

## UML – Anwendungsfalldiagramm

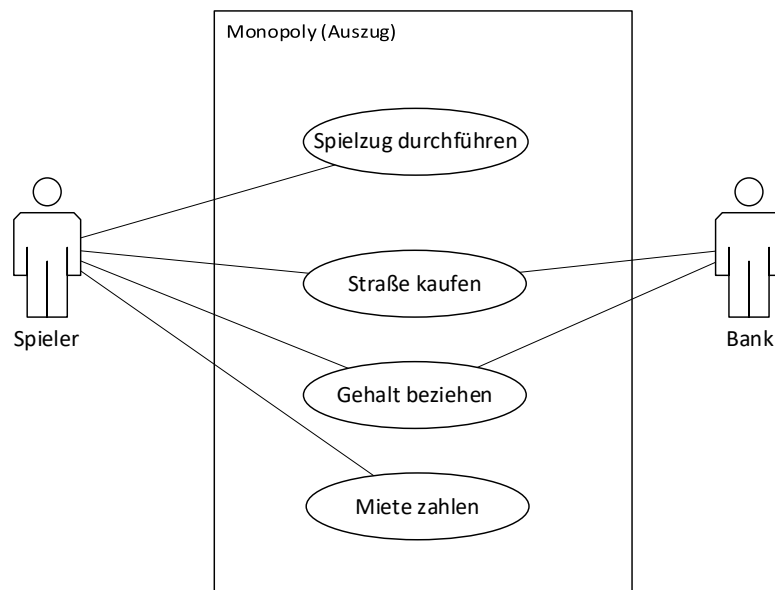
### Einsatzzweck

Das Anwendungsfalldiagramm, das auch Use-Case-Diagramm genannt wird, beschreibt das Zusammenspiel von Akteuren und Geschäftsvorfällen. Es wird kein Verhalten oder Ablauf dargestellt. Nur das „Was“ steht im Mittelpunkt der Betrachtung, nicht das „Wie“. Es kann insofern von einem „Blackboxcharacter“ gesprochen werden.

Das Diagramm dient primär der Kommunikation zwischen zukünftigen Anwendern und Programmierern. Missverständnisse, die in dieser frühen Phase der Softwareentwicklung entstehen, werden ein kostspieliges Nachspiel haben. Genau dieses Risiko soll mit der Visualisierung reduziert werden.<sup>1</sup> Anwendungsfalldiagramme können aus den Ergebnissen der Anforderungsanalyse abgeleitet werden und dienen sodann als Grundlage zur Gestaltung von Mock-Ups.<sup>2</sup>

### Grundaufbau

*Beispiel Monopoly: Zu sehen sind hier zwei Akteure, die unterschiedliche Anwendungsfälle (Spielzug durchführen, Straße kaufen, Gehalt beziehen, Miete zahlen) anstoßen oder bei deren Erfüllung auf irgendeine Art und Weise in Kenntnissetzung mitwirken, bzw. von der Auswirkung der Durchführung betroffen sind.*



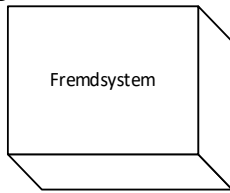
Alle Anwendungsfälle befinden sich in einem mit einem Namen versehenen Rahmen (Systemgrenze genannt). Dieser Rahmen verbundenen mit einem passenden Namen entspricht den Ausschnitt der Realität, den das System abbilden soll.

Jeder **Anwendungsfall** wird in Form einer Kombination aus Substantiv + Verb notiert. Ein Anwendungsfall kann als eine Sammlung von Aktionen verstanden werden, die später bei der Programmierung zu einer Funktion gebündelt werden.

<sup>1</sup> Vgl. für folgende Beispiele Software-Entwurf mit UML 2, Seemann&Wolff von Gudenberg, S.15ff.

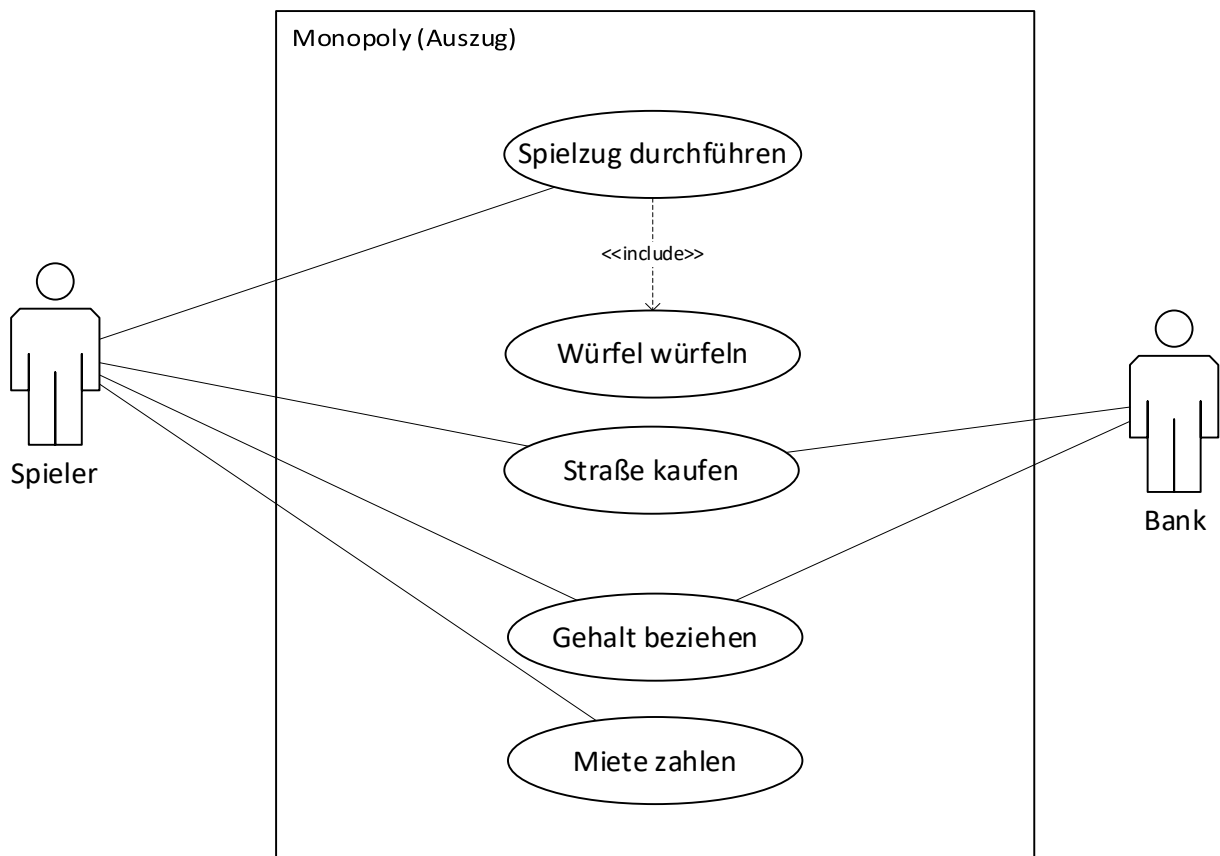
<sup>2</sup> Vgl. Usability und UX kompakt, Richter, Flückiger, S. 85ff.

Die menschlichen Akteure sind im Regelfall einfache Strichmännchen. Ein Akteur könnte natürlich auch nicht menschlicher Natur sein, z.B. ein Drittsystem, das zur Aufgabenerfüllung genutzt wird. In diesem Fall wird ein 3-D Kasten mit den Namen des Drittsystem verwendet.



### Benutzung eines anderen Anwendungsfalls (Include-Beziehung)

Die nachfolgende Abbildung zeigt, dass jede Spielzugdurchführung immer „Würfel würfeln“ als weitere Funktionalität erfordert. Da der Anwendungsfall „Würfel würfeln“ auch von einem anderen Anwendungsfall genutzt werden könnte, wird dies als eigenständiger Anwendungsfall aufgeführt und vom Anwendungsfall „Spielzug durchführen“ immer inkludiert.

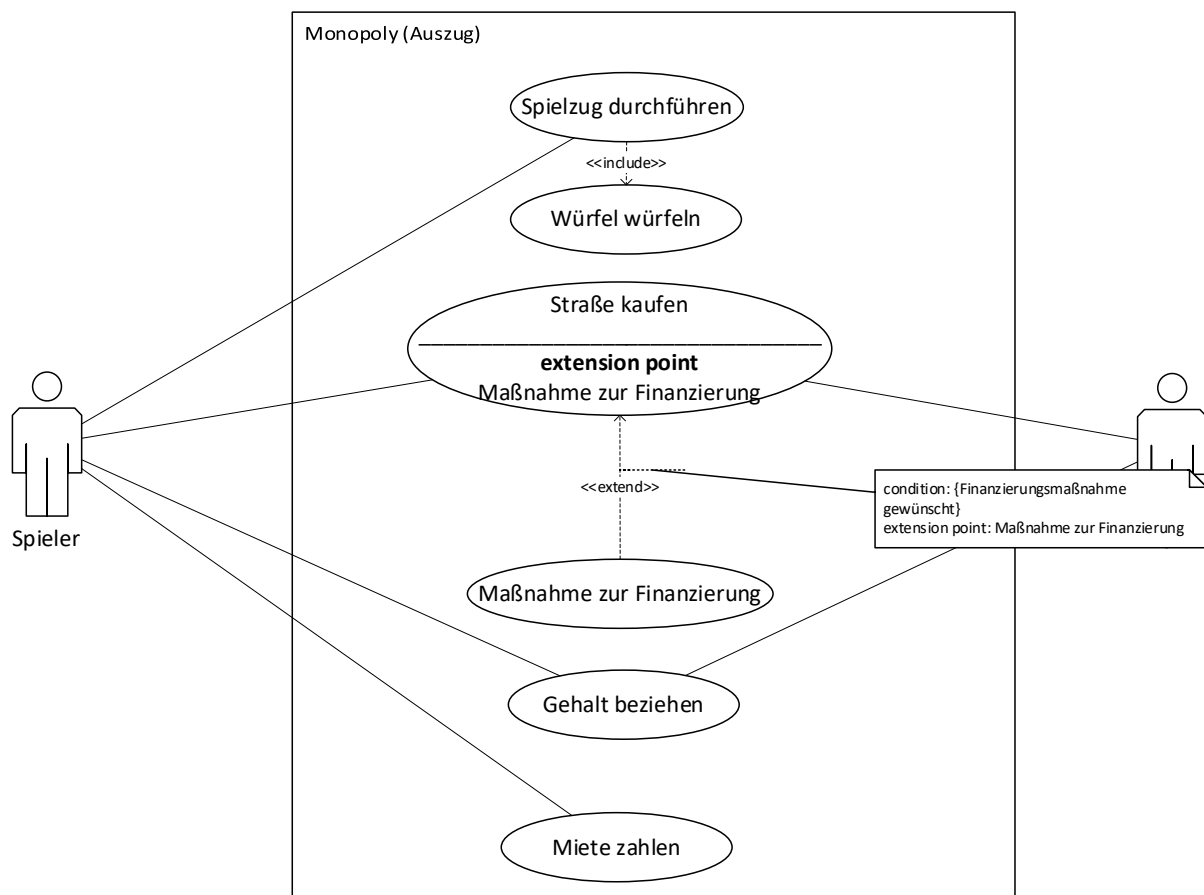


Eine Include-Beziehung modelliert die **unbedingte Einbindung der Funktionalität eines Anwendungsfalls** in einem anderen Anwendungsfall.

Genutzt wird hier ein gestrichelter Pfeil, der vom Hauptfall auf den Erweiterungsfall zeigt.

## Optionale Anwendungsfälle (Extend-Beziehung)

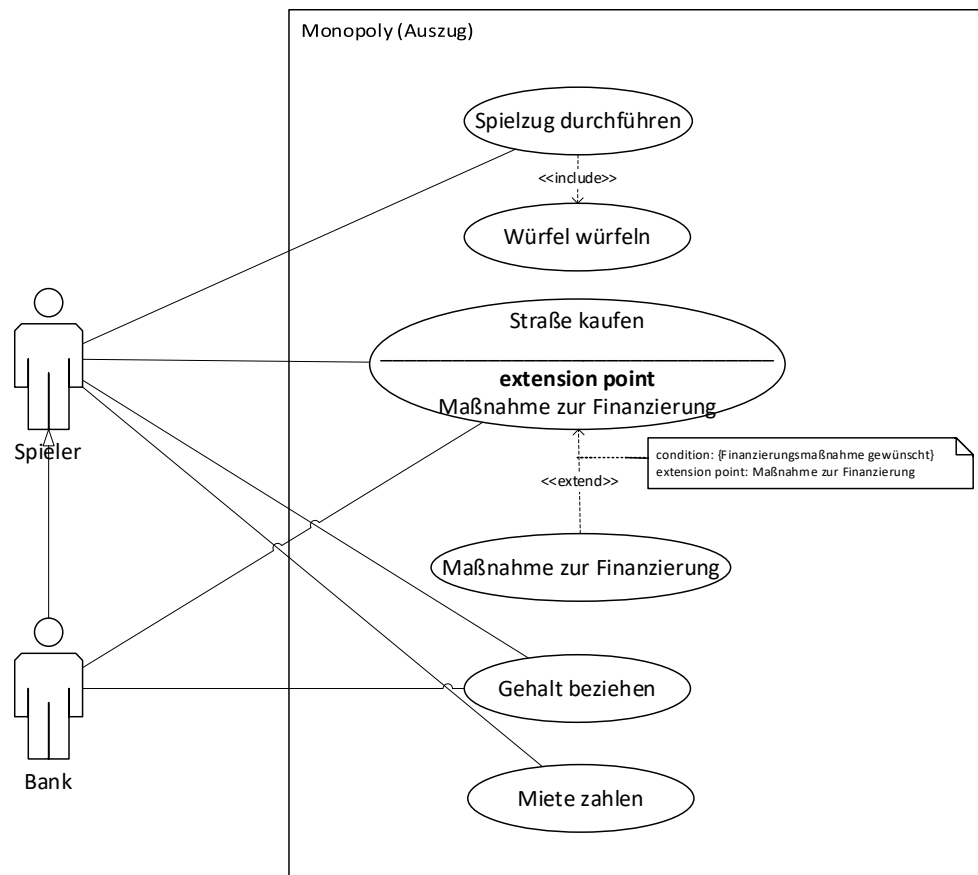
Die nachfolgende Erweiterung zeigt eine bedingte Einbindung der Funktionalität eines Anwendungsfalls in einem weiteren Anwendungsfall. Es wird hierbei von einer „Kann“-Einbindung gesprochen, da diese Einbindung in Abhängigkeit des Eintretens der jeweiligen Bedingung ist. In diesem Beispiel tritt der Anwendungsfall (Hypothek aufnehmen) nur ein, wenn der Wunsch des Spielers besteht, eine Maßnahme zur Finanzierung des Straßenkaufs durchzuführen.



Eine Extend-Beziehung modelliert die **bedingte Einbindung der Funktionalität eines Anwendungsfalls** in einem anderen Anwendungsfall.

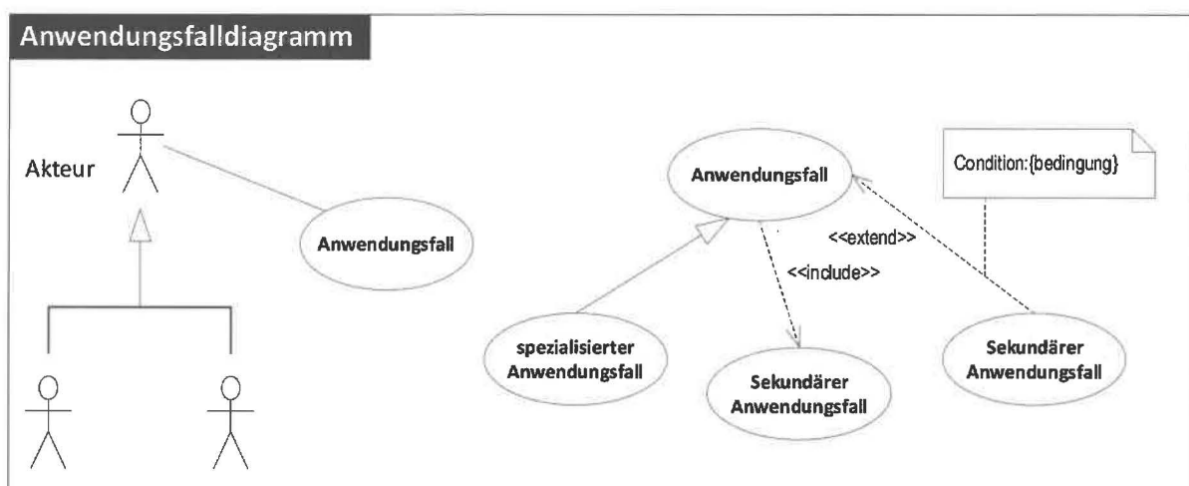
Gleichwohl eine Extend-Beziehung mit einer Bedingung eine Art „If-Then-Else“-Modellierung darstellt, ist es wichtig bei diesem Diagramm, keine weiteren Programmabläufe darzustellen, sondern sich nur darauf zu konzentrieren, einen Überblick über die benötigte Funktionalität eines Systems zu verschaffen.

## Generalisierung von Akteuren



Bei Monopoly wird ein Spieler bestimmt, der die Bank verwaltet. Dieser Akteur ist ein Spieler und kann alle Anwendungsfälle, die für Spieler spezifiziert wurden, ausführen. Außerdem kann er auch alle Anwendungsfälle der Bank ausführen. Es handelt sich hierbei insofern um eine klassische Vererbung.

## Belegsatz der IHK



Gleichwohl der IHK-Belegsatz hier etwas vom [UML-Standard](#) bei den Extend-Beziehungen abweicht, bitten wir Sie, den UML-Standard wie zuvor beschrieben einzuhalten.



## Übungen zum Anwendungsfalldiagramm

### 1. Einfache Kundenverwaltung (CRM)

Es soll ein Programm entwickelt werden, mit dem es möglich wird, verschiedene Kundendaten (CRM) zu verwalten. Im Genaueren soll folgendes möglich sein:

- Der Vertriebsmitarbeiter führt mit einem Kunden ein Gespräch, das mit dem Kunden zusammen dokumentiert wird. Während das Gespräch nicht Bestandteil des Systems ist, wird das Gespräch im System dokumentiert.
- Der Vertriebsmitarbeiter kann sich die komplette Kundenliste oder die Daten eines einzelnen Kunden anzeigen lassen. In beiden Fällen werden die Kundendaten zuvor vom Programm geladen.
- Ein Vertriebsmitarbeiter kann weiterhin Kundendaten löschen. Handelt es sich dabei um einen Großkunden, ist eine Freigabe von einem Teamleiter nötig.
- Insofern ist klar, dass ein Teamleiter Löschungen freigeben kann. Der Teamleiter kann im Übrigen auch alles, was ein „normaler“ Kundenbetreuer auch kann.
- Ein Vertriebsmitarbeiter kann weiterhin Auswertungen des Kundenstammes vornehmen.

### 2. E-Mailprogramm

Der Benutzer kann mit Hilfe seines clientseitigen Emailprogramms entweder eine E-Mail verfassen, eine E-Mail lesen oder eine E-Mail löschen. Zur Durchführung jeder dieser Vorgänge ist es erforderlich, dass der Nutzer sich vorab hierfür mit seinem Kennwort angemeldet hat. Im Rahmen der Löschung einer E-Mail, kann sich der Nutzer optional auch dafür entscheiden, alle Emails zu löschen. Ein Administrator, der selbst auch ein E-Mailaccount nutzen kann, hat die Möglichkeit, ein Kennwort auf Verlangen eines Nutzers zurückzusetzen.

### 3. Materialwirtschaft (optionale Aufgabe)

In der Materialwirtschaft einer Firma kommt es immer wieder zu ungeplanten Entnahmen und Zugängen.

- Die Mitarbeiter der Warenannahme buchen den ungeplanten Zugang.
- Dabei wird automatisch nach einem Lagerplatz gesucht.
- Wird kein Lagerplatz gefunden, so sucht der Mitarbeiter der Warenannahme manuell nach einem freien Lagerplatz.
- Der Materialverwalter kümmert sich ausschließlich um die Buchung ungeplanter Entnahmen.
- Zudem kümmert er sich um das Umlagern von Waren. Dies beinhaltet die manuelle Suche nach einem Lagerplatz.

Erstellen Sie anhand der vorliegenden Informationen ein UML-Anwendungsfalldiagramm für die Software „Materialwirtschaft (ungeplant)“.

### 4. Kurierfahrt

Das zu modellierende Programm soll die Verwaltung von klassischen Kurierfahrten darstellen. Ein Kurierdienst erhält von einem Auftraggeber (Sender) folgende Beschreibung von den bei einer Kurierfahrt auszuführenden Tätigkeiten:

- Erfassung der Daten der Sendung, die der Kurierdienst vom Sender erhält.
- Abholung der Sendung beim Sender
- Ablieferung der Sendung beim Empfänger.
- Persönliche Übergabe der Sendung an den Empfänger, sofern vom Auftraggeber gefordert.
- Abschließende Rechnungserstellung der Kurierfahrt.

**5. E-Mailadressen erstellen** (optionale Zusatzaufgabe)

Sie sollen ein Anwendungsfalldiagramm für folgende Beschreibung erstellen: Ein Mitarbeiter der Personalabteilung stellt eine Liste neuer Mitarbeiter bereit. Auf dieser Liste sind die Vor- und Nachnamen der neuen Mitarbeiter vermerkt. Ein Mitarbeiter der IT-Abteilung lädt diese Liste (CSV-Datei). Anschließend lässt er aus den vorhandenen Vor- und Nachnamen geeignete Emailadressen([vorname.nachname@unternehmen.de](mailto:vorname.nachname@unternehmen.de)) erstellen. In jedem Fall wird bei der Erstellung überprüft, ob Umlaute(ä,ü,ö,ß) vorhanden sind. Sofern dies der Fall ist, werden diese entsprechend angepasst. Der IT-Mitarbeiter kann sich zudem die erstellten Emailadressen anzeigen lassen.

*Freiwillige Erweiterungen:* Erstellen Sie in Ihrer Programmiersprache eine entsprechende Umsetzung obiger Aufgabe. Erstellen Sie hierzu zuerst CSV-Datei.

Vorname,Nachname Sandy, Andersen Brigitte, Öttinger
---