

# 6. UML – Verhaltensmodellierung

## 5 UML – Strukturdiagramme

- 5.1 Einführung UML
- 5.2 Objektdiagramm, Klassendiagramm
- 5.3 Komponentendiagramm

## 6 UML – Verhaltensmodellierung

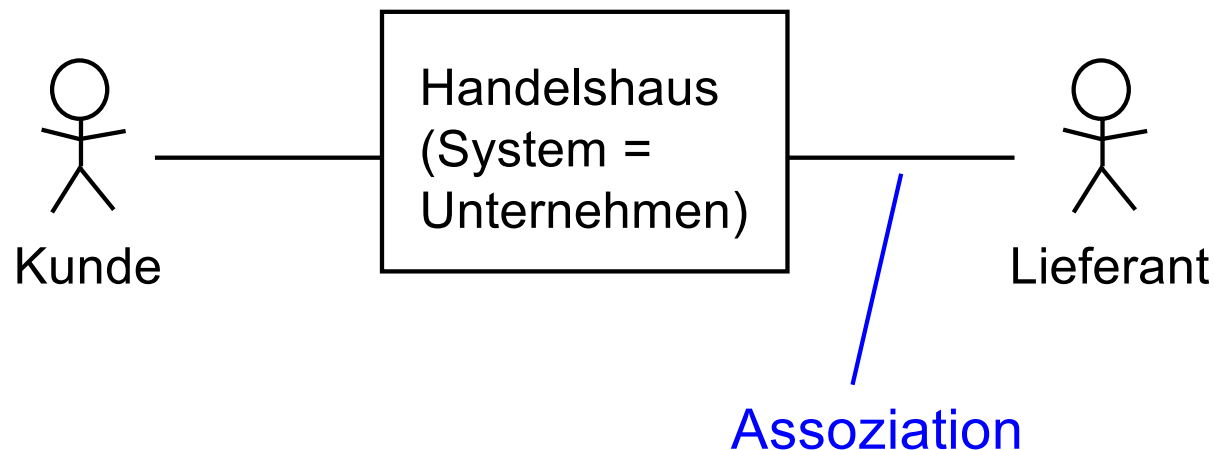
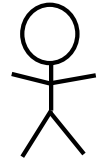
- 6.1 Use Case-Diagramm
- 6.2 Aktivitätsdiagramm
- 6.3 Kommunikationsdiagramm
- 6.4 Sequenzdiagramm
- 6.5 Zustandsdiagramm

# 6.1 Use Case-Diagramm

- Use Case-Diagramm (Anwendungsfall-Diagramm)
  - „Was leistet mein System für seine Anwender?“
- Beschreibung dessen, was ein System leisten soll
  - Semiformale Notation
  - Hohes Abstraktionsniveau
  - Weglassung funktionaler Details
  - Einfach zu verstehen, Beispielhaftigkeit
  - Ergänzung oder Vorstufe weiterer Modelle
- Gut geeignet als erster Schritt beim Einsatz der OOA (Anforderungsanalyse)

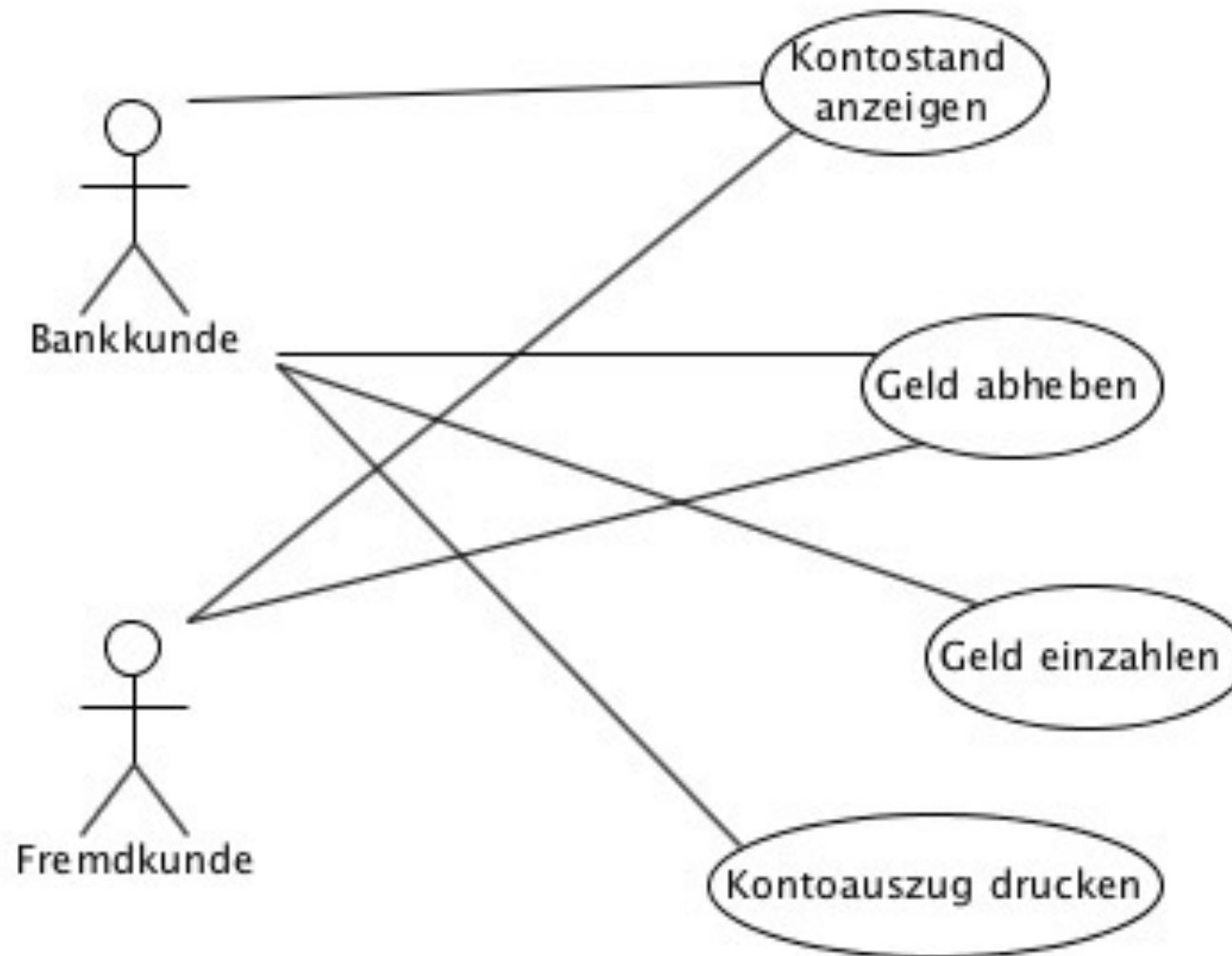
# Use Case-Diagramm

- Akteur
  - Benutzer des Systems
  - Person, Organisationseinheit oder externes System



# Use Case-Diagramm

## Beispiel Geldautomat

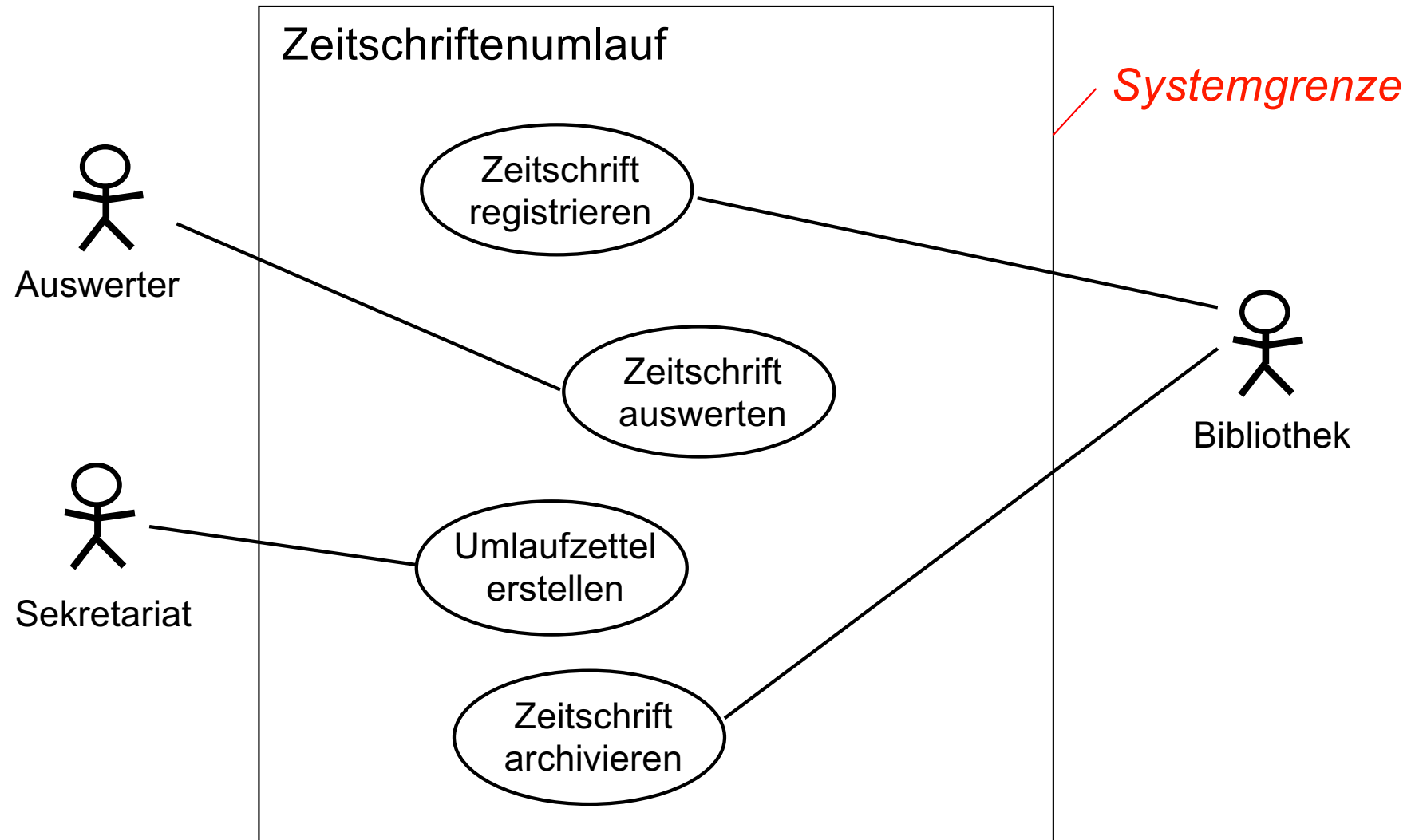


# Use Case-Diagramm

- Name des Anwendungsfalls
  - Aktives Verb, das die Tätigkeit bzw. den Ablauf beschreibt
  - Substantiv, das den Gegenstand beschreibt, der bearbeitet wird
  - Bsp: Kfz reservieren, Ware anliefern
- Eigenschaften von Use Cases
  - An jedem Use Case ist mindestens ein Akteur beteiligt
  - Maximaler Grad an Intuitivität und Abstraktion
  - Keine Modellierung von Kontroll- oder Datenfluss!
  - Relevanten Ablauf beschreiben, nicht z.B. „Knopf drücken“ !
- Use Case-Diagramme können durch andere UML-Dynamikmodelle verfeinert werden
  - Sequenzdiagramme
  - Aktivitätsdiagramme

# Use Case-Diagramm

## Beispiel Zeitschriftenumlauf



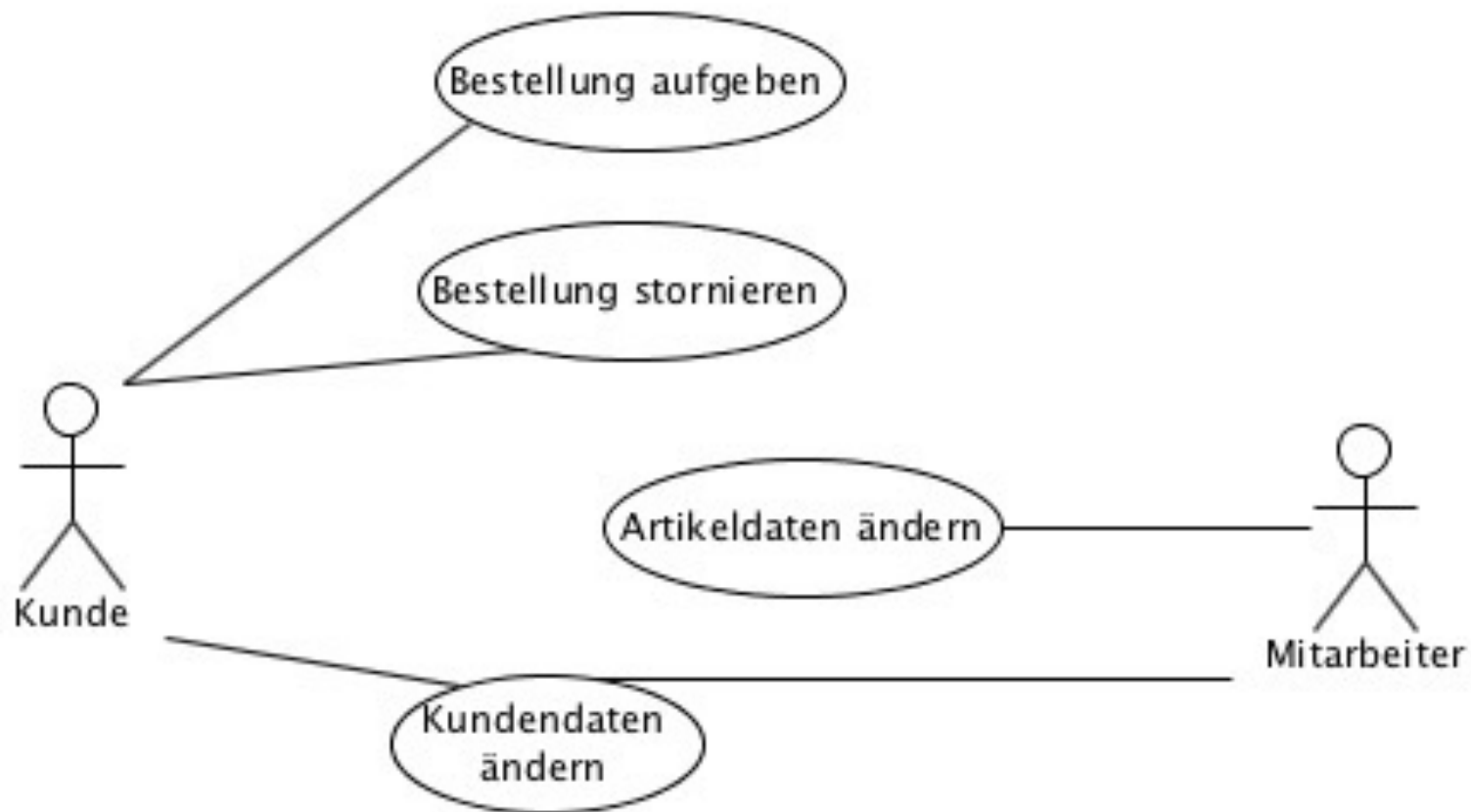
# Use Case-Diagramm

## Beispiel Homebanking



# Use Case-Diagramm

## Beispiel Webshop



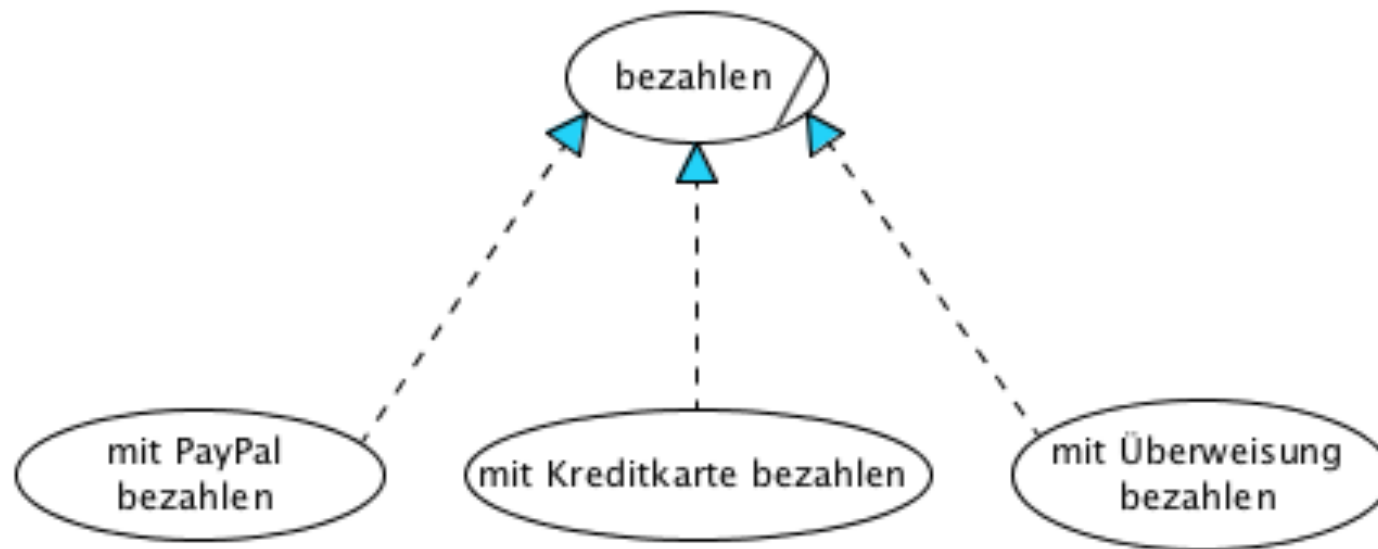


# Arten von Anwendungsfälle

- Geschäftsanwendungsfall (Business Use Case)
  - Beschreibt geschäftlichen Ablauf
  - Keine Berücksichtigung möglicher systemtechnischer Umsetzungen
- Systemanwendungsfälle
  - Beschreibt Interaktion mit einem (Hardware- oder Software-) System
  - Vorwiegend benutzte Art von Anwendungsfall
- Sekundäre Anwendungsfälle
  - Beschreibt unvollständigen Teilablauf
  - Teil von mehreren anderen Anwendungsfällen

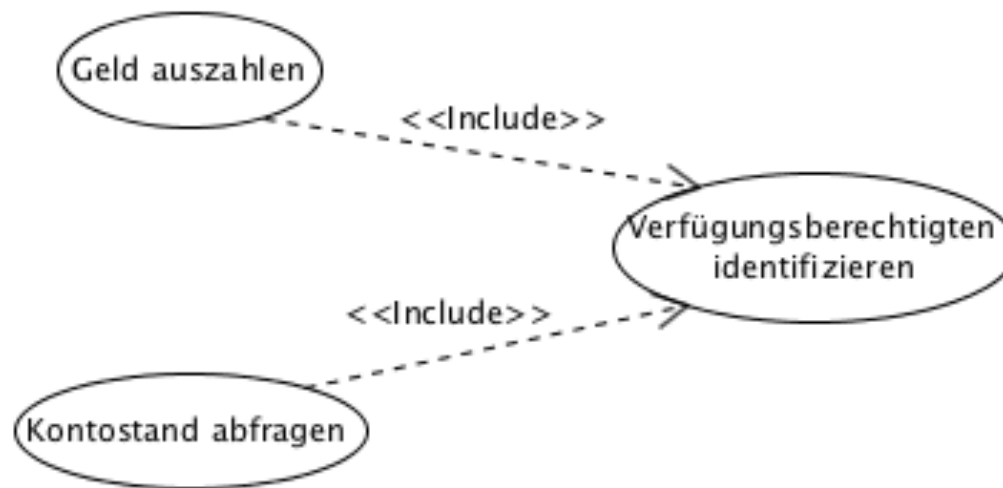
# Realisierung für Anwendungsfälle

- Eine *Realisierung* ist eine Beziehung zwischen
  - einem Element, das eine Anforderung beschreibt
  - und einem Element, das diese Anforderung umsetzt



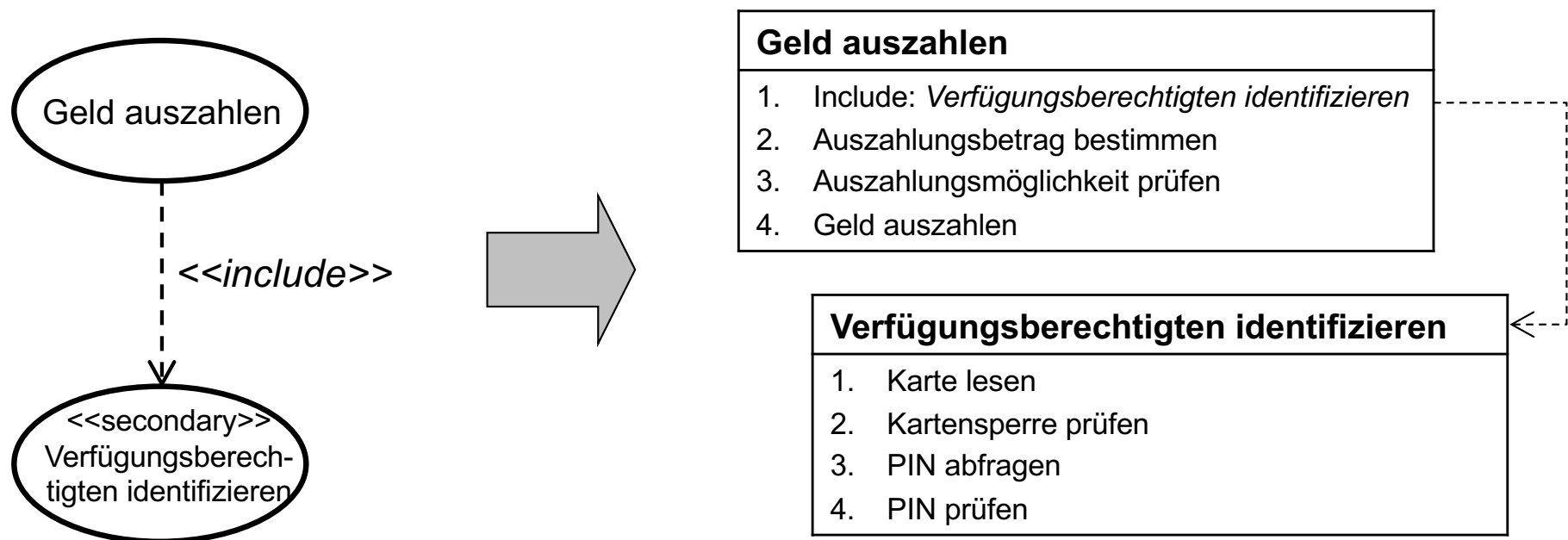
# Include für Anwendungsfälle

- Eine *Include* beschreibt
  - dass innerhalb des Ablaufs eines Use Cases ein anderer Use Case aufgerufen wird
  - Ein Use Case kann dadurch an einer zentralen Stelle beschrieben und beliebig oft genutzt werden



# Enthältbeziehung in Anwendungsfälle

- Enthältbeziehung zwischen Anwendungsfällen
  - Ein Anwendungsfall wird eingebunden in einen anderen Anwendungsfall und damit logischer Teil von diesem



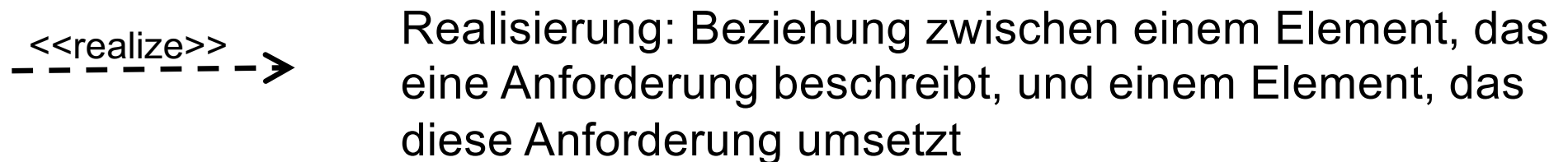
# Use Case-Diagramm

## Notation

- Verbindungen zwischen Use Cases, Akteure und Systeme

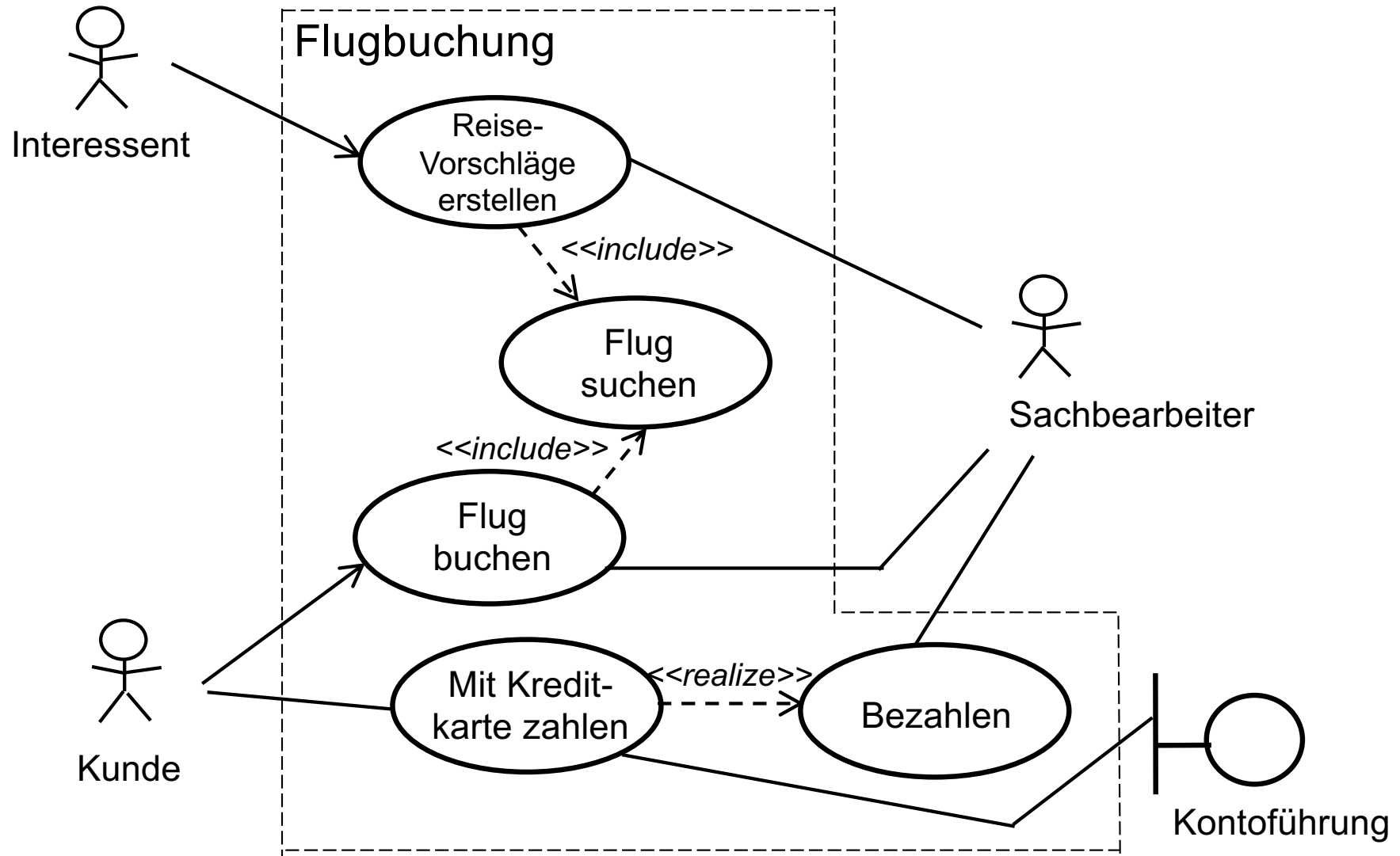


- Beziehungen zwischen Anwendungsfällen



# Use Case-Diagramm

## Beispiel Flugbuchung



# Natürlichsprachliche Anwendungsfallbeschreibung

<b>Af. 2, Vers. 2</b>		<b>Mitglied neu aufnehmen</b>
Kurzbeschreibung		Eine Person wird als Mitglied aufgenommen
Beteiligte Akteure		Ma                      Mitarbeiter der Mitgliederbetreuung
Auslöser, Vorbedingungen		Ein ausgefüllter Teilnahmeantrag geht ein
Ergebnisse, Nachbedingungen		Person wurde als Mitglied aufgenommen Dem neuen Mitglied wurden der Teilnahmevertrag zugesendet.
<b>Ablauf:</b>		
<b>1.</b>		<b>Erfassung und Prüfung der Antragsdaten</b>
1.1.	Ma	Die Daten des Teilnahmeantrags werden ins System eingegeben.
1.2.	System	Das System prüft, ob die Daten vollständig sind (Name, Vorname, Straße, Plz, Ort, Geburtsdatum, Geschlecht, Beginndatum, Aufnahmegebühr, Kapitaleinlage und Verzinsung sind Pflicht).
1.3.	System	Ausnahme: <b>Daten unvollständig</b>
1.4.	System	Für Aufnahmegebühr, Kapitaleinlage und Verzinsung wird überprüft, ob dies tariflich zulässige Werte sind. Wenn die Daten okay sind, wird die Schaltfläche "Freigeben" bedienbar.
1.5.	System	Ausnahme: <b>Daten unzulässig</b>
<b>2.</b>		<b>Freigabe der Mitgliedschaft</b>
2.1.	Ma	Schaltfläche "Freigeben" wird gedrückt.
...	...	...
<b>Ausnahmen, Varianten:</b>		
1.3.		<b>Daten unvollständig</b>
1.3.1.	Ma	Der Mitarbeiter entscheidet, ob er telefonisch oder schriftlich nachfragt. Sofern die fehlenden Daten sofort ermittelt werden, wird Schritt 1 mit entsprechend ergänzten Daten wiederholt. Anderenfalls wird die Aufnahme abgebrochen und der Antrag in die manuelle Wiedervorlage gelegt.
1.5.		<b>Daten unzulässig</b>

Quelle: [www.oose.de](http://www.oose.de)

# UML – Übersicht

## 5 UML – Strukturdiagramme

- 5.1 Einführung UML
- 5.2 Objektdiagramm, Klassendiagramm
- 5.3 Komponentendiagramm

## 6 UML – Verhaltensmodellierung

- 6.1 Use Case-Diagramm
- 6.2 Aktivitätsdiagramm
- 6.3 Kommunikationsdiagramm
- 6.4 Sequenzdiagramm
- 6.5 Zustandsdiagramm



## 6.2 Aktivitätsdiagramme (activity diagram)

- Ein Aktivitätsdiagramm beschreibt einen Ablauf
- Aktivität
  - Beschreibt Ausführung von Funktionalität
  - Zur Beschreibung von Use Cases geeignet
- Aktion
  - Kleinste ausführbare Funktionseinheit innerhalb einer Aktivität
  - Vorsicht: Begriffe waren in UML 1.x anders definiert
- Ein Aktivitätsdiagramm besteht aus
  - Start- und Endknoten
  - Reihe von (Aktions-, Kontroll- und Objekt-) Knoten
  - Objekt- und Kontrollflüssen

# Aktivitätsdiagramm

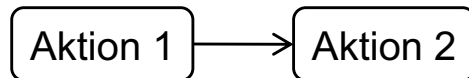
## Notation



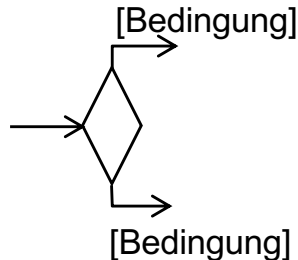
Aktion



Kontrollfluss (unbedingt oder bedingt)



Sequenz von Aktionen



Alternative (OR-SPLIT)

Guard:

- Bedingung in eckigen Klammern [ ]
- muss erfüllt sein, damit Ablauf fortgesetzt



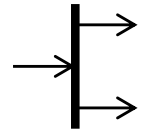
Startknoten



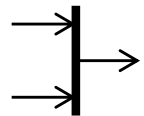
Endknoten für Aktivitäten

# Aktivitätsdiagramm

## Notation



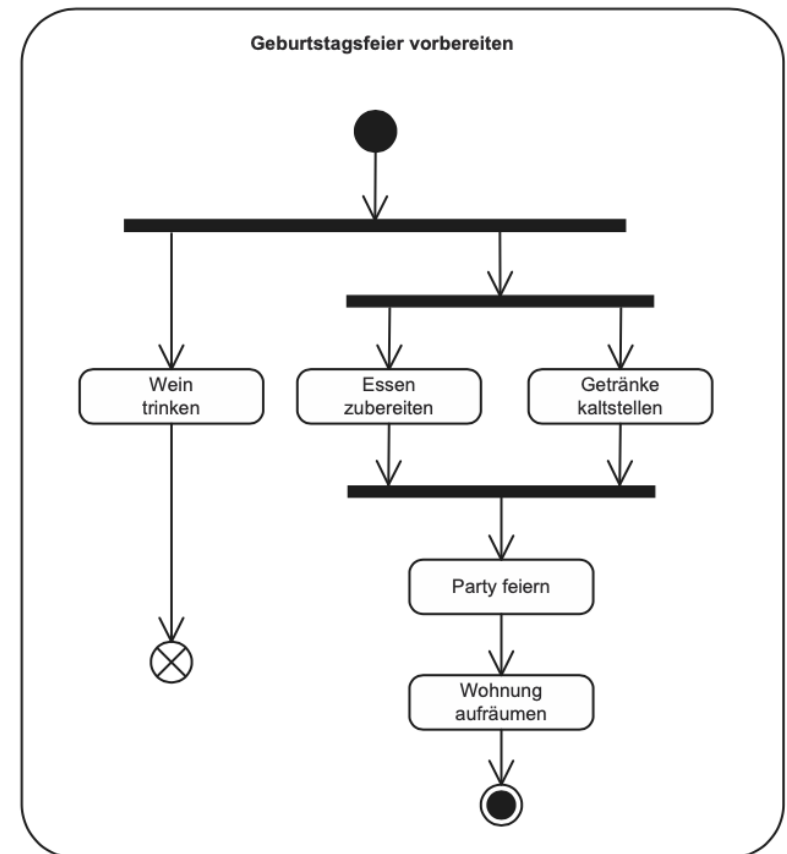
Parallelisierung



Synchronisation



Endknoten für Kontrollflüsse



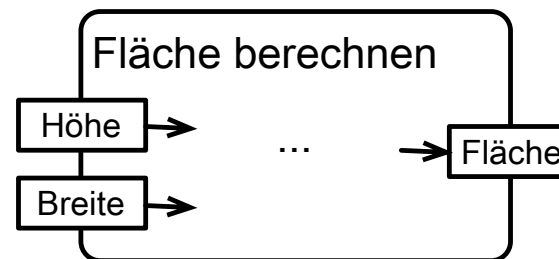
Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Aktivitätsdiagramm

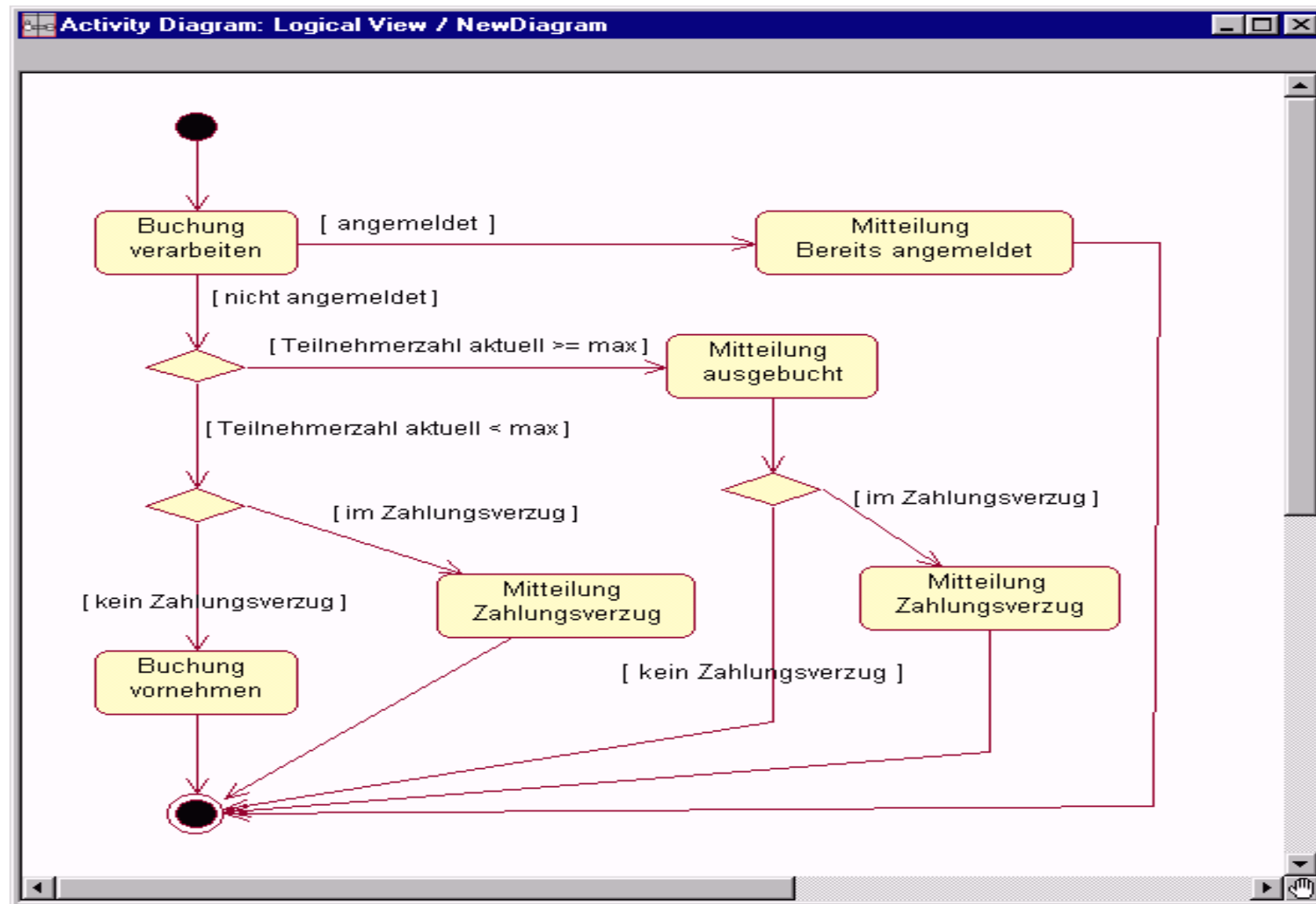
- Aktion
  - Keine weitere Zerlegung



- Aktivitäten
  - Verschachtelte Aktivitäten
  - Übergabe von Parametern

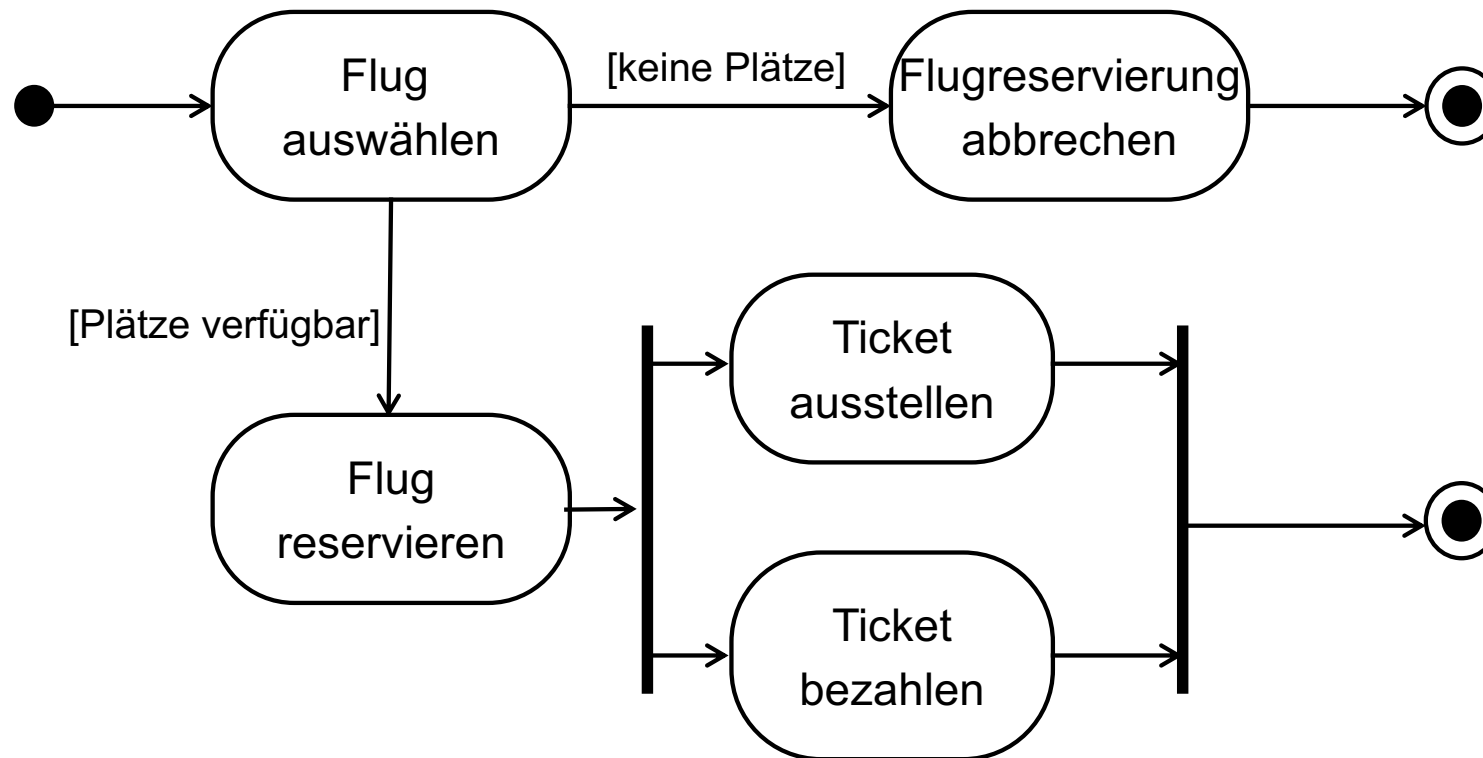


# Tool-Beispiel für ein Aktivitätsdiagramm

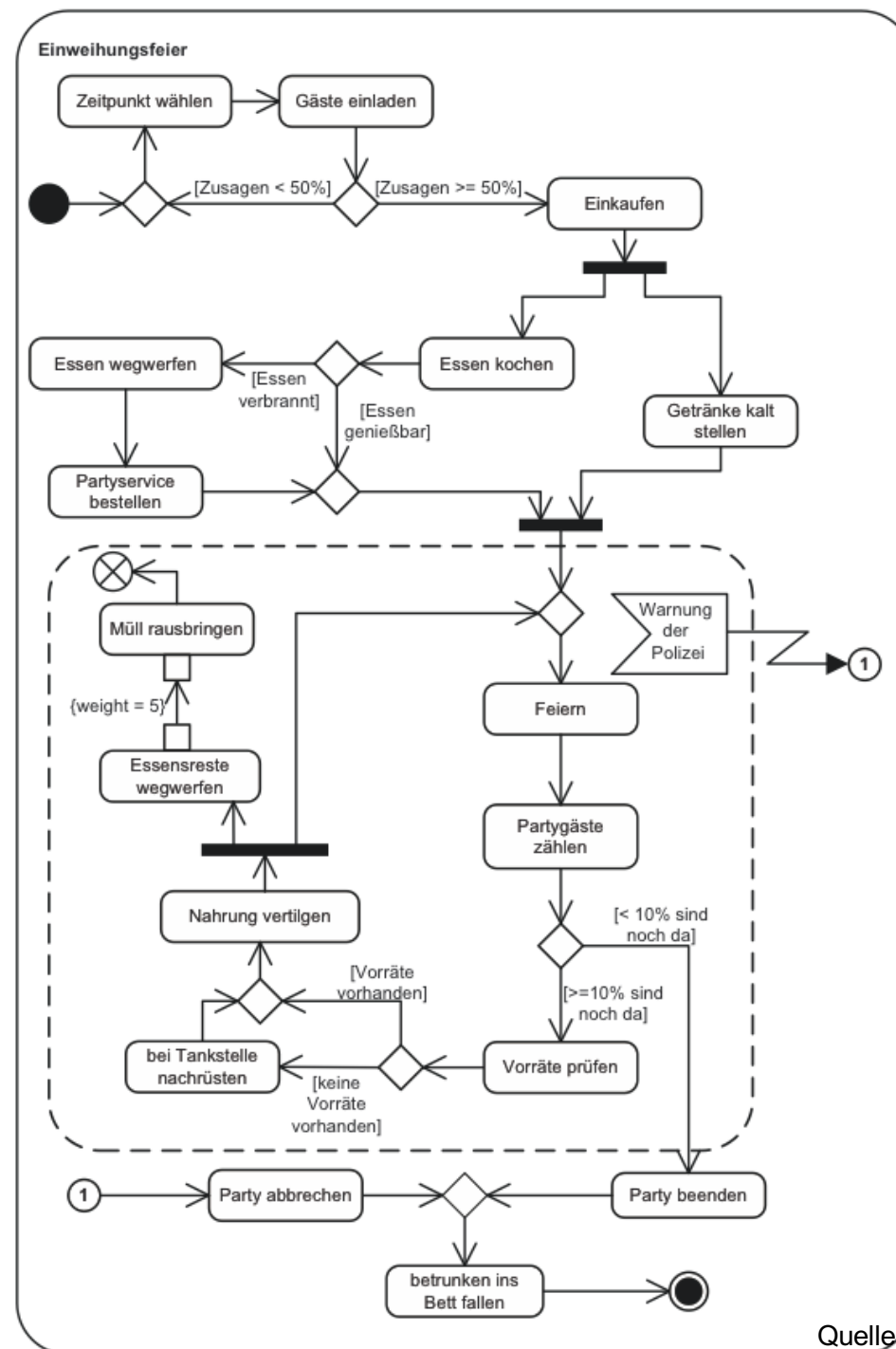


# Aktivitätsdiagramm

## Beispiel



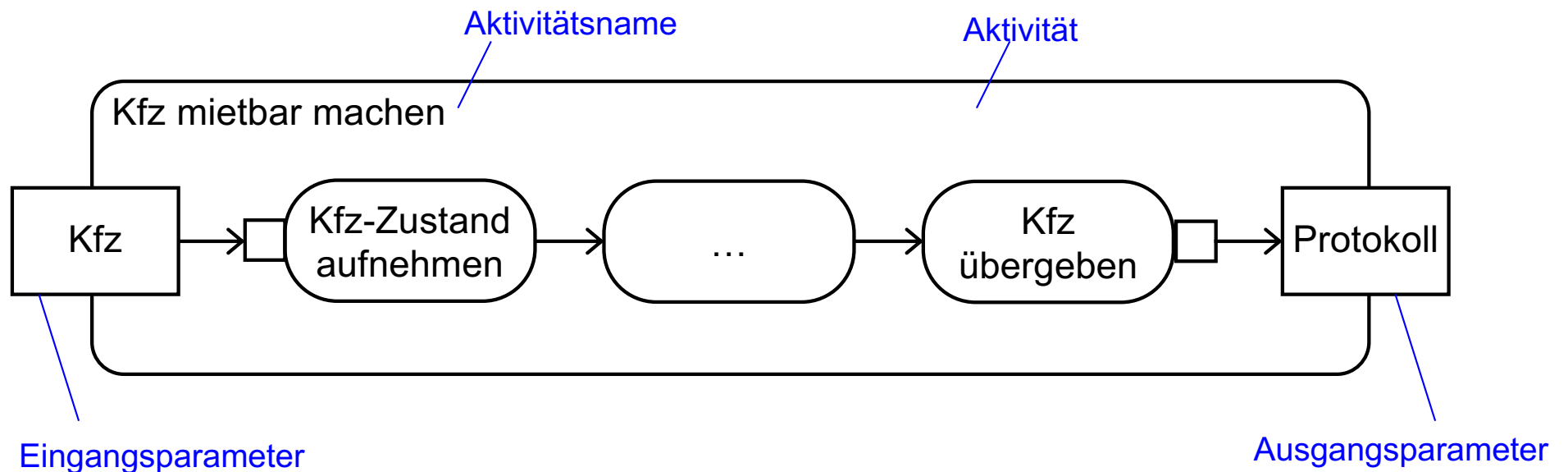
# Beispiel



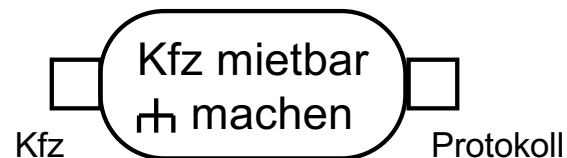
Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Notation von Aktivitätsparametern

- Ein- oder ausgehende Objekte im Aktivitätsdiagramm

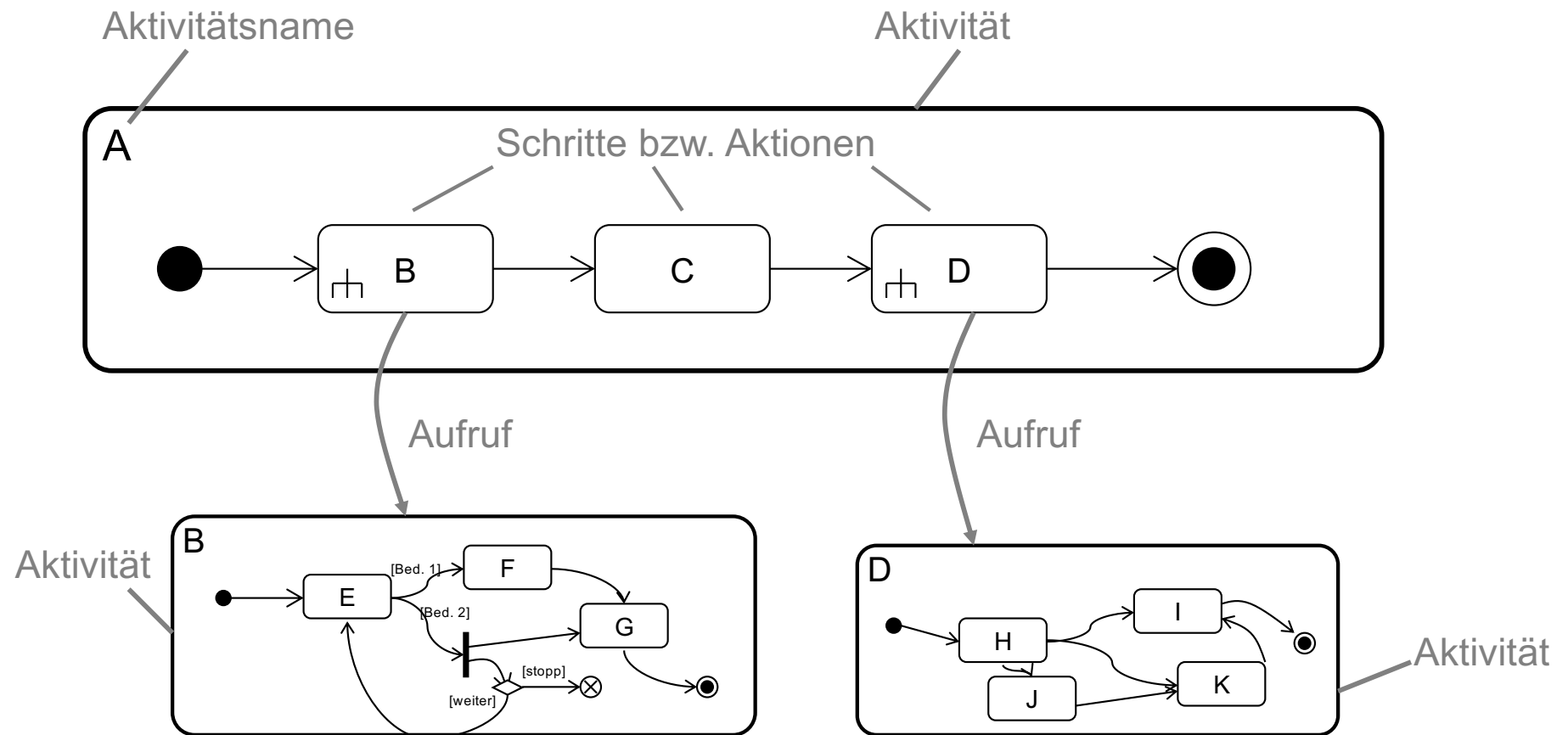


Aufrufweise als Einzelnoten:





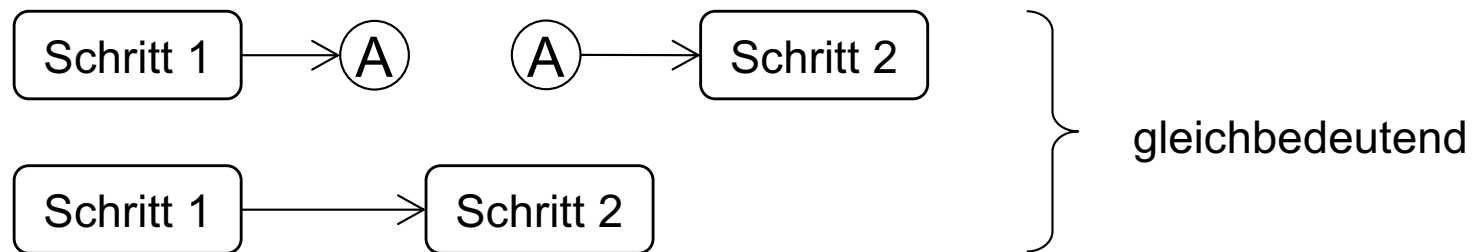
# Verschachtelung von Aktivitäten



# Aktivitätsdiagramme

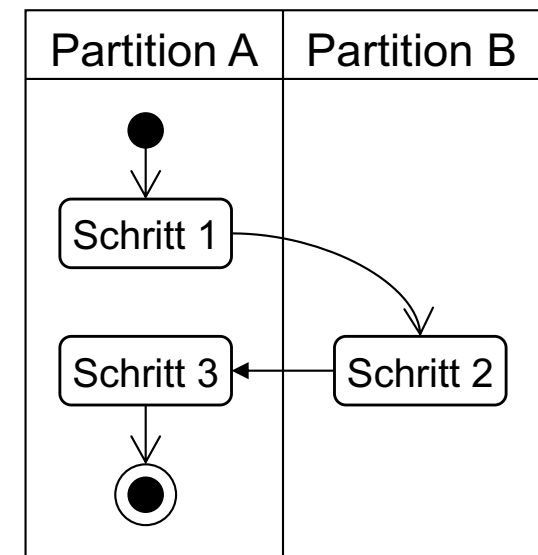
- Konnektoren

- Vermeidung langer, quer durchs Diagramm laufender Linien



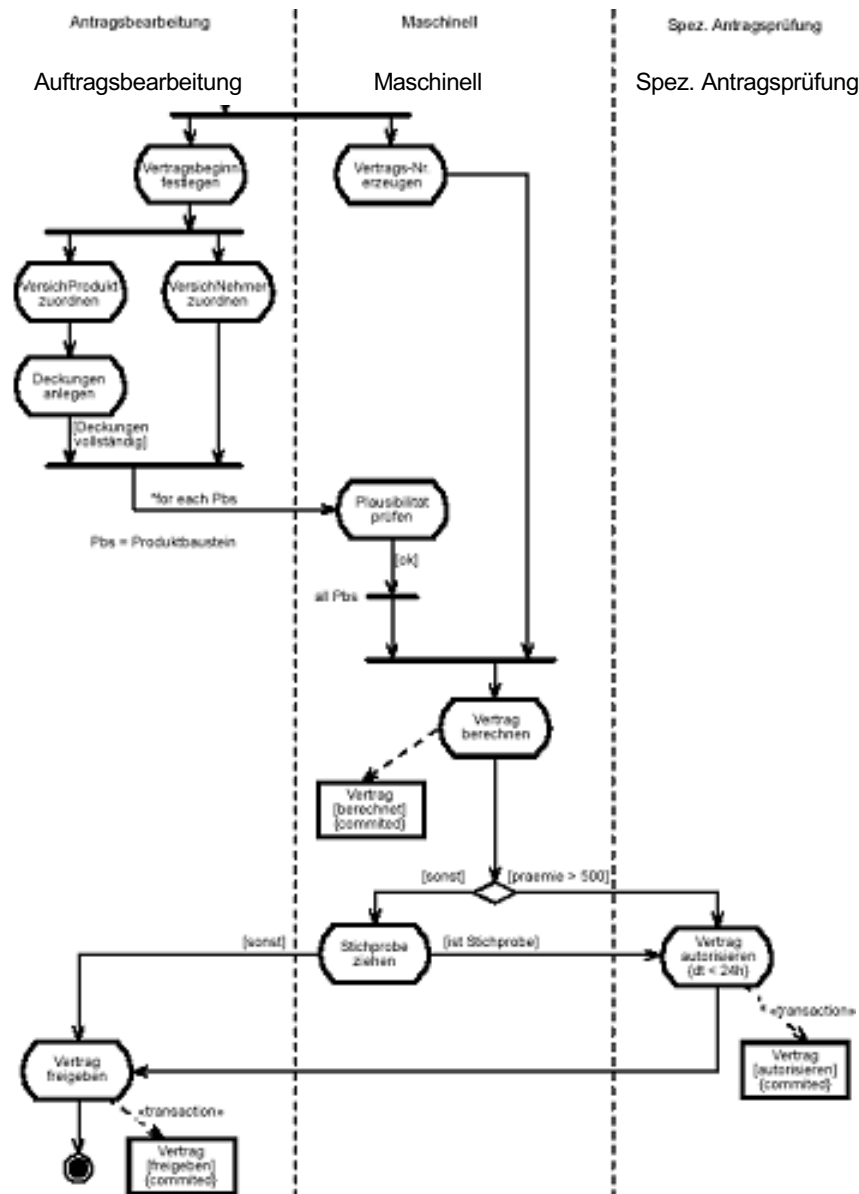
- Partitionen (swimlane, Schwimmbahnen, Eigenschaftsbereiche)

- Beschreibung, wer oder was für einen Knoten verantwortlich ist
- z.B. Organisationseinheit, Standort, Verantwortungsbereich
- Darstellung durch senkrechte Linien



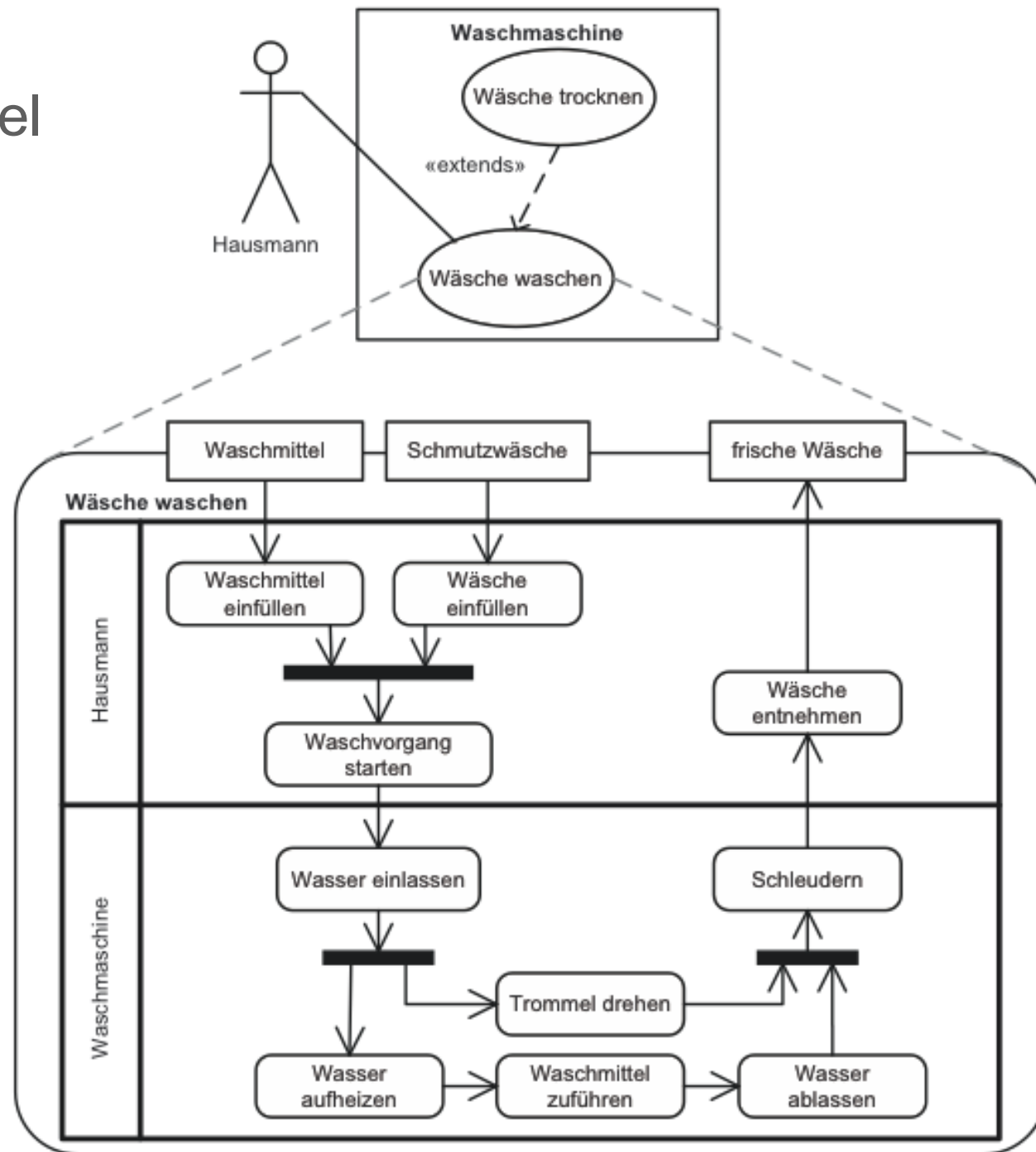
# Partitionen (swimlanes)

## Beispiel



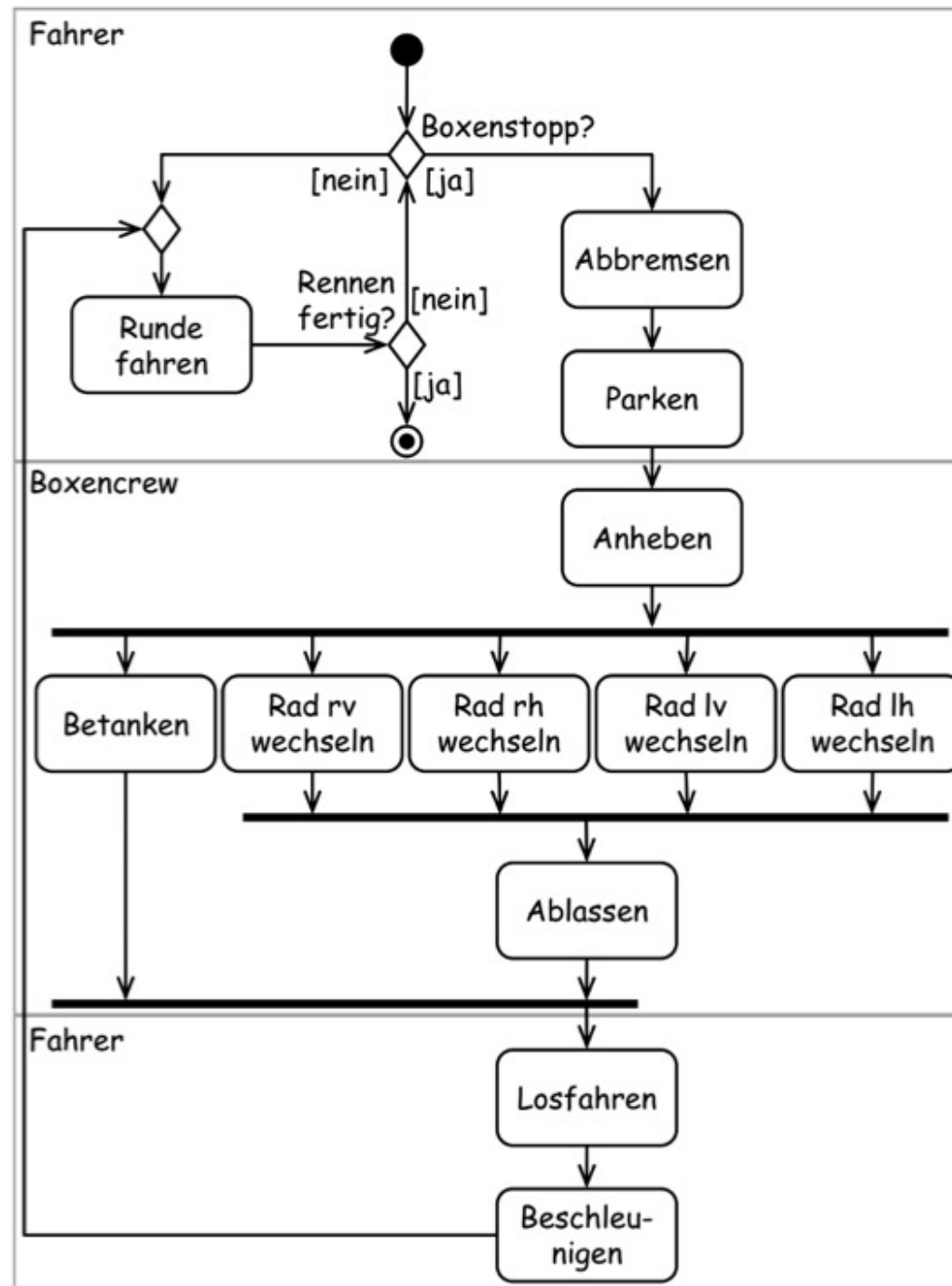
Quelle: [www.oose.de](http://www.oose.de)

# Beispiel



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Beispiel



Quelle: studwww.informatik.kit.edu

# Aktivitätsdiagramm

## Beispiel

Die Bearbeitung von Bewerbungen wird in einem Unternehmen auf folgende Weise durchgeführt:

Bewerbungen werden zunächst von der Sekretärin der Personalabteilung erfasst und in einer zentralen Bewerberkartei gespeichert. Offensichtlich ungeeignete Bewerber werden bereits hier aussortiert. Der Abteilungsleiter des zuständigen Bereichs gibt anschließend für eine Bewerbung seine schriftliche Einschätzung ab. Der Geschäftsführer des Unternehmens entscheidet daraufhin, ob ein Bewerber zu einem Vorstellungsgespräch geladen wird. Wenn sich der Geschäftsführer nach dem Vorstellungsgespräch für die Einstellung des Bewerbers entscheidet, erhält der Bewerber vom Leiter der Personalabteilung einen Arbeitsvertrag zugesendet. Abgelehnte Bewerber erhalten ihre Bewerbungsunterlagen von der Sekretärin der Personalabteilung zurück.

Modellieren Sie ein Aktivitätsdiagramm mit Swimlanes.

# UML – Übersicht

## 5 UML – Strukturdiagramme

- 5.1 Einführung UML
- 5.2 Objektdiagramm, Klassendiagramm
- 5.3 Komponentendiagramm

## 6 UML – Verhaltensmodellierung

- 6.1 Use Case-Diagramm
- 6.2 Aktivitätsdiagramm
- 6.3 Kommunikationsdiagramm
- 6.4 Sequenzdiagramm
- 6.5 Zustandsdiagramm

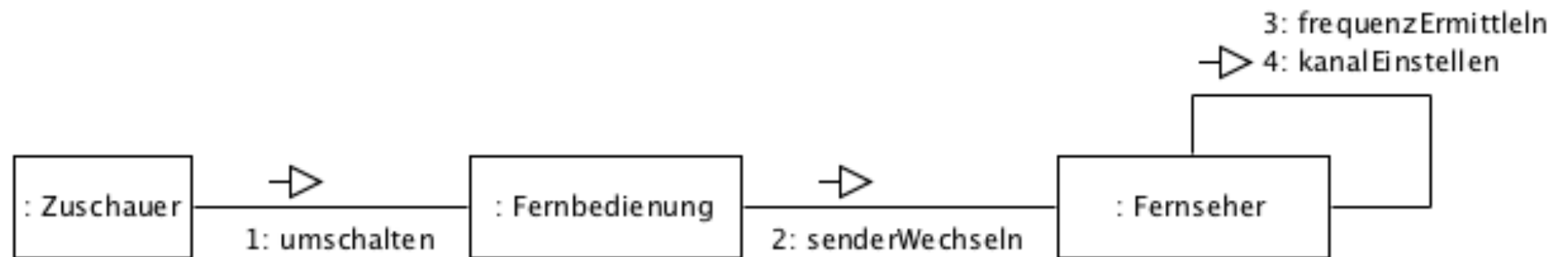
## 6.3 Kommunikationsdiagramm

- Darstellung Zusammenspiel (Nachrichtenaustausch) zwischen Kommunikationspartner und Verantwortlichkeiten
- **Nicht** im Vordergrund:
  - Kontrollsequenzen (Alternative, Schleifen)
  - Parallelität
  - Zeitliche Übergänge
- Wichtig ist das grundsätzliche Verständnis und weniger die Details
- Reihenfolge der Nachrichten durch Nummerierung
- Konsistenz zum Klassendiagramm

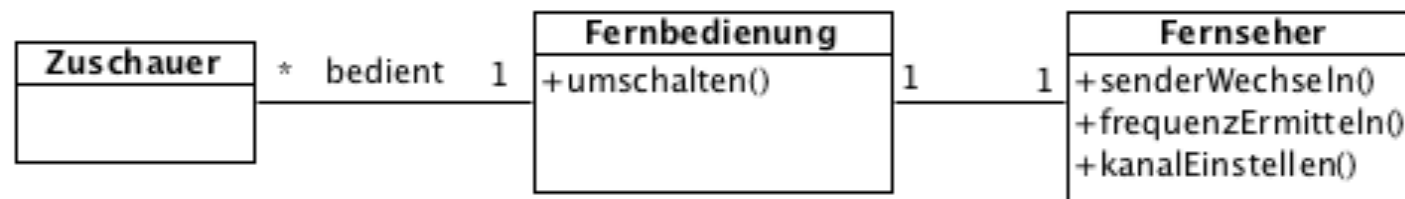


# Kommunikationsdiagramm

- Beispiel

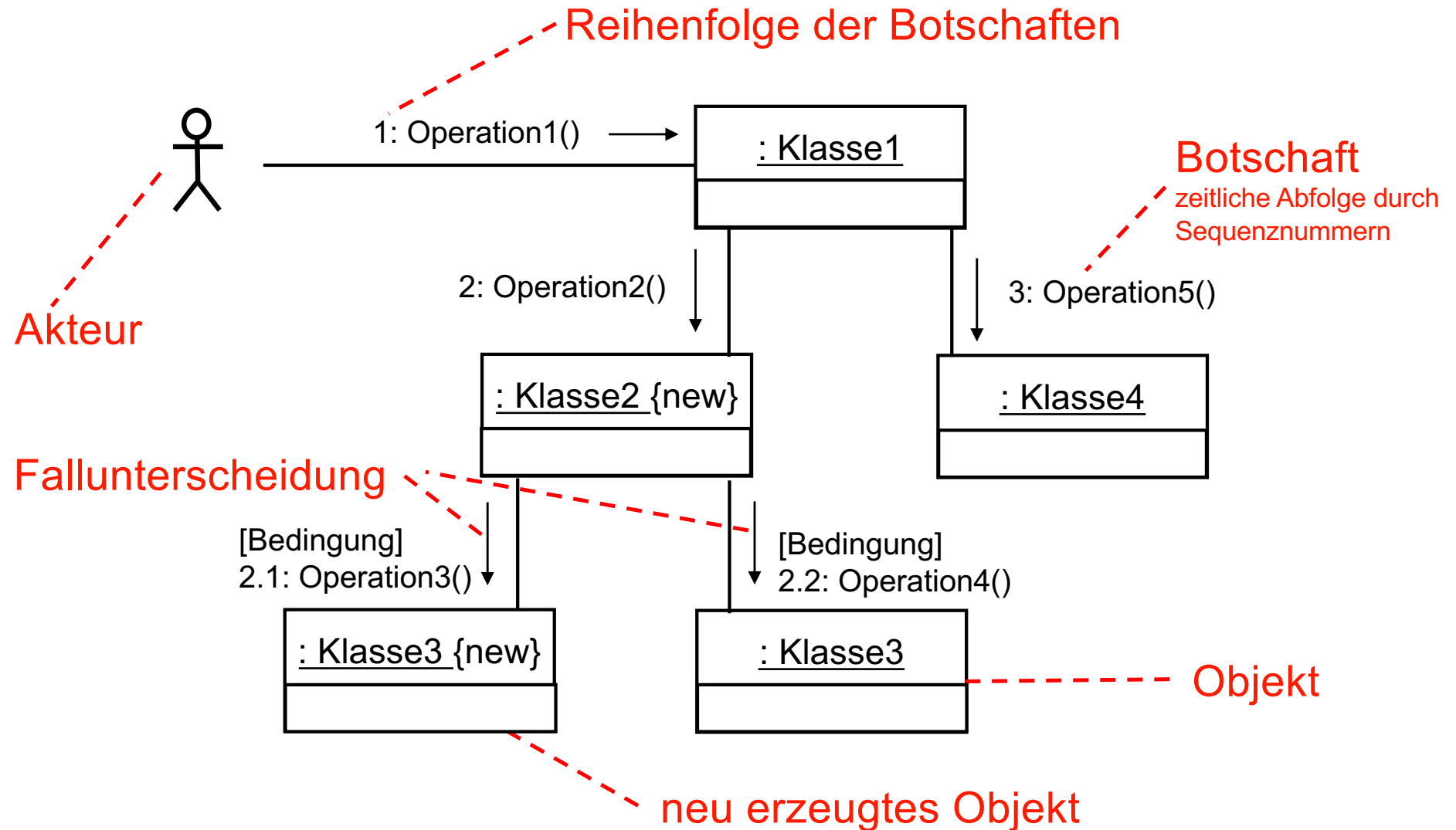


## Konsistenz zum Klassendiagramm



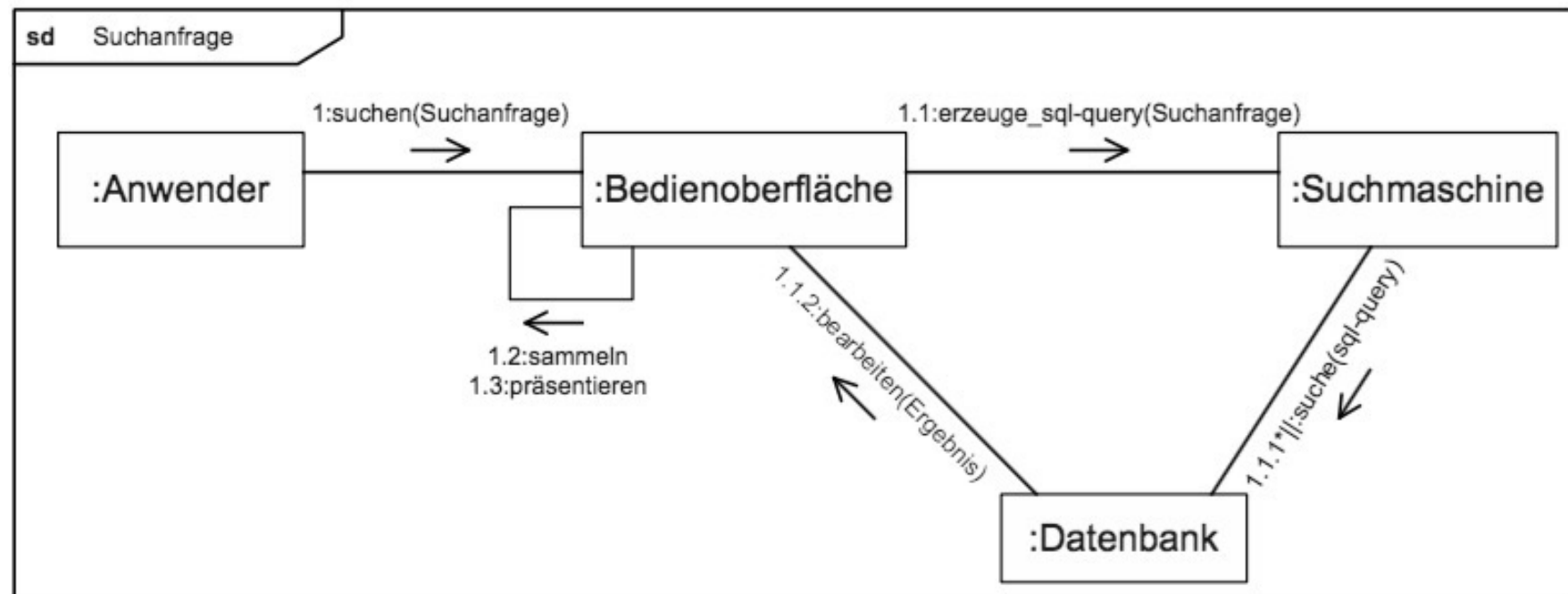
Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Kommunikationsdiagramme



# Kommunikationsdiagramm

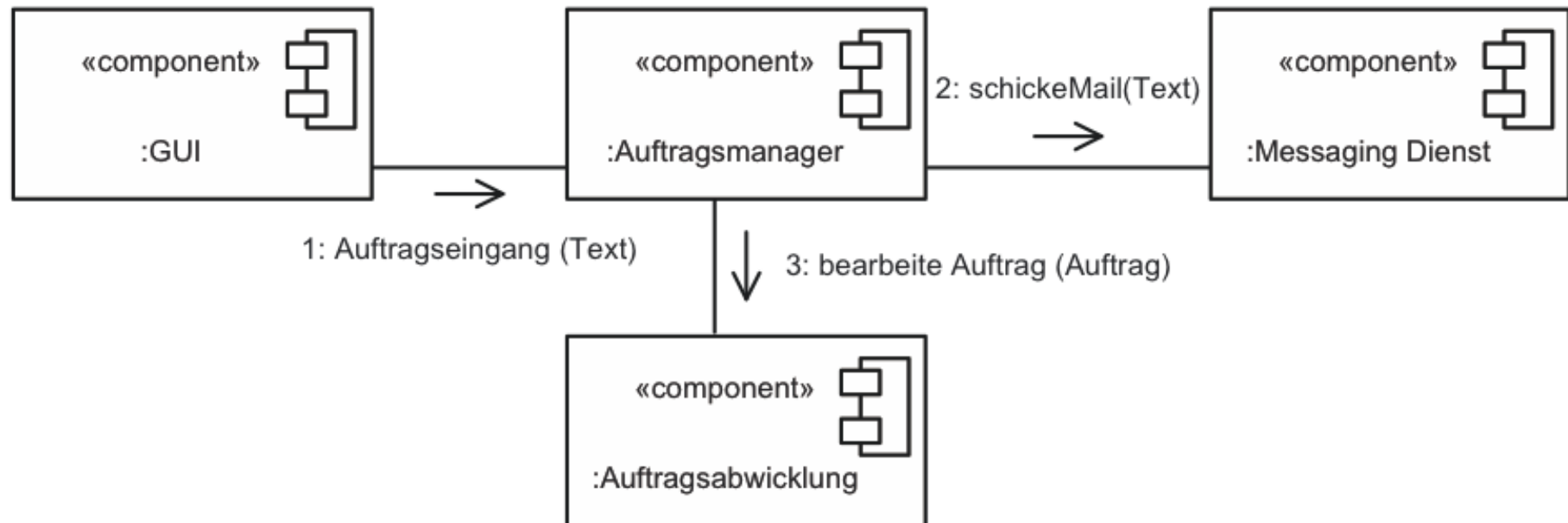
- Ziele
  - Einfache Darstellung des Zusammenwirkens von Teilen einer komplexen Struktur
  - Darstellung **eines** Zusammenspiels, nicht mehrerer Varianten



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Kommunikationsdiagramm

- Darstellung schwierig durchschaubarer Strukturen



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# UML – Übersicht

## 5 UML – Strukturdiagramme

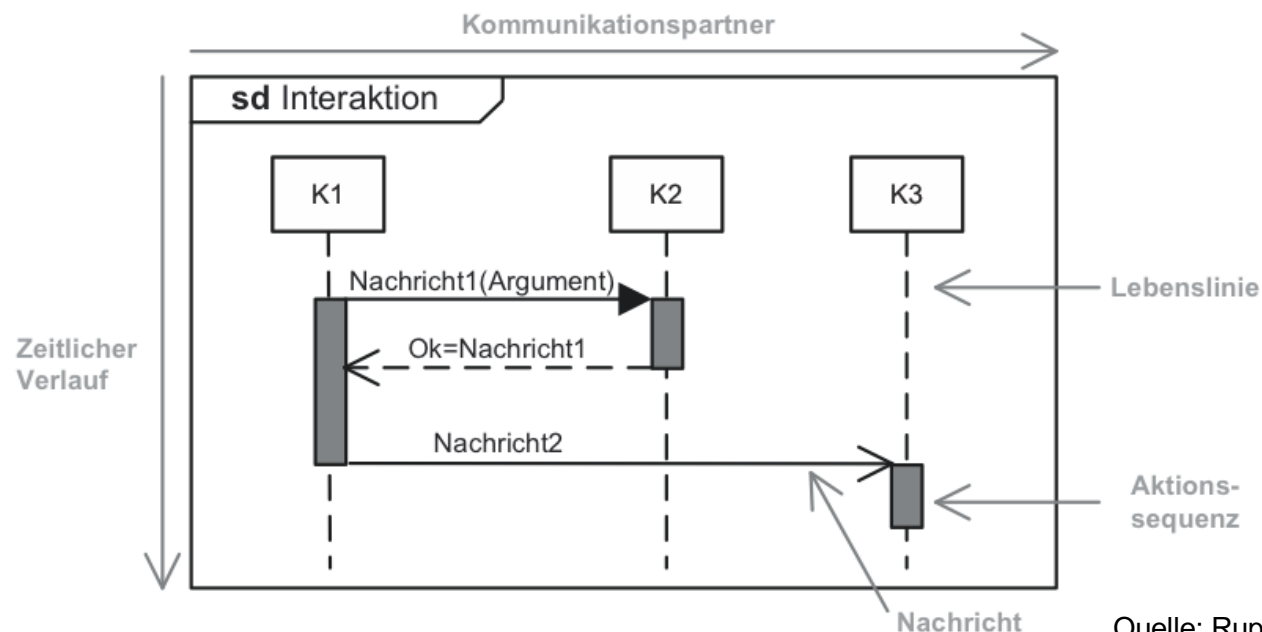
- 5.1 Einführung UML
- 5.2 Objektdiagramm, Klassendiagramm
- 5.3 Komponentendiagramm

## 6 UML – Verhaltensmodellierung

- 6.1 Use Case-Diagramm
- 6.2 Aktivitätsdiagramm
- 6.3 Kommunikationsdiagramm
- 6.4 Sequenzdiagramm
- 6.5 Zustandsdiagramm

## 6.4 Sequenzdiagramm (sequence diagram)

- Modellierung von Interaktionen zwischen mehreren Kommunikationspartnern
- Betonung liegt auf zeitlichen Ablauf
  - Anordnung auf einer Zeitachse
- Vorsichtige Verwendung
  - Für komplexe Abläufe ist Aktivitätsdiagramm besser geeignet



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Sequenzdiagramme

## Notation

Objekt : Klasse

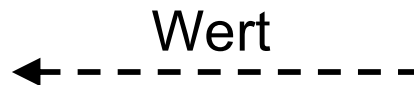
- Objekt (mit Klassenzugehörigkeit)



- Synchrone Nachricht an ein anderes Objekt



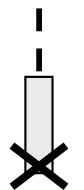
- Asynchrone Nachricht an ein anderes Objekt



- Rückgabe eines Wertes

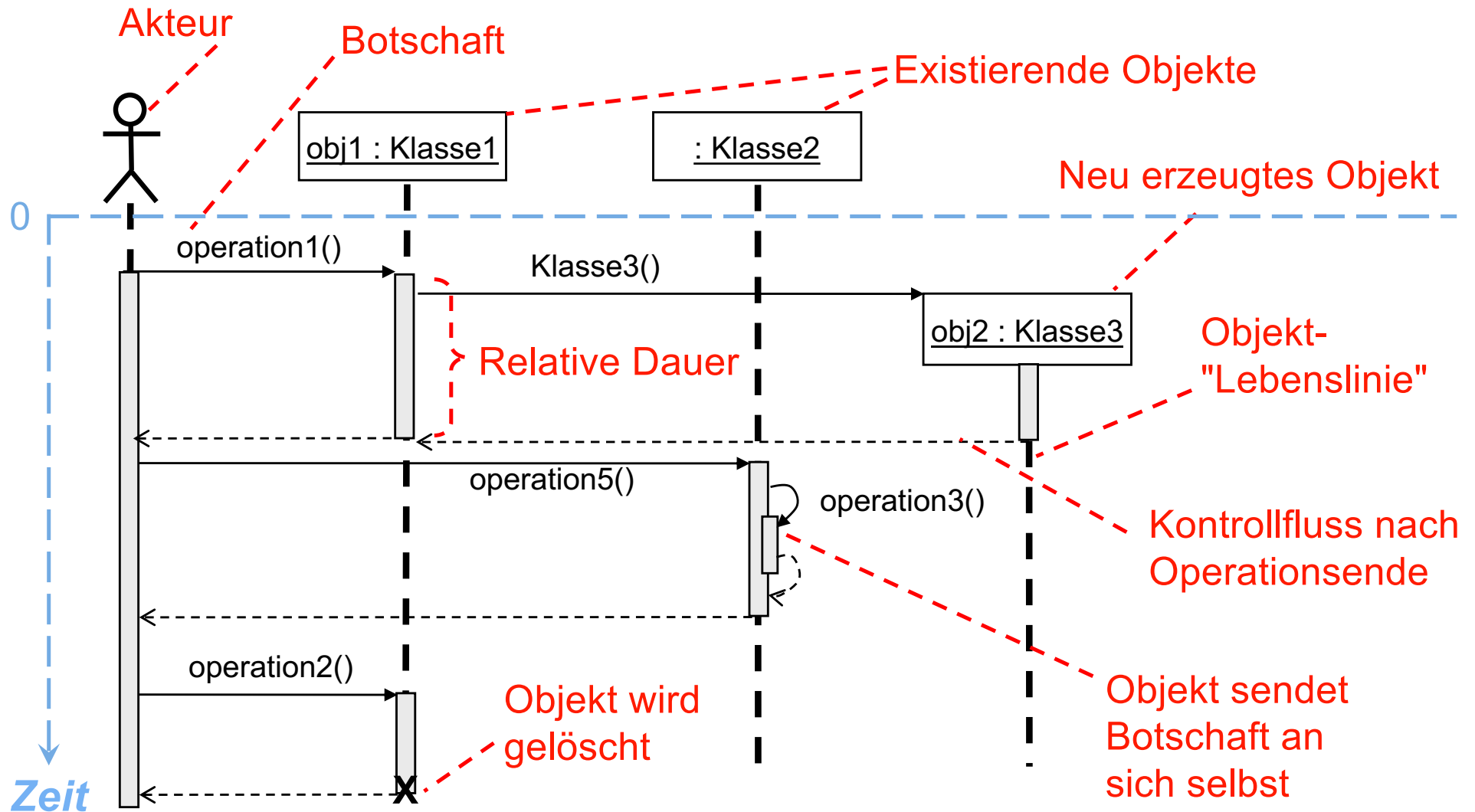


- Akteur oder Benutzer des Systems  
(= externe Schnittstelle)



- Lebenslinie eines Objekts,  
  ▮ zeigt Steuerungsfokus (Programmkontrolle) an  
  X zeigt Destruktor an

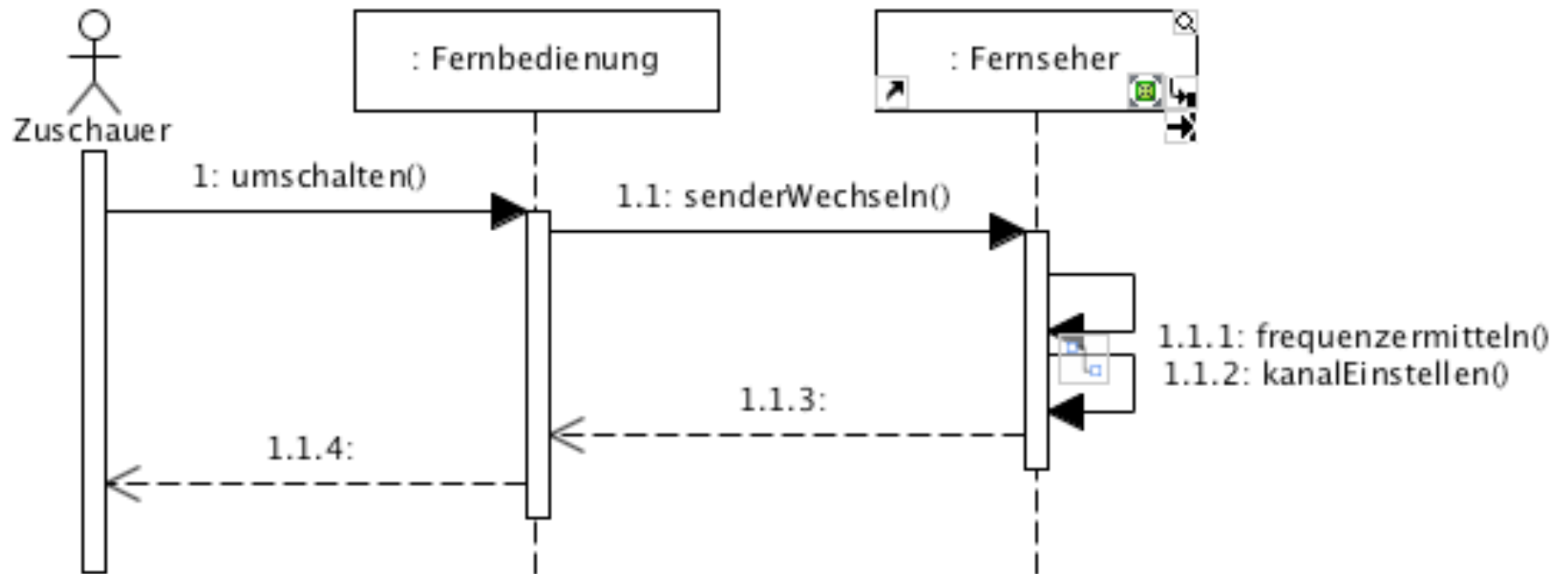
# Notation





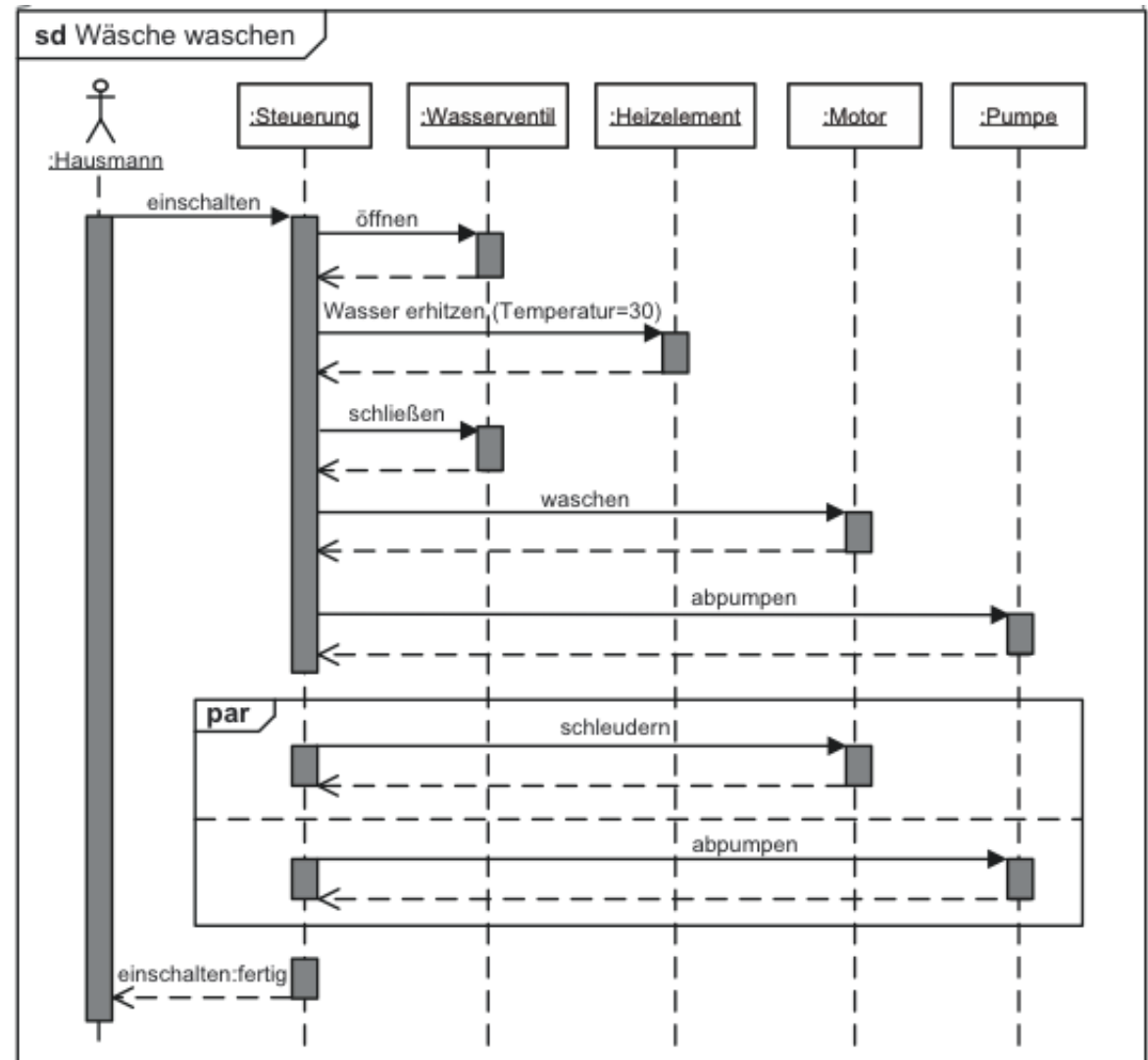
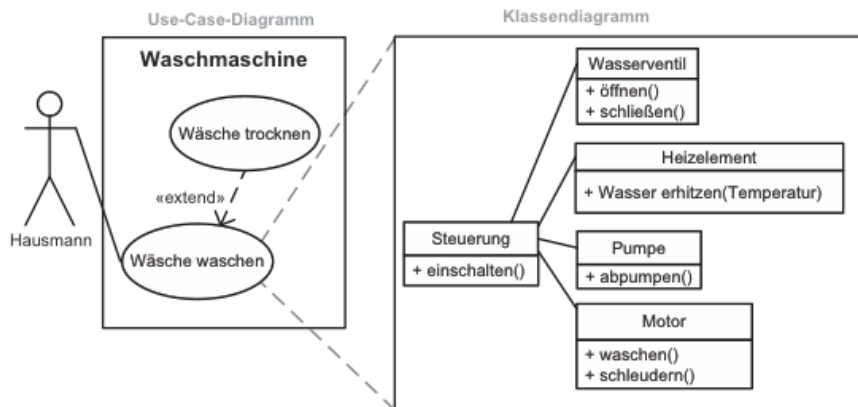
# Sequenzdiagramm

- Beispiel Sequenzdiagramm



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

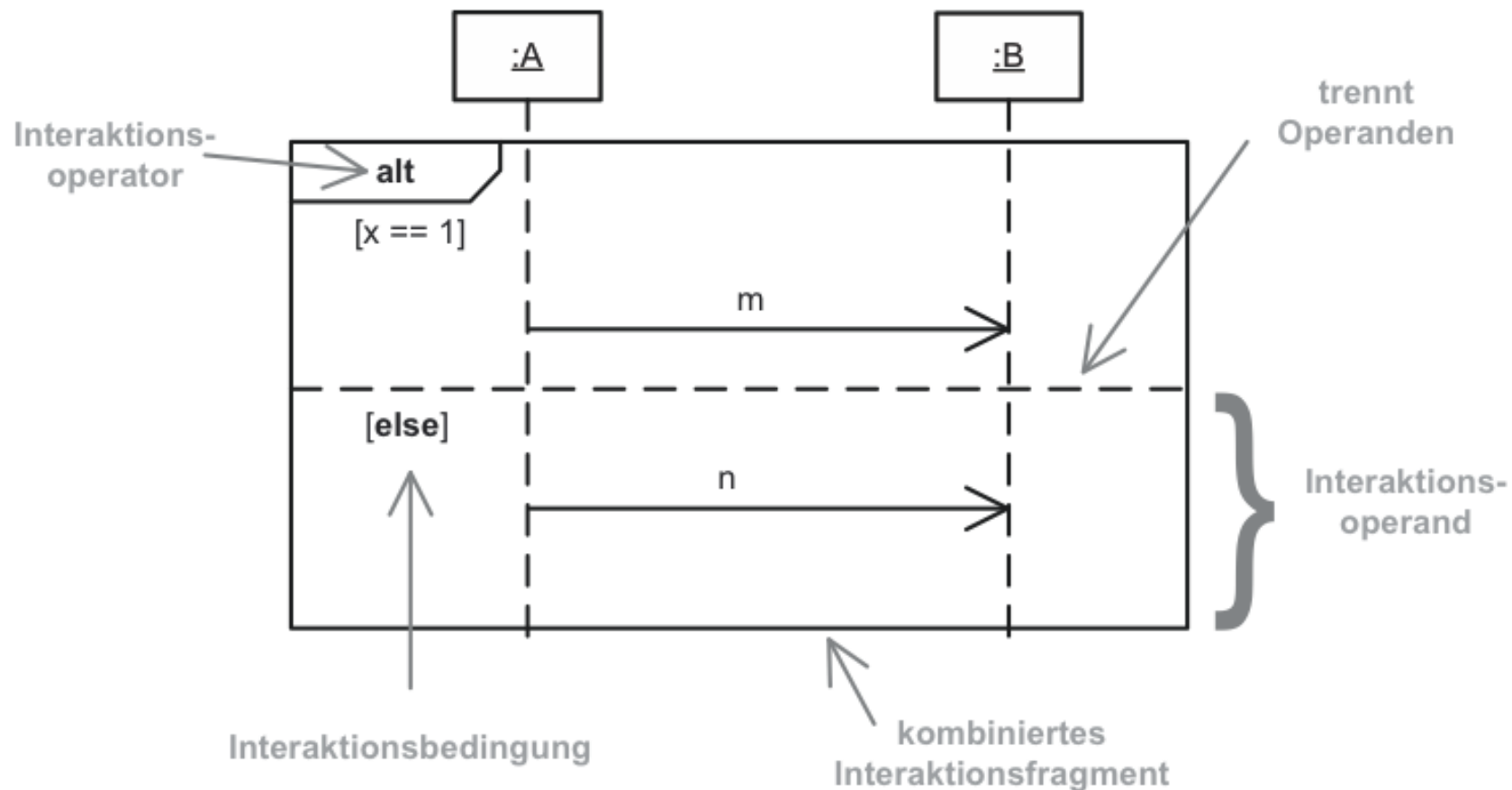
# Sequenzdiagramm - Beispiel



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Kombiniertes Fragment

- Kombiniertes Fragment (combined fragment)
  - Verwendung eines Interaktionsoperators als Kürzel



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Interoperationsoperatoren

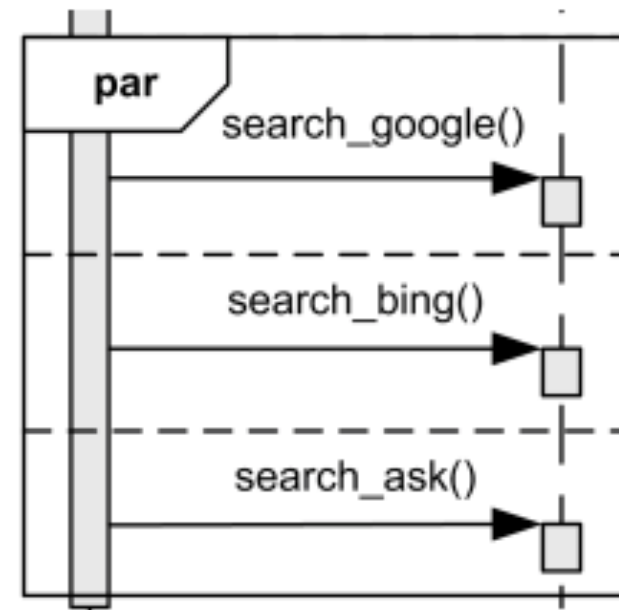
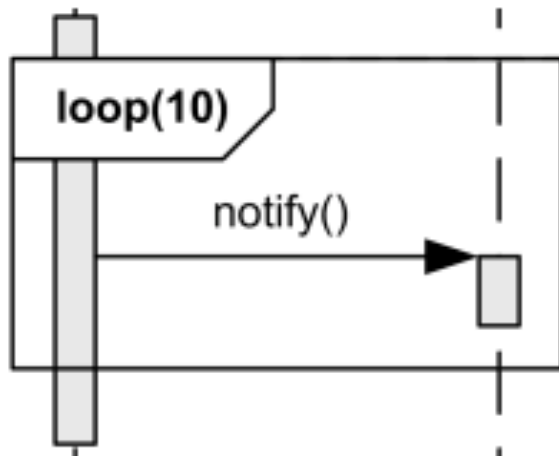
Deutsche Bezeichnung	Englische Bezeichnung	Kürzel im Diagramm	Damit modellieren Sie ...
Alternative Fragmente	Alternative	alt	... alternative Ablaufmöglichkeiten
Optionales Fragment	Option	opt	... optionale Interaktionsteile
Abbruchfragment	Break	break	... Interaktionen in Ausnahmefällen
Negation	Negative	neg	... ungültige Interaktionen
Schleife	Loop	loop	... iterative Interaktionen
Parallele Fragmente	Parallel	par	... nebenläufige Interaktionen
Lose Ordnung	Weak Sequencing	seq	... von Lebenslinie und Operanden <i>abhängige</i> chronologische Abläufe
Strenge Ordnung	Strict Sequencing	strict	... von Lebenslinie und Operanden <i>unabhängige</i> chronologische Abläufe
Kritischer Bereich	Critical Region	critical	... atomare Interaktionen
Irrelevante Nachrichten	Ignore	ignore	... Filter für unwichtige Nachrichten
Relevante Nachrichten	Consider	consider	... Filter für wichtige Nachrichten
Sicherstellung	Assertion	assert	... unabdingbare Interaktionen

Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Kombinierte Fragmente

## Beispiele

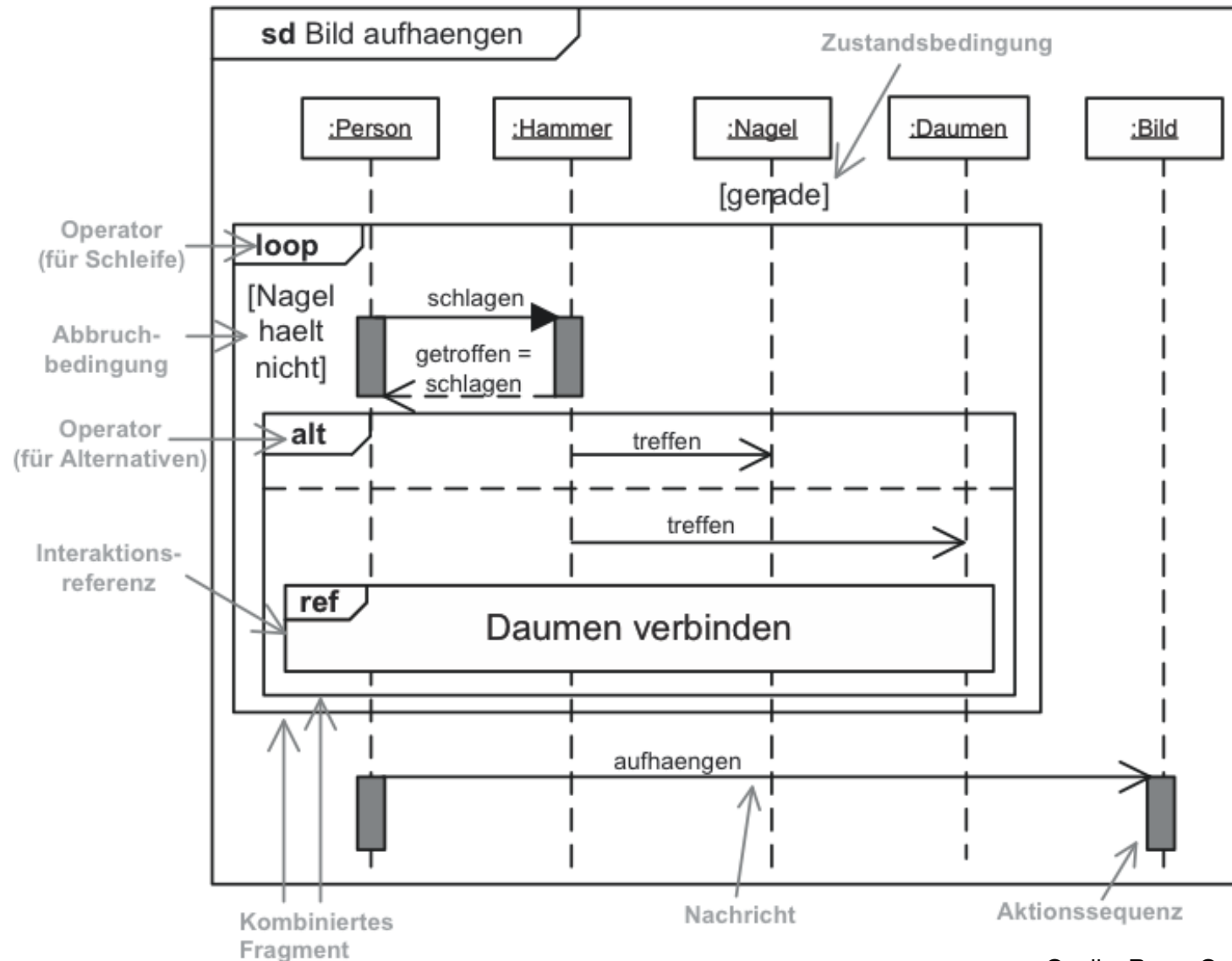
- Interaktionsoperatoren
  - loop: Schleife
  - par: Parallele Fragmente



Quelle: [www.uml-diagrams.org](http://www.uml-diagrams.org)

# Kombinierte Fragmente

## Beispiele



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Übungsbeispiele

Beschreiben Sie folgende Problemstellung als Aktivitäts-Kommunikations- und als Sequenzdiagramm.

Beschreiben Sie die zugehörigen Use Case- und Klassendiagramme und achten Sie auf Konsistenz zwischen den Diagrammen.

Ein Kunde gibt in ein Onlineshop eine Bestellung auf. Der Bestellung wird eine oder mehrere Bestellpositionen hinzugefügt. Bei einem bestellten Artikel wird nachgeschlagen, ob der Artikel überhaupt vorrätig ist. Wenn er lieferbar ist, wird die Bestellposition bestätigt. Am Ende der Bestellung wird eine Rechnung erstellt.

# UML – Übersicht

## 5 UML – Strukturdiagramme

- 5.1 Einführung UML
- 5.2 Objektdiagramm, Klassendiagramm
- 5.3 Komponentendiagramm

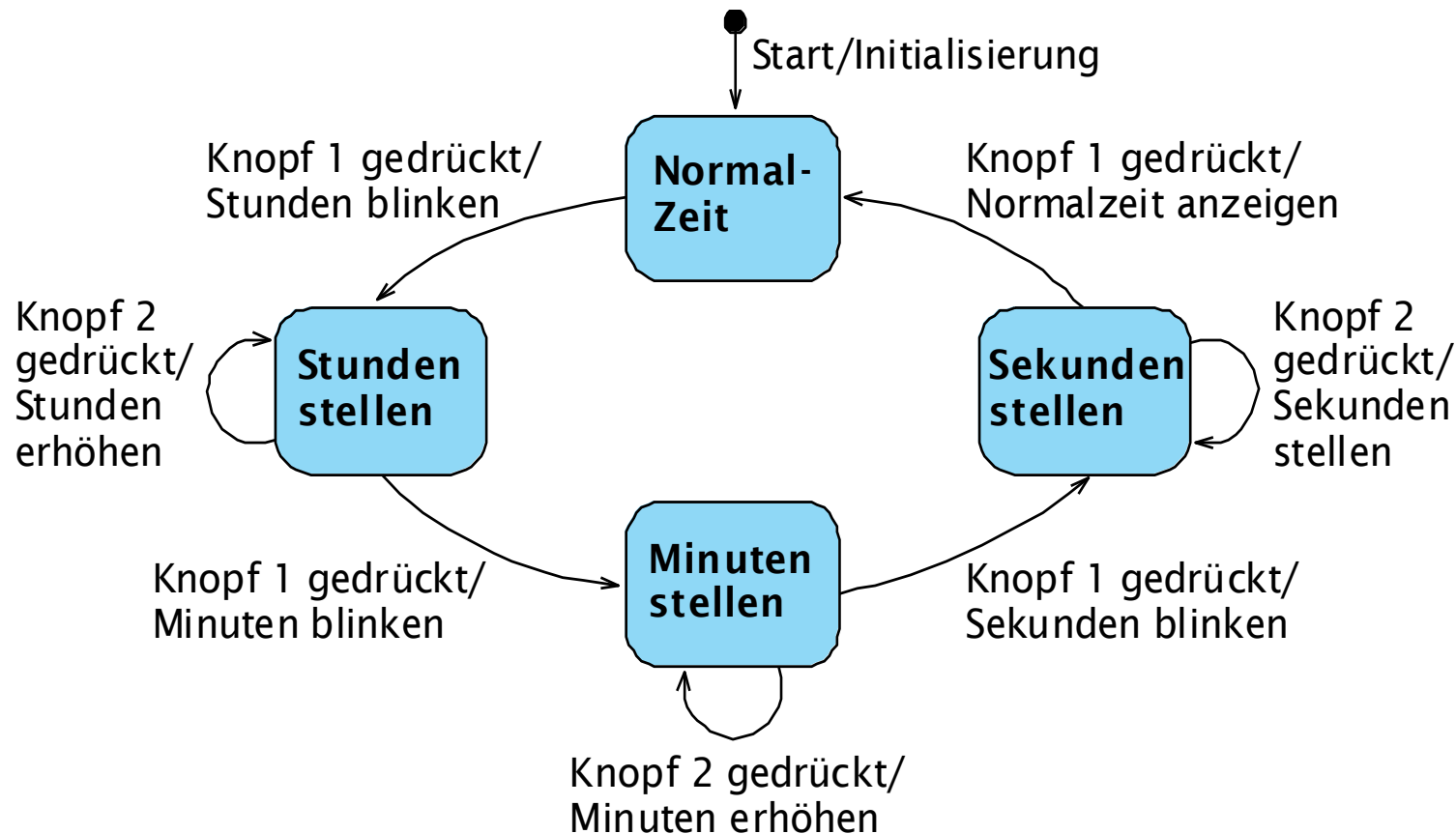
## 6 UML – Verhaltensmodellierung

- 6.1 Use Case-Diagramm
- 6.2 Aktivitätsdiagramm
- 6.3 Kommunikationsdiagramm
- 6.4 Sequenzdiagramm
- 6.5 Zustandsdiagramm



## 6.5 Zustandsdiagramm (state diagram)

- Modellierung des 'Lebenszyklus' eines Objektes
- Wiederholung aus Kap. 4



## 6.5 Zustandsdiagramm (state diagram)

- Modellierung der Zustände
  - einer Klasse
  - einer Komponente
  - des Gesamtsystems
- Zuordnung Zustandsdiagramm zu Klasse
  - Alle Objekte einer Klasse besitzen denselben Zustandsautomaten
  - Jedes Objekt kann jedoch einen individuellen Zustand einnehmen
- OO-Zustandsautomaten
  - Es ist nicht notwendig, für jede Klasse ein Zustandsdiagramm aufzustellen
  - Als Aktionen und Aktivitäten sind nur Operationen der jeweiligen Klasse zulässig

# Zustandsautomaten

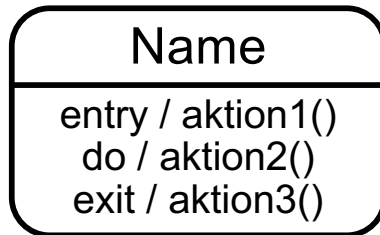
- Zustand
  - Gehört zu einem Zustandsautomaten beispielsweise einer Klasse
  - Fachlich motivierte Abstraktion bzw. Zusammenfassung einer Menge von möglichen Attributwerten
  - Nicht jede Änderung eines Attributwertes eines Objekts wird als Zustandsänderung angesehen
  - Modellierung: wann verhält sich das System ähnlich = Zustände
- Ereignis, Event, Trigger
  - Beachtendes Vorkommnis mit Bedeutung
  - Ursachen: Eine Bedingung wird erfüllt oder Objekt erhält Nachricht
- Ereignisse können Aktionen auslösen
  - entry: löst automatisch beim Eintritt einen Zustand aus
  - exit: löst automatisch beim Verlassen eines Zustands aus
  - do: wird immer wieder ausgelöst, solange Zustand aktiv ist

# Zustandsdiagramm

## Notation



Zustand



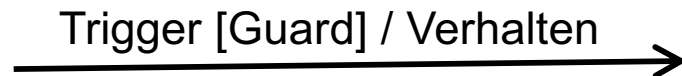
Zustand mit internen Aktionen



Anfangszustand



Endzustand



Zustandsübergang

# Elemente im Zustandsautomaten

- Trigger
  - Signaltrigger: Eingehendes Signal
  - CallTrigger: Aufruf einer Operation
  - TimeTrigger: Ablauf einer bestimmten Zeitdauer oder zu einem bestimmten Zeit
  - ChangeTrigger: Änderung des Wertes einer Variablen
- Guard
  - Guard wird nach Auftreten des Triggers ausgewertet
  - Wenn Guard true ergibt, wird Transition ausgeführt
- Verhalten
  - Verhalten wird durchgeführt, wenn Trigger eingetreten ist und Bedingung in der Guard erfüllt ist

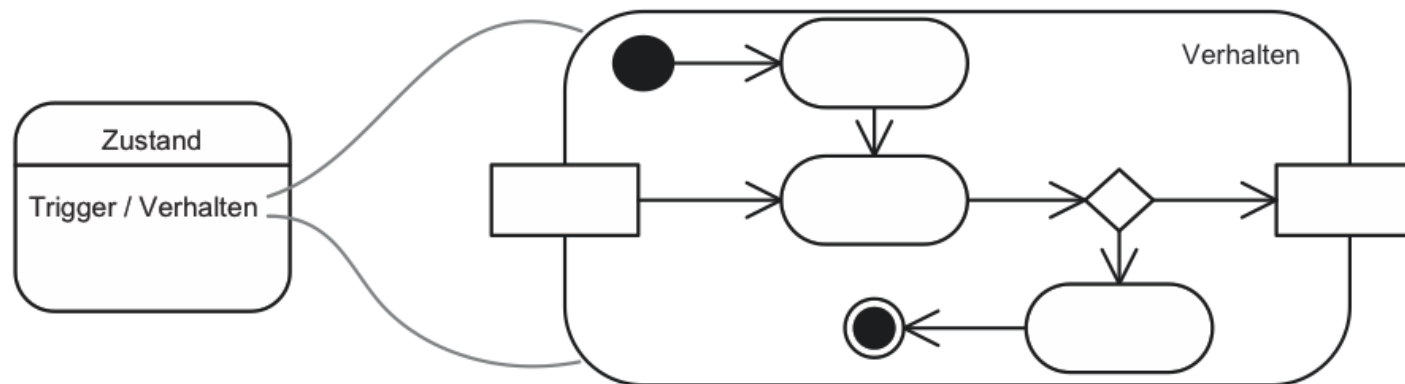
# Zustandsdiagramm

## Notation

- Eintritts-, Zustands- und Austrittsverhalten



- Verhalten in einem Zustand

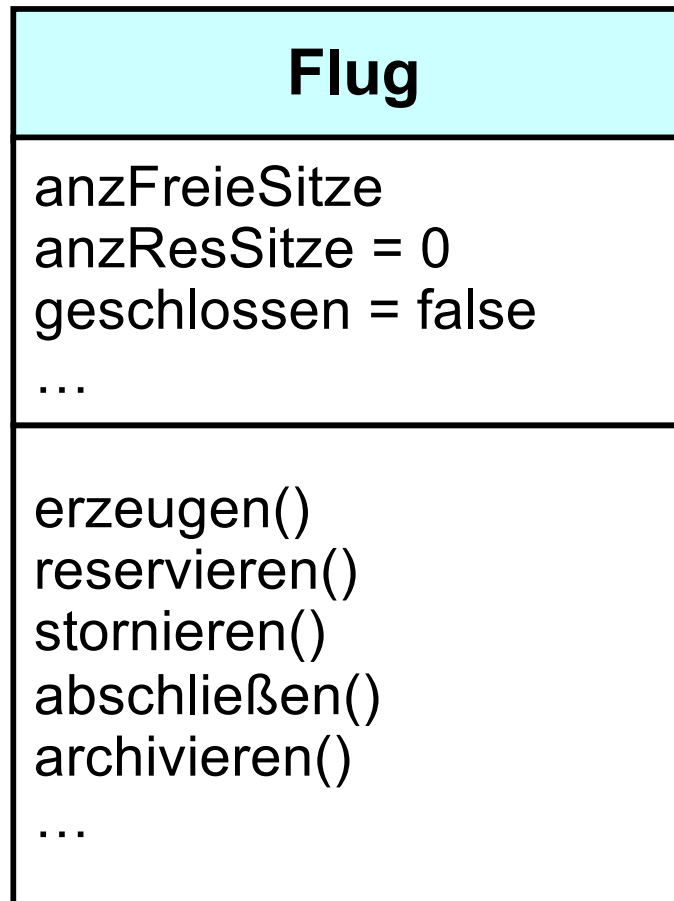


Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Zustandsautomaten

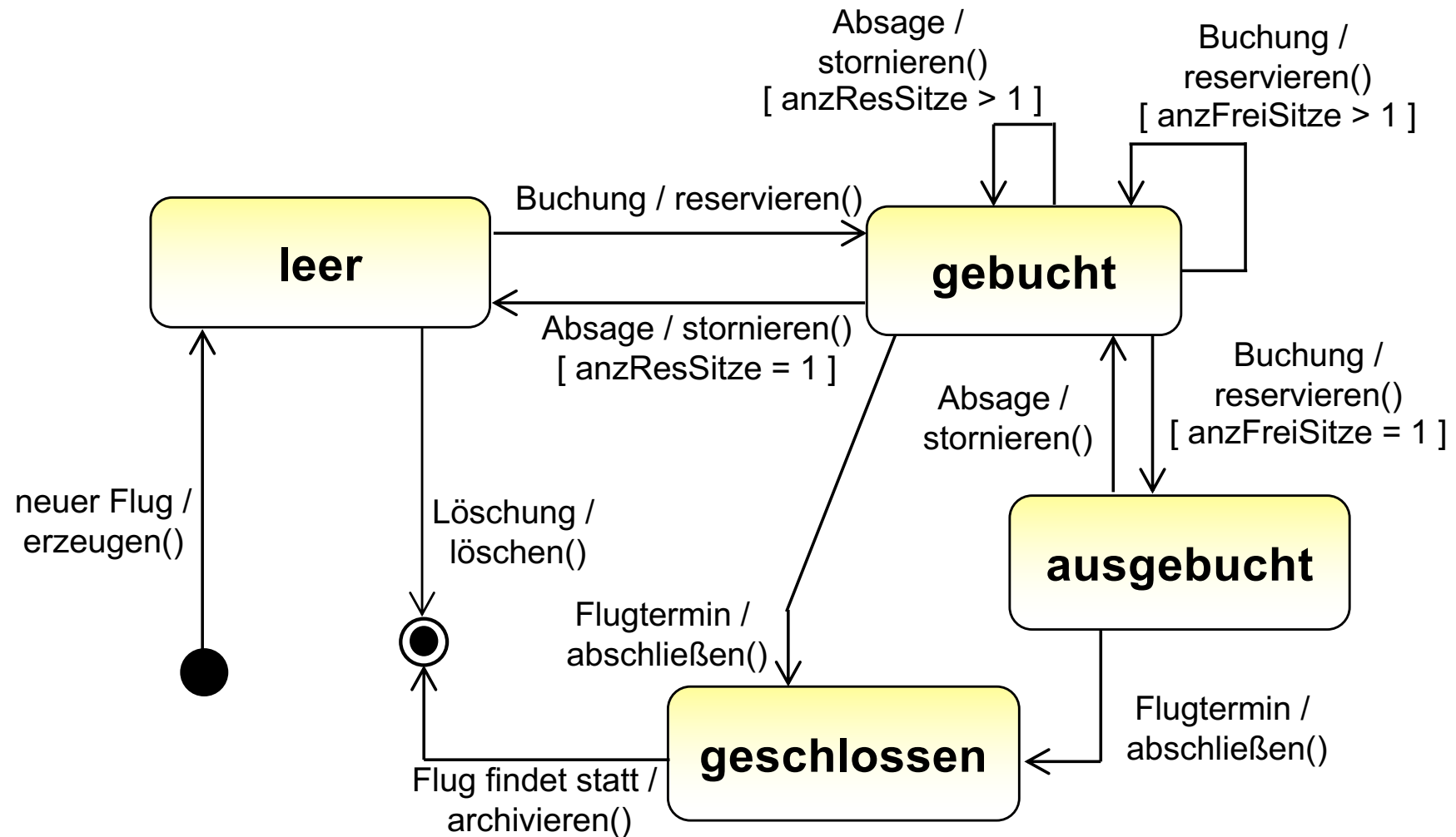
## Klasse

- Klassendefinition mit verwendeten Attributen und Operationen



# Zustandsdiagramm

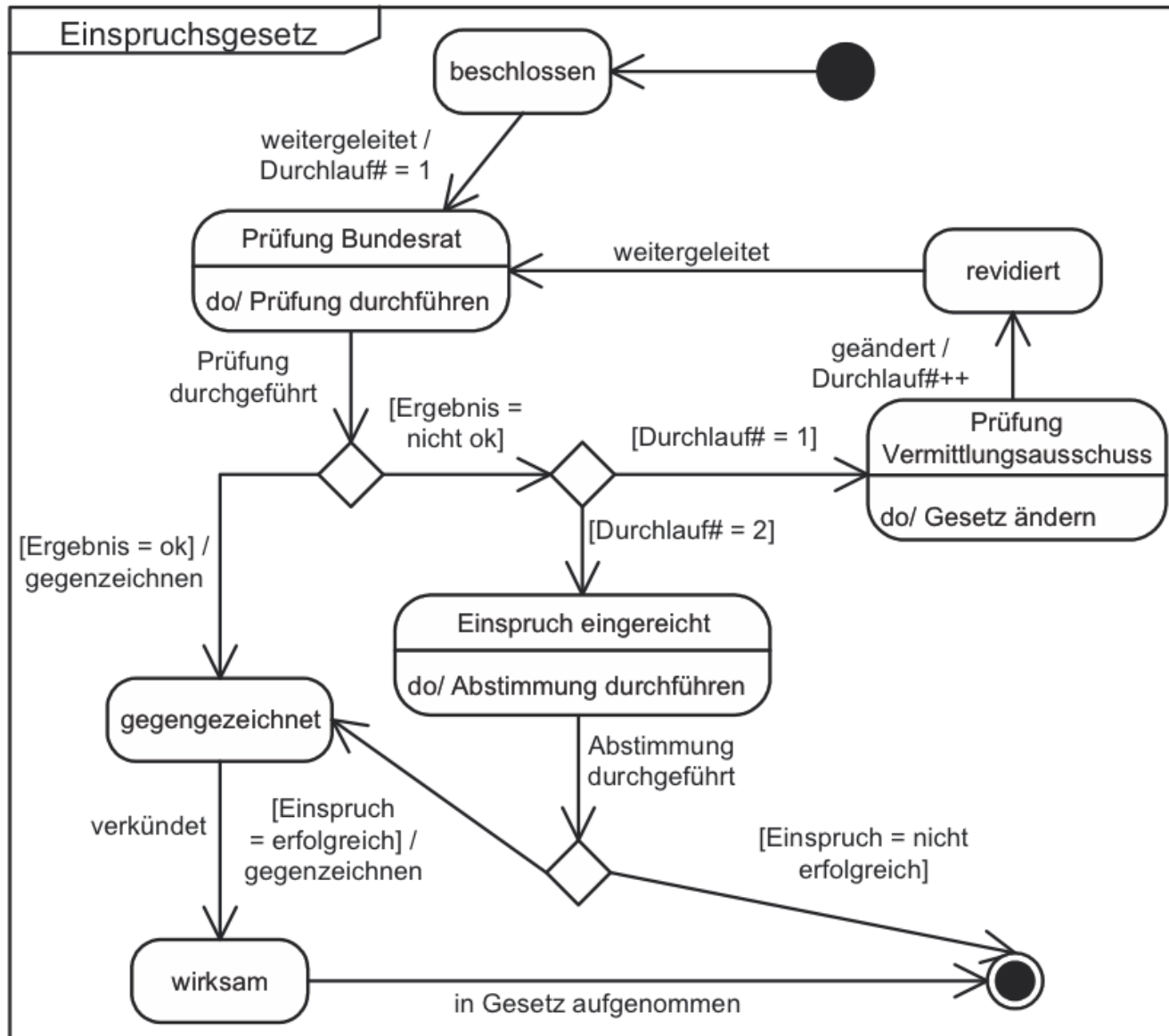
## Beispiel





# Zustandsdiagramm

## Beispiel



Quelle: Rupp, Queins, die Sophisten: UML2 glasklar

# Aufgabe (10 Punkte) aus Klausur SS03

Ergänzen Sie die folgende Diagrammskizze zu einem (vollständigen!) UML-Zustandsdiagramm (mit Ausgabe beim Zustandsübergang), durch das der Lebenszyklus eines Buches in einer Leihbibliothek beschrieben wird



# Klausur SS05 - Zustandsdiagramm

Moderne Fahrrad-Computer zeigen nicht nur die Geschwindigkeit und die Anzahl zurückgelegter Kilometer an, sondern können auch den Puls des Sportlers angeben. Hierbei misst ein Pulsmesser den Puls des Sportlers und übermittelt die Daten per Funk an den Fahrrad-Computer. Die Funktionsweise des Fahrrad-Computers ist wie folgt gegeben:

- Der Fahrrad-Computer hat eine "ON"-Taste, durch die man den Fahrrad-Computer jeder Zeit ein- oder ausschalten kann
  - Wenn der Fahrrad-Computer eingeschaltet ist und kein Signal vom Pulsmesser empfängt, wird die Zahl "0" blinkend angezeigt
  - Wenn der Computer ein Signal vom Pulsmesser empfängt, wird dieser Wert nicht-blinkend angezeigt
  - Sobald der Computer kein Signal mehr empfängt, wird wieder die blinkende "0" angezeigt
  - Wenn der Puls über einem vorgegebenen Höchstwert liegt, wird ein Hinweis auf der Anzeige angegeben, welcher dem Sportler signalisiert, dass sein Puls zu hoch ist
  - Die Warn-Anzeige wird erst dann beendet, wenn der Puls wieder unterhalb des vorgegebenen Höchstwertes liegt
- a) Geben Sie ein UML-Zustandsdiagramm für die Puls-Anzeige eines Fahrrad-Computers an. Beschreiben Sie dabei sowohl die beteiligten Ereignisse, als auch die Aktionen.
- b) Erstellen Sie ein Klassendiagramm für die Klasse Fahrrad-Computer, in dem die Klasse alle notwendigen Attribute und Operationen besitzt.