einem Akteur interagieren
 - Abstrakte Anwendungsfälle, bei denen sämtliche Spezialisierungen mit einem Akteur interagieren.
- Anwendungsfalldiagramm
 - Umfang: maximal ca. 10 Anwendungsfälle
- Konsistenz Anwendungsfalldiagramm und -Beschreibungen
 - Sind alle Akteure der Anwendungsfallbeschreibungen auch im Anwendungsfalldiagramm eingetragen und umgekehrt?