

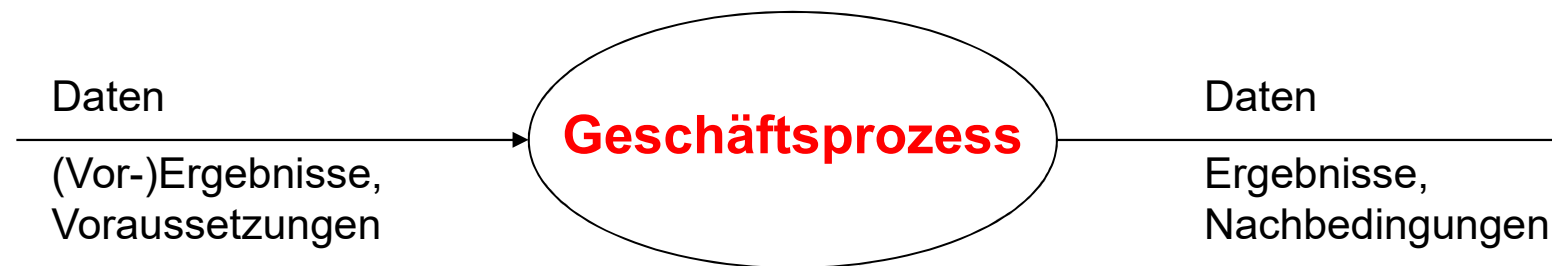
Grundlagen der Objektorientierung

Use-Cases

Was sind Use-Cases und wozu dienen sie?

Was sind Use-Cases?

- (engl.) *Use Case* = **Geschäftsprozess, Anwendungsfall**



Beispiele:

- Rechnung erstellen
- Einzelteil aus Lager holen
- Stückliste erstellen
- Buchung aufnehmen
- Buch ausleihen
- Klausur schreiben

Was sind Use-Cases (2) ?

- Erste allgemein verwendbare Notation von Ivar Jacobson (1992)
- Jacobson unterscheidet zwischen Use-Cases in einem
 - Informationssystem
 - ➔ „Sequenz von zusammengehörigen Transaktionen, die von einem Akteur im Dialog mit einem System ausgeführt werden, um dem Akteur ein Ergebnis von messbarem Wert zu erstellen“
 - Unternehmen
 - ➔ „Sequenz von Transaktionen in einem System (=Unternehmen) ➔ „*Business Process*“ (Geschäftsprozess)
 - Die ausgeführte Aufgabe soll für den Akteur des Unternehmens von messbarem Wert sein

Was sind Use-Cases (3) ?

Ein Use-Case beschreibt einen **speziellen Aspekt** eines Systems.

Beispiele:

1. die Anmeldung eines Users am System
2. die Reaktion eines Regelsystems auf das Überschreiten eines gemessenen Grenzwertes
3. Jede beliebige Funktion eines Systems

Wozu Use-Cases und Use-Case-Diagramme?

Zu Beginn der Entwicklung:

- sämtliche Funktionalitäten und Abläufe des zu entwickelnden Systems erfassen („*was soll das System bieten?*“)
- alle Benutzerrollen identifizieren („*wer nutzt das System?*“)
- Alle Funktionalitäten/Abläufe für jede Benutzerrolle festlegen („*wer darf was?*“)
- Übersichtliche grafische Darstellung (u.U. über mehrere Hierarchien hinweg)

Wie werden Use-Cases erfasst?

- Ermitteln, welche **Aufgaben** mit dem neuen Softwaresystem zu bewältigen sind, um die gewünschten Ergebnisse zu erzielen.
- Es soll nicht nur die Funktionalität der Software spezifiziert werden, sondern die **ergebnisorientierten Abläufe bei der Benutzung** dieser Software!
- In der Regel werden diese Use-Cases zunächst auf Karteikarten stichwortartig erfasst, um sie dann bei Bedarf in ein Diagramm umzusetzen.
- Diese Use-Cases müssen am Beginn eines Projekts gesammelt werden, um alle möglichen Fälle abzudecken.
- Es werden **Szenarios** erstellt, die den Use-Case repräsentieren, der modelliert werden soll.

Szenarios ... !?

Was ist ein Szenario?

- Ein Szenario ist ein Bestandteil eines Use-Cases
- Ein Use-Case wird durch ein Szenario oder einer Kollektion von Szenarios dokumentiert.
- Ein Szenario ist eine **Sequenz von Verarbeitungsschritten**, die unter bestimmten Bedingungen auszuführen ist.
- Diese Schritte sollen das Hauptziel eines Akteurs realisieren und ein entsprechendes Ergebnis liefern.
- Sie beginnen mit einem auslösenden Ereignis und werden fortgesetzt, bis das Ziel erreicht ist oder aufgegeben wird.
- Jedes Szenario wird durch eine oder mehrere Bedingungen definiert, die zu einem speziellen Ablauf des jeweiligen *Use Cases* führen.

Szenario-Beispiel *Telefonverbindung (UML-Spec.)*

Anrufer hebt Hörer ab
Wählton (Freizeichen) beginnt
Anrufer wählt Ziffer (5)
Wählton (Freizeichen) endet
Anrufer wählt Ziffer (4)
Anrufer wählt Ziffer (3)
Anrufer wählt Ziffer (2)
Angerufenes Telefon beginnt zu klingeln
Anrufendes Telefon signalisiert Rufton
Angerufener Teilnehmer meldet sich
Angerufenes Telefon hört auf zu klingeln
Anrufendes Telefon signalisiert Rufton nicht mehr
Telefone werden verbunden
Angerufener Teilnehmer hängt ein
Telefone werden getrennt
Anrufer hängt ein

Szenario-Beispiel *Telefonverbindung (verbessert)*

BEGINN Telefonverbindung

Anrufer hebt Hörer ab
Wählton (Freizeichen) beginnt
Anrufer wählt erste Ziffer (z.B. 5)
Wählton (Freizeichen) endet
Anrufer wählt restliche Ziffern (4, 3, 2, ..)
Angerufenes Telefon beginnt zu klingeln
Anrufendes Telefon signalisiert Rufton

WENN sich angerufener Teilnehmer meldet:

Angerufenes Telefon hört auf zu klingeln
Anrufendes Telefon signalisiert Rufton nicht mehr
Telefone werden verbunden

WENN angerufener Teilnehmer ODER Anrufer den Hörer auflegt:
Telefone werden getrennt

ENDE-WENN

SONST

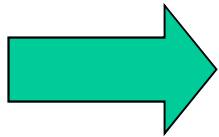
Anrufer legt auf

ENDE-WENN

ENDE Telefonverbindung

Szenario - Übungsbeispiele

Für die einzelnen Lastenheftanalysen:

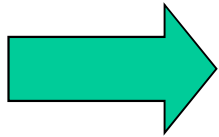


1. **Zooverwaltung: „Tier anlegen“**
2. **Vereinsverwaltung: „Mitglied anlegen“**
3. **Arztpraxis: „Arztbesuch“**

Hinweis:

bei allen Übungen sollen die Elemente „komplett“ angelegt werden, d.h. mit allen Referenzen zu anderen Elementen.

Szenario – weitere Übungsbeispiele



1. *Bei Firma bewerben*
2. *Vorstellungsgespräch „erleben“*
3. *Vorlesung besuchen*
4. *Prüfung schreiben*
5. *Praxis-/Bachelorarbeit anfertigen*

Hinweis:

bei allen Übungen sollen die Elemente „komplett“ angelegt werden, d.h. mit allen Referenzen zu anderen Elementen.

Formalisierte Beschreibung von Use-Cases

Use-Case-Beschreibung

Formalisierte Darstellung (Schablone, Template) zur besseren Übersicht:

- **Name** des Use-Cases
- **Ziel** des Use-Cases im Kontext (der SW)
- Liste der beteiligten **Akteure**
- **Trigger** (Auslöser, auslösendes Ereignis)
- **Vorbedingungen**
- **Nachbedingungen** (a) bei Erfolg und (b) bei Fehlschlag
- Essenzielle **Schritte** (als Szenario, z.B. mit Pseudocode)
- **Erweiterungen**
- **Alternativen**
- Bedingungen (optional)
- Häufigkeit (optional)

Use-Case-Beschreibung

Formalisierte Darstellung (Schablone, Template) zur besseren Übersicht:

Name des Use-Cases	(2-3 Wörter, was wird getan?)
Ziel des Use-Cases im Kontext (der SW)	globale Zielsetzung bei erfolgreicher Ausführung des UC
Liste der beteiligten Akteure	Rollen von Personen und anderen Systemen, die den Use-Case auslösen oder daran beteiligt sind
Trigger (Auslöser, auslösendes Ereignis)	Wenn dieses Ereignis eintritt, dann wird der Use-Case initiiert
Vorbedingungen	Erwarteter Zustand, bevor der Use-Case beginnt Der betrachtete Use-Case kann nur ausgeführt werden, wenn die genannte Vorbedingung erfüllt ist.

Use-Case-Beschreibung

Formalisierte Darstellung zur besseren Übersicht (Forts.):

Nachbedingungen bei Erfolg	Erwarteter Zustand nach erfolgreicher Ausführung des Use-Case → Ergebnis des Use-Case
Nachbedingungen bei Fehlschlag	<p>Erwarteter Zustand, wenn Ziel nicht erreicht werden kann</p> <p>Die Nachbedingung eines Use-Cases A kann für einen Use-Case B eine Vorbedingung bilden. → Reihenfolge von Use-Cases teilweise bestimmbar!</p>
Essenzielle Schritte	1. erste Aktion, 2. zweite Aktion, ... (ugs., Szenario)
Erweiterungen	<p>1a. Erweiterungen des Funktionsumfangs der 1. Aktion</p> <p>2a. Erweiterungen des Funktionsumfangs der 1. Aktion usw.</p>
Alternativen	<p>1a. Alternative Ausführung der ersten Aktion</p> <p>1b. Weitere Alternativen zur ersten Aktion</p>

Use-Case-Templates (5)

Beispiel für Erweiterungen und Alternativen



Essenzielle Schritte:

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:

- 1b Neukunden erfassen
- 3a Rechnung mit Nachnahme erstellen
- 3b Rechnung mit Bankeinzug erstellen

Use-Case-Beschreibung (**Beispiel**)

Name	Taxi bestellen
Ziel im Kontext	Ein Kunde bestellt ein Taxi
Akteure	Kunde, Taxivermittler (TV)
Trigger	Kunde ruft den Taxivermittler an
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Ein Kunde ruft den Taxivermittler an, um ein Taxi zu bestellen. 2. Der Kunde teilt dem Taxivermittler die Bestelldetails mit. 3. Der Taxivermittler registriert den Auftrag im System. 4. Der Taxivermittler bestätigt dem Kunden den Bestellauftrag.
Erweiterungen	<ol style="list-style-type: none"> 3a. Der Taxivermittler stellt fest, dass kein Taxi verfügbar ist: 3a1. Der TV stellt Anfrage an den übergeordneten Taxiverband 3a2. Der TV erhält die Bestätigung vom Taxiverband. 3b. Der TV stellt fest, dass Kunde nicht vertrauenswürdig ist: 3b1. Der TV lehnt den Auftrag ab.

Use-Case-Beispiel *einfacher Internetshop*

Erfassung der Use-Cases

Name	Use Case 1: Bestellung auf einer Internetseite
Ziel im Kontext	Ein Kunde bestellt Waren auf einer Internetseite
Akteure	Kunde
Trigger	Der Kunde legt eine ausgewählte Ware in den Warenkorb
Essenzielle Schritte	1. Der Kunde gibt eine Warenbestellung auf der Internetseite ein 2. Beim Bezahlvorgang muss sich der Kunde mit seinem Benutzernamen einloggen
Erweiterungen	2.a wenn der Benutzer noch nicht registriert ist, muss er sich registrieren 2.b wenn der Benutzer seine Login-Daten nicht mehr weiß ...
Bedingungen	Um einen bereits gespeicherten Zahlungsweg zu nutzen, ist die Anmeldung mit Passwort notwendig

Use-Case-Beispiel *einfacher Internetshop*

Erfassung der Use-Cases (Forts.)

Name	Use Case 2: Versand einer Bestellung (auf einer Internetseite)
Ziel im Kontext	Ein Händler versendet bestellte Waren
Akteure	Händler
Trigger	Ein Kunde hat eine Bestellung getätigt
Essenzielle Schritte	1. Der Händler prüft die (Bestell-)Daten des Kunden 2. Sind die Daten in Ordnung, verpackt er die Ware und versendet sie
Erweiterungen	2.a. Ist die Ware als Geschenk erwünscht, wird sie (zusätzlich) als Geschenk verpackt

Use-Case-Beispiel *einfacher Internetshop*

Erfassung der Use-Cases (Forts.)

Use-Case 3: Kundendaten prüfen

Use-Case 4: Waren verpacken

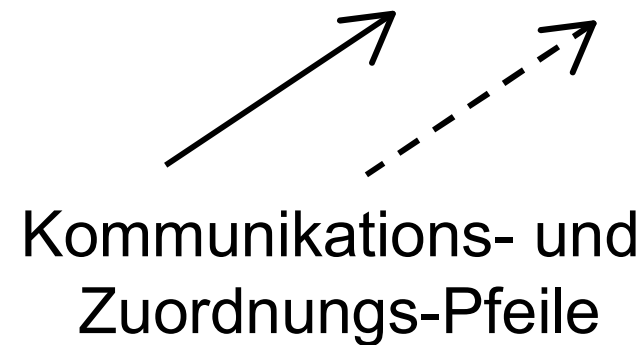
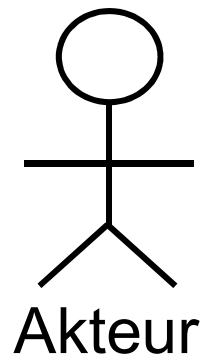
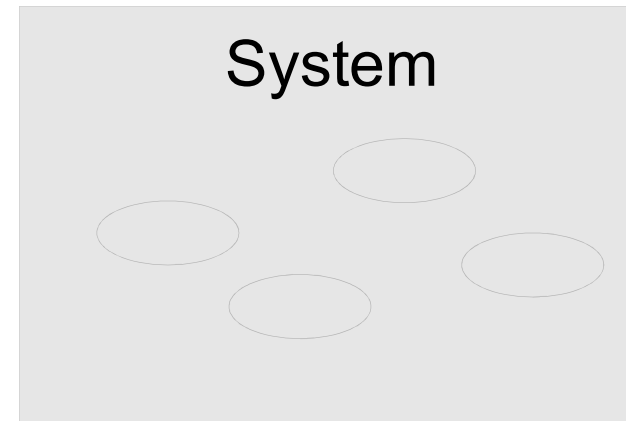
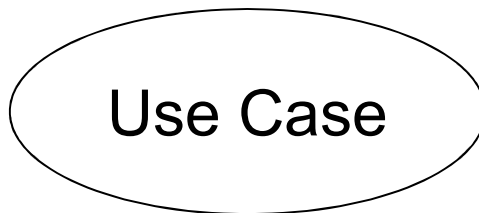
Use-Case 5: Waren als Geschenk verpacken

Use-Case 6: In System einloggen (Kunde)

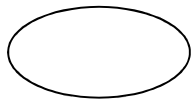
Use-Case 7: Am System registrieren (Kunde)

Grundelemente von Use-Case-Diagrammen

Use-Case-Elemente



Use-Case-Elemente



Standard-
Darstellung

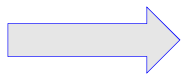
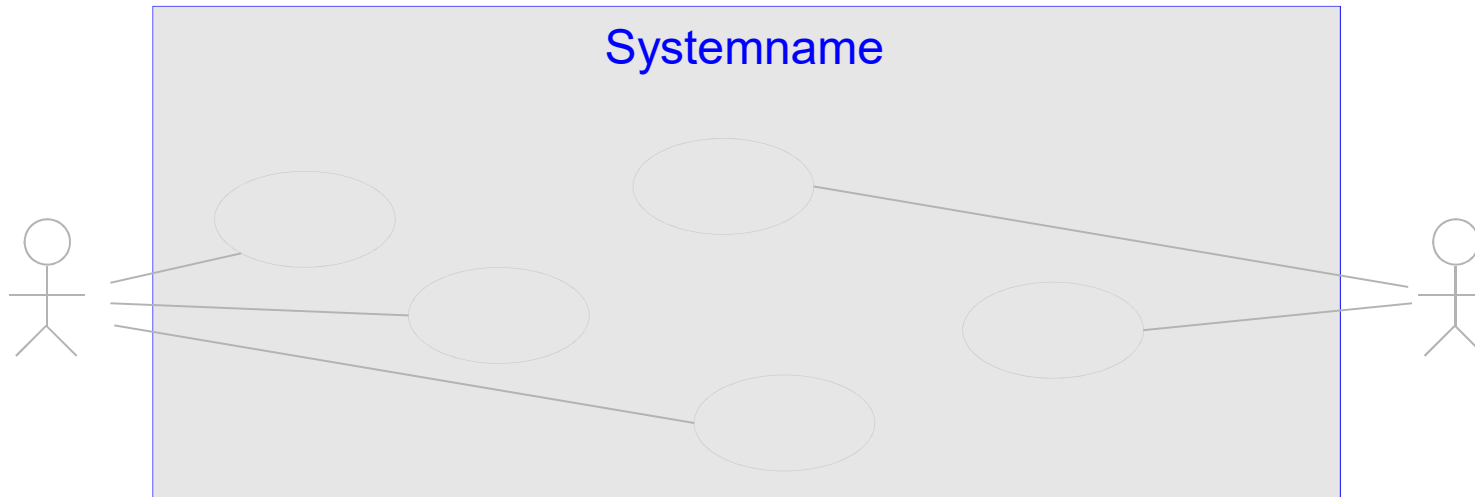


stereotypisierte
Darstellung

- Ein **Use-Case** besteht aus mehreren zusammenhängenden Aufgaben, die von einem **Akteur** durchgeführt werden, um ein Ziel zu erreichen bzw. ein gewünschtes Ergebnis zu erzielen.
- Beschreibung eines Use-Cases:
 - Ein einfacher Anwendungsfall kann umgangssprachlich beschrieben werden
 - Bei einer umfangreichen Spezifikation ist der Einsatz von *Use-Case-Templates* erforderlich.
 - Ebenso können Zustands- und Sequenzdiagramme, Pseudocode, PAP, Struktogramme usw. verwendet werden.

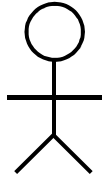
System - Systemgrenze

- System(-grenze), **in** dem die Use-Cases stattfinden
- Darstellung als Rechteck mit Benennung
- Jeder **Akteur** befindet sich stets **außerhalb** des Systems
- Jeder Use-Case kann selbst durch ein eigenes System definiert sein („Subsystem“).

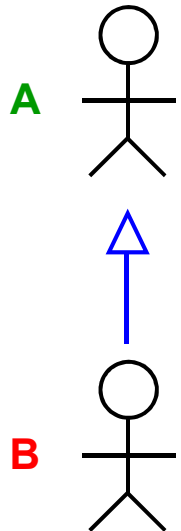


Die zu entwickelnde Software

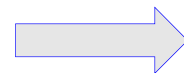
Akteure



- Ein **Akteur** (actor) ist eine Rolle, d.h. Menschen oder auch automatisierte Systeme, die ein **Benutzer** des Systems spielt.
 - er kommuniziert **mit** dem System
 - hat einen gewissen Einfluss auf das System.



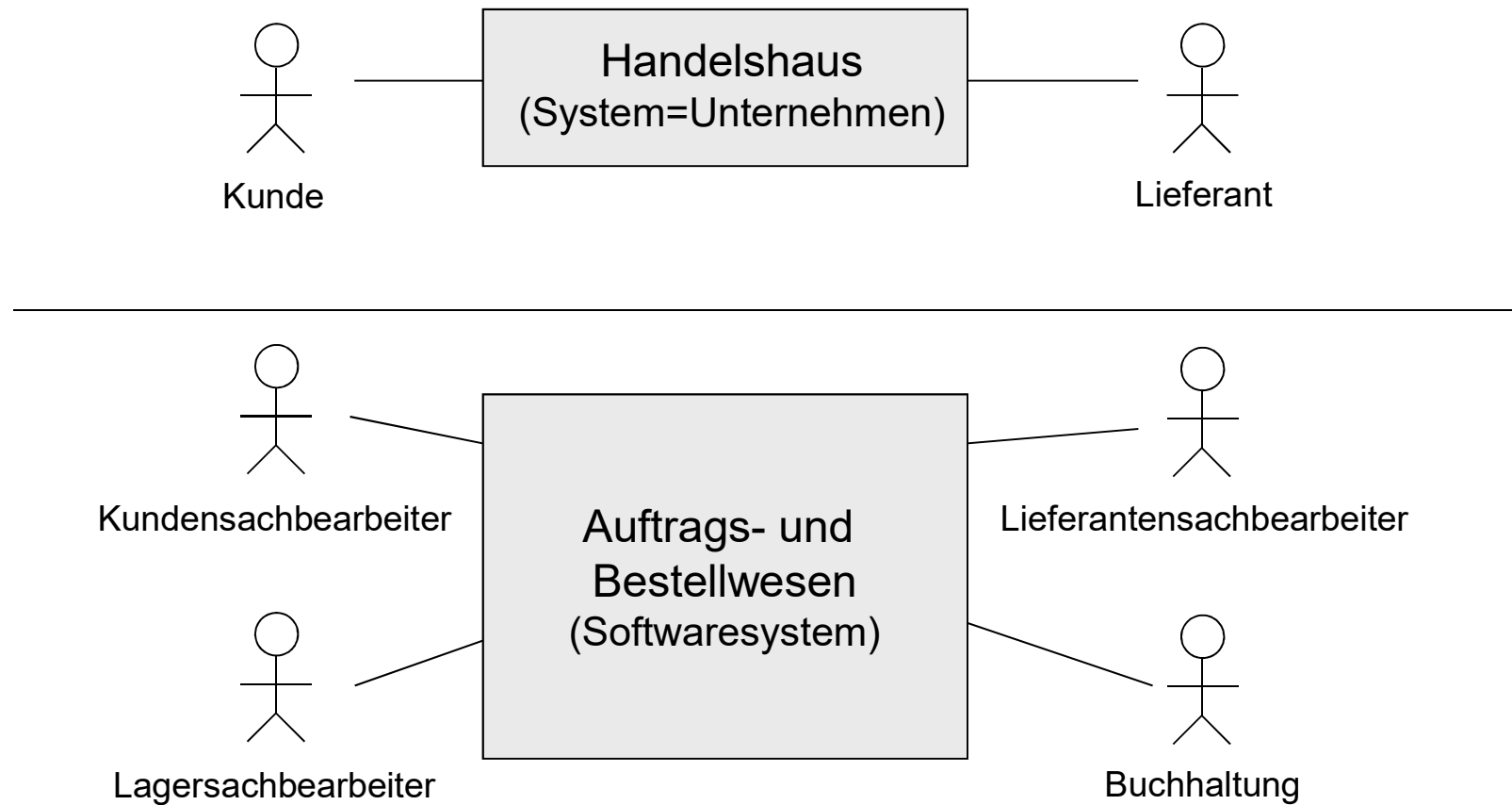
- Vererbung bzgl. Verantwortlichkeit ist möglich (Vererbungspfeil).
 - ➔ Bedeutung:
eine Instanz von Akteur **B** kann mit denselben Use-Cases kommunizieren wie eine Instanz von Akteur **A** (und evtl. mehr)



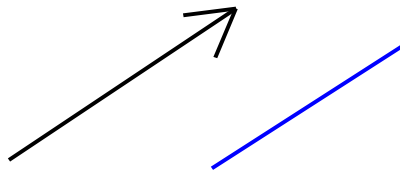
Anwender des Systems

Akteure und Systeme

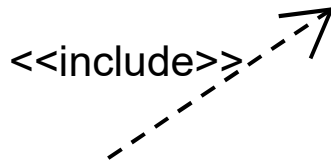
Beispiele für Akteure



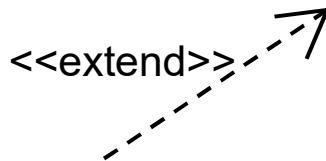
Assoziationen



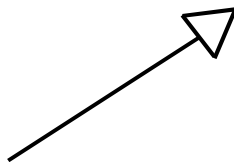
- Assoziation eines **Akteurs** mit einem **Use-Case**. Pfeile können die Kommunikationsrichtung hervorheben



- **include**-Beziehung (Auslagern gemeinsamer Funktionalität mehrerer Use-Cases)

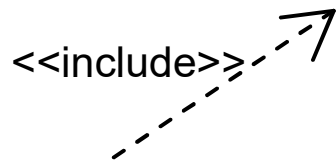


- **extend**-Beziehung (spezialisierte Erweiterung eines Use-Case)

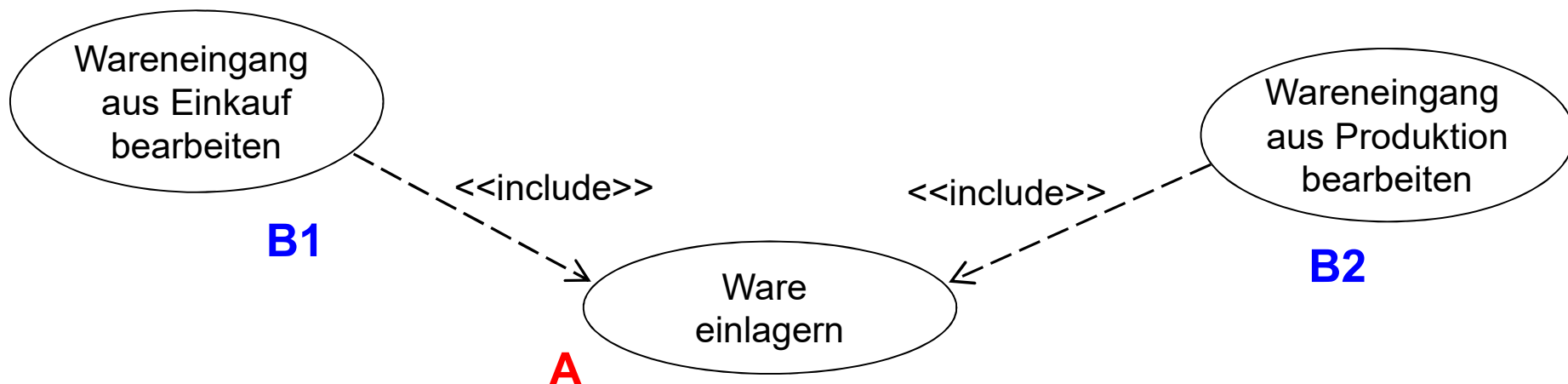


- Generalisierung (Vererbung). (Nur für Akteure sinnvoll, für Use-Cases in der UML nicht sauber definiert)

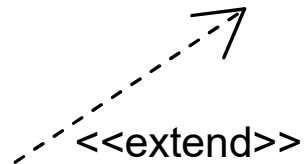
<<include>>-Assoziation



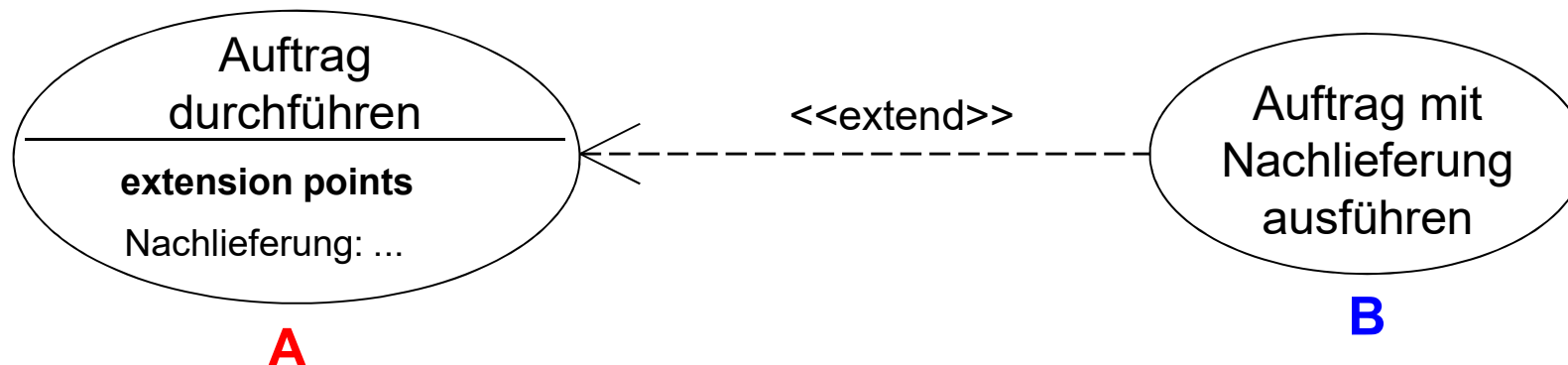
- Eine **include**-Beziehung liegt vor, wenn Use-Cases (z.B: **B1** und **B2**) ein gemeinsames (Teil-)Verhalten besitzen, das in dem Use-Case **A** spezifiziert ist.
- Use-Case **A** wird analog zu einem Unterprogramm oder einem Macro aufgerufen bzw. benutzt.
- Die **include**-Beziehung erspart die mehrmalige redundante Beschreibung des gleichen Verhaltens.
- Der inkludierte Use-Case ist nicht optional und wird **auf jeden Fall** aufgerufen (Workaround: Kommentar)



<<extend>>-Assoziation



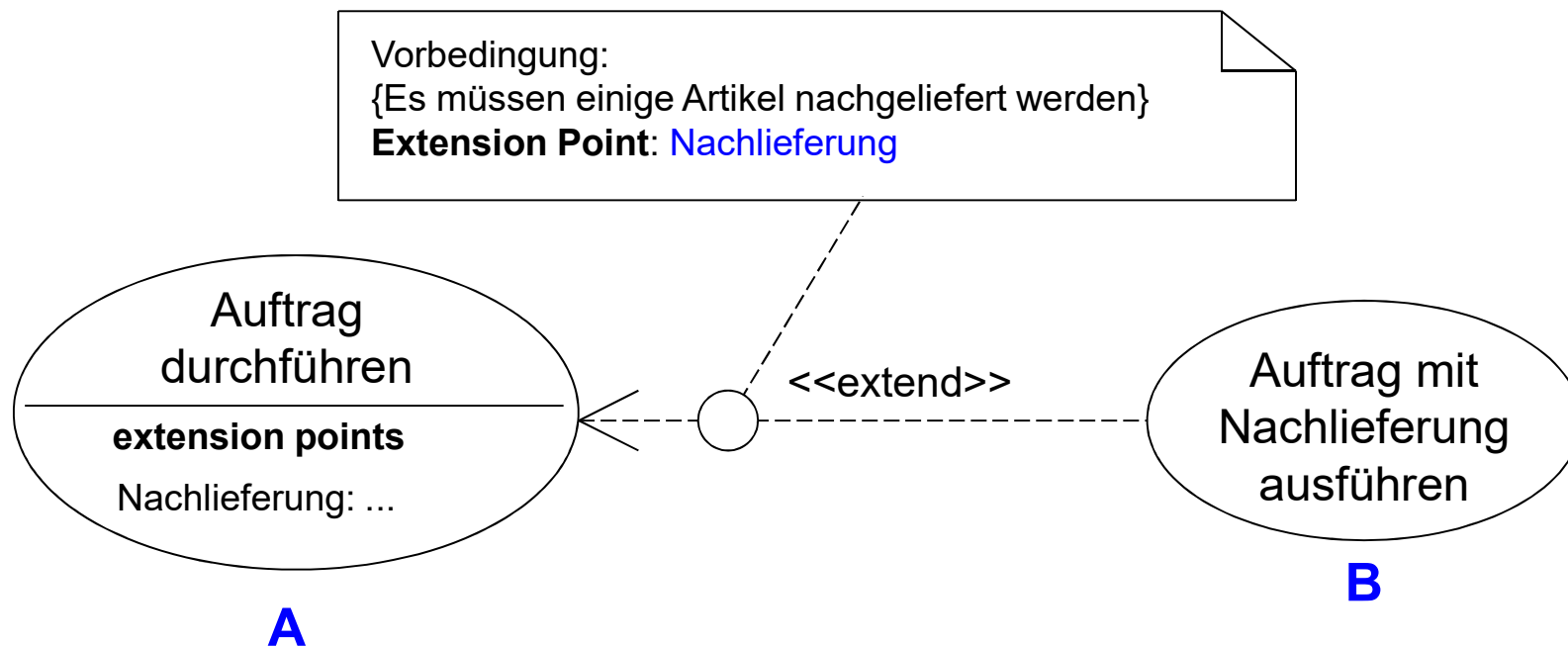
- Eine **extend**-Beziehung liegt vor, wenn ein Use-Case **B** ähnlich einem Use-Case **A** ist, aber „*etwas mehr tut als A*“.
- Der erweiternde Use-Case **B** ist optional und **ergänzt** den vorhandenen Use-Case **A**
- Die **extend**-Beziehung ermöglicht es, einen komplexen Use-Case zunächst in vereinfachter Form zu spezifizieren, und komplexe Sonderfälle in die Erweiterungen zu verlagern (s. auch *Extension Point* in übernächster Folie)
- Der erweiternde Use-Case wird **nur unter bestimmten Bedingungen** aufgerufen



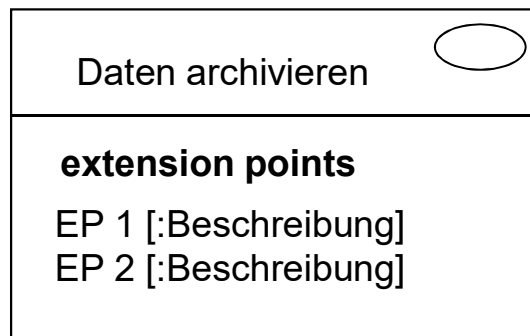
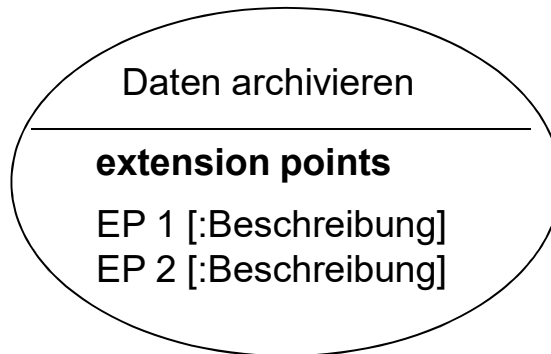
<<extend>>-Assoziation (Vorbedingungen)

- Die **Vorbedingungen**, die zur Verwendung eines erweiternden Use-Cases führen und die entsprechenden *Extension Points* werden als **Notiz** an die **extend**-Beziehung gehängt.

↗
<<extends>>



<<extend>>-Assoziation (*Extension Points*)



- Ein Use-Case kann mehrere *Extension Points* (**Erweiterungspunkte**) enthalten.
- Ein Erweiterungspunkt ist eine Referenz zu einem Ort innerhalb eines Use-Cases, wo Aktions-Sequenzen und Verhalten anderer Use-Cases eingefügt werden können.
- Jeder Erweiterungspunkt hat innerhalb eines Use-Cases einen **eindeutigen Namen** und eine **Beschreibung**, an welcher Stelle des Use-Cases dieser um einen anderen Use-Case erweitert wird.

Use-Case-Diagramm

Beispiel:
einfacher Internetshop

Use-Case-Beispiel einfacher Internetshop

Erfassung der Use-Cases (Forts.)

Use-Case 1: **Bestellung** auf einer Internetseite

Use-Case 2: **Versand** einer Bestellung

Use-Case 3: **Kundendaten** prüfen

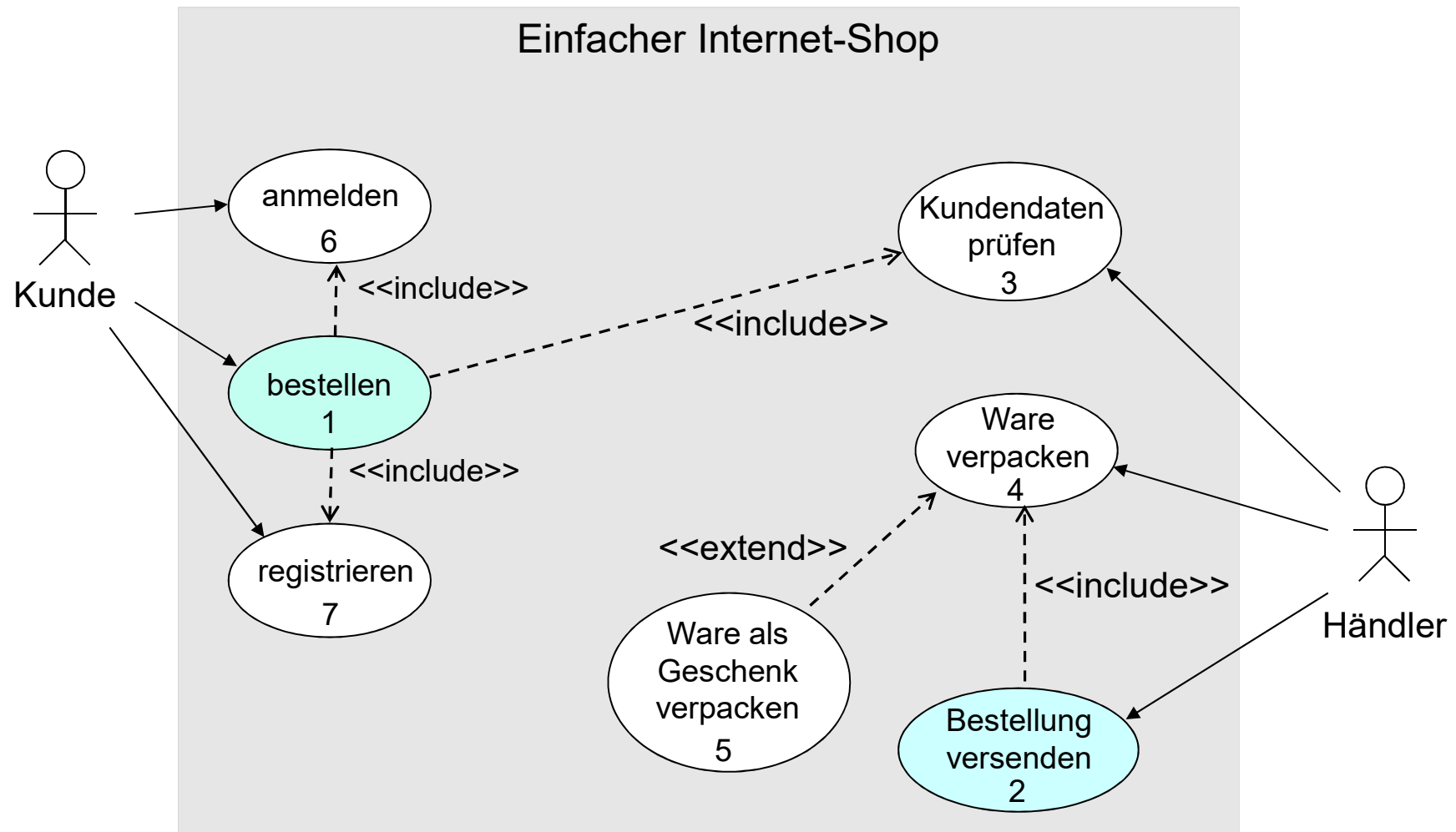
Use-Case 4: **Waren** verpacken

Use-Case 5: **Waren** als Geschenk verpacken

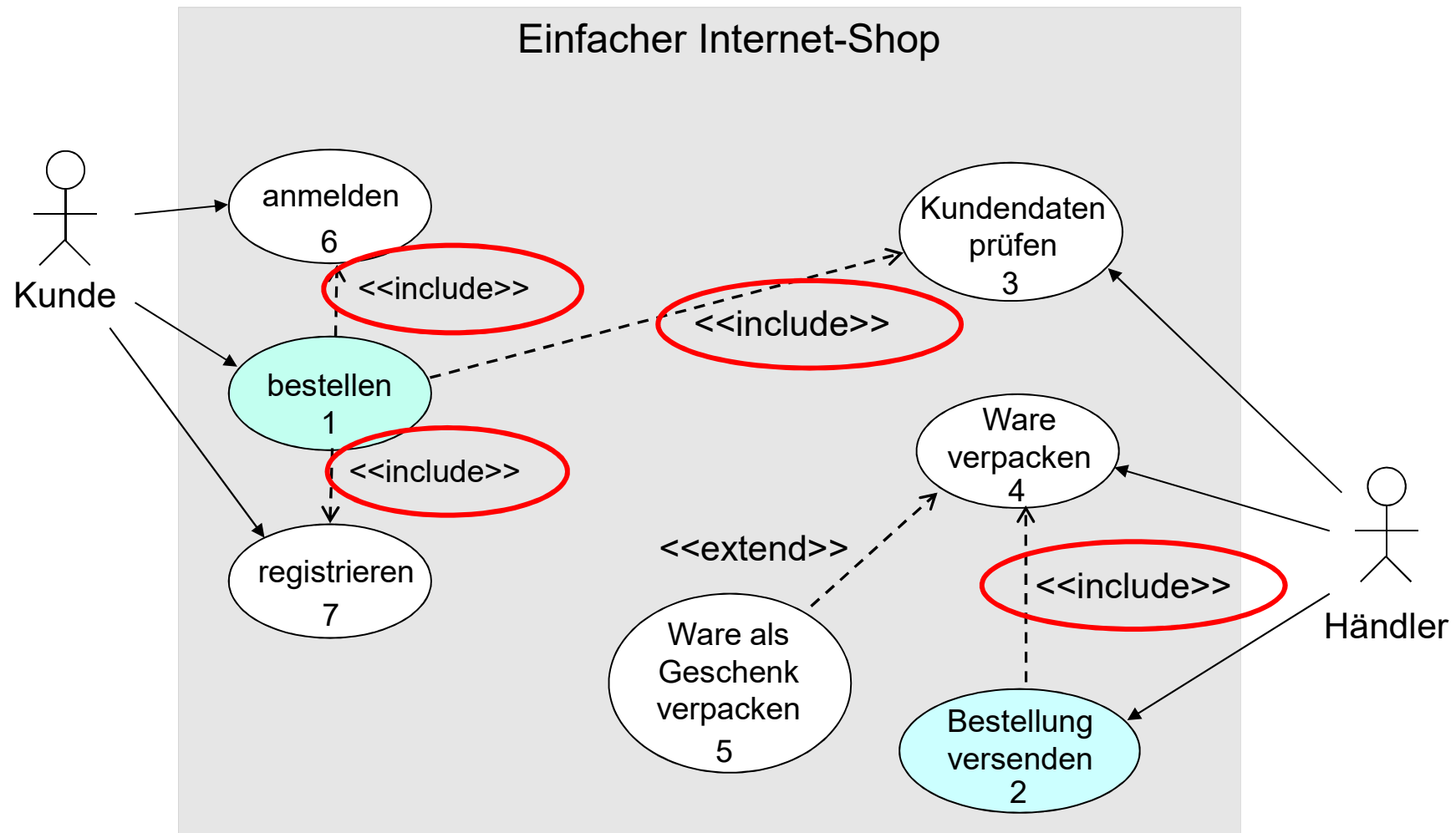
Use-Case 6: **In System** einloggen (Kunde)

Use-Case 7: **Am System** registrieren (Kunde)

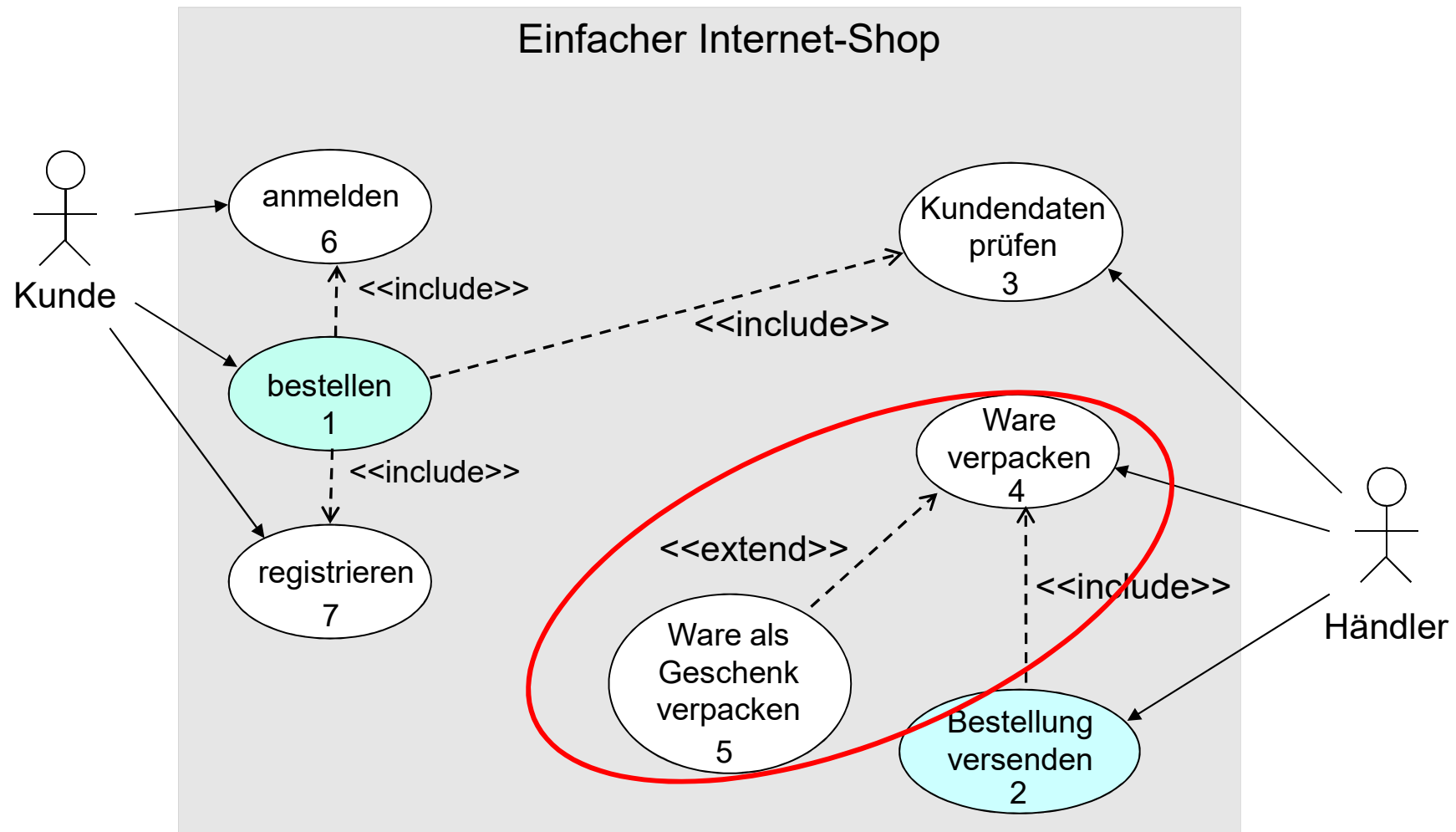
Use-Case-Beispiel: einfacher Internet-Shop



<<include>>-Assoziation



<<extend>>-Assoziation

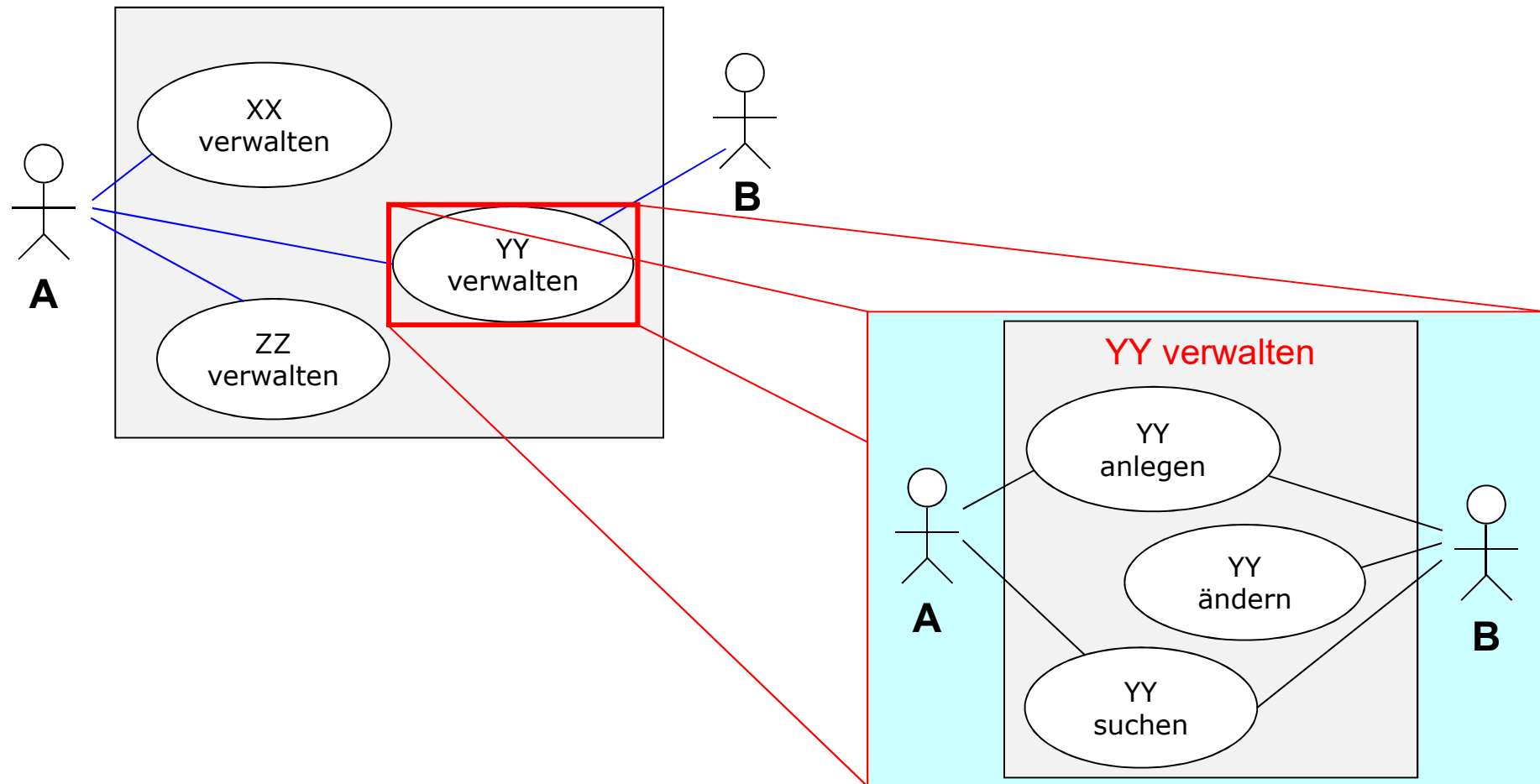


und schließlich:

Hierarchisierung (Verfeinerung)
von
Use-Case-Diagrammen

Hierarchisierung von Use-Cases

→ analog zu Datenflussdiagrammen



Was noch zu beachten ist beim Erstellen von Use-Case-Diagrammen

Zu beachten!

- Erstellen Sie Use-Cases aus der **Sicht des Anwenders** und nicht aus der Sicht des Entwicklers
- Verwenden Sie Use-Cases **nicht** zur **detaillierten** Beschreibung von Operationen und Funktionen
- Modellieren Sie nicht zu viele Use-Cases (Empfehlung: max. 10 Use-Cases pro Diagramm)
- Verfeinern Sie die Use-Cases durch eine geeignete Technik.
- Benennen Sie den Akteur immer eindeutig

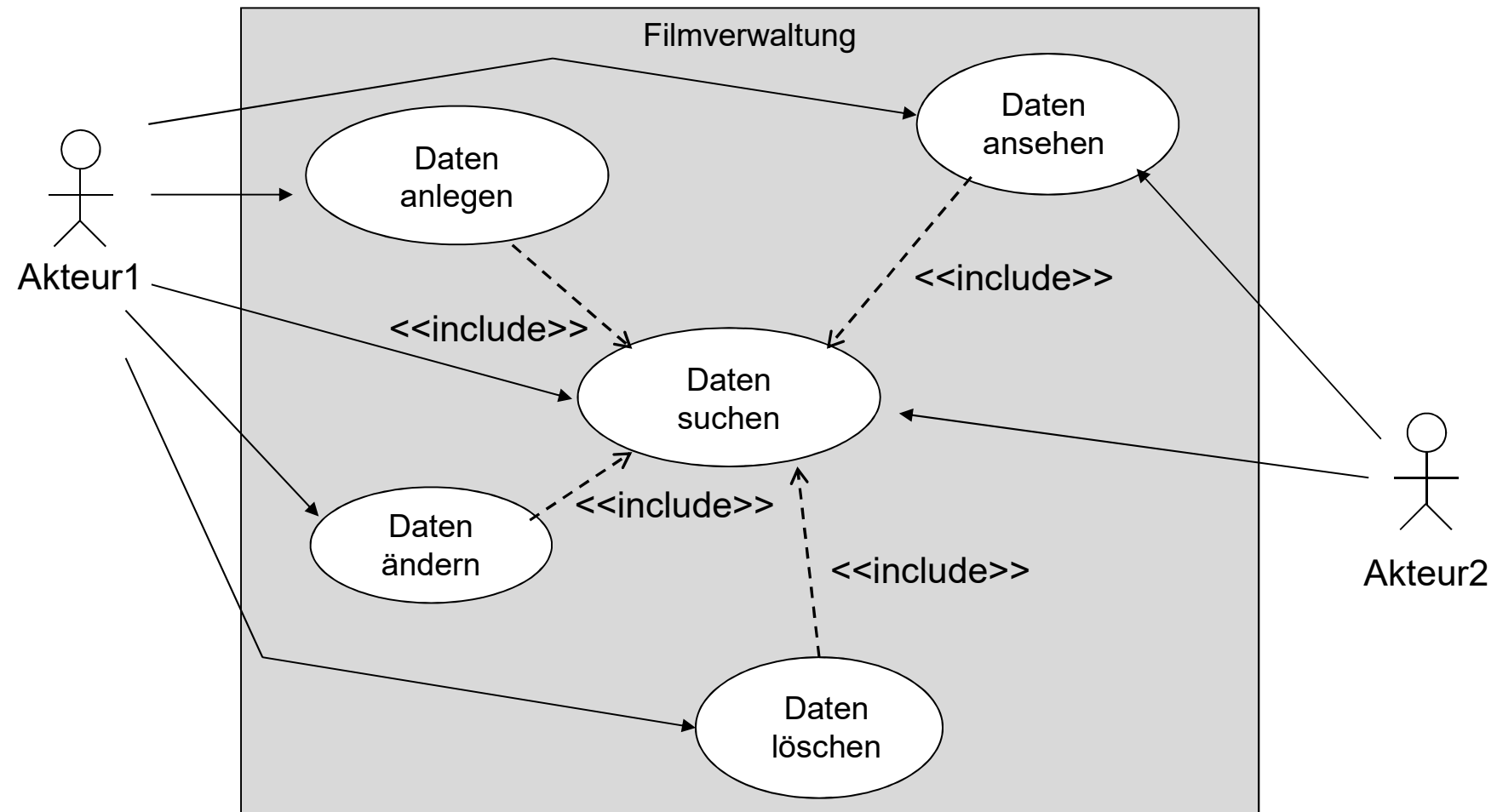
Wozu dienen Use-Cases **nicht**?

- Use-Cases beschreiben **nicht** die innere Struktur eines Software-Systems.
- Mit Use-Cases ist es **nicht** möglich, eine (vollständige) Beschreibung eines GUI-Designs zu erstellen (die funktionalen Anforderungen sind größtenteils unabhängig von der GUI).
- Use-Cases beschreiben **keine** Programmabläufe
→ Aktivitäts- und Sequenzdiagramme
- Use-Cases beschreiben **keine** generellen nichtfunktionalen Anforderungen (gesetzliche Regelungen, Qualitätsanforderungen, HW-Anforderungen, Standards etc.)

Use-Case-Diagramm: 2 Ansätze

- Funktionsorientierter Ansatz
 - Die reine Funktionalität steht im Vordergrund der Modellierung
 - ==> höherer Dokumentationsaufwand
 - ==> i.d.R. höherer Verfeinerungsgrad
- Elementorientierter Ansatz
 - Die Verwaltung der Elemente steht im Vordergrund
 - ==> Verfeinerungen sind sehr ähnlich (=> copy/paste)
 - ==> geringerer Dokumentationsaufwand

Funktionsorientierter Ansatz



Elementorientierter Ansatz

