

5 Anwendungsfalldiagramm

Inhalt

5.1 Überblick Use Case / Anwendungsfall	-2
5.2 Definitionen.....	-7
5.3 Akteure, Anwendungsfälle und Kommunikationsbeziehungen.....	-8
5.4 Beziehungen zwischen Anwendungsfällen	-12
5.5 Beziehungen zwischen Akteuren	-19
5.6 Beschreibung eines Anwendungsfalls	-21
5.7 Identifizierung von Anwendungsfällen	-24
5.8 Die wichtigsten Regeln zur Anwendungsfallmodellierung	-25

5.1 Überblick Use Case / Anwendungsfall

- ❑ Use Cases oder Anwendungsfälle sind Ausgangspunkt vieler objektorientierten Entwicklungsmethoden.
- ❑ Zusätzlich dienen sie oft auch als Basiskonzept das sich über den kompletten Analyse- und Designprozess hinweg spannt.
- ❑ Die Ausgangsfragen für den Einsatz von Anwendungsfällen sind:
 - Warum verwendet man Anwendungsfälle?
 - Wie sehen Anwendungsfälle aus?
 - Was macht man mit Anwendungsfällen, wenn man sie einmal definiert hat?
- ❑ Anwendungsfälle konzentrieren sich auf das fundamentale Problem bei der Entwicklung eines Systems: der Entwicklung einer Lösung für den Kunden bzw. Anwender, die der Kunde bzw. der Anwender auch gewünscht hat.

- Zu einem besseren Verständnis des Ansatzpunktes von Anwendungsfällen vergleichen wir, wie zwei Modellierer - einer mit Schwerpunkt Strukturierte Analyse und einer mit Schwerpunkt Objektorientierung - ein Auto sehen:

Modellierer	Beschreibung eines Autos
Strukturierte Analyse	Ein Benzintropfen gelangt vom Tank über die Benzinleitung zum Motor. Der Motor wandelt das Benzin in Energie um, wodurch sich der Motor zu bewegen beginnt. Dadurch dreht sich auch das Getriebe um ein Stück. Dies veranlasst auch eine kleine Drehung der Kurbelwelle. Das an der Kurbelwelle angebrachte Differential dreht sich somit auch ein Stück. Das Differential sitzt auf den Reifen wodurch sich auch die Reifen um ein Stück drehen. Dadurch bewegt sich das Auto ein Stück vorwärts
Objekt orientiert	Der Fahrer steigt auf das Gaspedal, wodurch sich das Auto fortbewegt. Der Motor stoppt sobald der Fahrer die Zündung ausschaltet.

- Somit ergibt sich auch ein unterschiedlicher Ansatz in der Systementwicklung zwischen strukturierter Analyse und einem Anwendungsfall-zentriertem Ansatz.

- Dabei können wir die Entwicklung eines Systems in zwei Phasen beschreiben.
 - Die erste Phase dient zur Beschreibung der fokussierten Geschäftsumgebung
 - Die zweite Phase dient zur Umsetzung dieser Geschäftsumgebung in einem System bzw. in einer Anwendung.
- In der ersten Phase erfolgt
 - die Identifizierung des Problems
 - analytisch ausgerichtete Erfassung der Systemanforderungen mittels
 - Process Decomposition
 - Functional Decomposition
 - Business Flow-Charts
 - Workflow-Diagrammen
 - Business Process Reengineering Aktivitäten
 - etc.
- Am Ende der ersten Phase entsteht ein guter Überblick über die Geschäftsumgebung. Es entstehen Ansatzpunkte zur Schaffung eines Systems.

- ❑ Während der ersten Phase ist der Anwender ständig eingebunden. Der Anwender treibt dabei den Prozess.
- ❑ Auf Basis der Erkenntnisse der ersten Phase gestaltet sich die zweite Phase:
 - Zuerst erfolgt die Modellierung von einer datenorientierten Sichtweise: Es entstehen ER-Diagramme und Datenflussdiagramme.
 - Auf Basis der datenorientierten Diagramme werden Diagramme erstellt, die festhalten wie die Daten manipuliert werden.
 - Danach muss auf diese Diagramme noch ein entsprechendes User Interface aufgesetzt werden.
 - Letztendlich entsteht am Ende der zweiten Phase das Anwendungssystem.

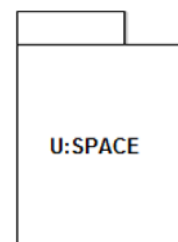
- ❑ Mit den Prozessen in der zweiten Phase ist vorwiegend die IT-Abteilung befasst. Dadurch ergeben sich folgende Probleme:
 - Die IT-Abteilung interpretiert die Anwenderanforderungen aus ihrer eigenen, meist rein technischen Sichtweise
 - Dadurch erfolgt die Systementwicklung vor allem von einer IT-Perspektive und nicht von einer Anwender-Perspektive
 - Das Risiko, ein dem Anwenderwunsch nicht entsprechendes System zu schaffen, ist hoch.
 - Weiters können durch eine scharfe Trennung der beiden Phasen neue Erkenntnisse und Änderungen in der Geschäftsumgebung nicht in das sich entwickelnde System einfließen
- ❑ Es bedarf einer Entwicklungsmethode, in welcher der Anwender bis zum Ende involviert bleibt und die flexibel genug ist, um auf Umgebungsänderungen zu reagieren.
- ❑ Die Verwendung einer Anwendungsfall-zentrierten Methode erfüllt diese Anforderungen. Sämtliche entstehende Dokumente der Systemdokumentation müssen immer gegen den entsprechenden Anwendungsfall evaluiert werden.
- ❑ Eine Anwendungsfall-zentrierte Methode ist eine inkrementelle und iterative Entwicklungsmethode.

5.2 Definitionen

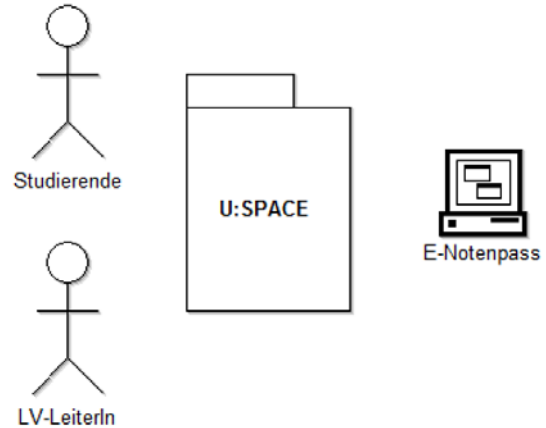
- ❑ Das Konzept der Anwendungsfälle stammt von Ivar Jacobson's "Object-Oriented Software Engineering" (OOSE). Ein Anwendungsfall ist dort definiert als eine Sequenz von Transaktionen innerhalb eines Systems, deren Aufgabe es ist einen für den einzelnen Akteur (Anwender) identifizierbaren Nutzen zu erzeugen.
- ❑ Ein Akteur ist eine Rolle, die jemand oder etwas einnimmt und die in Beziehung zum Geschäftsbereich steht. Oder alternativ:
 - Alles, das Informationen mit dem System austauschen muss
 - Alles, das mit dem System interagiert
- ❑ Transaktionen innerhalb eines Systems implizieren, dass den Akteuren eine Reihe von Möglichkeiten geboten werden um mit dem System zu kommunizieren und dass durch sie ein messbarer Nutzen erzeugt wird.
- ❑ Ein messbarer Nutzen impliziert, dass die Ausführung einer Transaktion eine sichtbare, quantifizierbare und/oder qualifizierbare Auswirkung auf jene Dinge hat, die außerhalb des Systems liegen, im speziellen auf den Akteur.
- ❑ Eine Transaktion ist definiert als eine Menge von untrennbaren Aktivitäten, die entweder alle zusammen oder gar nicht ausgeführt werden.

5.3 Akteure, Anwendungsfälle und Kommunikationsbeziehungen

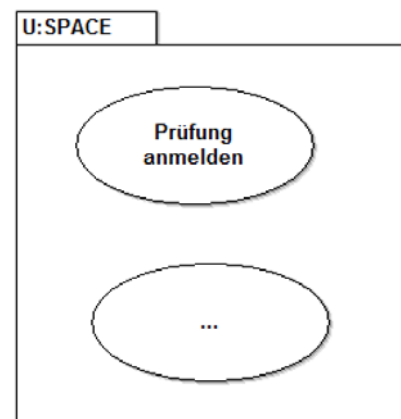
- ❑ Ein System kann durch seine zur Verfügung zu stellenden Anwendungsfälle und durch seine Grenzen beschrieben werden.
- ❑ Ein System wird durch ein Rechteck dargestellt, das die Anwendungsfälle enthält.
- ❑ Der Name des Systems wird innerhalb des Rechtecks angegeben, die Kanten des Rechtecks symbolisieren die Systemgrenzen
- ❑ Durch Akteure wird festgelegt wer mit dem System interagieren soll.
- ❑ Akteure stehen klar außerhalb des Systems. Sie benutzen das System, indem sie die Ausführung von Anwendungsfällen initiieren, oder sie werden vom System benutzt, indem sie selbst Funktionalität zur Realisierung einzelner Anwendungsfälle zur Verfügung stellen.



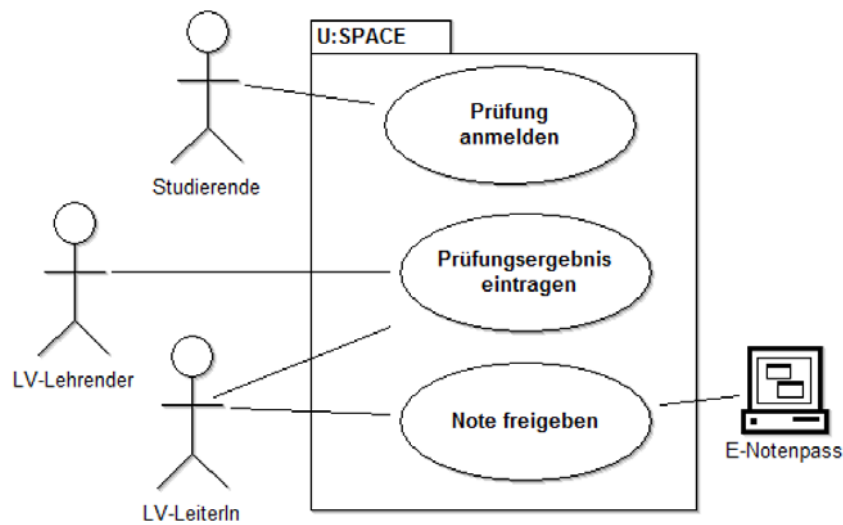
- ❑ Obwohl es für das Auffinden von Akteuren hilfreich ist, von konkreten Individuen, die mit dem System arbeiten, auszugehen, repräsentieren Akteure **keine** Objekte, sondern *Rollen*, die von Objekten gespielt werden können. Jeder Akteur hat einen Namen, der seiner Rolle entspricht.
- ❑ Akteure werden durch ein “Strichmännchen” (oder Symbole) dargestellt. Der Name des Akteurs wird unterhalb der Figur angegeben.
- ❑ Ein Anwendungsfall beschreibt ein bestimmtes Verhalten, das von einem zu entwickelnden System erwartet wird.
- ❑ Die Summe aller Anwendungsfälle zusammen macht die Systemfunktionalität des Gesamtsystems aus.
- ❑ Die Schwierigkeit in der Praxis ist es oft sämtliche Anwendungsfälle vollständig zu dokumentieren.



- ❑ Ein Anwendungsfall wird durch eine Ellipse dargestellt. Der Name des Anwendungsfalles wird innerhalb oder unterhalb der Ellipse angegeben.
- ❑ Die bloße “Ellipse” ist wertlos, wenn nicht die Semantik jedes Anwendungsfalles (in Prosa) festgehalten wird.
- ❑ Die identifizierten Anwendungsfälle, das sie umschließende System und die kommunizierenden Akteure werden in einem Anwendungsfalldiagramm zusammengefasst.
- ❑ Jeder Akteur muss zu mindestens einem Anwendungsfall eine Kommunikationsbeziehung haben. Er kann aber auch mit mehreren Anwendungsfällen kommunizieren.
- ❑ Eine Kommunikationsbeziehung wird zwischen genau einem Akteur und genau einem Anwendungsfall spezifiziert.
- ❑ Kommunikationsbeziehungen sind normalerweise ungerichtet, d.h. die Kommunikation kann in beide Richtungen erfolgen.



- Wenn auch nicht in UML festgelegt, wird öfters auch eine gerichtete Kommunikationsbeziehung verwendet, falls die Kommunikation nur in eine Richtung erfolgt.

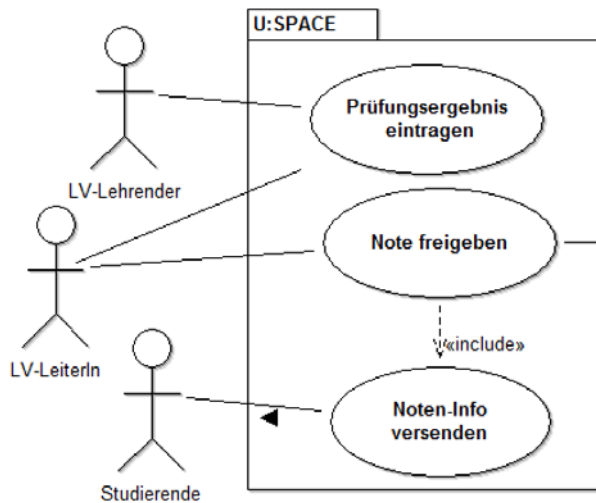


5.4 Beziehungen zwischen Anwendungsfällen

- Anwendungsfälle können nicht nur eine Kommunikationsbeziehung zu Akteuren haben, sondern können auch untereinander in Beziehung stehen.
- Es werden 3 Arten von Beziehungen zwischen Anwendungsfällen unterschieden:
 - Include-Beziehung
 - Extend-Beziehung
 - Generalisierungsbeziehung

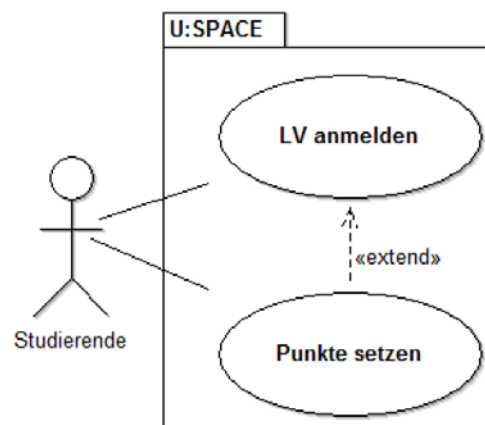
Include-Beziehung

- ❑ Die Include-Beziehung ist eine gerichtete Beziehung zwischen zwei Anwendungsfällen.
- ❑ Eine Include-Beziehung von Anwendungsfall A nach Anwendungsfall B besagt, dass das Verhalten von B in A eingefügt *wird*.
- ❑ Die Include-Beziehung von A nach B besagt, dass jedesmal wenn A ausgeführt wird auch der Anwendungsfall B ausgeführt wird.
- ❑ Die Include-Beziehung wird durch einen Abhängigkeitspfeil dargestellt, der mit dem Schlüsselwort <<include>> beschriftet ist.

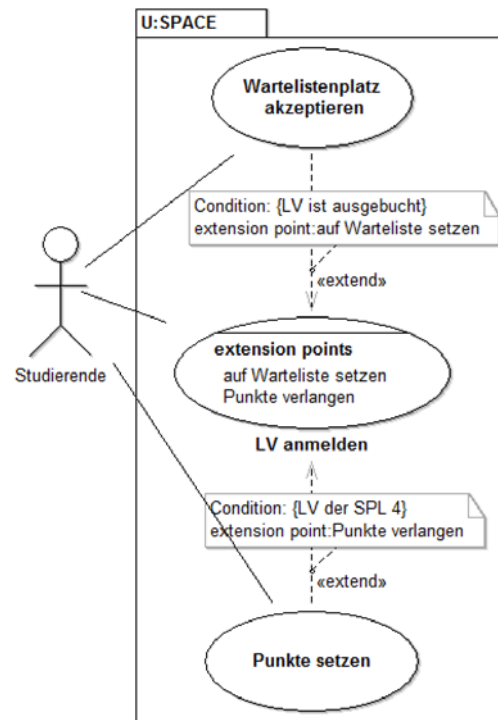


Extend-Beziehung

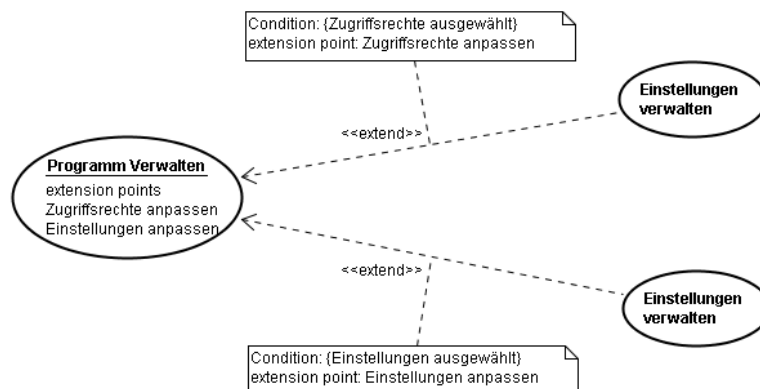
- ❑ Die Extend-Beziehung ist ebenfalls eine gerichtete Beziehung zwischen zwei Anwendungsfällen.
- ❑ Eine Extend-Beziehung von Anwendungsfall B nach Anwendungsfall A besagt, dass das Verhalten von B in A eingefügt werden *kann*.
- ❑ Die Extend-Beziehung wird ebenfalls durch einen Abhängigkeitspfeil dargestellt, der mit dem Schlüsselwort <<extend>> beschriftet ist.
- ❑ Für jede Extend-Beziehung können eine oder mehrere Erweiterungsstellen im zu erweiternden Anwendungsfall definiert werden, die angeben, wo der zu erweiternde Anwendungsfall einzufügen ist.



- ❑ Die Erweiterungsschnittstellen, die sogenannten Extension Points, sind innerhalb des Anwendungsfalles A eindeutig identifiziert und können in diesem in einem eigenen Abschnitt des Anwendungsfallsymbols aufgelistet werden.
- ❑ Die Erweiterungsschnittstellen müssen nicht unbedingt mit den Namen der zu erweiternden Anwendungsfälle übereinstimmen.
- ❑ Werden mehrere Erweiterungsschnittstellen spezifiziert, muss eine Zuordnung zwischen dem zu erweiternden Anwendungsfall und einer Erweiterungsstelle, wo er eingefügt werden soll, vorgenommen werden. Dies geschieht indem der Name der Erweiterungsstelle in Klammern bei der <<extend>>-Beschreibung spezifiziert wird.



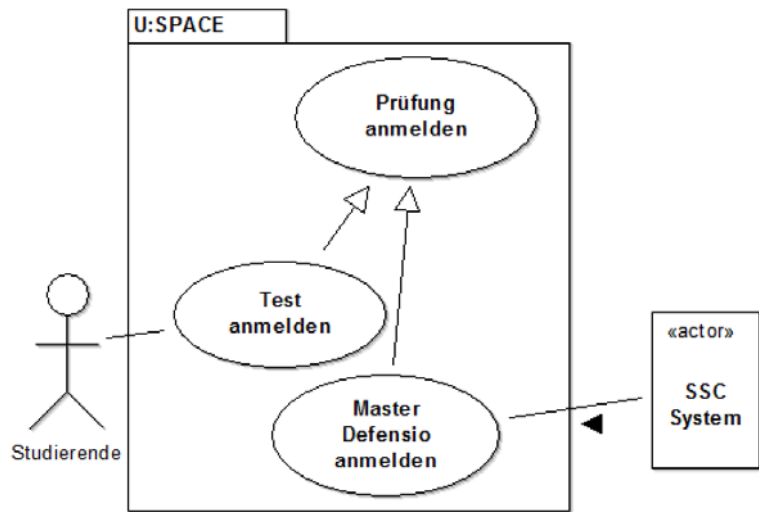
- ❑ Da die Extend-Beziehung eine optionale Beziehung zum Ausdruck bringt, muss für jeden zu erweiternden Anwendungsfall auch eine Bedingung spezifiziert werden, die erfüllt sein muss, damit der Anwendungsfall eingefügt werden kann.
- ❑ Die Bedingung wird in Klammern ebenfalls bei der Extend-Beziehung vermerkt. Die Spezifikation einer entsprechenden Einschränkung in einer Notiz und die Annotation dieser Notiz zur dazugehörigen Extend-Beziehung ist eine weitere Darstellungsmöglichkeit.
- ❑ UML 2.x Notation:



Generalisierungsbeziehung

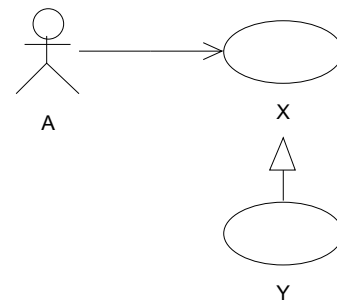
- Eine Generalisierungsbeziehung zwischen zwei Anwendungsfällen hat eine zur Generalisierungsbeziehung zwischen Klassen äquivalente Semantik.

- Der erbbende Anwendungsfall erbt das gesamte Verhalten des vererbenden Anwendungsfalls und kann dieses überschreiben oder ergänzen.

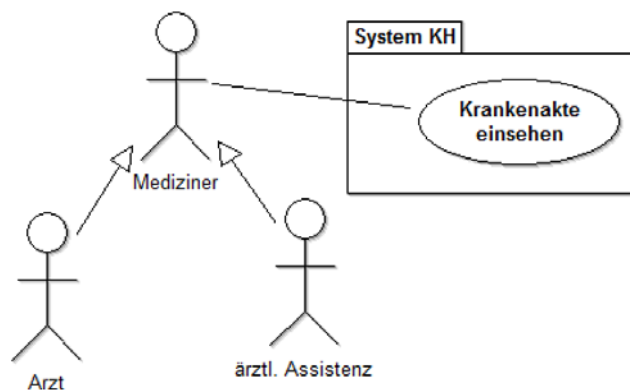


- Die Darstellung der Generalisierungsbeziehung ist ebenfalls identisch zu der zwischen Klassen.

- Ein interessanter Aspekt bei der Generalisierungsbeziehung zwischen Anwendungsfällen ist die Vererbung der Kommunikationsbeziehung. Damit erbt der spezialisierte Anwendungsfall auch die Kommunikationsbeziehungen des generellen Anwendungsfalls. Somit müssen die Kommunikationsbeziehungen zum Akteur im spezialisierten Anwendungsfall nicht noch einmal eingezeichnet werden.



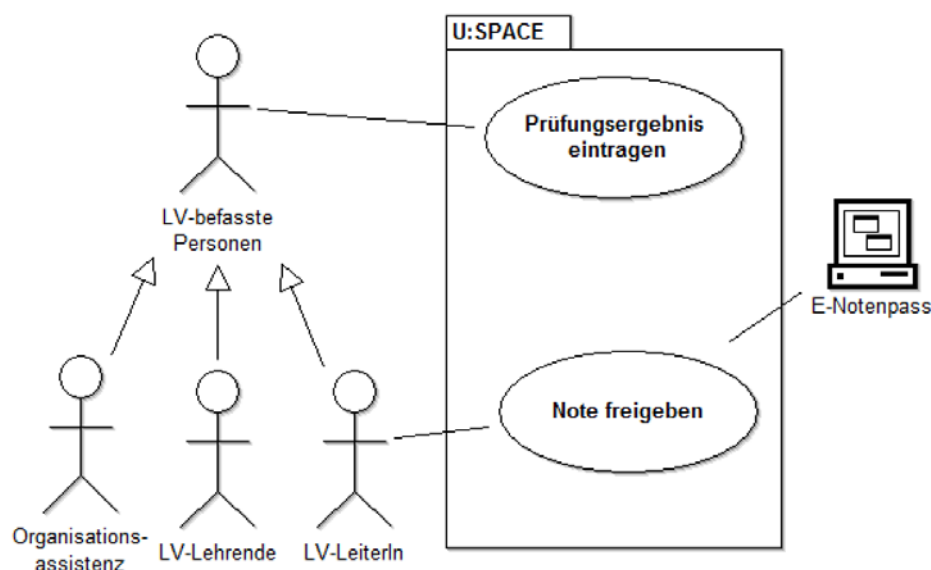
- Generalisierungsbeziehungen können auch zwischen Akteuren bestehen (vgl. Folie 7-23).



5.5 Beziehungen zwischen Akteuren

- ❑ Zwischen Akteuren gibt es als einzige Beziehungsart die Generalisierungsbeziehung.
- ❑ Eine Generalisierungsbeziehung zwischen dem generellen Akteur A und dem speziellen Akteur B besagt, dass B mit denselben Anwendungsfällen wie A kommunizieren kann.
- ❑ Ein Akteur kann von mehreren Akteuren erben und kann selbst mehrfach beerbt werden.
- ❑ Die Generalisierung von Akteuren kann auch benutzt werden, um zu unterscheiden, ob mehrere Akteure gemeinsam mit einem Anwendungsfall kommunizieren müssen oder ob mehrere Akteure mit demselben Anwendungsfall kommunizieren können.
- ❑ Sollten mehrere Akteure gemeinsam mit einem Anwendungsfall kommunizieren, so wird direkt eine Kommunikationsbeziehung zwischen jedem Akteur und dem Anwendungsfall definiert.

- ❑ Sollte nur einer von mehreren Akteuren mit einem Anwendungsfall kommunizieren müssen, so können die einzelnen Akteure zu einem (eventuell abstrakten) generellen Akteur zusammengefasst werden, der mit dem Anwendungsfall kommuniziert.



5.6 Beschreibung eines Anwendungsfalls

- ❑ Die Abbildung eines Anwendungsfalls in einem Anwendungsfalldiagramm ist wertlos, wenn nicht klar ist, welche Transaktionen im entsprechenden Anwendungsfall ausgeführt werden sollen.
- ❑ Um eine konsistente Beschreibung von mehreren Anwendungsfällen durch verschiedene Personen zu garantieren eignet es sich in einem Projekt ein Template für die Beschreibung zu verwenden.
- ❑ Das Template soll dazu dienen eine semiformale Beschreibung der Anforderungen unter konsistenter Verwendung einer Prosa zu garantieren.
- ❑ Die Beschreibungen innerhalb eines Templates sollen NICHT auf Implementierungsdetails Rücksicht nehmen. Bemerkungen zur Implementierung sollten getrennt dokumentiert werden.
- ❑ Das nachfolgende Schema stellt einen Vorschlag für ein Template dar. Es kann aber kein semiformales, allumfassendes Template zur Erfassung aller möglicher Strukturen, die in irgendeinem Anwendungsfall vorkommen können, geben.

Title	Titel des Anwendungsfalls umfasst üblicherweise wenige Worte, maximal jedoch einen Satz
Summary	Enthält eine Zusammenfassung der unten folgenden Details. Die Zusammenfassung kann oft in drei Punkte unterteilt werden, die in Form einer Liste angeführt werden. Diese Punkte sind der Trigger, die Aktion und das Resultat. 1. Dieser Anwendungsfall beginnt mit ... 2. Dieser Anwendungsfall umfasst ... 3. Dieser Anwendungsfall endet mit ...
Description	Eine kurze, präziser Paragraph der die Natur des Anwendungsfalles in Prosa festhält. Der Inhalt dieser Beschreibung enthält: 1. zeitlichen Bezugspunkt 2. die Akteure 3. die Aktion 4. die Einschränkungen
Use Case Goal	Aussage über das Geschäftsziel des Akteurs, das befriedigt werden muss. Falls der Akteur ein Anwender ist, so sollte dieses Ziel in seiner Sprache ausgedrückt werden. Stilistisch könnte begonnen werden mit "Ich möchte ... "
Actors	Eine Liste mit einem oder mehreren Akteuren, die am Anwendungsfall teilnehmen.
Assumptions / Constraints	Eine oder mehrere Aussagen, die das Systemverhalten eines Anwendungsfalles betreffen, z.B. Vorbedingungen die eingehalten werden müssen

Basic course of action	Eine geordnet Liste aller Schritte, die innerhalb des Anwendungsfalles ausgeführt werden müssen. Diese optimale Schrittfolge zeigt jene Schritte an, die zur Erfüllung des Anwendungsfalles führen. Aktionen werden generell aus der Sicht des Akteurs und nicht aus der Sicht des Analysten beschrieben. Details, welche die Implementierung betreffen, sollten vermieden werden. Jeder Schritt sollte zum Zwecke einer späteren Referenzierung nummeriert werden.
Alternative Aktionen	Jede alternative Menge an Schritten stellen Ausnahmen zum obigen Normalablauf dar. Der erste Schritt eines Alternativen Ablaufes sollte einen entsprechenden Schritt im Normalablauf referenzieren. Alternative Schrittfolgen, können auch darstellen, bei welchem Schritt in der Normalfolge wieder aufgesetzt werden muss, nachdem ein korrigierenden Schritt durchgeführt wurde. Sie können aber auch ohne Wiedereintritt in den Normalablauf enden. Alternative Abläufe können auch zur Reduktion der Gesamtanzahl an Anwendungsfällen genutzt werden.
Questions	Alle ungelösten Punkte, die im Rahmen der Beschreibung des Anwendungsfalles aufgetreten sind und einer weiteren Klärung bedürfen.
Business Owner	Wirtschaftlich Verantwortlicher für den Anwendungsfall. Er ist üblicherweise nicht der Akteur.
Autor	Analyst der den Anwendungsfall dokumentiert hat.
Version und Date	Datum der Beschreibung des Anwendungsfalles und eventuell Versionsnummer der Beschreibung.

5.7 Identifizierung von Anwendungsfällen

- ❑ Das Identifizieren von Anwendungsfällen sollte niemals ohne die Anwender des Systems erfolgen
- ❑ Es erweist sich als geeignet mit der Identifizierung der Akteure zu beginnen:
 - Welche Anwendergruppen gibt es im System?
 - Wer innerhalb der Anwendergruppen wird das System verwenden?
 - Gibt es irgendwelche bestehende Systeme die berücksichtigt werden müssen?
- ❑ Danach erfolgt die Identifizierung der Anwendungsfälle. Es erweist sich als nützlich, die Trigger für Anwendungsfälle zu suchen. Trigger sind jene Ereignisse, die eintreten müssen, damit das System veranlasst wird ein Ergebnis zu produzieren. Der Aufruf des Systems erfolgt oft durch einen Akteur, der damit Akteur des Anwendungsfalles wird. Es werden folgende Trigger unterschieden: interne, externe und zeitliche Trigger.

5.8 Die wichtigsten Regeln zur Anwendungsfallmodellierung

- ❑ Die wichtigsten funktionalen Anforderungen müssen in den Anwendungsfällen festgehalten werden.
- ❑ Ein Anwendungsfall beschreibt eine Transaktion für die der Auftraggeber bezahlt.
- ❑ Ein Anwendungsfall beschreibt einen typischen Fall ein System zu verwenden und nicht mehr.
- ❑ Ein Anwendungsfall ist wie ein Theaterstück. Die Anwendungsfallbeschreibung enthält die Choreographie.
- ❑ Ein Anwendungsfall hat eine Einleitung einen Hauptteil und einen Schluss.
- ❑ Ein Anwendungsfall soll so einfach wie möglich beschrieben sein.
- ❑ Die Beschreibung eines Anwendungsfalles sollte 2 Seiten nicht überschreiten.
- ❑ Ein Anwendungsfall muss präzise definiert sein.
- ❑ Ein Anwendungsfall ist dann fertig beschrieben, wenn der Kunde, die Anwender und die Softwareentwickler ihn akzeptieren.
- ❑ Ein Anwendungsfall stellt die Grundlage für einen Systemtest dar.



Literatur und Quellen:

- ❑ OMG Unified Modeling Language™ (OMG UML), Version 2.5
Normative Reference: <http://www.omg.org/spec/UML/2.5>
OMG Document Number formal/2015-03-01
- ❑ <http://www.uml.org>
- ❑ Kapitel 4.1 in:
Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, Werner Retschitzegger:
UML @ Work - Objektorientierte Modellierung mit UML2.
dpunkt Verlag 2005 / 3. aktualis. u. überarb. Aufl. 2005.
ISBN-13: 9783898642613 ISBN-10: 3898642615
- ❑ Tool zur Vertiefung: BEE-UP Modelling Tool:
<http://austria.omilab.org/psm/content/bee-up/info>