

Objektorientierte Modellierung

ANWENDUNGSFALLDIAGRAMM

Anwendungsfälle

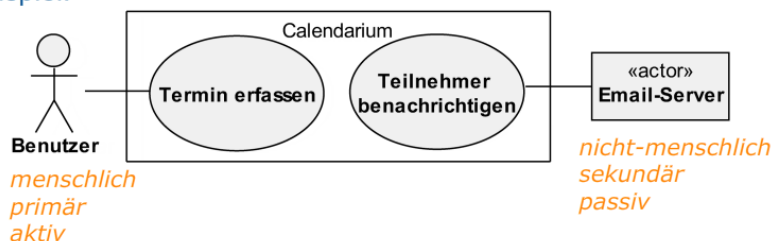
- sind Ausgangspunkt vieler objektorientierter Entwicklungsmethoden, haben aber nicht unbedingt etwas mit objektorientierter Programmierung zu tun
- konzentrieren sich auf das fundamentale Problem bei der Entwicklung eines Systems
- **repräsentieren Anforderungen des Kunden**
- sind typische Fälle, die das System behandeln muss
- **sind Interaktionen zwischen User und System (alles, was ich vornehmen kann, ist ein Use Case)**

Akteure

- **interagieren mit dem System im Kontext der Anwendungsfälle**
 - indem sie das System benutzen
 - indem sie vom System benutzt werden
- werden durch Assoziationen mit Anwendungsfällen verbunden (kommunizieren mit System)
- **müssen mindestens mit einem Anwendungsfall kommunizieren**
- Assoziationen sind binär und können Multiplizitäten aufweisen
 - default: Akteur 1 : N Anwendungsfall
 - **Multiplizitäten werden nur auf der Seite des Akteurs geändert**
- repräsentieren Rollen der Benutzer
 - konkrete Benutzer können gleichzeitig mehrere Rollen haben, annehmen, ablegen
- **befinden sich klar außerhalb der Systemgrenzen**
- Benutzerdaten werden innerhalb des Systems als Klassendiagramm modelliert
- Klassifikation:
 - **Menschlich:** Merkmal des Akteurs
 - **Nicht-Menschlich:** Merkmal des Akteurs
 - **Primär:** Hauptnutznießer der Anwendung, Merkmal der Beziehung
 - **Sekundär:** notwendig für das Funktionieren des Systems, Merkmal der Beziehung
 - **Aktiv:** stößt selbst Anwendungsfälle an, Merkmal der Beziehung
 - **Passiv:** initiiert selbst keine Anwendungsfälle, Merkmal der Beziehung



Beispiel:



Anwendungsfälle:

- 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



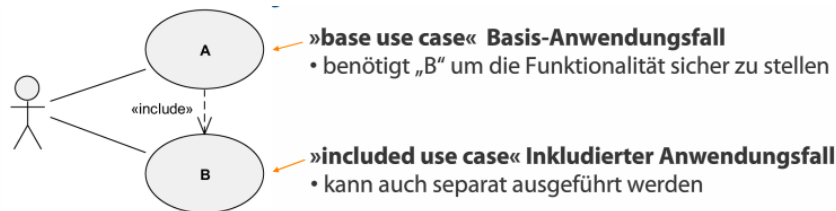
Termin erfassen

»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.«

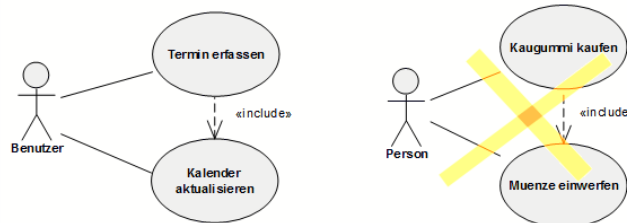
Beziehungen zwischen Anwendungsfällen

include-Beziehung:

- das Verhalten des benutzten Anwendungsfalls (included use case) wird in den benutzenden Anwendungsfall (base use case) eingebunden
- B ist unbedingt notwendig, um die Funktionalität von A sicherzustellen
- B kann separat ausgeführt werden
- Java: direkter Unterprogrammaufruf
- Satzbau: Basis Use Case A inkludiert den Inkludierten Use Case B

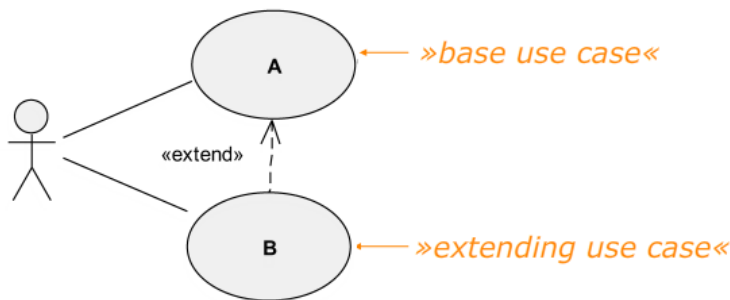


Bsp.

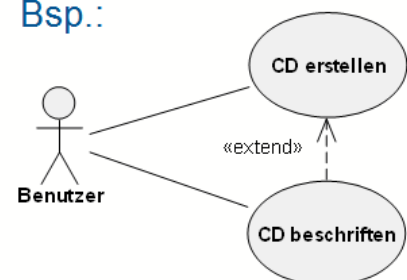


extends-Beziehung:

- das Verhalten von B kann in A inkludiert werden (A entscheidet, ob B ausgeführt wird)
- B kann von A aktiviert werden, muss aber nicht
- A bzw. B können auch separat ausgeführt werden
- Satzbau



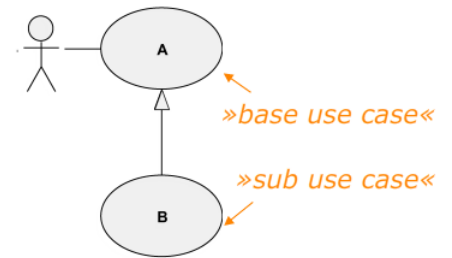
Bsp.:



- mehrere Erweiterungsstellen (extension points) je nach Anwendungsfall möglich
- Namen von Erweiterungsstellen müssen eindeutig sein, aber nicht mit dem Namen der erweiternden Anwendungsfälle übereinstimmen
- Java: Unterprogrammaufruf mittels Condition
- Satzbau: Der erweiternde Use Case B erweitert den Basis Use Case A

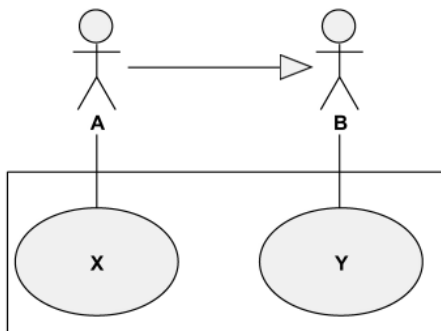
- **Generalisierung bei Anwendungsfällen:**

- 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} sind nicht ausführbar

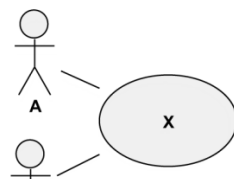


- **Generalisierung zwischen Akteuren:**

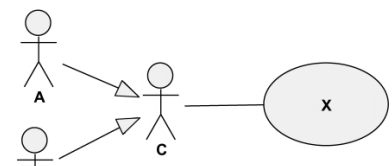
- Akteur A erbt von Akteur B
- A kann mit den Anwendungsfällen X und Y kommunizieren
- Mehrfachvererbung ist erlaubt



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



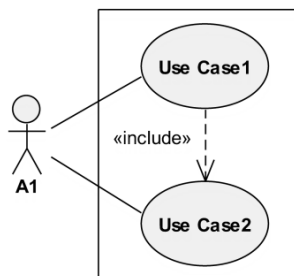
A und B kommunizieren mit X



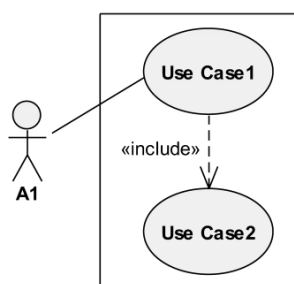
A oder B kommuniziert mit X

- **include/extends – Best Practice:**

Standard



Best Practice



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 muss das System kommunizieren?
- 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
 - es gibt interne, externe und zeitliche Trigger

Typische Modellierungsfehler:

- Anwendungsfälle modellieren keine Abläufe!
- Akteure sind immer außerhalb der Systemgrenzen!
- Generalisierungen müssen richtig verwendet werden!
- Triviale Anwendungsfälle sollen zu einem abstrakten Anwendungsfall zusammengefasst werden!
- Einzelne Schritte eines Anwendungsfalles sollen nicht extra modelliert werden!
- Die Anwendungsfälle sollten auf der gleichen Abstraktionsebene sein!
- Es sollen nur Anwendungsfälle und Akteure modelliert werden, die relevant sind!

Anwendungsfallbeschreibung

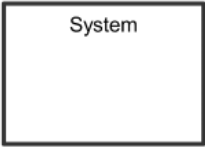
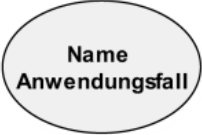

■ Strukturierter Text





- 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

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.

ZUSAMMENFASSUNG

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

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

CHECKLISTE

- 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.