

### Use-Case-Diagramm

---

Grundlagen (241,242) | Use-Case (246,247) | System (249,250) | Akteur (251,252) | «include»-Beziehung (256) | «extend»-Beziehung (258) | Vergleich «include»-/«extend»-Beziehung (262)

### Grundlagen (S. 241, 242)

---

- Beginn jeder Systementwicklung
- Use-Case-Diagramm zeigt das externe Verhalten eines Systems aus der Sicht der Nutzer
- Use-Case kapselt geschlossene Sammlung von Aktionen
- Akteur liefert oder empfängt Signale bzw. Daten zum oder vom System
- Use-Case-Ablauf ist abgeschlossener Vorgang mit einem Auslöser und einem Ergebnis
- Akteur initiiert Use-Case

### Use-Case (S. 246, 247)

---

- *A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system*
- **Notation:** Ellipse, Name inner- oder unterhalb
- Menge von Aktionen, die, schrittweise ausgeführt, ein spezielles Verhalten formen

### System (S. 249, 250)

---

- *Subject: Extending a classifier with the capability to own use cases*
- **Notation:** rechteckiger Kasten, Kanten sind Systemgrenzen
- Einheit, die Use-Cases realisiert und anbietet

### Akteur (S. 251, 252)

---

- *An actor specifies a role played by a user or any other system that interacts with the subject.*
- **Notation:** Strichmännchen, Namen oberhalb oder unterhalb
- Interagiert mit System, steht immer außerhalb
- Rolle
- In einem Use-Case-Diagramm interagiert ein Akteur mit einem Use-Case, indem er dessen Ausführung anstößt oder an der Ausführung aktiv oder passiv beteiligt ist. Zwischen dem Use-Case und dem Akteur findet ein wechselseitiger Austausch von Signalen und Daten statt (binäre Assoziation)

### «include»-Beziehung (S. 256)

---

- *An include relationship defines that a use case contains the behavior defined in another use case*
- **Notation:** unterbrochene gerichtete Kante mit dem Schlüsselwort «include»
- Die «include»-Beziehung visualisiert, dass ein Use-Case A das Verhalten eines anderen Use-Case B importiert

### «extend»-Beziehung (S. 258)

---

- *Extend: 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*  
*Extension Point: An extension point identifies a point in the behavior of a use case where that behavior can be extended by the behavior of some other (extending) use case, as specified by an extend relationship*
- **Notation:** unterbrochene gerichtete Kante mit der Bezeichnung «extend» vom erweiternden Use-Case zum erweiterten Use-Case
- Die «extend»-Beziehung zeigt an, dass das Verhalten eines Use-Case (A) durch einen anderen Use-Case (B) erweitert werden kann

## Vergleich «include»-/«extend»-Beziehung (262)

- **Notation**



- **Bedeutung**

«include»-Beziehung: Ablauf von A schließt immer Ablauf von B ein  
«extend»-Beziehung: Ablauf von A kann, muss aber nicht durch Ablauf von B erweitert werden

- **Nutzung der Beziehung**

«include»-Beziehung: Ablauf von B kann in verschiedenen Use-Cases genutzt werden  
«extend»-Beziehung: A besitzt neben Normalverhalten auslagerbare Sonderfälle

- **Bedeutung für Modellierung**

«include»-Beziehung: A ist meist unvollständig und wird erst durch Inklusion B zu einem vollständigen Ablauf  
«extend»-Beziehung: A ist vollständig und kann durch B optional erweitert werden

- **Abhängigkeiten**

«include»-Beziehung: A muss B bei der Modellierung berücksichtigen  
B wird unabhängig von A modelliert, um die Nutzung durch weitere Use-Cases sicherzustellen (Wiederverwendbarkeit),  
B muss in sich nicht vollständig sein ("B weiß nicht, durch wen er inkludiert wird")  
«extend»-Beziehung: A muss durch Angabe von Erweiterungspunkten auf die Erweiterung durch B vorbereitet werden  
B wird in sich vollständig und unabhängig von A modelliert ("B weiß nicht, wen er erweitert")