



Objektorientierte Systemanalyse

Systemanalyse

Teil 2: Use Cases

IT works.

Agenda



- ✦ Einführung
 - Basiskonzepte, UML
 - Fokus in der OOA

- ✦ **Use Cases**
 - Use Case Diagramme
 - Textschablone

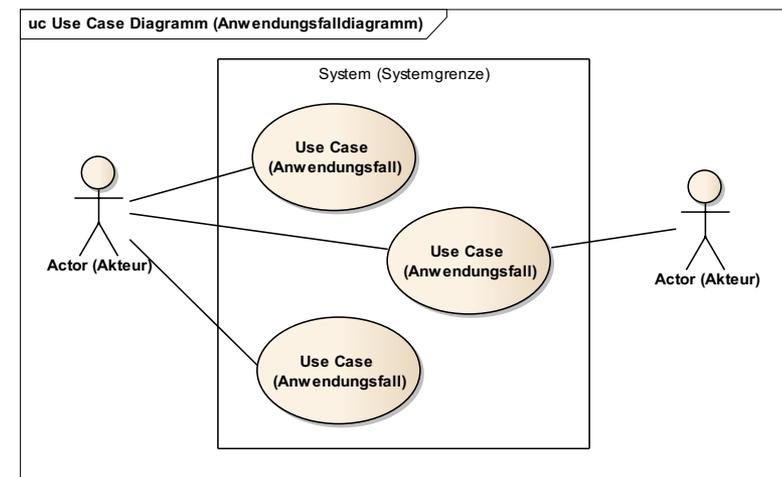
- ✦ **Aktivitätsdiagramme**
 - Grundkonzepte
 - Erweiterungen

- ✦ **Klassendiagramme**
 - Klassendiagramme
 - Paketdiagramme, Komponentendiagramme
 - Objektmodelle, Szenarien, Zustandsautomaten
 - CRC-Karten

- ✦ **UI-Design**
 - Ergonomie, Usability, Gestaltung
 - Ableitung aus Klassendiagrammen
 - Beschreibung über Zustandsautomaten

Use Cases und Use Case Diagramme

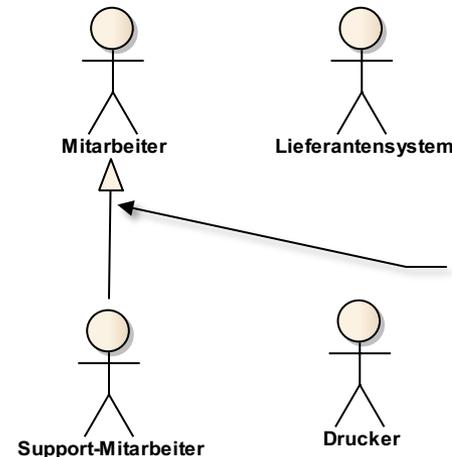
- Use Cases und Use Case-Diagramme haben das Ziel, das Verhalten eines Systems **aus Anwendersicht** auf einem sehr hohen Abstraktionsniveau darzustellen und zu beschreiben. Dies dient
 - dem Verständnis der fachlichen Anforderungen durch die Entwickler und
 - der Vorstellung des Auftraggebers über das Systemverhalten
- Dabei erfolgt die Abgrenzung:
 - Was liegt außerhalb des Systems? Wer interagiert mit dem System? Diese werden als **Akteure** bezeichnet.
 - Was leistet das System? Welche fachlichen, in sich abgeschlossenen, Funktionalitäten werden angeboten. Diese werden als **Use Cases** (oder auf Deutsch: **Anwendungsfälle**) bezeichnet.
- Beispiel für ein Use Case Diagramm mit
 - Akteuren
 - der Darstellung der Systemgrenze, bzw. des Systems
 - Use Cases
 - der Darstellung, welche Akteure mit welchen Use Cases interagieren



Akteure

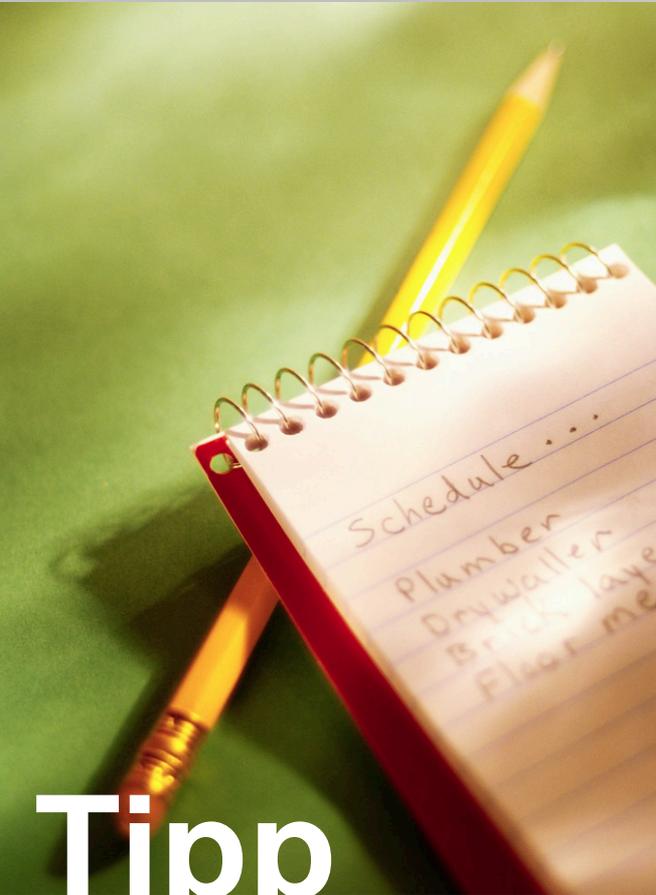
- Ein Akteur ist eine Rolle, die ein Benutzer des Systems wahrnimmt und über die er durch Interaktion Einfluss auf das System nimmt.
 - Akteure müssen nicht unbedingt reale Personen darstellen, sondern sie können auch andere Systeme und somit Schnittstellen repräsentieren.
 - Akteure befinden sich grundsätzlich außerhalb des zu erstellenden Systems und sind daher nicht Bestandteil des Systems. Trotzdem kann es im System eine Implementierung geben, die dem Akteur entspricht, z.B. der Benutzer, der sich authentifiziert oder die Schnittstellen-Implementierung zu einem System.
 - Akteure tauschen mit dem System Informationen aus bzw. erhalten vom System Informationen oder senden Information an das System.

- Im Use Case-Diagramm werden Akteure als Strichmännchen dargestellt.
 - Dabei können auch Spezialisierungen von Rollen modelliert werden (siehe Beispiel Mitarbeiter und die spezialisierte Rolle Support-Mitarbeiter).



Der Pfeile zeigt stets von der Spezialisierung zur Generalisierung.

Wer ist Akteur? Wie identifiziere ich Akteure?



Tipp

Folgende Fragestellungen können bei der Identifikation von Akteuren im realen Aufgabenumfeld helfen:

- Welche Personen führen welche Aufgaben durch und besitzen daher wichtige Kenntnisse über die notwendigen Arbeitsabläufe? Welche Rollen nehmen diese Personen ein?
- Welche Personen werden zukünftig diese Aufgaben durchführen und welche Rollen spielen diese Personen?
- Wo befindet sich die Schnittstelle des betrachteten Systems bzw. was gehört nicht mehr zum System, wo verläuft die Systemgrenze?
- Wer nutzt Leistungen des Systems?
- Welche IT-Systeme benutzt das System? Welche IT-Systeme beliefert es?

Übung 1: Identifikation von Akteuren



Aufgabe: Identifizieren Sie in der folgenden Problembeschreibung mögliche Akteure

Eine Bibliothek ist zu verwalten. Jeder registrierte Leser kann sich Bücher ausleihen.

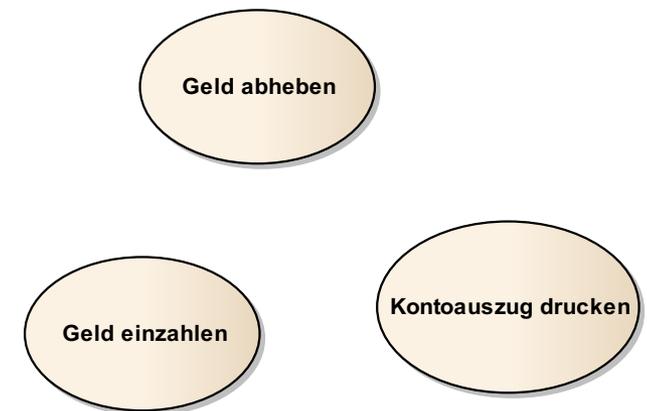
Sind vom gewünschten Buch alle Exemplare verliehen, so kann ein Exemplar vorbestellt werden. Ein Buch kann zu einem Zeitpunkt von mehreren Lesern vorbestellt sein, d.h. es wird eine Warteliste gebildet. Wird ein vorbestelltes Buch zurückgegeben, dann wird der erste Leser auf der Warteliste benachrichtigt und das Buch wird vom Mitarbeiter an der Ausgabetheke zurückgelegt. Reservierte Bücher, die nach einer Woche nicht abgeholt wurden, werden wieder zur Ausleihe bereitgestellt oder der nächste Leser auf der Warteliste wird informiert. Bücher, die nicht in der Bibliothek vorhanden sind, können über Fernleihe in anderen Bibliotheken gesucht und ggf. angefordert werden.

Für jeden Leser werden der Name und die Adresse gespeichert. Bei der Ausgabe der Bücher wird das Ausleihdatum und das Rückgabedatum gespeichert. Bei allen Büchern, deren Ausleihfrist um eine Woche überschritten ist, werden deren Leser automatisch gemahnt.

Zurückgegebene Bücher werden von den Bibliotheksmitarbeitern zunächst auf Beschädigung geprüft, bevor sie wieder bereitgestellt werden. Neue Bücher bzw. Buchexemplare werden vom Bibliothekar inventarisiert, bevor sie zur Ausleihe bereitgestellt werden.

Use Case (dt. Anwendungsfall)

- Ein Use Case steht stellvertretend für mehrere zusammenhängende Aufgaben oder Arbeitsschritte, welche von einem Akteur mit dem Ziel durchgeführt werden, ein bestimmtes fachliches Ergebnis zu erreichen. Diese werden unter einem beschreibenden Namen zusammengefasst, der aus einem Subjekt und einem aktiven Verb bestehen sollte (z.B. Geld abheben, Buchung stornieren, Bestellung aufgeben).
- Ein Use Case steht für
 - eine spezifische Art der Benutzung eines Softwaresystems
 - die Anforderungen an das Softwaresystem im Hinblick auf sein externes Verhalten
 - den Ablauf eines Vorgangs durch die Interaktion des Akteurs mit dem System
- Ein Use Case hat immer das Ziel, ein sichtbares und sinnvolles fachliches Ergebnis zu erreichen.
- Use Cases werden in einem weiteren Schritt durch eine textuelle Beschreibung des Ablaufs (Standardfall) und alternativer Abläufe (Varianten) weiter detailliert.



Wie finde ich Use Cases?

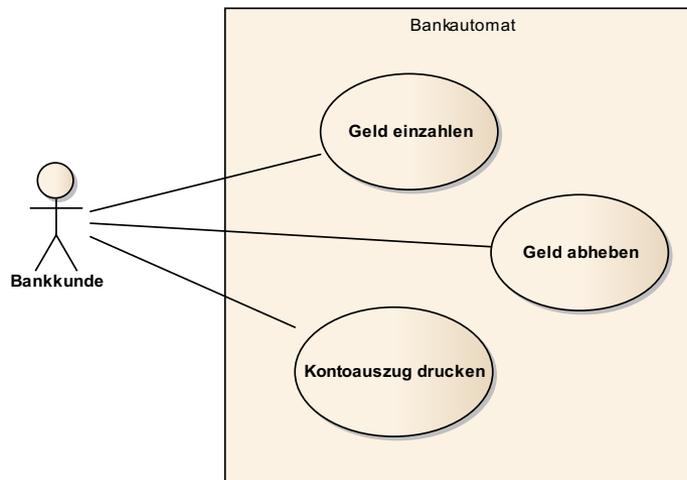


Jacobson empfiehlt, bei der Identifikation von Anwendungsfällen von den Akteuren des Systems auszugehen. Hierbei helfen folgende Fragestellungen:

- Welches Ereignis löst einen Arbeitsablauf aus?
- Welche Eingabedaten werden benötigt?
- Welche Aktivitäten hat der Akteur mit dem System auszuführen?
- Informiert der Akteur das System über Veränderungen in der Umgebung?
- Welche Informationen über Veränderungen im System benötigt der Akteur?

Use Case Diagramm

- Das Use Case Diagramm gibt einen Überblick über das System, seine Schnittstellen und die Interaktionen mit den Akteuren und stellt dies auf einem sehr hohen Abstraktionsniveau dar. Use Case Diagramme umfassen Akteure, Use Cases und Systemgrenzen. Die Linie (genannt Assoziation) zwischen Akteur und Use Case bedeutet Interaktion/Kommunikation des Akteurs mit dem System im Rahmen des im jeweiligen Use Case beschriebenen Szenarios.
- Die Schwierigkeit besteht darin, das richtige Abstraktionsniveau zu treffen. In der Praxis findet man sehr häufig Use Case-Diagramme auf unterschiedlichsten Abstraktionsniveaus. Der häufigste Fehler besteht darin, dass Use Cases zu feingranular definiert werden und nur noch wenige oder einen einzigen Arbeitsschritt beinhalten.



Wichtig:

Die Anordnung der Use Cases im Diagramm stellt keinen zeitlichen Ablauf dar!
Jeder Use Case ist in sich abgeschlossen und führt zu einem eigenen fachlichen Ergebnis.

Übung 2: Use Cases und Use Case-Diagramm



Aufgabe: Identifizieren Sie aus folgender Problemstellung mögliche Use Cases und stellen Sie diese zusammen mit den Akteuren in einem Use Case-Diagramm dar.

Eine Bibliothek ist zu verwalten. Jeder registrierte Leser kann sich Bücher ausleihen.

Sind vom gewünschten Buch alle Exemplare verliehen, so kann ein Exemplar vorbestellt werden. Ein Buch kann zu einem Zeitpunkt von mehreren Lesern vorbestellt sein, d.h. es wird eine Warteliste gebildet. Wird ein vorbestelltes Buch zurückgegeben, dann wird der erste Leser auf der Warteliste benachrichtigt und das Buch wird vom Mitarbeiter an der Ausgabetheke zurückgelegt. Reservierte Bücher, die nach einer Woche nicht abgeholt wurden, werden wieder zur Ausleihe bereitgestellt oder der nächste Leser auf der Warteliste wird informiert. Bücher, die nicht in der Bibliothek vorhanden sind, können über Fernleihe in anderen Bibliotheken gesucht und ggf. angefordert werden.

Für jeden Leser werden der Name und die Adresse gespeichert. Bei der Ausgabe der Bücher wird das Ausleihdatum und das Rückgabedatum gespeichert. Bei allen Büchern, deren Ausleihfrist um eine Woche überschritten ist, werden deren Leser automatisch gemahnt.

Zurückgegebene Bücher werden von den Bibliotheksmitarbeitern zunächst auf Beschädigung geprüft, bevor sie wieder bereitgestellt werden. Neue Bücher bzw. Buchexemplare werden vom Bibliothekar inventarisiert, bevor sie zur Ausleihe bereitgestellt werden.

Use Case Beschreibung

- ✚ Use Case Beschreibungen stellen eine weitere Detaillierung der einzelnen Use Cases dar.
 - Zu jedem Use Case wird der normale Arbeitsablauf (der Standardfall, 90%-Fall, ...) eines Use Case in nummerierten Einzelschritten beschrieben. D.h.: Wie bzw. durch welche Arbeitsschritte erreicht der Akteur in Zusammenarbeit mit dem System das fachlich vorgegebene Ziel.
 - Zusätzlich können alternative Abläufe (Sonderfälle) beschrieben werden.

- ✚ In der Praxis haben sich Formulare zur Beschreibung von Use Cases durchgesetzt. Diese umfassen neben dem eigentlichen Ablauf noch weitere Informationen, wie z.B. das fachliche Ziel, das auslösende Ereignis, eventuelle Vorbedingungen etc.

- ✚ Die Use Case-Beschreibung sollte ausschließlich aus Sicht des Anwenders erstellt und in seinem Fachvokabular geschrieben werden. Technische Implementierungen/Begriffe sollen nicht verwendet werden.
 - **Richtig:** Der Anwender wählt die Art seiner Zahlungsweise. Möglich sind Vorkasse, Kreditkarte, Paypal.
 - **Falsch:** Der Anwender selektiert die Zahlungsweise mit der Maus aus einer Combobox mit den Inhalten Vorkasse, Kreditkarte und Paypal.

Use Case Beschreibung: Die gängige Textschablone

Ziel:

Grobe Zielsetzung bei erfolgreicher Ausführung des Use Case.

Akteure:

Rollen von Personen oder Systemen, die den Use Case auslösen oder beteiligt sind.

Auslösendes Ereignis:

Wenn dieses Ereignis eintritt, wird der Use Case initiiert.

Vorbedingung

Erwarteter Zustand des Systems vor der Ausführung des Use Case

Nachbedingung

Erwarteter Zustand des Systems nach Ausführung des Use Case

Ablauf

Textuelle Beschreibung des internen Ablaufs. Diese Beschreibung sollte durch Unterpunkte hierarchisch gegliedert sein.

1. ...
 2. ...
- usw.

Alternative Abläufe

- 2.a ...
- 2.b ...

Use Case Beschreibung: Die gängige Textschablone – ein Beispiel

Ziel:

Überweisung am Automaten durchführen

Akteure:

Bankkunde

Auslösendes Ereignis:

Der Kunde möchte eine Überweisung durchführen.

Vorbedingung

Die EC-Karte des Kunden ist bereits eingeschoben und der Kunde hat seine PIN korrekt eingegeben.

Nachbedingung

Der Überweisungsbetrag wurde an das Buchungssystem der Bank übertragen.

Ablauf:

1. Der Kunde gibt die Kontodaten des Empfängers mit Name und IBAN ein.
2. Der Kunde gibt den Betrag für die Überweisung in Euro ein.
3. Der Kunde gibt den Bemerkungstext für die Überweisung ein (max. 120 Zeichen).
4. Der Kunde wählt das Ausführungsdatum für die Überweisung (sofort oder Datum).
5. Der Kunde bestätigt die Eingaben.
6. Das System prüft die eingegebenen Daten auf Vollständigkeit sowie Korrektheit und ermittelt die zur IBAN gehörige Bank. Die vervollständigten Daten werden angezeigt.
7. Der Kunde bestätigt nochmals den Überweisungsauftrag.
8. Das System überträgt die Daten für den Überweisungsauftrag an das Buchungssystem der Bank.
9. Das System informiert den Kunden über die erfolgreiche Datenübertragung.

Übung 3: Use Case Beschreibungen erstellen



Aufgabe: Beschreiben Sie die Use Cases zur Registrierung eines Lesers und zur Rückgabe eines Buches. Nutzen Sie zur Beschreibung der beiden Use Cases die vorgestellte Text-Schablone.

Eine Bibliothek ist zu verwalten. Jeder registrierte Leser kann sich Bücher ausleihen.

Sind vom gewünschten Buch alle Exemplare verliehen, so kann ein Exemplar vorbestellt werden. Ein Buch kann zu einem Zeitpunkt von mehreren Lesern vorbestellt sein, d.h. es wird eine Warteliste gebildet. Wird ein vorbestelltes Buch zurückgegeben, dann wird der erste Leser auf der Warteliste benachrichtigt und das Buch wird vom Mitarbeiter an der Ausgabetheke zurückgelegt. Reservierte Bücher, die nach einer Woche nicht abgeholt wurden, werden wieder zur Ausleihe bereitgestellt oder der nächste Leser auf der Warteliste wird informiert. Bücher, die nicht in der Bibliothek vorhanden sind, können über Fernleihe in anderen Bibliotheken gesucht und ggf. angefordert werden.

Für jeden Leser werden der Name und die Adresse gespeichert. Bei der Ausgabe der Bücher wird das Ausleihdatum und das Rückgabedatum gespeichert. Bei allen Büchern, deren Ausleihfrist um eine Woche überschritten ist, werden deren Leser automatisch gemahnt.

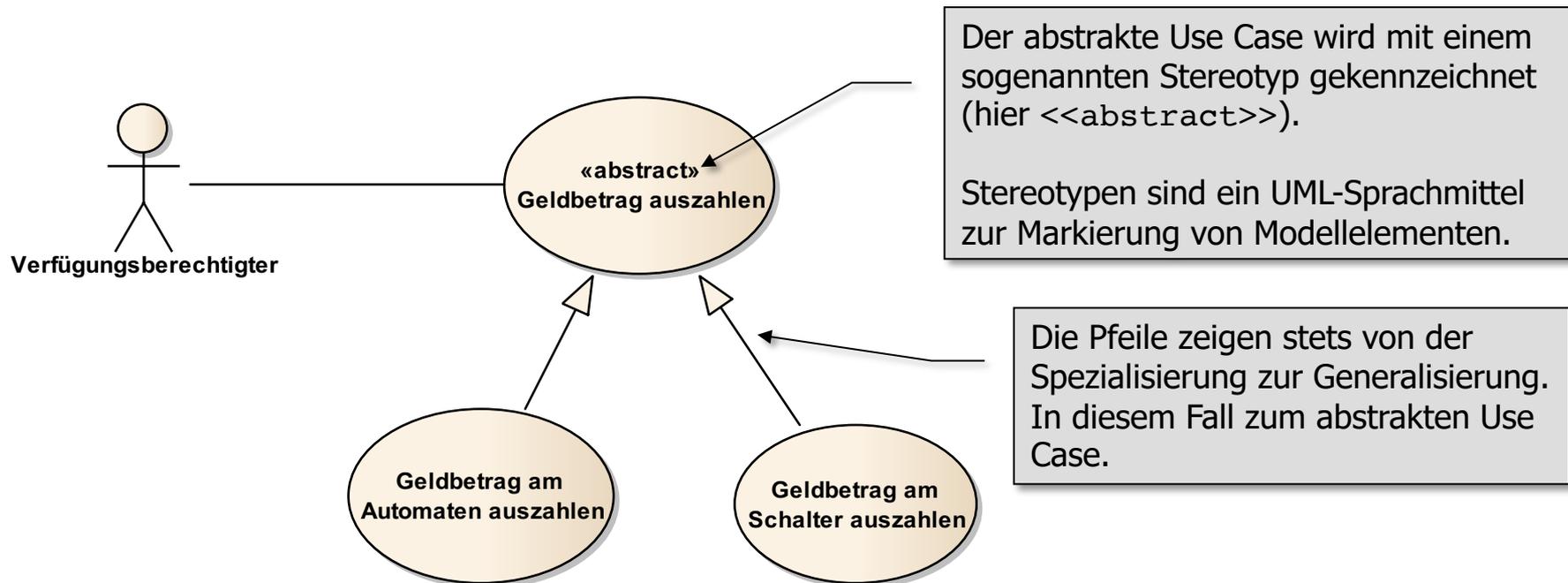
Zurückgegebene Bücher werden von den Bibliotheksmitarbeitern zunächst auf Beschädigung geprüft, bevor sie wieder bereitgestellt werden. Neue Bücher bzw. Buchexemplare werden vom Bibliothekar inventarisiert, bevor sie zur Ausleihe bereitgestellt werden.

Use Cases und ihre unterschiedlichen Ausprägungen

- Der normale, typische Use Case wird als konkreter oder **primärer Use Case** bezeichnet. Sein Ablauf führt mit konkreten und vollständigen Arbeitsschritten zu einem definierten fachlichen Ergebnis.
- Darüber hinaus gibt es sogenannte **abstrakte Use Cases**. Diese fassen generalisiert die Gemeinsamkeiten mehrerer normaler Use Cases unter einem Use Case zusammen. Die Bezeichnung als abstrakt kommt daher, dass sie keinen konkreten Arbeitsablauf beschreiben.
- Sogenannte **sekundärer Use Cases** beschreiben Teile oder einzelne Arbeitsschritte/-folgen, die in mehreren Use Cases gemeinsam vorkommen (include) oder einen Use Case unter bestimmten Voraussetzungen ergänzen (extend). Ein sekundärer Use Case ist fachlich immer unvollständig.
- Darüber hinaus können Use Cases ganz unterschiedlich verwendet werden, z.B.
 - eine Ist-Situation beschreiben (Ist-Anwendungsfall)
 - einen Soll-Zustand beschreiben (Soll-Anwendungsfall)
 - eine Essenz (nur den wesentlichen Ablauf) beschreiben (essentieller Anwendungsfall)
 - den fachlichen Kontext beschreiben, d.h. die Einbettung des zu entwickelnden Systems seine Umwelt
 - nur die durch Software unterstützten Sachverhalte beschreiben
 - auch außerhalb der Software allgemeine geschäftliche Anwendungsfälle darstellen (Business Use Case, Geschäftsprozess)
 - ganz einfach und rudimentär verwendet werden (Name und Kurzbeschreibung, kein Ablauf)

Use Cases und ihre unterschiedlichen Ausprägungen - Beispiele

Abstrakter Use Case

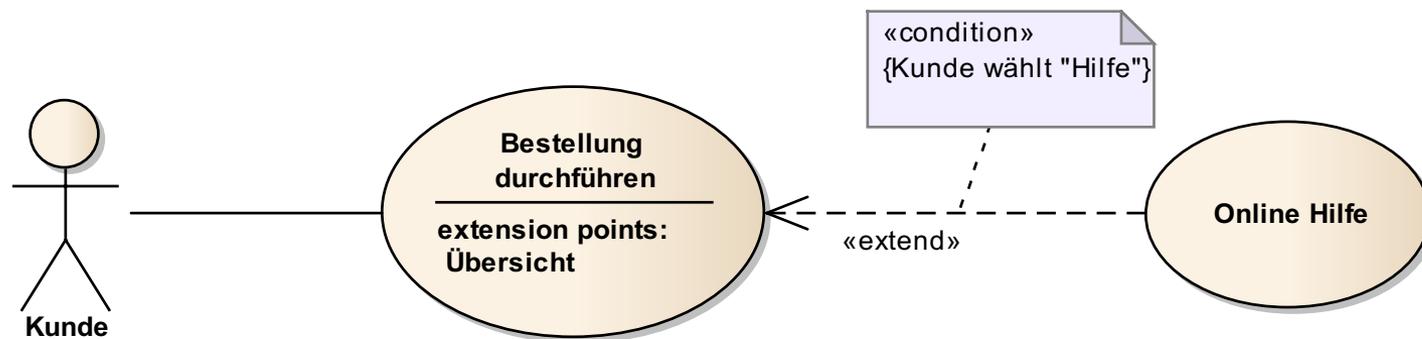


Use Cases und ihre unterschiedlichen Ausprägungen - Beispiele

Sekundärer Use Case durch <<extend>>-Beziehung

Extends sind eine Möglichkeit, einen primären Use Case durch einen sekundären Use Case fachlich zu erweitern. Das bedeutet, dass unter einer angegebenen Bedingung zusätzliche Arbeitsschritte ausgeführt werden.

Hierbei gilt, dass der primäre Use Case auch ohne den Extend vollständig ist. Man spricht von einer losen Kopplung zwischen den Use Cases.



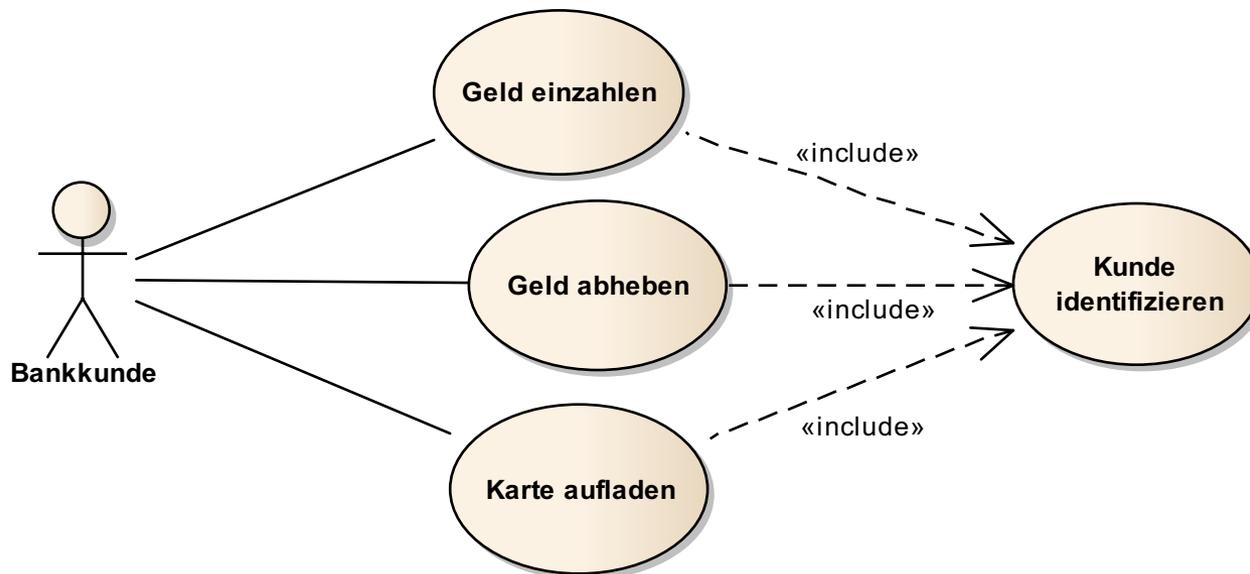
Vorsicht: Es besteht die Gefahr, dass man eine funktionale Dekomposition durchführt. Das heißt, man zerlegt Use Cases unnötigerweise in einzelne Teile (sekundäre Use Cases), ähnlich Unterprogrammen in der strukturierten Programmierung, ohne dass dies notwendig wäre.

programmieren in der strukturierten Programmierung, ohne dass dies notwendig wäre.
man zerlegt Use Cases unnötigerweise in einzelne Teile (sekundäre Use Cases), ähnlich Unter-

Use Cases und ihre unterschiedlichen Ausprägungen - Beispiele

Sekundärer Use Case durch <<include>>-Beziehung

Wird eine bestimmte Arbeitsschrittfolge in mehreren Use Cases benötigt, kann sie separiert und als sekundärer Use Case beschrieben werden. Der sekundäre Use Case wird mit einer <<include>>-Beziehung in die primären Use Cases eingebunden. Man spricht hier von einer starken Kopplung, da die primären Use Cases ohne die Arbeitsschritte des sekundären Use Case fachlich unvollständig sind.



Vorsicht: Auch hier besteht die Gefahr einer funktionalen Dekomposition!

Use Cases und ihre typischen Probleme

In der Praxis gibt es häufig Probleme in der Anwendung von Use Cases. Diese sind in der Regel begründet durch eine fehlende Standardisierung von

- Detaillierungsgraden
- Abstraktionsebenen, Granularitäts- und Zerlegungsstufen
- Fachliche Terminologie und Sprachverwendung
- Strukturierungs- und Gliederungsschemata

Use Cases treten aus diesem Grund in extrem unterschiedlicher Qualität auf.
Daher sind sie nicht oder nur schwer zu bewerten oder zu vergleichen.

Fehler, Lücken und Redundanzen oft nur schwer erkennbar.

Fehler, Lücken und Redundanzen oft nur schwer erkennbar.

Use Cases und ihre korrekte Verwendung



Tipp

- Sinnvolle Bezeichnung für einen Use Case wählen: mindestens ein echtes Substantiv und ein aktives Verb („Antrag erfassen“).
- Zuerst den fachlichen Auslöser und das fachliche Ergebnis definieren, um den Anfang und das Ende des Use Case zu definieren.
- Den Use Case so abstrakt wie möglich, aber so konkret wie nötig beschreiben. Nicht immer einfach ... zugeben.
- Am Anfang keine Zerlegung in abstrakte oder sekundäre Use Cases. Besser ist es, auf abstrakte und sekundäre Use Cases ganz zu verzichten.
- Die Sprache in den Use Cases standardisieren. Einfache, knappe Sätze bilden. Gleiche Begriffe für die selbe Sache verwenden, keine Synonyme. Wertausprägungen/Auswahlmöglichkeiten möglichst vollständig aufzählen.
- Use Cases eignen sich nicht zur funktionalen Zerlegung, d.h. ein Use Case beschreibt keine einzelnen Schritte, Operationen, etc. (z.B. „Vertrag drucken“), sondern relativ große Abläufe (z.B. „neuen Kunden aufnehmen“).
- Use Cases belassen das Sprachmonopol beim Stakeholder. Keine technischen Begriffe der Umsetzung verwenden. Die Fachsprache des Stakeholders verwenden! Ggf. hilft ein separat angelegtes Glossar.

Typische Fehlerquellen in der Anwendung von Use Cases



Tipp

Typische Fehler bei der Identifikation und Beschreibung von Anwendungsfällen sind:

- zu kleine und damit zu viele Anwendungsfälle
- zu frühe Betrachtung von Sonderfällen
- zu detaillierte Beschreibung der Anwendungsfälle
- verwechseln von include- und extend-Beziehungen
- Beschreibung von Dialogabläufen durch die Anwendungsfälle

Besser:

- Essentielle Use Cases mit der Beschreibung des Standardablaufs (ohne alternative Abläufe)
- Möglichst knappe Beschreibungen der Arbeitsschritte (max. 3-5 Sätze als Ziel).
- Verfeinerung der Use Cases im nächsten Schritt über Aktivitätsdiagramme. Hierbei können dann alle alternativen Abläufe/Varianten berücksichtigt werden.

Weiterführende Aufgaben

- Die Übungsaufgaben in dieser Präsentation bearbeiten. Den Link zu einem Dokument mit den dazugehörigen Lösungen finden Sie auf meiner Homepage. Die Lösungsansätze werden dort auch diskutiert, da es insbesondere in der Modellierung immer unterschiedliche Ansätze geben kann und beispielhafte Problemstellungen stets Interpretationsspielraum zulassen.

- Kapitel 3.1 (Seiten 57-68) im Buch „Objektorientierte Systemanalyse“ lesen sowie die Aufgaben 2 und 3 in Kapitel 3.1.5 bearbeiten.

- Weiterführende Recherche:
 - Ivar Jacobson ist der „Erfinder“ der Use Cases. Recherchieren Sie auf seiner Webseite <https://usecase.ivarjacobson.com/> zu seinem neuesten Ansatz „Use Cases 2.0“. Was versteht er darunter? Worin unterscheidet sich der Ansatz und wofür ist er gedacht?
 - Suchen Sie nach wissenschaftlichen Artikeln zu Use Cases, die sich kritisch mit dem Ansatz auseinandersetzen. Worin werden Probleme gesehen, welche Vorteile werden beschrieben? Mögliche Ansatzpunkte sind folgende Artikel:
 - [A Critique of UML's Definition of the Use Case Class](#) (Conference Paper, Lecture Notes in Computer Science, 2003)
 - [Auslaufmodell Use Case?](#) (Artikel in der Fachzeitschrift netzwelt)

Vielen Dank für Ihre Aufmerksamkeit. Haben Sie noch Fragen?



Kontakt

Dr. Nikolaus Mairon

klaus@mairon-online.de
Tel. +49 (0) 160 96678776
Skype-ID: klaus@mairon-online.de

msg nexinsure ag

Geschäftsstelle St. Georgen
Leopoldstr. 1
78112 St. Georgen

nikolaus.mairon@msg.group
Tel. +49 (0) 89 96101-3004
Mobil +49 (0) 171 9716462

IT works.