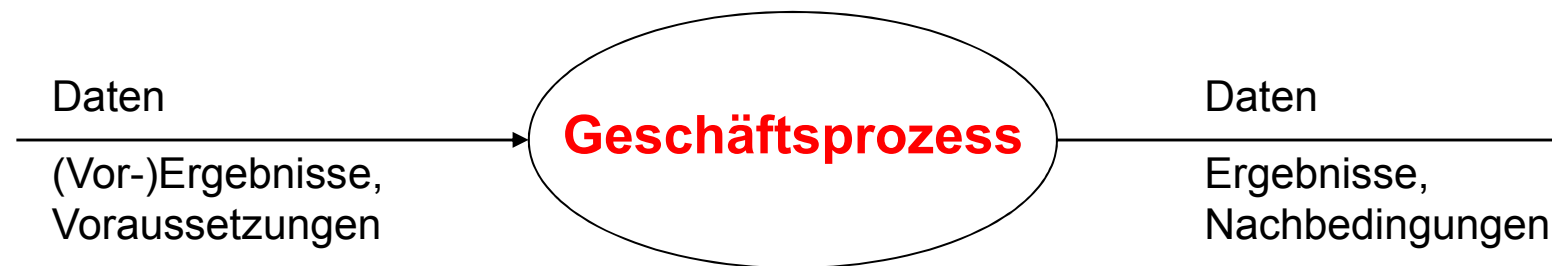


Grundlagen der Objektorientierung

Use Cases

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)
 - Unterscheidung (Jacobson) zwischen Use-Case 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

Wie werden Use-Cases erfasst?

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

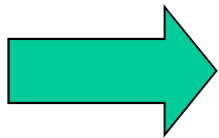
Was ist ein Szenario?

- 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*

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 - Übungsbeispiele

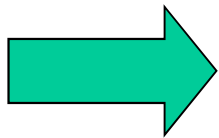


1. *Zooverwaltung: Tier anlegen*
2. *Vereinsverwaltung: Mitglied anlegen*
3. *Arztpraxis: Arztbesuch*
4. *Medienverwaltung: Medium anlegen*

Hinweis:

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

Szenario - Beispiele

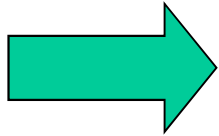


1. *Bei Firma bewerben*
2. *Vorstellungsgespräch „erleben“*
3. *Vorlesung besuchen*
4. *Prüfung schreiben*
5. *Praxis-/Bachelorarbeit anfertigen*

Hinweis:

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

Szenario - Beispiel *Assistentenprüfung*



Beispiel: einfacher Internet-Shop

Use-Case 1.0: Bestellung (auf einer Internetseite)

User: Kunde

Szenario:

Der Kunde gibt eine Warenbestellung auf der Internetseite ein. Beim Bezahlvorgang muss sich der Kunde mit seinem Usernamen einloggen oder neu registrieren.

Rationale (Grundprinzip):

Um einen bereits gespeicherten Zahlungsweg zu nutzen, ist die Anmeldung mit Passwort notwendig.

Frequency (Häufigkeit):

Jedes Mal, wenn der Kunde die Internetseite länger als 30 Minuten verlassen hat.

Beispiel: einfacher Internet-Shop

Use-Case 1.1: Versand einer Bestellung (auf einer Internetseite)

User: Händler

Szenario:

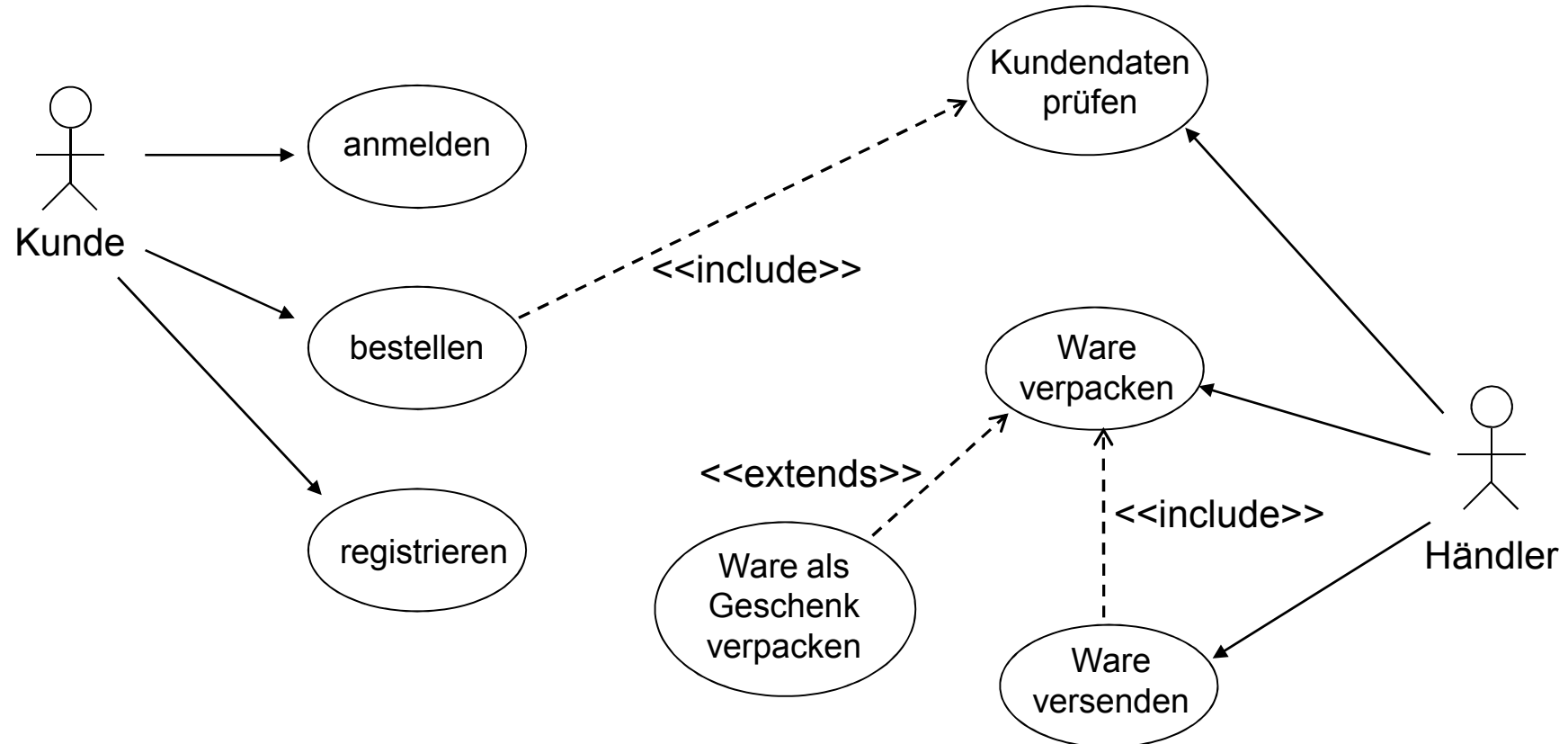
Der Händler prüft die (Bestell-)Daten des Kunden. Sind die Daten in Ordnung, verpackt er die Ware und versendet sie.

Ist die Ware als Geschenk erwünscht, wird sie (zusätzlich) als Geschenk verpackt.

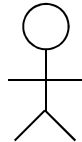
Frequency (Häufigkeit):

Jedes Mal, wenn der Kunde eine Bestellung getätigt hat.

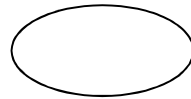
Use-Case-Beispiel



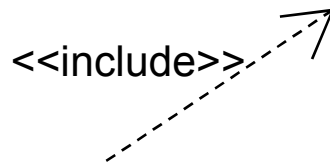
Elemente in Use-Case-Diagrammen



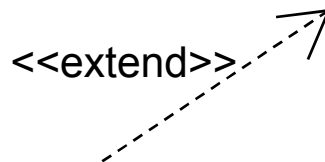
- Akteur (Rolle, d.h. Menschen, auch automatisierte Systeme)



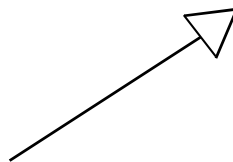
- Anwendungsfall bzw. Use-Case (Arbeitsablauf) (links stereotypisierte, rechts Standard-Darstellung)



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

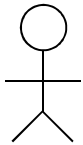


- Extend-Beziehung (spezialisierte Erweiterung eines Use-Case)

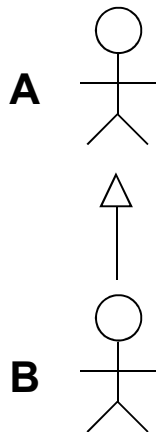


- Generalisierung (Vererbung)

Elemente in Use-Case-Diagrammen

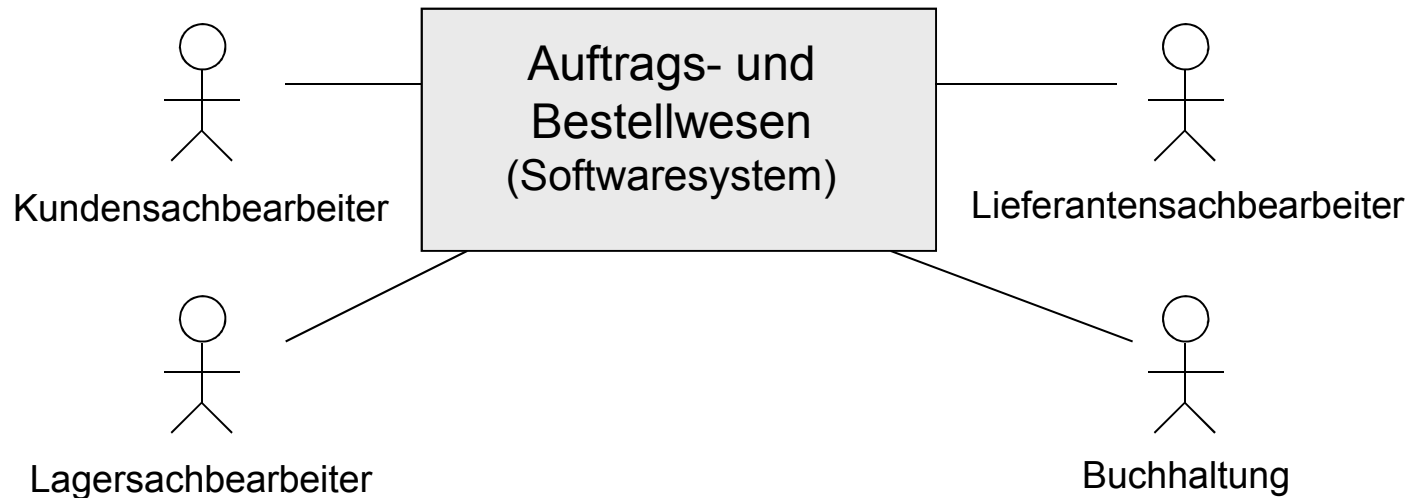
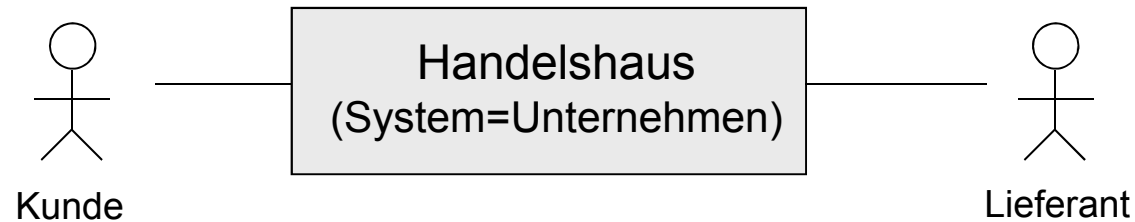


- Ein Akteur (actor) ist eine Rolle, d.h. Menschen oder auch automatisierte Systeme, die der Benutzer eines Systems spielt.
- Jeder Akteur hat einen gewissen Einfluss auf das System.
- Akteure befinden sich stets außerhalb des Systems.
- Vererbung (Vererbungspfeil).
→ Bedeutung: eine Instanz von Akteur A kann mit denselben Use-Case-Instanzen kommunizieren wie eine Instanz von Akteur B

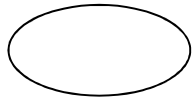


Elemente in Use-Case-Diagrammen

Wer ist Akteur?

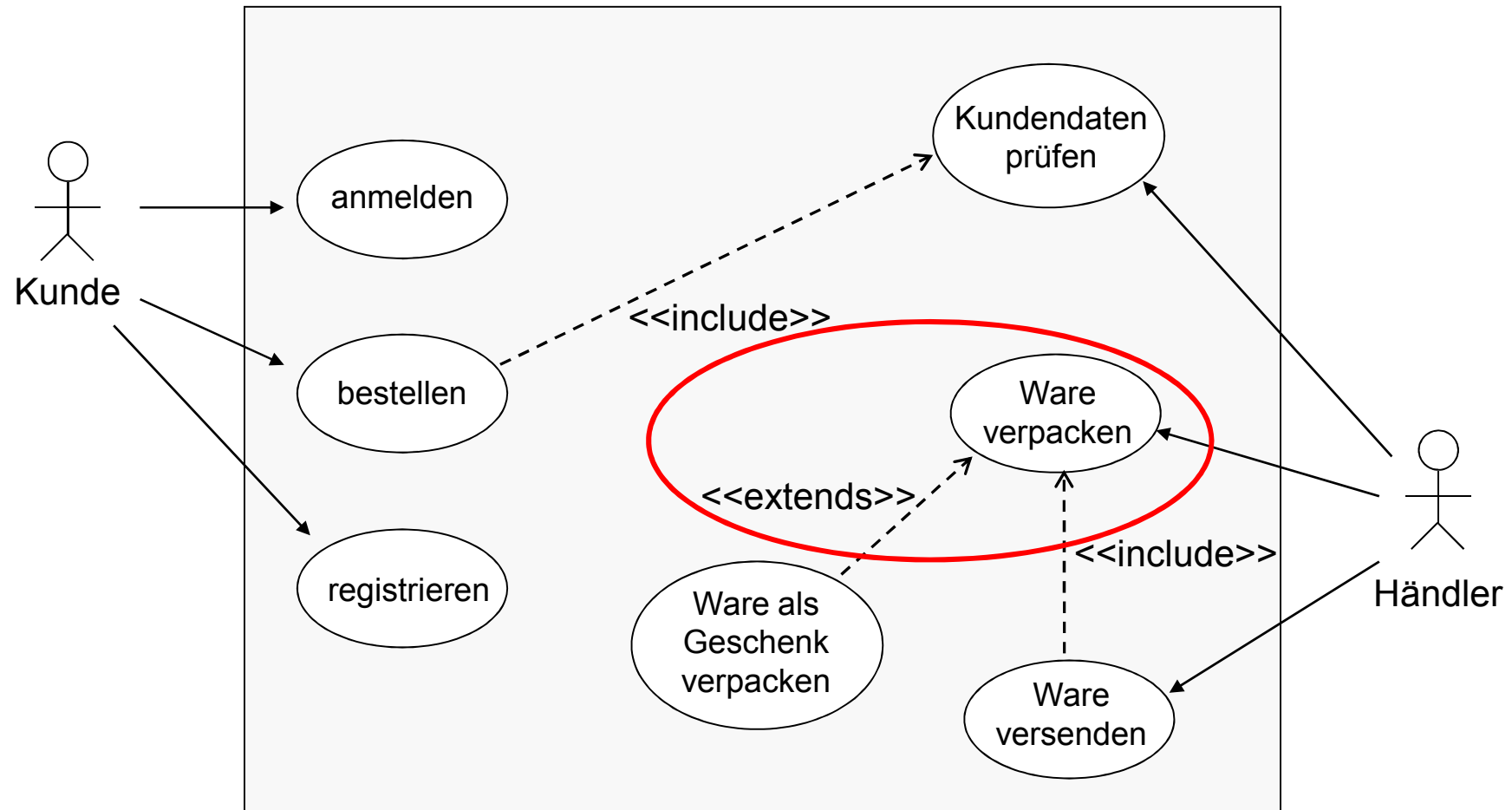


Elemente in Use-Case-Diagrammen

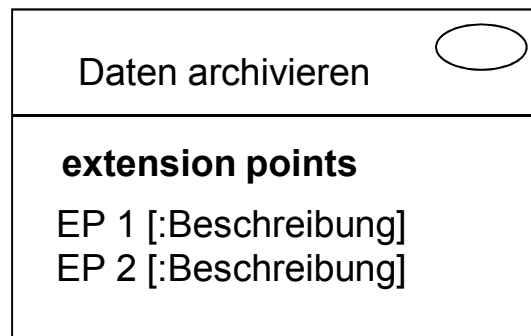
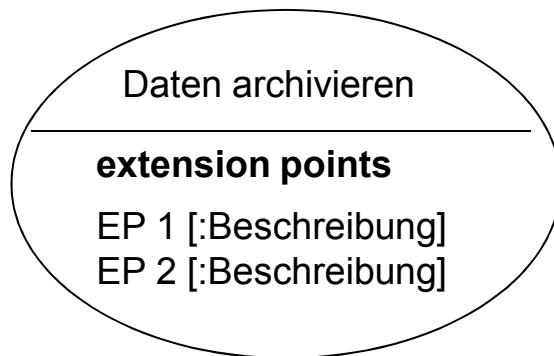


- Ein Anwendungsfall (*Use Case*, Geschäftsprozess) besteht aus mehreren zusammenhängenden Aufgaben, die von einem Akteur durchgeführt werden, um ein Ziel zu erreichen bzw. ein gewünschtes Ergebnis zu erstellen.
- Ein einfacher Anwendungsfall kann umgangssprachlich beschrieben werden,
- Bei einer umfangreichen Spezifikation ist der Einsatz von *Use Case Templates* erforderlich.
- Ebenso können Methoden, Zustands- und Sequenzdiagramme verwendet werden.

Use-Case-Beispiel – „extends“



Elemente in Use-Case-Diagrammen

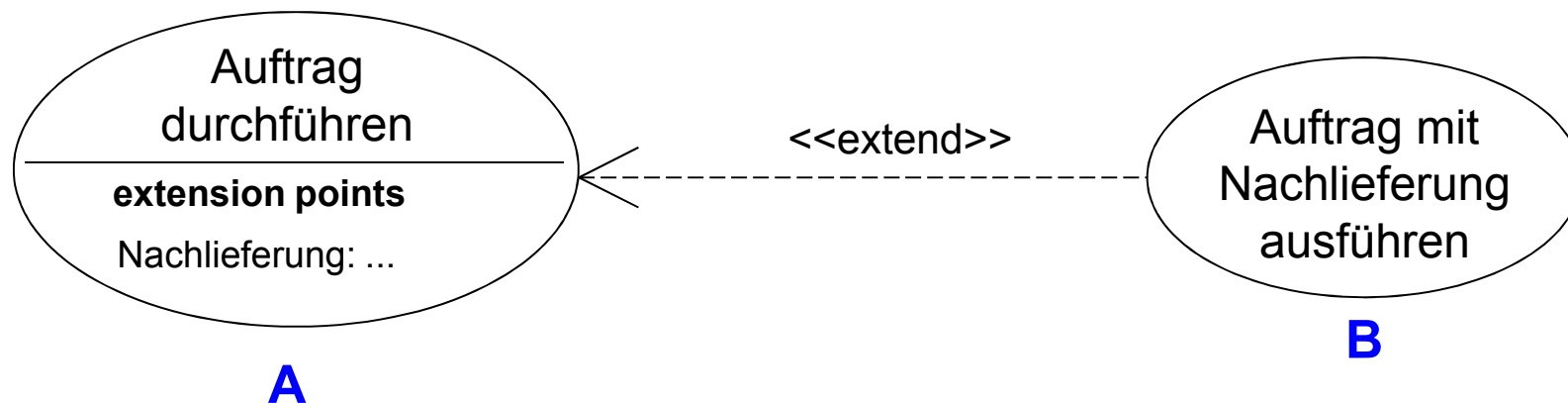


- Ein Anwendungsfall kann mehrere **Erweiterungspunkte** (*Extension Points*) enthalten.
- Ein Erweiterungspunkt ist eine Referenz zu einem Ort innerhalb eines Use-Cases, wo Aktions-Sequenzen und Verhalten anderer Use-Cases eingefügt werden können.
- Jeder Erweiterungspunkt hat innerhalb eines Use-Cases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

Elemente in Use-Case-Diagrammen

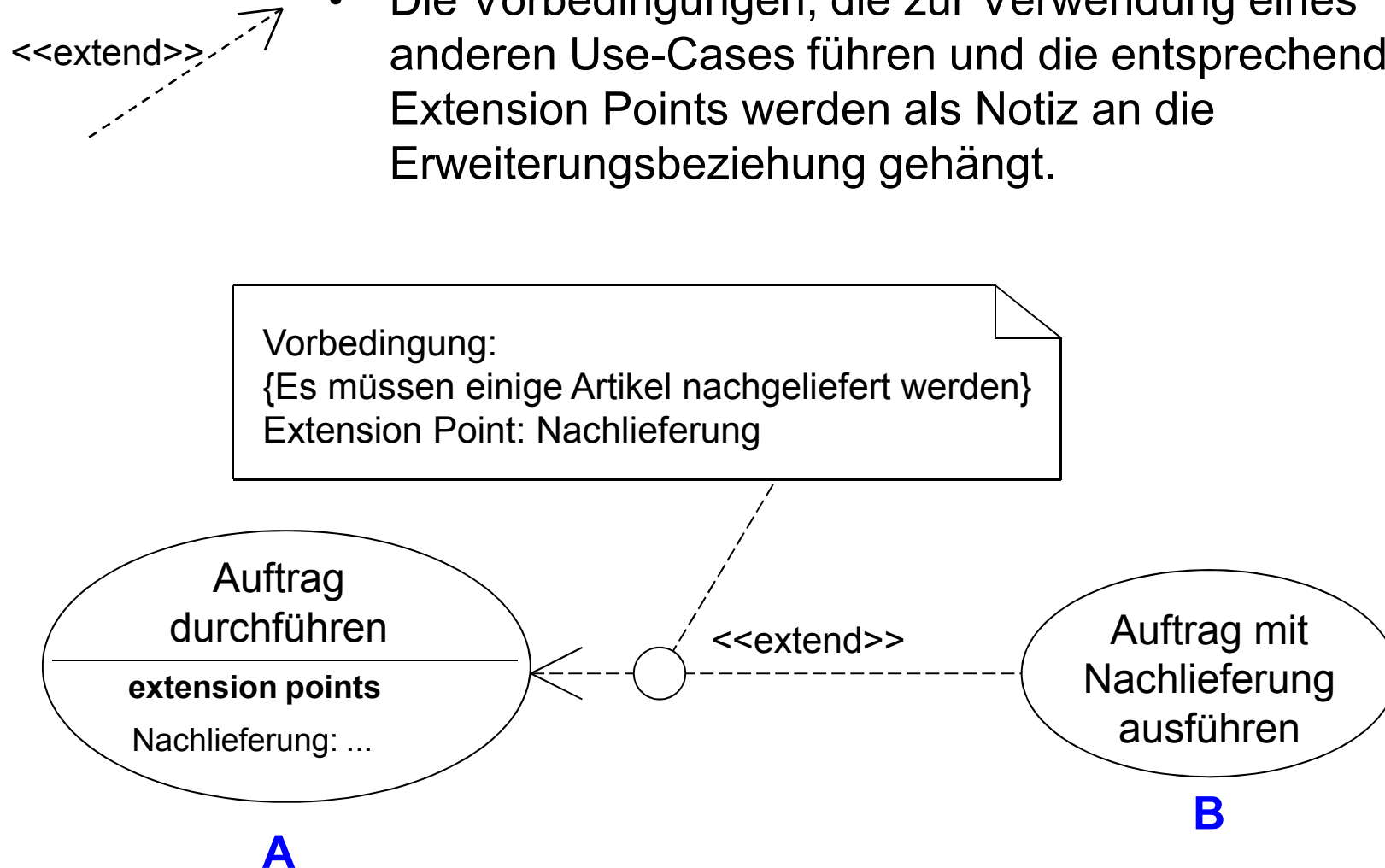
<<extend>>

- Eine **extend**-Beziehung liegt vor, wenn ein Use-Case **B** ähnlich einem Use-Case **A** ist, aber „etwas mehr als **A** tut“.
- 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*)

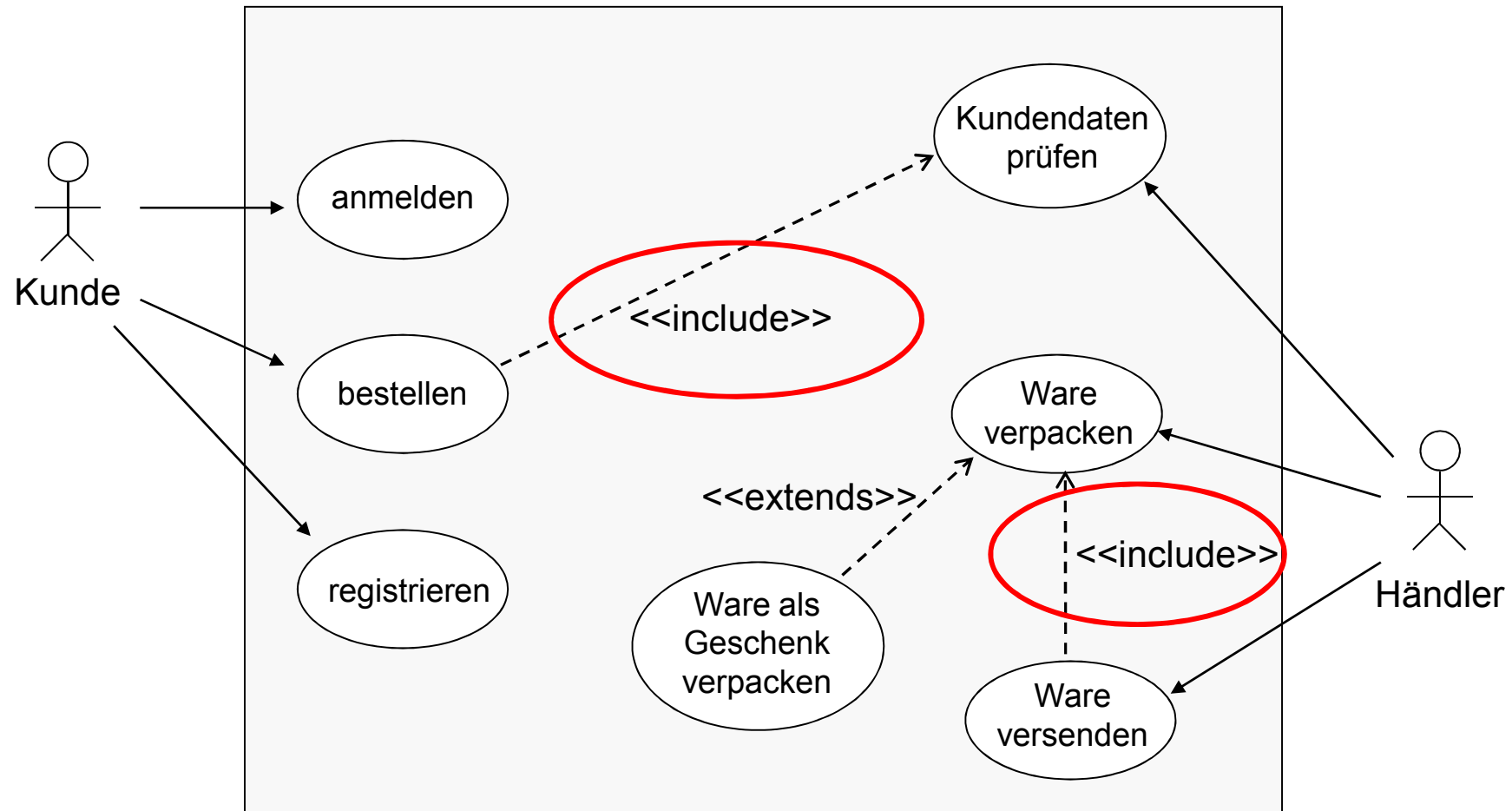


Elemente in Use-Case-Diagrammen

- Die Vorbedingungen, die zur Verwendung eines anderen Use-Cases führen und die entsprechenden Extension Points werden als Notiz an die Erweiterungsbeziehung gehängt.

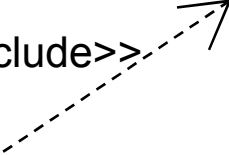


Use-Case-Beispiel – „include“

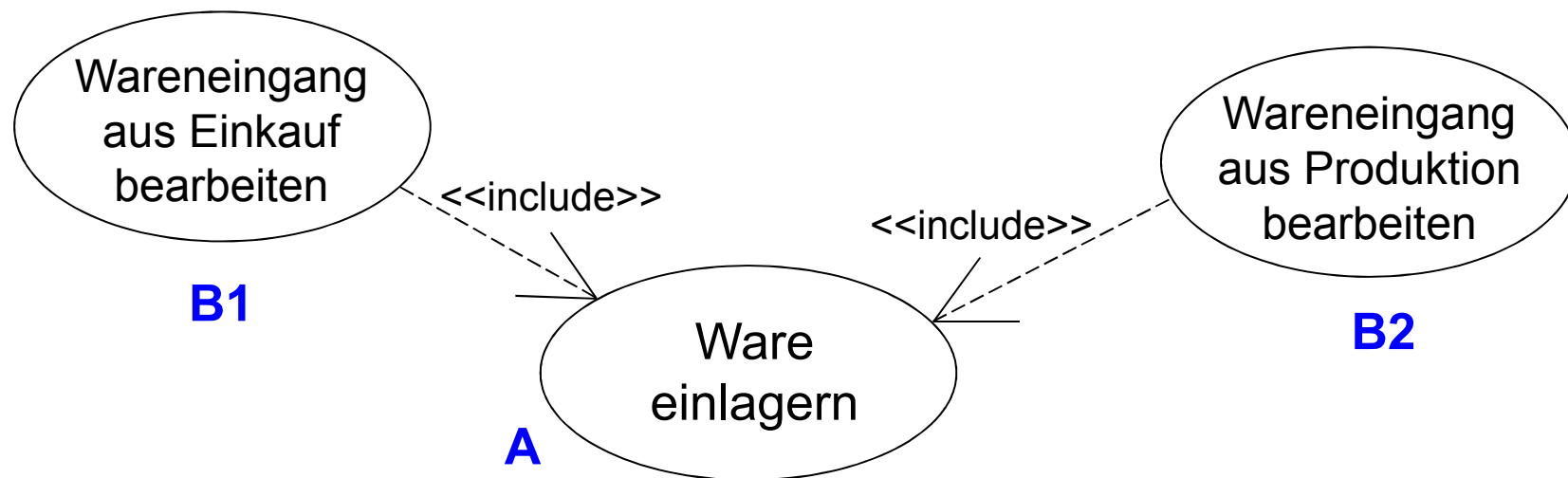


Elemente in Use-Case-Diagrammen

<<include>>



- Eine **include**-Beziehung liegt vor, wenn zwei Use-Cases **B1** und **B2** ein gemeinsames Verhalten besitzen, das in dem Use-Case **A** spezifiziert ist.
- Use-Case **A** wird analog zu einem Unterprogramm aufgerufen bzw. benutzt.
- Die **include**-Beziehung erspart die mehrmalige redundante Beschreibung des gleichen Verhaltens.



Use-Case-Templates

→ Zur detaillierten Beschreibung eines Use-Cases

Grundlegende Struktur:

1. Use-Case-Name
2. Ziel
3. Kategorie
4. Vorbedingung
5. Nachbedingung bei Erfolg
6. Nachbedingung bei Fehlschlag
7. Akteure
8. Auslösendes Ereignis
9. Beschreibung
10. Erweiterungen
11. Alternativen

Use-Case-Templates

→ Zur detaillierten Beschreibung eines Use-Cases

Use-Case: Name (2-3 Wörter, was wird getan?)

Ziel: globale Zielsetzung bei erfolgreicher Ausführung des UC

Kategorie: *primär:* Wenn der Use-Case *notwendiges* Verhalten beschreibt, das **häufig** benötigt wird

sekundär: ... das **selten** benötigt wird

optional: wenn er Verhalten beschreibt, das für das System zwar **nützlich**, aber **nicht unbedingt notwendig** ist

Use-Case-Templates (2)

→ Zur detaillierten Beschreibung eines Use-Cases

- Vorbedingung:** Erwarteter Zustand, bevor der Use-Case beginnt
- Nachbedingung bei Erfolg:** Erwarteter Zustand nach erfolgreicher Ausführung des Use-Case → Ergebnis des Use-Case
- Nachbedingung bei Fehlschlag:** Erwarteter Zustand, wenn das Ziel nicht erreicht werden kann
- Der betrachtete Use-Case kann nur ausgeführt werden, wenn die genannte **Vorbedingung** erfüllt ist.
 - Die **Nachbedingung** eines Use-Cases **A** kann für einen Use-Case **B** eine Vorbedingung bilden.
→ Reihenfolge von Use-Cases teilweise bestimmbar!

Use-Case-Templates (3)

→ Zur detaillierten Beschreibung eines Use-Cases

Akteure:	Rollen von Personen und anderen Systemen, die den Use-Case auslösen oder daran beteiligt sind.
Auslösendes Ereignis:	Wenn dieses Ereignis eintritt, dann wird der Use-Case initiiert.
Beschreibung:	1.erste Aktion, 2. zweite Aktion, ... (ugs.)
Erweiterungen:	1a. Erweiterungen des Funktionsumfangs der ersten Aktion
Alternativen:	1a. Alternative Ausführung der ersten Aktion 1b. Weitere Alternativen zur ersten Aktion

Use-Case-Templates (4)

Beispiel

Use-Case:	Auftrag ausführen
Ziel:	Ware an Kunden geliefert
Vorbedingung:	Auftrag muss gegeben sein
Nachbed. Erfolg:	Ware ausgeliefert (auch Teillieferungen), Rechnungskopie bei Buchhaltung
NB. Fehlschlag:	Mitteilung an Kunden, dass nichts lieferbar ist
Akteure:	Kundensachbearbeiter, Lagersachbearbeiter, Buchhaltung
Auslösendes Ereignis:	Bestellung des Kunden liegt vor

Use-Case-Templates (5)

Beispiel (Forts.)

Beschreibung:

1. Kundendaten abrufen
2. Lieferbarkeit prüfen
3. Rechnung erstellen
4. Auftrag vom Lager ausführen lassen
5. Rechnungskopie an Buchhaltung geben

Erweiterung:

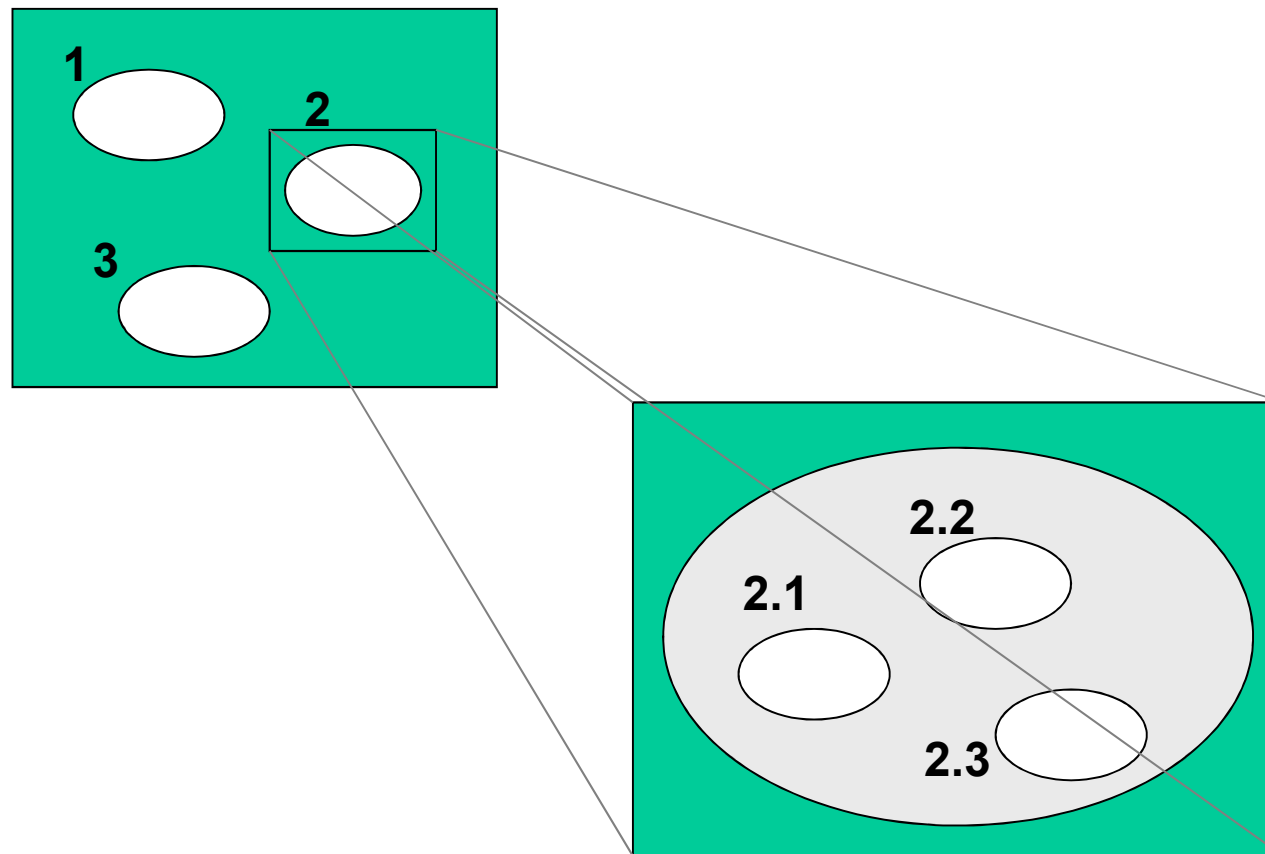
- 1a Kundendaten aktualisieren

Alternativen:

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

Hierarchisierung von Use-Cases

→ analog zu Datenflussdiagrammen



Zu beachten!

- Verwenden Sie Use-Cases **nicht** zur **detaillierten** Beschreibung von Operationen und Funktionen
- Erstellen Sie Use-Cases aus der **Sicht des Anwenders** und nicht aus der Sicht des Entwicklers
- Modellieren Sie nicht zu viele Use-Cases (Empfehlung: max. 10 Use-Cases pro Diagramm)
- Gehen Sie mit den erlaubten Beziehungen (include und extend) zwischen den Use-Cases sparsam um, denn sie reduzieren die intuitive Verständlichkeit für den Leser
- Verfeinern Sie die Use-Cases durch eine geeignete Technik.
- Benennen Sie den Akteur immer eindeutig

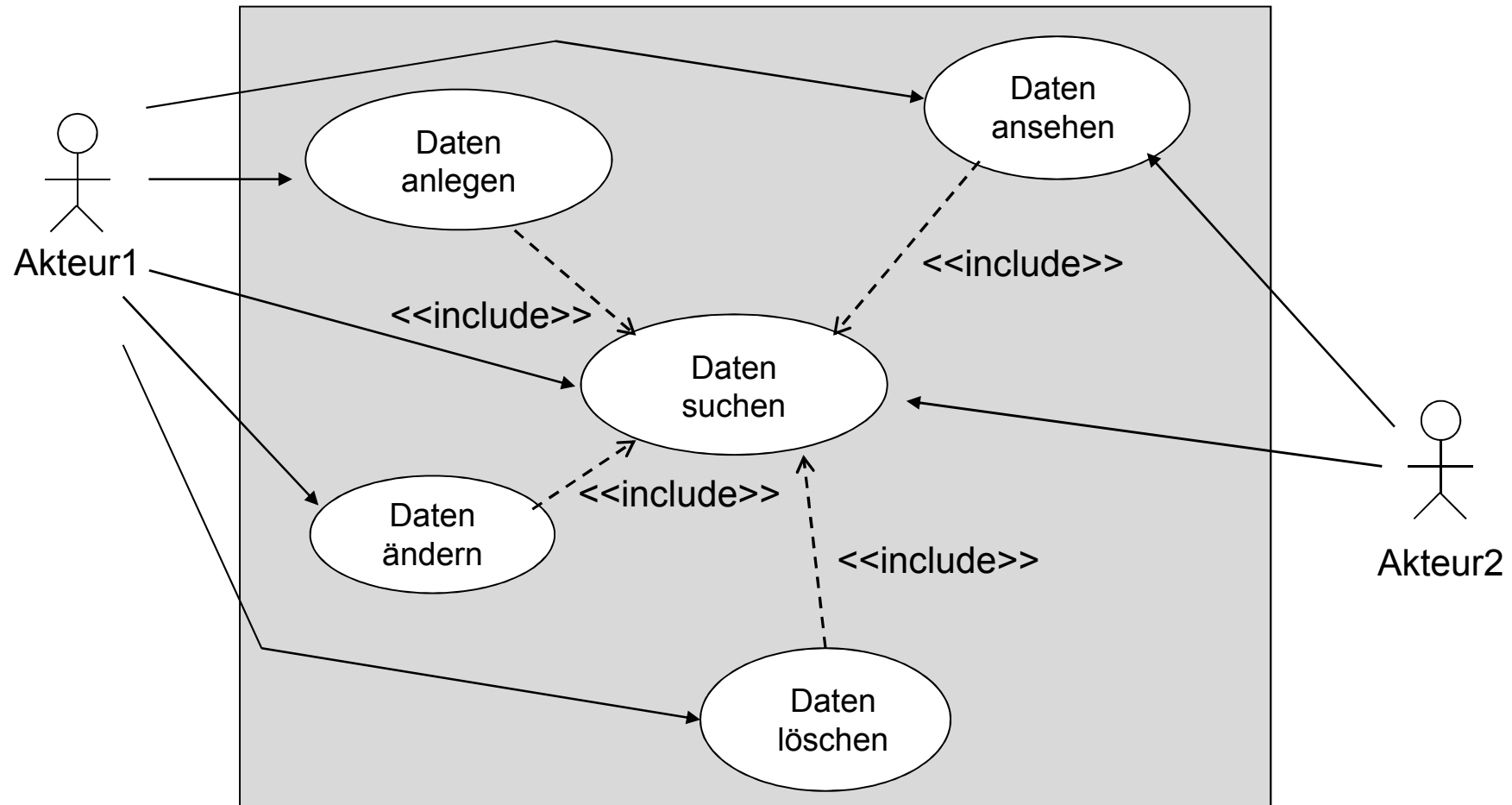
Wozu dienen Use-Cases **nicht**?

- 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.)

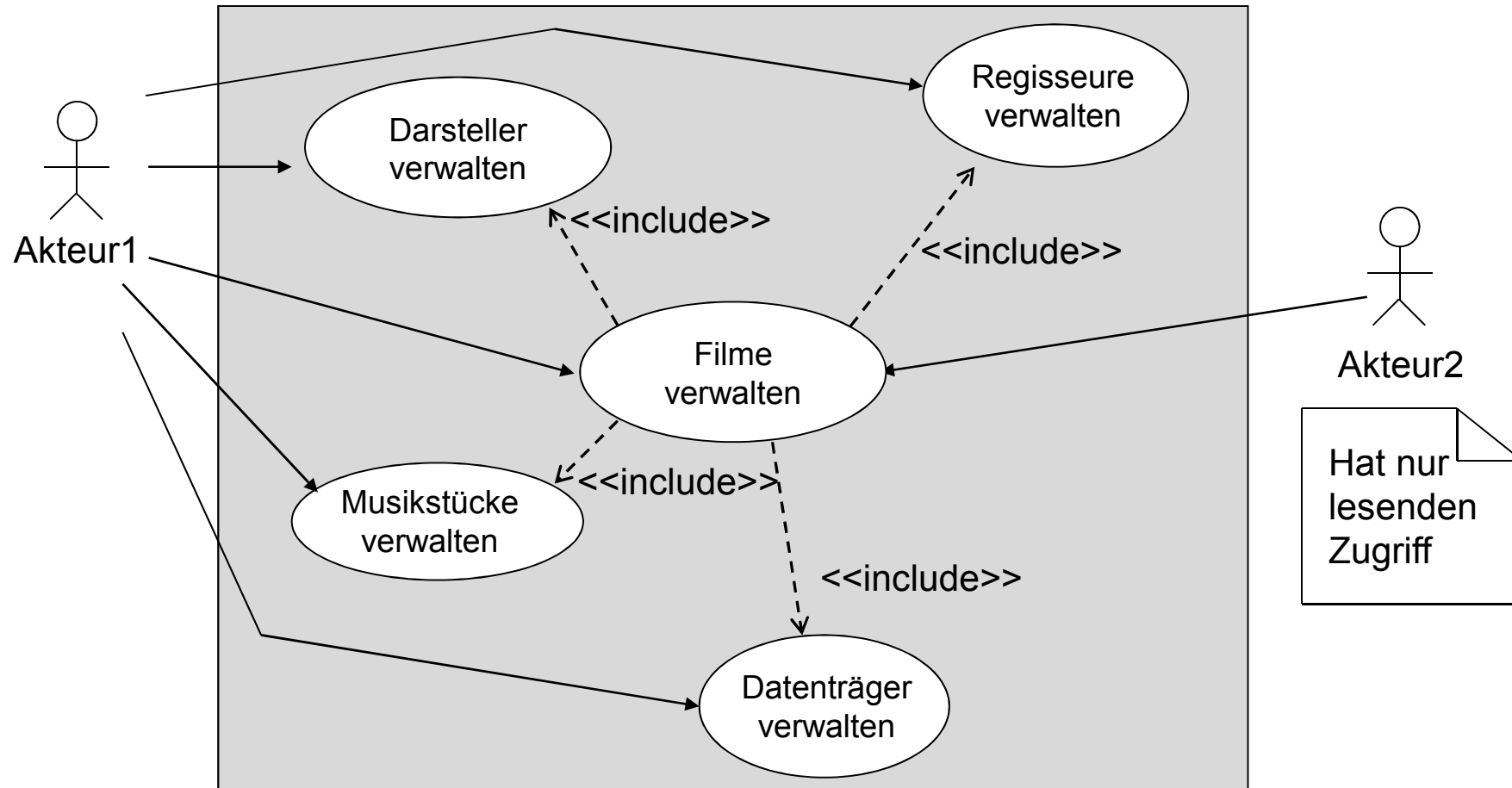
Use-Case-Diagramm: 2 Ansätze

- 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



<<include>>: es werden die „suchen“-Use-Cases verwendet.