



IREB Foundation Level

LE 3.4 Modellbasierte Arbeitsprodukte (K3)

IREB-FL Struktur

Tag	Inhalt
Tag 1	LE 1, LE 2, LE 3.1, LE 3.5 bis LE 3.8, LE 4.1 bis LE 4.2
Tag 2	LE 3.2 bis LE 3.4
Tag 3	LE 4.3 bis LE 4.4, LE 5, LE 6, LE 7
Tag 4	Zertifizierungsprüfung

IREB-FL Tagesablauf

Tag	Inhalt
9:00 – 10:45	LE 3.2 Natürlichsprachige Arbeitsprodukte LE 3.3 Vorlagenbasierte Arbeitsprodukte LE 3.3 Übung Satzschablone
10:45 – 11:00	Pause
11:00 – 12:30	LE 3.4.1 Die Rolle von Modellen im Requirements Engineering LE 3.4.2 Modellierung des Systemkontexts LE 3.4.2 Übung Use-Case Diagramm
12:30 – 13:30	Mittagspause
13:30 – 15:00	Drei Perspektiven auf Anforderungen LE 3.4.3 Modellierung von Struktur und Daten LE 3.4.3 Übung UML-Klassendiagramm
15:00 – 15:15	Pause
15:15 – 17:00	LE 3.4.4 Modellierung von Funktion und Ablauf LE 3.4.5 Zustands- und Verhaltensmodellierung LE 3.4.6 Modellierung von Zielen LE 3.4.5 Übung UML-Zustandsdiagramm

LE 3.4 Agenda Modellbasierte Arbeitsprodukte

- LE 3.4.1 Die Rolle von Modellen im Requirements Engineering
 - Zweck – Eigenschaften – Vor- und Nachteile von Modellen
- LE 3.4.2 Modellierung des Systemkontexts
 - Kontextmodelle – Datenflussdiagramme – UML Use-Case Diagramme
- Drei Perspektiven auf Anforderungen
 - Strukturperspektive – Funktionsperspektive – Verhaltensperspektive
- LE 3.4.3 Modellierung von Struktur und Daten (Strukturperspektive)
 - ER-Diagramme – UML-Klassendiagramm
- LE 3.4.4 Modellierung von Funktion und Ablauf (Funktionsperspektive)
 - UML-Aktivitätsdiagramme
- LE 3.4.5 Zustands- und Verhaltensmodellierung (Verhaltensperspektive)
 - Statecharts – UML-Zustandsdiagramm
- LE 3.4.6 Modellierung von Zielen
 - UND/ODER-Bäume

LE 3.4 Modellbasierte Arbeitsprodukte

Warum Anforderungen modellieren?

Textuell Anforderungen

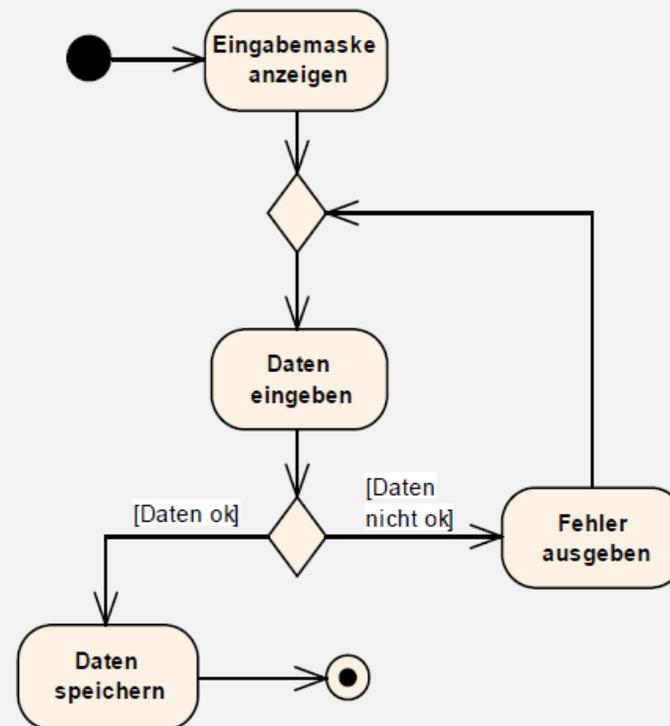
Req-1: Das System muss die Eingabemasken anzeigen.

Req-2: Nachdem die Aktion „Eingabemaske anzeigen“ beendet wurde, oder nachdem die Aktion „Fehler anzeigen“ beendet wurde, muss das System dem Benutzer die Möglichkeit bieten, Daten einzugeben.

Req-3: Nachdem die Aktion „Daten eingeben“ beendet wurde, und falls die Daten ok sind, muss das System die Daten speichern.

Req-4: Nachdem die Aktion „Daten eingeben“ beendet wurde, und falls die Daten nicht ok sind, muss das System die Fehler ausgeben.

Modellierte Anforderungen



LE 3.4 Modellbasierte Arbeitsprodukte

Warum Anforderungen modellieren?

Modelle sind:

- Übersichtlicher
- Strukturierter
- Verständlicher
- Nachvollziehbarer
- Komplexitäts reduzierend
- besser geeignet den Gesamtzusammenhang explizit ersichtlich darzustellen

im Vergleich zur textuellen, natürlichsprachigen Anforderungs-Darstellung

LE 3.4.1

Die Rolle von Modellen im Requirements Engineering

Zweck – Eigenschaften – Vor- und Nachteile von Modellen

LE 3.4.1 Modelle im Requirements Engineering

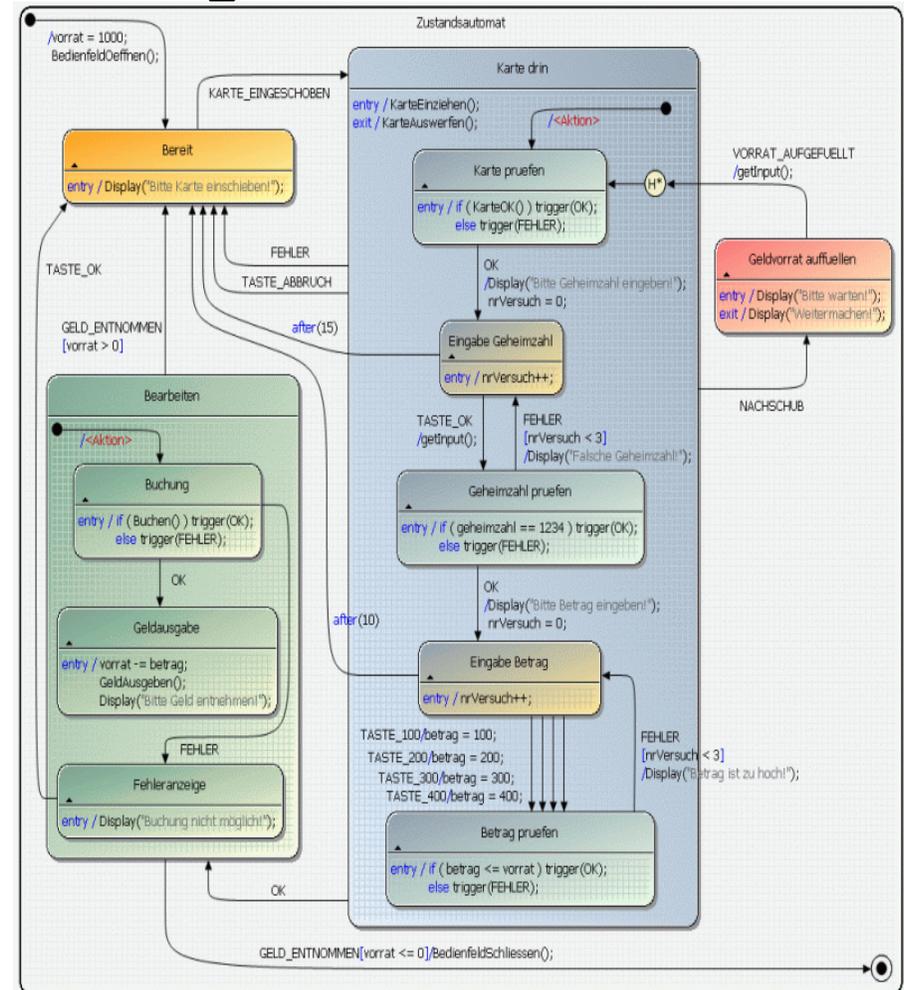
Modell Definition:

Ein Modell ist ein
abstraktes Abbild
einer existierenden oder
einer noch zu schaffenden
Realität

LE 3.4.1 Modelle im Requirements Engineering

Zweck von Modellen im Requirements Engineering:

- Darstellung von Anforderungen modellbasiert
- Modellierung von Anforderungen als Mittel zur *Spezifikation* von *Anforderungen*
- Modellierung existierender textueller Anforderungen zur *besseren Verständlichkeit*
- Modellierung existierender textueller Anforderungen zum *Zwecke* der *Prüfung*



LE 3.4.1 Modelle im Requirements Engineering

Eigenschaften von Modellen:

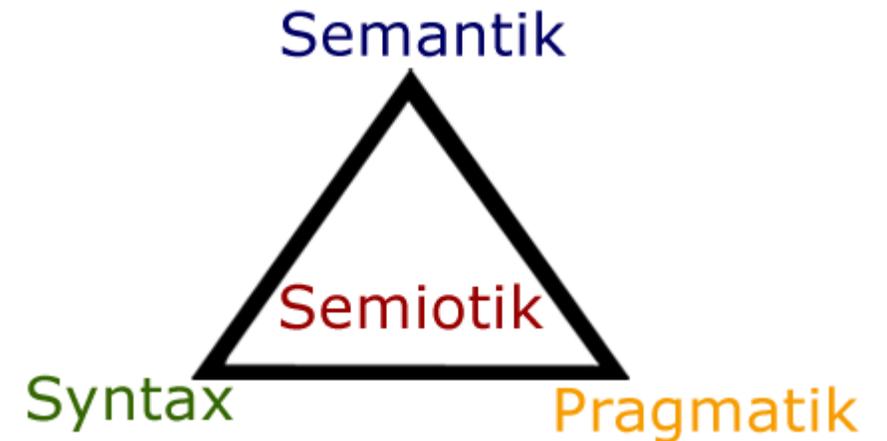
- **Abbild der Realität**
 - Deskriptiv (darstellend)
 - Präskriptiv (normative Beschreibung)
- **Verkürzung der Realität**
 - Selektiv (nur bestimmte Aspekte)
 - Verdichtung
- **Pragmatische Eigenschaft**
 - Verkürzung der Realität auf den Einsatzzweck



LE 3.4.1 Modelle im Requirements Engineering

Eigenschaften von Modellen:

- **Syntax:**
 - = Modellierungssprache
 - definiert Modellelemente
 - zulässige Kombinationen
- **Semantik:**
 - Bedeutung der Modellelemente
 - Basis für die Interpretation von Modellen
- **Pragmatik:**
 - Verwendungszweck



Semiotik: Zeichentheorie

<https://zeichnen-lernen.net/gestalten/semantik-bedeutung-inhalt-273.html>

LE 3.4.1 Modelle im Requirements Engineering

Eigenschaften von Modellen:

- **Zweck:** Ein Modell wird für einen bestimmten Zweck erstellt
 - Daher ist es wichtig, die richtige Modellart auszuwählen (z.B.: statisch vs. dynamisch)
- **Realität:** Ein Modell ist eine Darstellung der Wirklichkeit
 - Was soll dargestellt werden (z.B.: Ist vs. Soll)?
- **Reduktion:** Ein Modell dient dazu die Realität besser zu verstehen
 - Sinnvoll auf das Wesentliche zu reduzieren (Abhängig von der Sicht)
 - Reduktionsmöglichkeiten:
 - Kompression oder Aggregation
 - Selektion

LE 3.4.1 Modelle im Requirements Engineering

Vorteile von Modellen:

- Schnell zu erfassen (Ein Bild sagt mehr als tausend Worte)
- Definierte Modellierungssprache (Syntax & Semantik)
 - **Eindeutigkeit**
- Konzentration auf einen Aspekt (statisch vs. dynamische Sicht)
- Potenzial für die automatisierte Analyse und Verarbeitung von Anforderungen
- Gut in internationalen Teams einsetzbar

LE 3.4.1 Modelle im Requirements Engineering

Nachteile von Modellen:

- Modelle eignen sich primär für funktionale Anforderung, weniger für Qualitätsanforderungen
- Falsche Modellwahl erschwert die Modellierung
- Modellierungssprache muss verstanden werden
- Konsistenz zwischen Modellen ist sicherzustellen
- Mögliche Beschränkungen durch Modellsyntax
- Für effizienten Einsatz entsprechende Tools notwendig

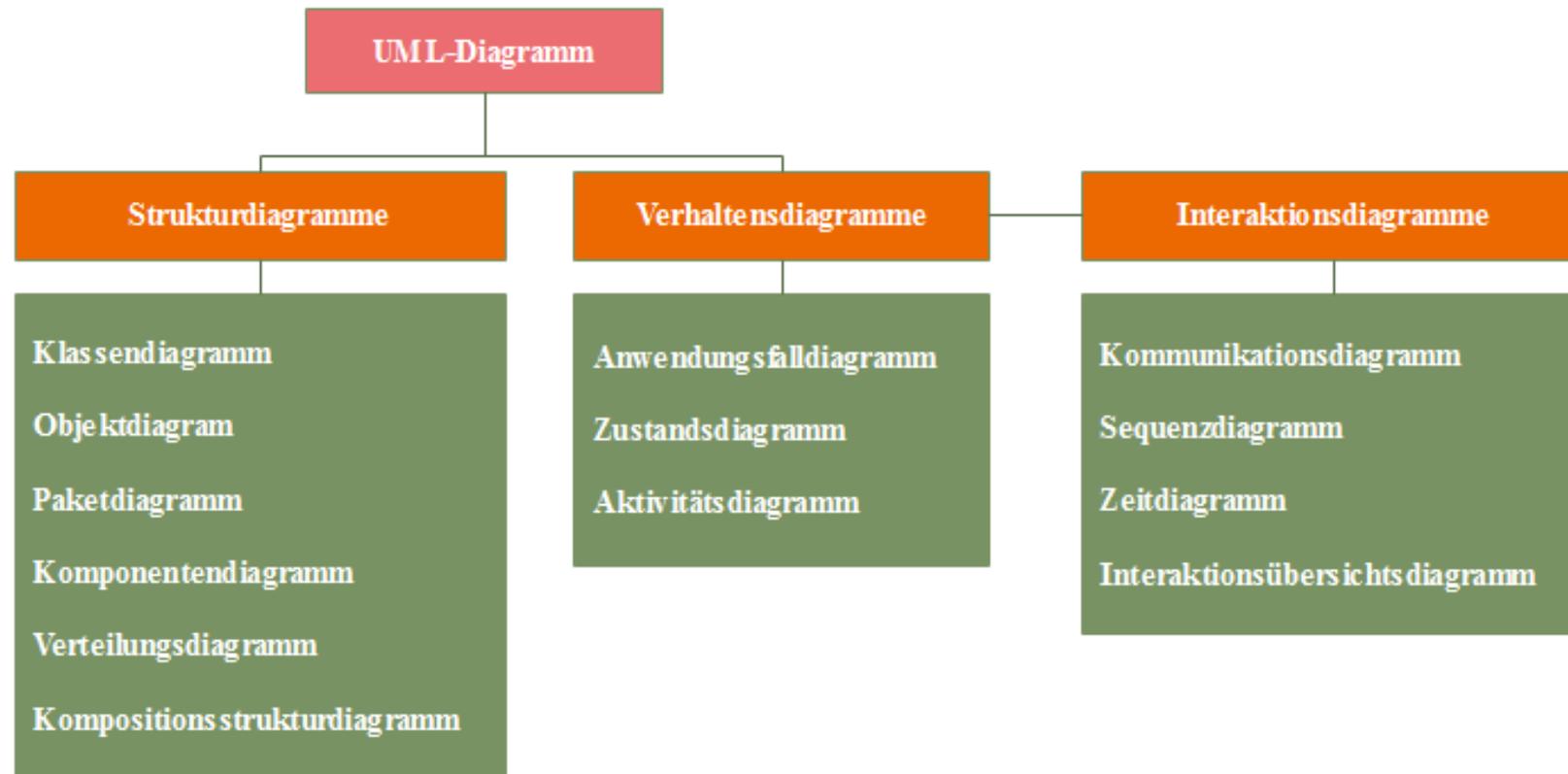
LE 3.4.1 Modelle im Requirements Engineering

Unified Modeling Language (UML):

- grafische Modellierungssprache
- Konstruktion und Dokumentation von Systemen
- Urväter: Grady Booch, Ivar Jacobson und James Rumbaugh (3 Amigos)
- Definition durch OMG (Object Management Group <https://www.omg.org/>)
- ISO-Norm (ISO/IEC 19505: <https://www.iso.org/standard/32624.html>)
- Modellierung aus verschiedenen Perspektiven (statisch und dynamisch)
- UML: Menge von komplementären, überlappenden Modellierungsmodellen
 - Aktivitätendiagramme
 - Sequenzdiagramme
 - Zustandsautomaten
 - Datenflussdiagramme
 - etc... (siehe nächste Seite)

LE 3.4.1 Modelle im Requirements Engineering

Unified Modeling Language (UML):



LE 3.4.2

Modellierung des Systemkontexts

Kontextmodelle – Datenflussdiagramme – UML Use-Case Diagramme

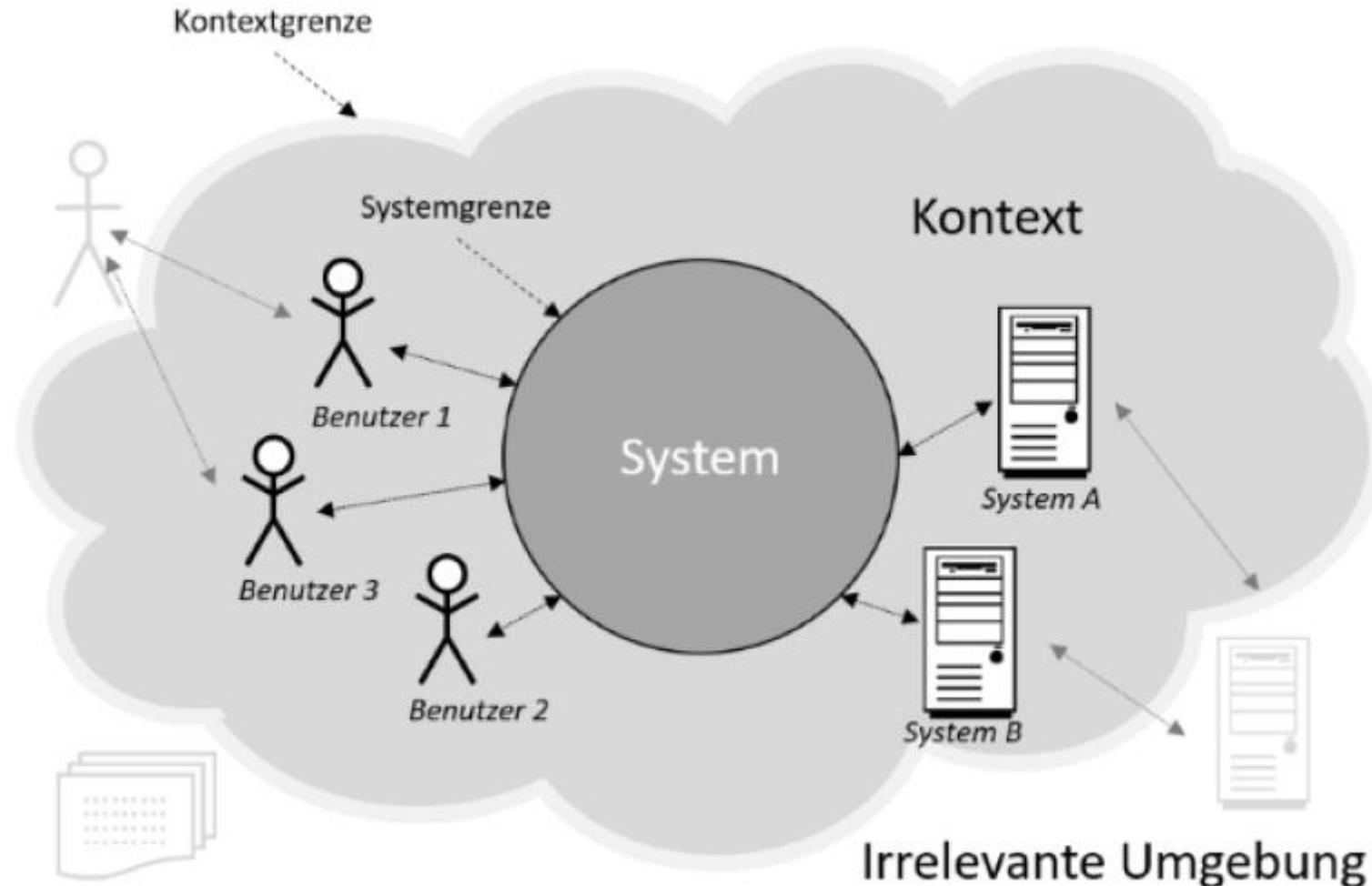
LE 3.4.2 Modellierung des Systemkontexts

Kontextmodelle:

- spezifizieren die *strukturelle Einbettung* des Systems in seine *Umgebung*
- Spezifizieren die *Interaktionen* mit den *Nutzern*
- Spezifizieren die *Interaktionen* mit neuen oder bestehenden *Systemen*
- dienen daher der *Identifikation* der *Anforderungsquellen*
- und deren *Interaktion* mit dem System (Benutzer- bzw. System-*Schnittstellen*)

LE 3.4.2 Modellierung des Systemkontexts

Kontextmodelle schematische Darstellung:



LE 3.4.2 Modellierung des Systemkontexts

Modellsprachen für Kontextmodelle:

- Datenflussdiagramme aus der strukturierten Analyse [DeMa1978]
- UML Use-Case Diagramme [OMG2017]
- Maßgeschneiderte Box-and-Line-Diagramme [Glin2019]

[DeMa1978] Tom DeMarco: Structured Analysis and System Specification. New York: Yourdon Press, 1978

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document, formal/2017-12-05. <https://www.omg.org/spec/UML/About-UML/>.

[Glin2019] Martin Glinz: Requirements Engineering I. Kursnotizen, Universität Zürich, 2019. <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources>.

LE 3.4.2 Modellierung des Systemkontexts

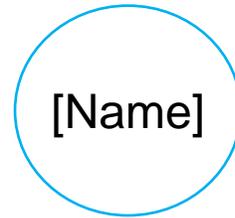
Datenflussdiagramme:

- Modellierung von
 - Funktionen
 - Datenspeicher
 - Daten- / Informationsflüsse
 - Quellen und Senken des Systemkontext
- Modellelemente nach Tom DeMarco [DeMarco 1978]
 - Prozess
 - Datenspeicher
 - Datenfluss
 - Quelle/Senke

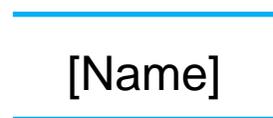
LE 3.4.2 Modellierung des Systemkontexts

Notation Datenflussdiagramm:

- Prozess/Funktion



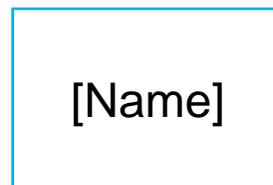
- Datenspeicher



- Datenfluss

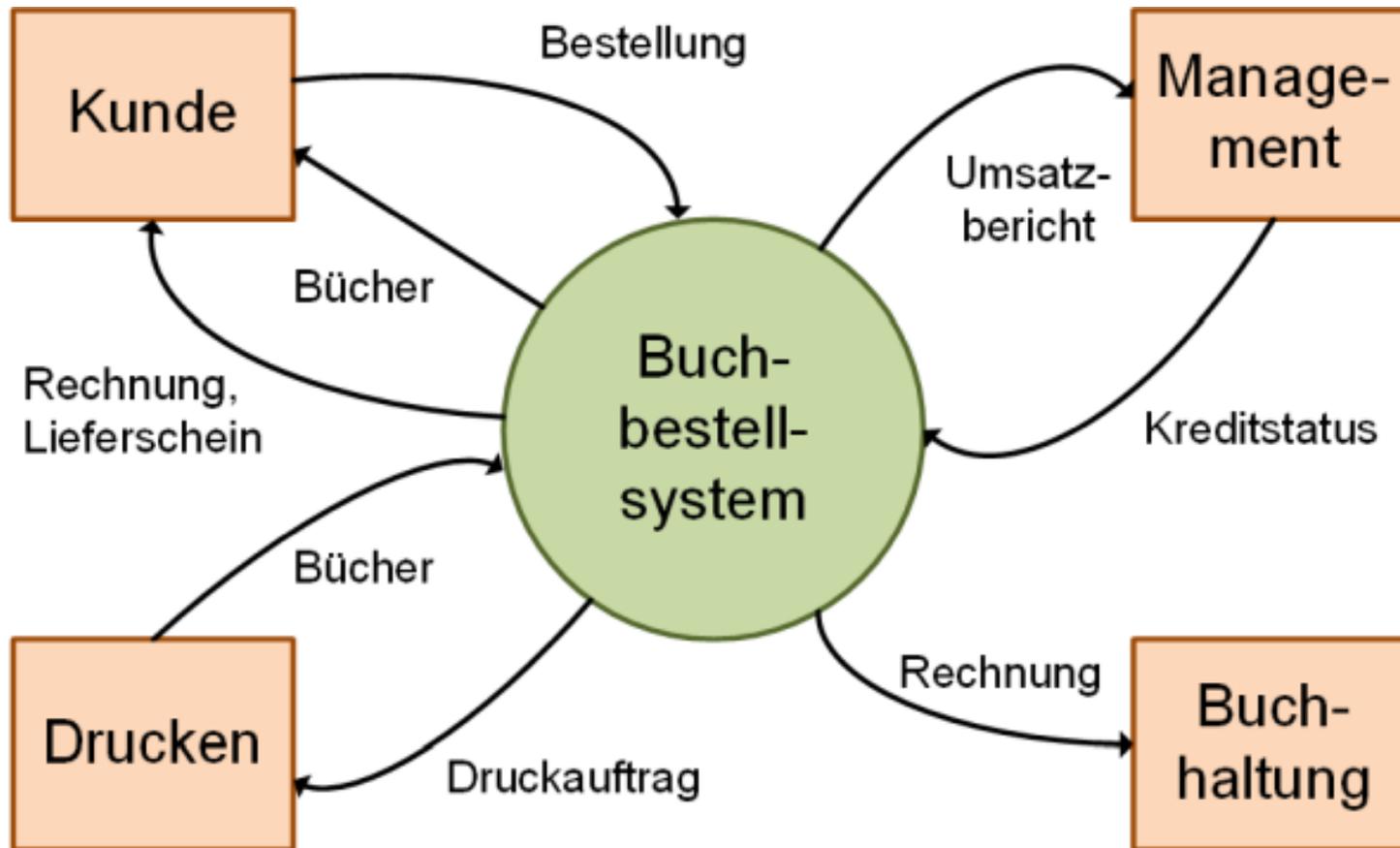


- Quelle/Senke (Terminator)



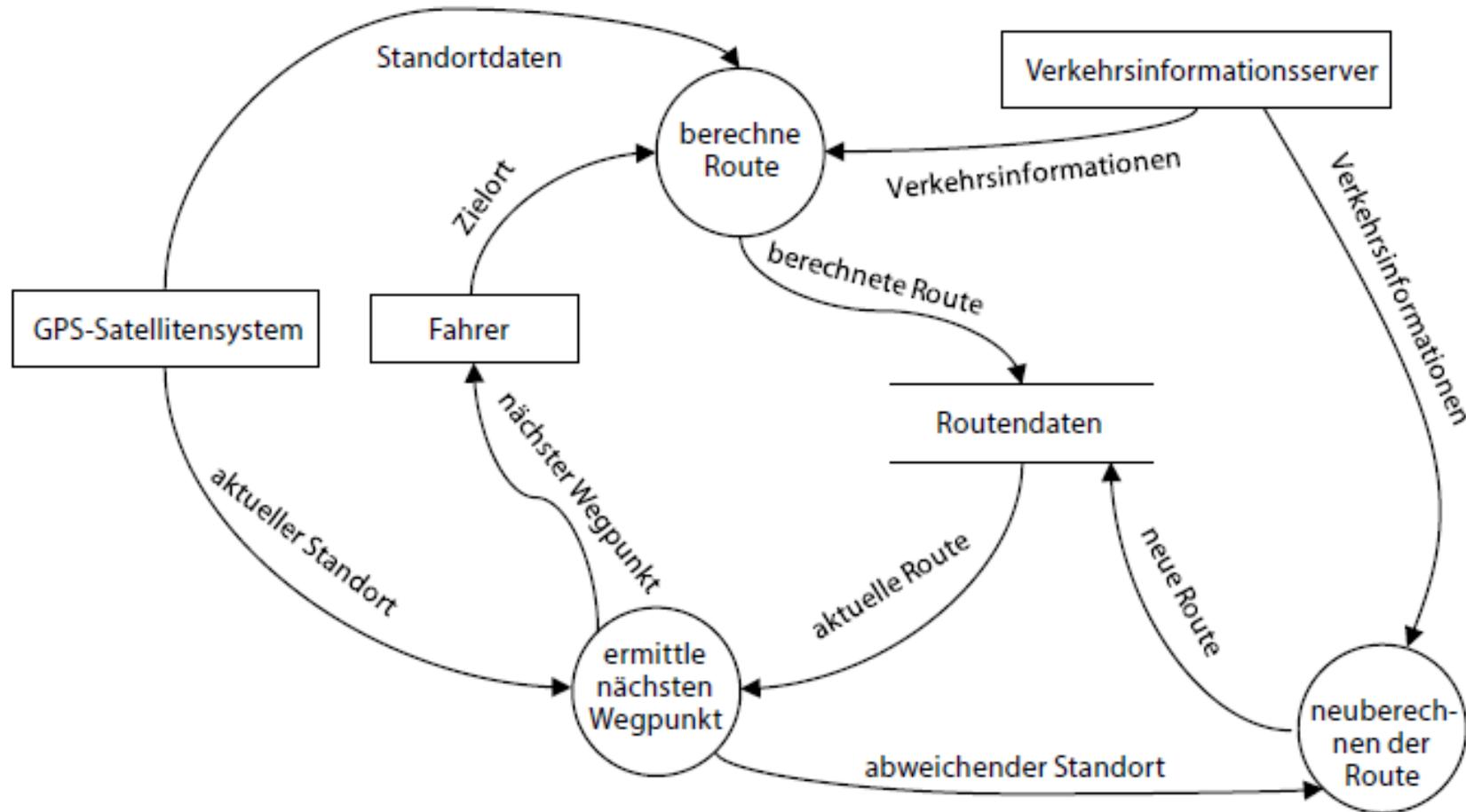
LE 3.4.2 Modellierung des Systemkontexts

Beispiel Datenflussdiagramm als Kontextdiagramm:



LE 3.4.2 Modellierung des Systemkontexts

Beispiel Datenflussdiagramm:



LE 3.4.2 Modellierung des Systemkontexts

UML Use-Case Diagramme:

- Use Case = Anwendungsfall
- Darstellung des Systemkontext anhand von Akteuren
- Darstellung der Systemabgrenzung
- Darstellung der Beziehungen zwischen Anwendungsfällen im System
(Aber **keine** Darstellung von **Abläufen!**)

- Anforderungsbeschreibung:
 - Diagramm (Use Case Modell)
 - Textuell (Use Case Beschreibung)

LE 3.4.2 Modellierung des Systemkontexts

UML Use-Case Beschreibung (Vorlagen basiert):

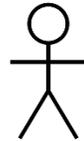
- Name und Identifikationsnummer
- Status (Entwurf, Review, Version, ...)
- Beschreibung
- Beteiligte Akteure
- Verwendete Anwendungsfälle
- Auslöser (Wann, Warum wird Anwendungsfall ausgelöst)
- Vorbedingungen
- Nachbedingung/Ergebnis
- Standardablauf
- Alternative Abläufe/Fehlerfälle
- Bemerkungen/Hinweise
- Änderungsgeschichte

LE 3.4.2 Modellierung des Systemkontexts

Notation UML Use-Case:

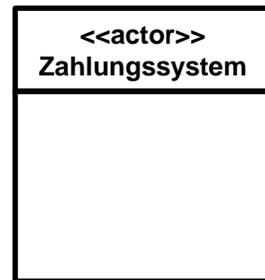
- **Akteur** (unterschiedliche Darstellung)

- als Person



Mitarbeiter

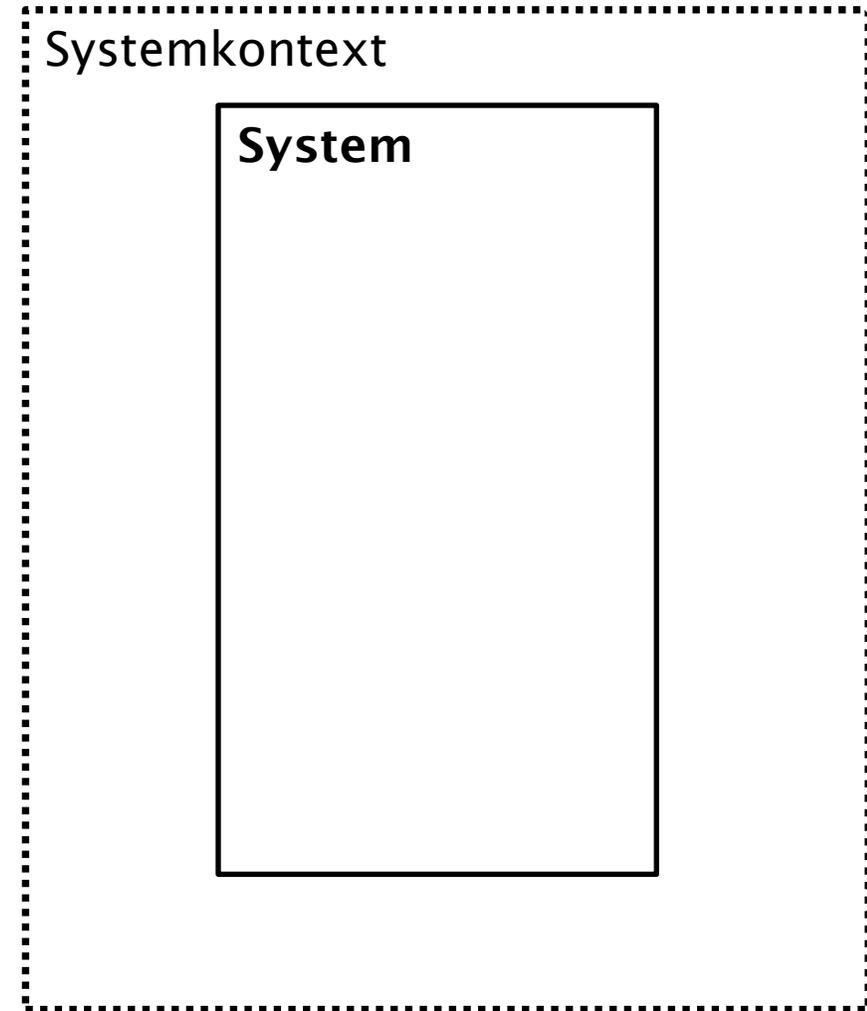
- als Rechteck



- als Icon



Arbeitsplatz
Kundenberater



LE 3.4.2 Modellierung des Systemkontexts

Notation UML Use-Case:

- *Use Case / Anwendungsfall*



- *Kommentar*

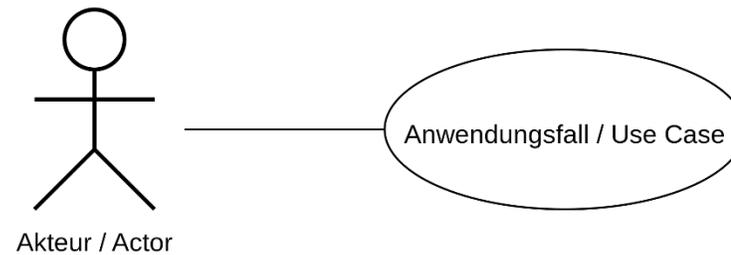


LE 3.4.2 Modellierung des Systemkontexts

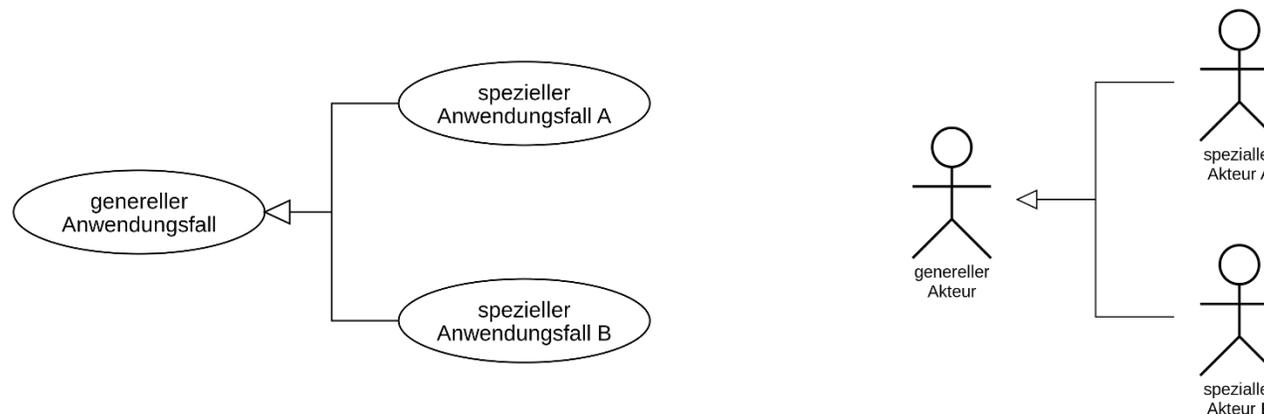
Notation UML Use-Case:

- *Beziehungen*

- *Assoziation/Kommunikation*



- *Generalisierung/Spezialisierung (Anwendungsfall, Akteur)*

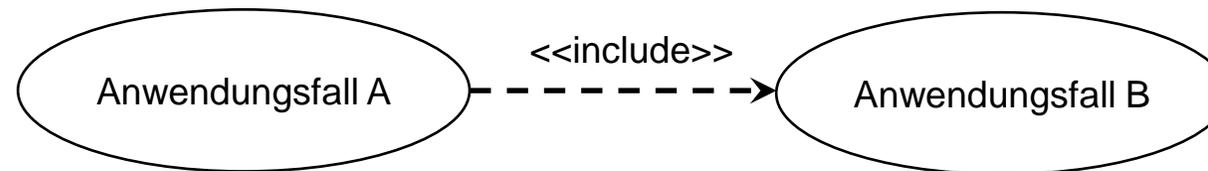


LE 3.4.2 Modellierung des Systemkontexts

Notation UML Use-Case:

- **Beziehungen**

- *Include Beziehung*



Anwendungsfall A beinhaltet (immer) Anwendungsfall B

- *Extend Beziehung*

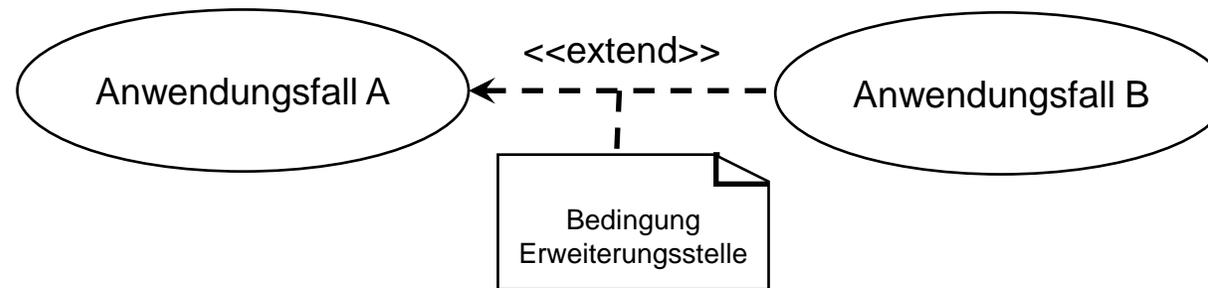


Anwendungsfall A erweitert (gegebenenfalls) um Anwendungsfall B

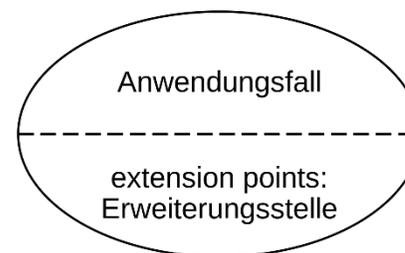
LE 3.4.2 Modellierung des Systemkontexts

Notation UML Use-Case:

- *Varianten Extend Beziehung*



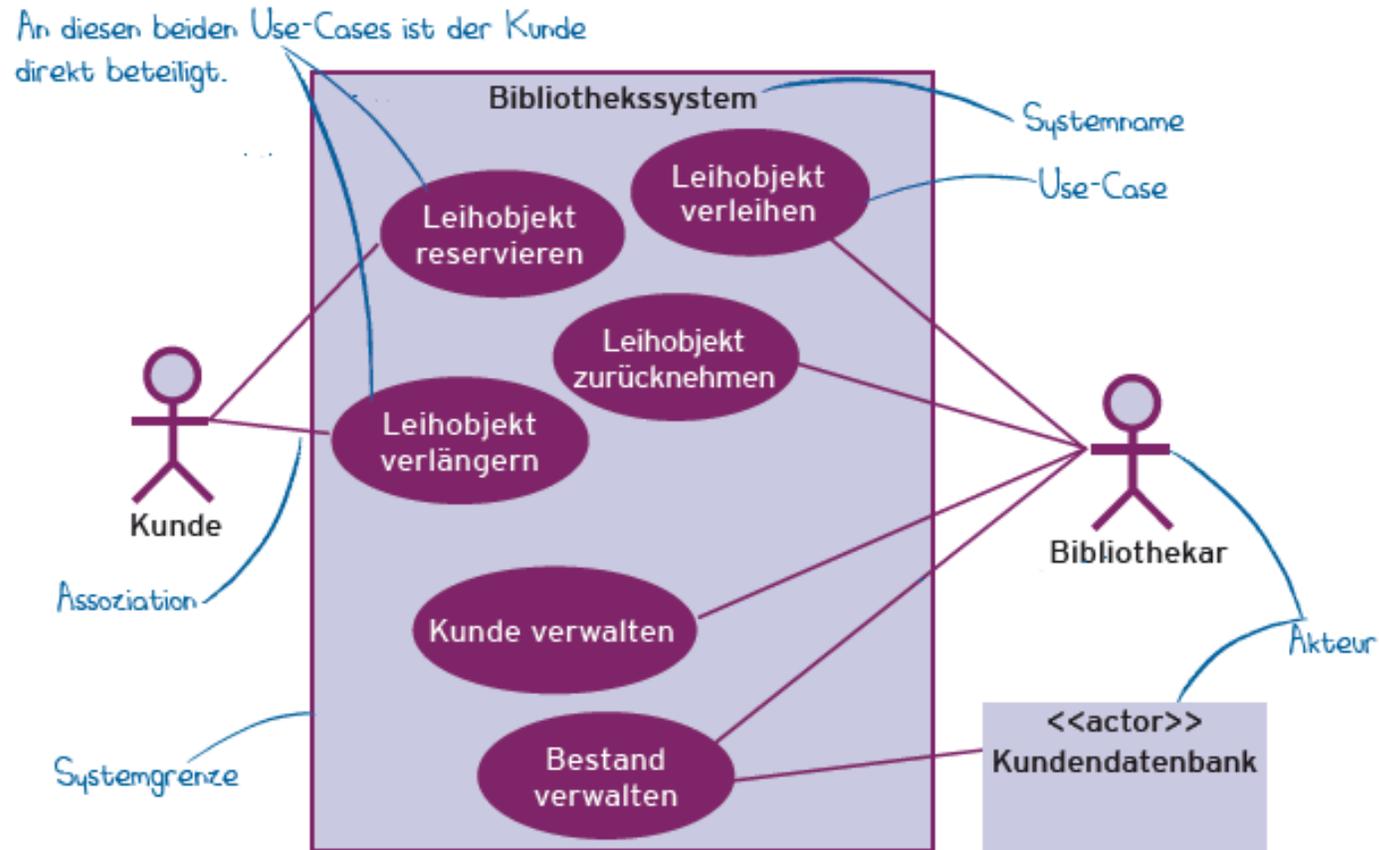
Anwendungsfall A erweitert um Anwendungsfall B unter der angegebenen Bedingung



Anwendungsfall erweitert unter der angegebenen Bedingung

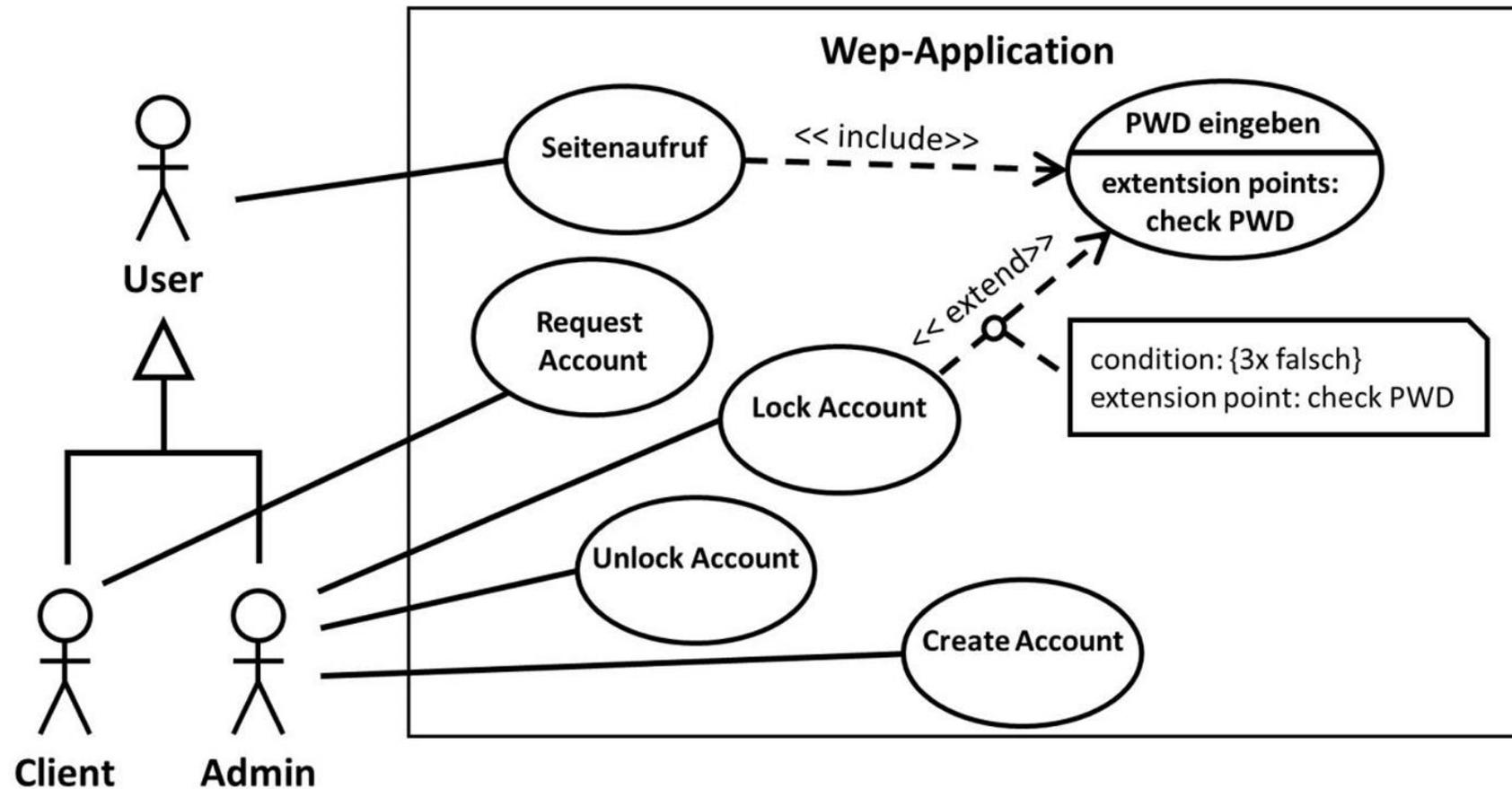
LE 3.4.2 Modellierung des Systemkontexts

Beispiel UML Use-Case Diagramm als Kontextdiagramm:



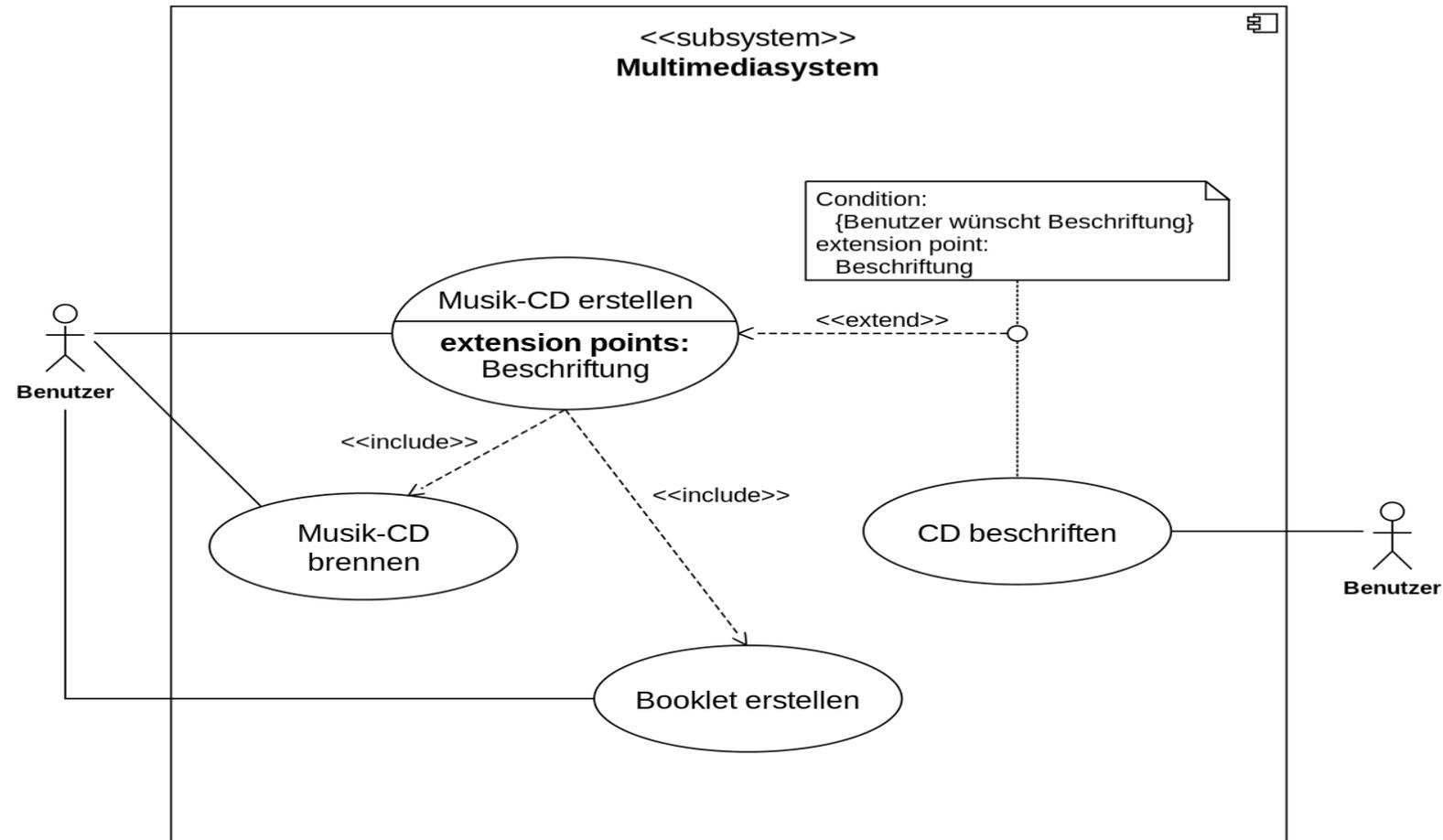
LE 3.4.2 Modellierung des Systemkontexts

Beispiel UML Use-Case Diagramm:



LE 3.4.2 Modellierung des Systemkontexts

Beispiel UML Use-Case Diagramm:



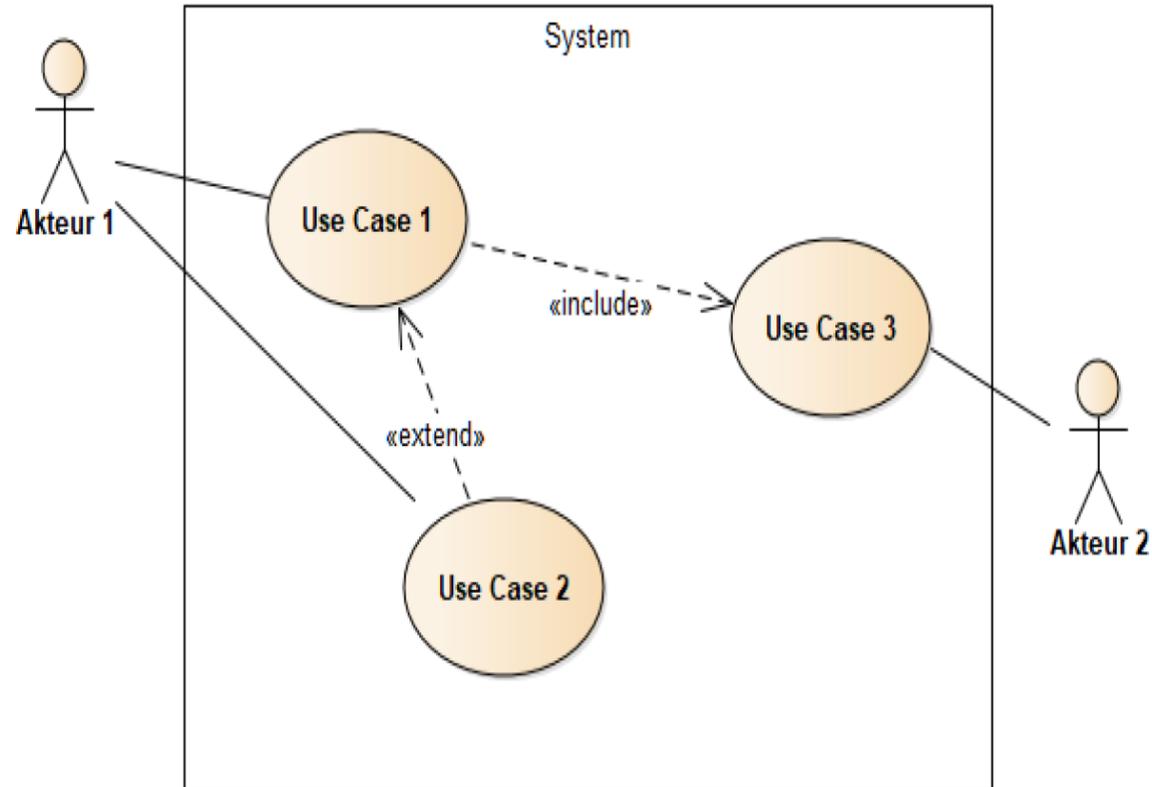
LE 3.4.2 Modellierung des Systemkontexts

Beispiel UML Use-Case Beschreibung:

Thema	Beschreibung
Identifikation	UC_47-11
Status:	Review
Name:	Bücher oder Medien ausleihen
Akteure:	Kunde, Bibliothekar*in, Kundendatenbank
Beschreibung:	Kunde leiht Bücher oder Medien in einer Bibliothek aus Bibliothekar*in verwaltet Kunden und Leihobjekte
Vorbedingung:	Bibliothekar*in ist am System angemeldet
Auslöser:	Bibliothekar*in nach Kundenwunsch
Ablauf:	<ul style="list-style-type: none">• Leihobjekt reservieren, wenn verfügbar• Leihobjekt verlängern, wenn nicht bereits vorbestellt• Leihobjekt in Kundendatenbank vermerken und verleihen• Leihobjekt zurücknehmen und in Kundendatenbank vermerken• Bibliothekar*in verwaltet Kunden und Leihobjekte
Ergebnis:	<ul style="list-style-type: none">• Leihobjekt verliehen• Leihobjekt verlängert• Leihobjekt zurückgenommen• Kunden und Leihobjekte administriert

LE 3.4.2 Modellierung des Systemkontexts

Beispiel Prüfung UML Use-Case:



Richtig	Falsch
<input type="checkbox"/>	A) Use-Case 2 wird nur durch Use-Case 1 ausgelöst.
<input type="checkbox"/>	B) Use-Case 1 kann durch Akteur 1 ausgelöst werden.
<input type="checkbox"/>	C) Use-Case 3 kann ablaufen, ohne von Use-Case 1 ausgelöst worden zu sein.
<input type="checkbox"/>	D) Während des Ablaufs von Use-Case 1 wird Use-Case 3 aufgerufen.
<input type="checkbox"/>	E) Use-Case 1 wird immer ausgelöst, wenn Use-Case 2 abläuft.

LE 3.4.2 Modellierung des Systemkontexts

Übung UML Use-Case Diagramm:

- Ziel: *Use-Case für Bankomat*
- Funktionen:
 - Geld abheben
 - Kontostand ausgeben
 - Kontoauszüge ausdrucken
 - Authentisierung für alle oben genannten Funktionen erforderlich!
- **Aufgabe** (Gruppenarbeit max. 3 Teilnehmer)
 - Erstellen eines Use Case Diagramms (Aktoren, Use-Cases)
 - <https://app.diagrams.net/>
 - Präsentation der Ergebnisse

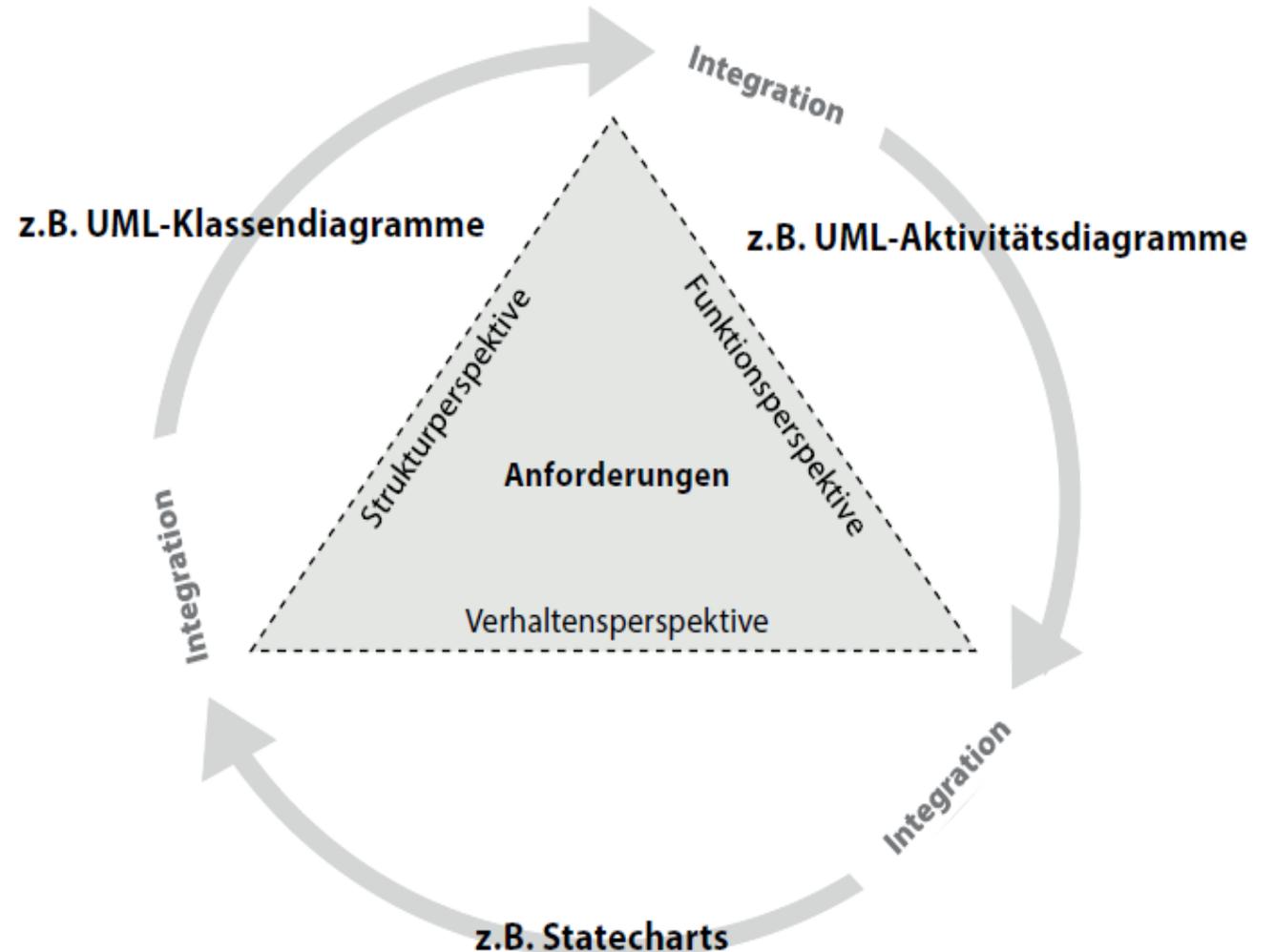
Drei Perspektiven auf Anforderungen

Strukturperspektive – Funktionsperspektive – Verhaltensperspektive

Drei Perspektiven auf Anforderungen

Modellierung Perspektiven:

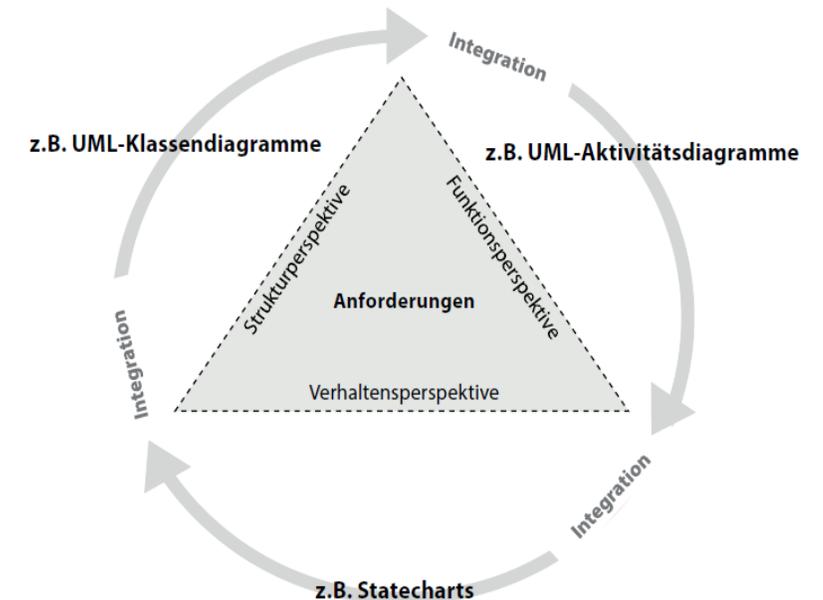
- Strukturperspektive
 - Funktionsperspektive
 - Verhaltensperspektive
-
- Abbildung der Anforderungen aus verschiedenen Sichten heraus
 - Abbildungen überlappen sich teilweise
 - Überlappung kann/soll zur Konsistenzprüfung verwendet werden



Drei Perspektiven auf Anforderungen

Modellierung aus der Strukturperspektive:

- Abbildung der *Struktur* von *Daten*
- Abbildung von *Abhängigkeitsbeziehungen* zwischen Daten
- Beschreibungsmodelle:
 - *Entity-Relationship Diagramme (ER-Diagramme)*
 - *UML-Klassendiagramme*
 - SysML-Blockdefinitionsdiagramme

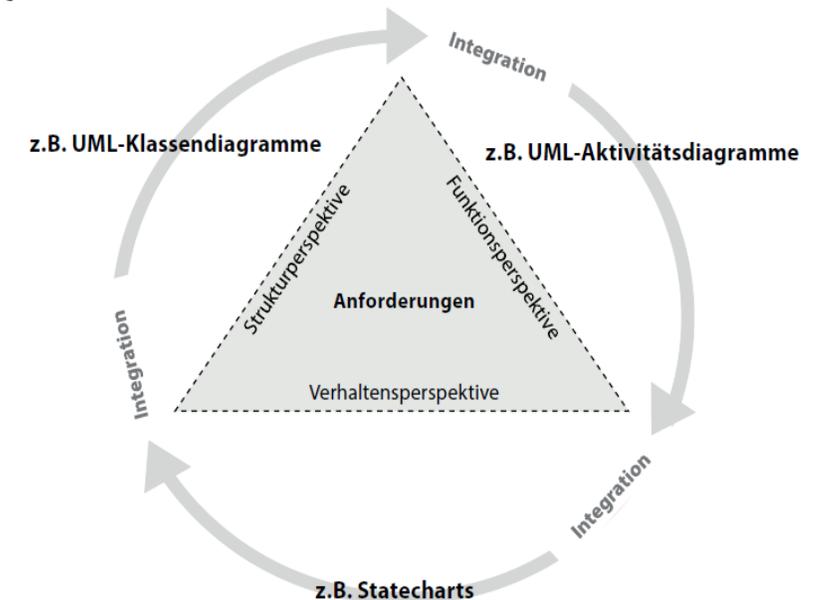


Basiswissen Requirementsengineering; Pohl, Rupp

Drei Perspektiven auf Anforderungen

Modellierung aus der Funktionsperspektive:

- Fokus auf *Verarbeitung* von Ein- zu Ausgabedaten
- Beschreibung der Verarbeitung
- Beschreibung des Kontrollflusses
- *Funktionsmodelle* werden als *Flussdiagramme* dargestellt
- Beschreibungsmodelle:
 - *UML-Aktivitätsdiagramme*
 - UML-Sequenzdiagramme
 - UML Use-Case Diagramme (siehe LE 3.4.2)
 - Datenflussdiagramme (siehe LE 3.4.2)
 - Business Process Modeling Notation (BPMN)
 - Domänen-Story-Modelle

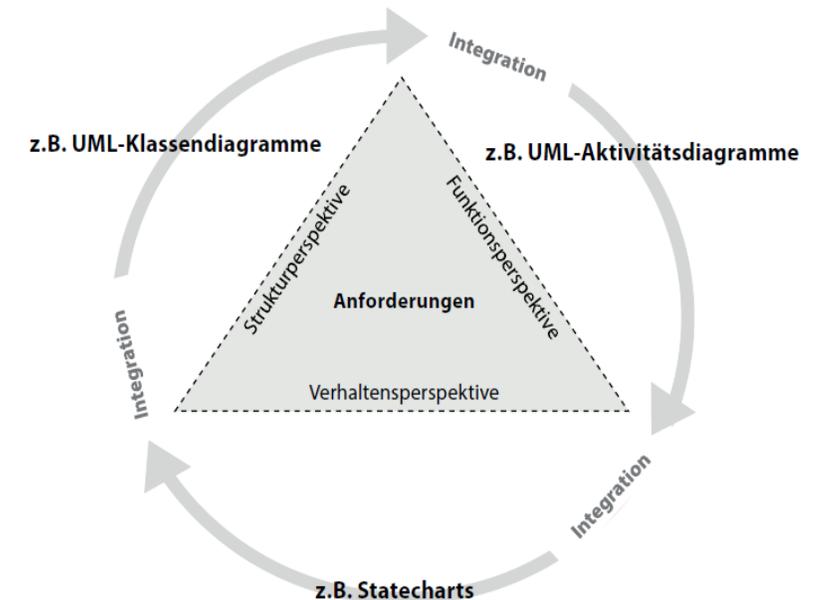


Basiswissen Requirementsengineering; Pohl, Rupp

Drei Perspektiven auf Anforderungen

Modellierung aus der Verhaltensperspektive:

- Modellierung des *dynamischen Verhaltens* des Systems
- Fokus auf *Systemzustände* und
- *Ereignisse* die zu Zustandswechsel führen
- Beschreibungsmodelle:
 - *Statecharts (Zustandsdiagramme)*
 - *UML-Zustandsdiagramme*
 - UML-Sequenzdiagramm



Basiswissen Requirementsengineering; Pohl, Rupp

LE 3.4.3

Modellierung von Struktur und Daten (Strukturperspektive)

ER-Diagramme – UML-Klassendiagramme

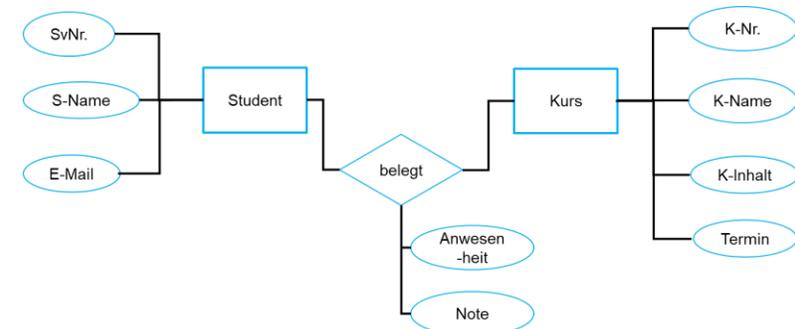
LE 3.4.3 Modellierung von Struktur und Daten

Modellierung aus der Strukturperspektive:

- Abbildung der *Struktur* von *Daten*
- Abbildung von *Abhängigkeitsbeziehungen* zwischen Daten

Modellsprachen für die Strukturperspektive:

- Entity-Relationship-Diagramme (ERD) [Chen1976]
- UML-Klassendiagramme [OMG2017]
- SysML-Blockdefinitionsdiagramme [OMG2018]



[Chen1976] Peter P.-S. Chen: The Entity-Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems 1976, 1(1), 9–36..

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document, formal/2017-12-05. <https://www.omg.org/spec/UML/About-UML/>.
Object Management Group: OMG Systems Modeling Language (OMG SysML™), version 1.6. OMG document, ptc/2018-12-08. <https://www.omg.org/spec/SysML/About-SysML/>

LE 3.4.3 Modellierung von Struktur und Daten

Modellierung Entity-Relationship-Diagramme:

Notation (basierend auf [CHEN 1976])

- Entitätstyp (Objektklasse)



- Attribut



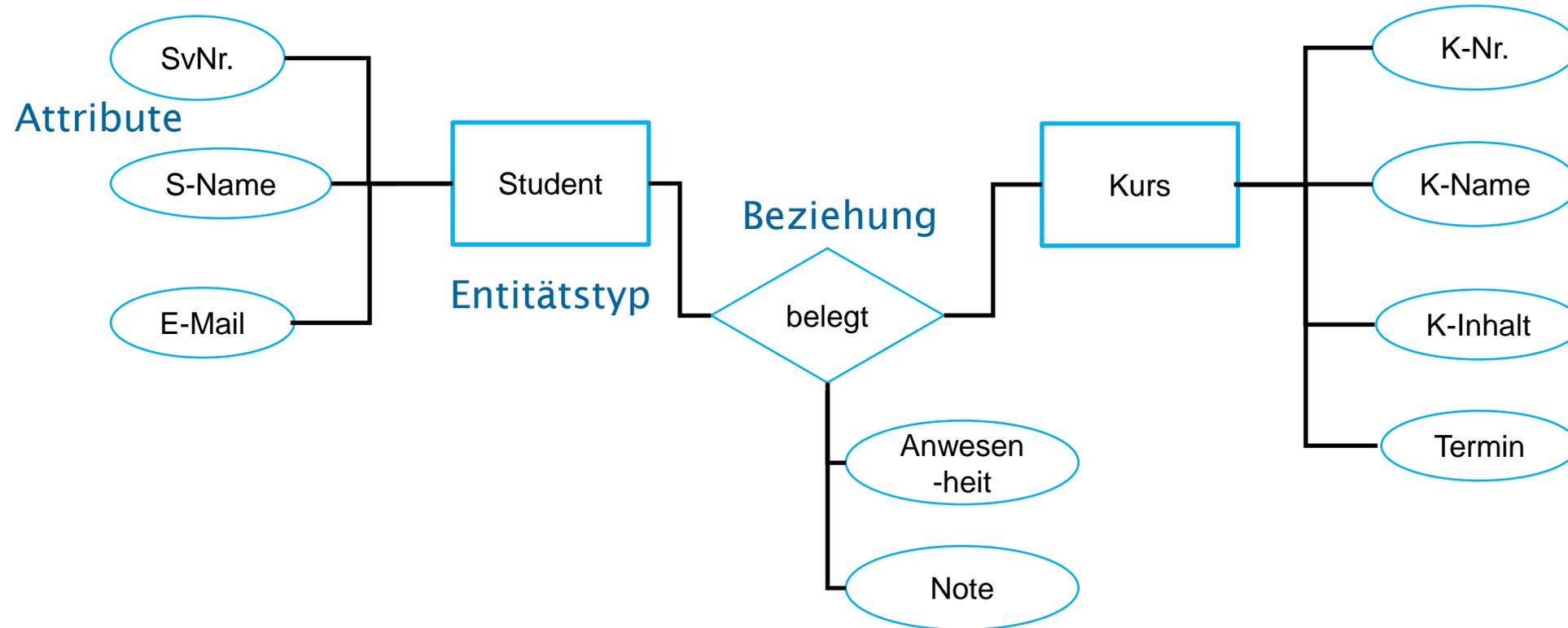
- Beziehung



[Chen 1976] P. Chen: The Entity-Relationship Specification – Toward a Unified View of Date. ACM Transactions on Database Systems, Vol. 1, Nr. 1, 1976, S. 9–38.

LE 3.4.3 Modellierung von Struktur und Daten

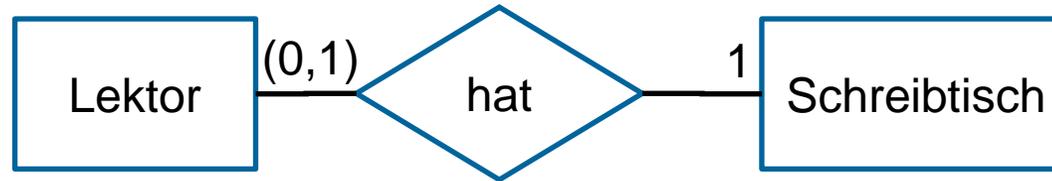
Beispiel Entity-Relationship Diagramm:



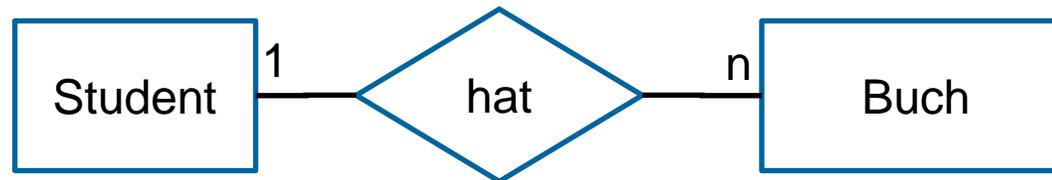
LE 3.4.3 Modellierung von Struktur und Daten

ER-Diagramm Beziehungstypen (Kardinalitäten):

- 1 : 1 Beziehung



- 1 : n Beziehung



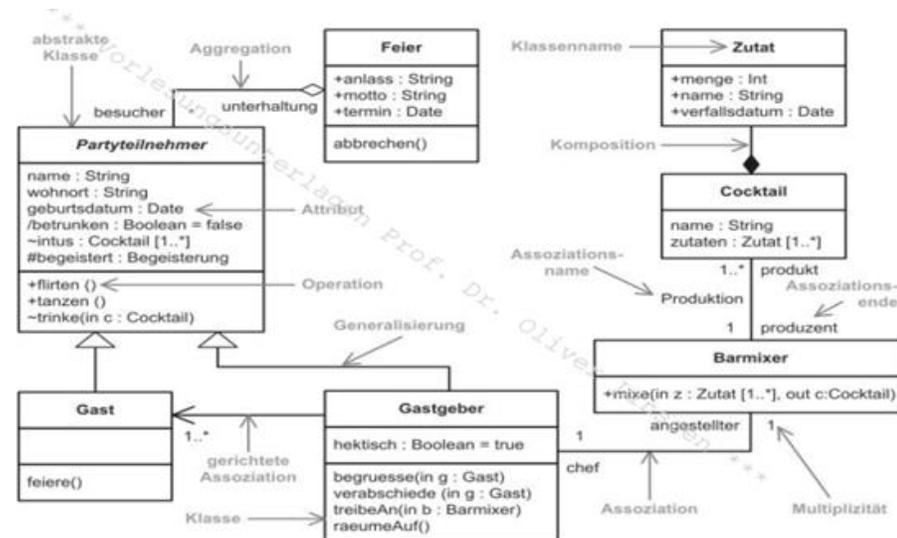
- m : n Beziehung



LE 3.4.3 Modellierung von Struktur und Daten

Modellierung UML-Klassendiagramme:

- Darstellung von Klassen, Schnittstellen und deren Beziehungen
- Ähnlich ER-Diagrammen mit Klassen und Assoziationen
- Zusätzliche Modellierungselemente (zur Spezifikation von Operationen auf Klassen)



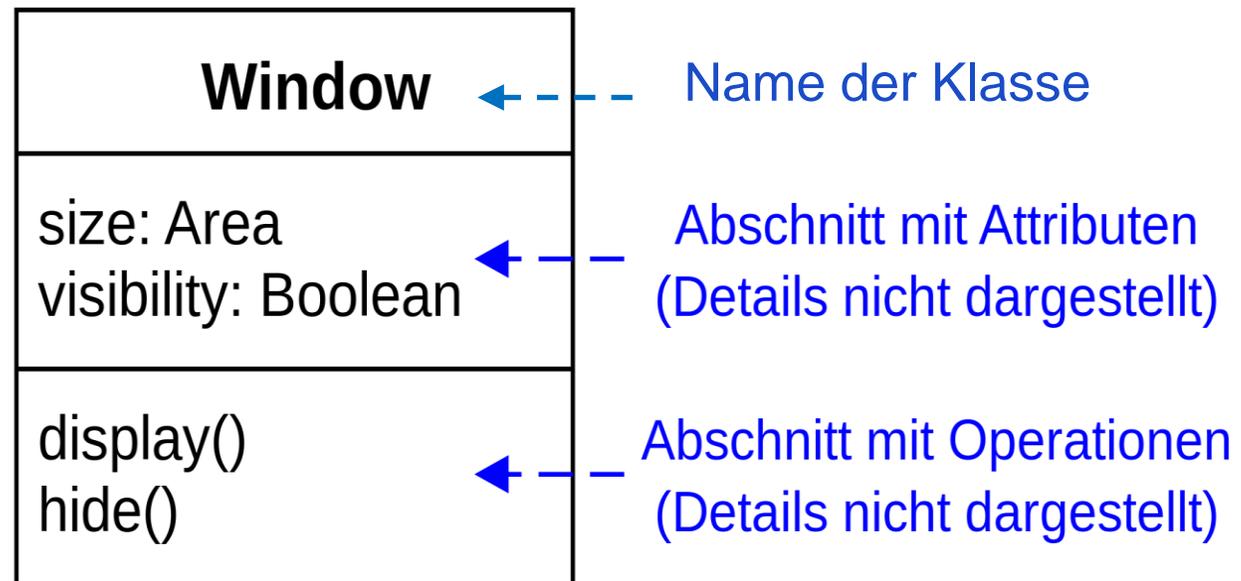
<https://www.repetico.de/card-76929800>

LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Klasse**

- Als Rechteck dargestellt mit *Klassenname*
- Attribute (optional)
- Operationen (optional)

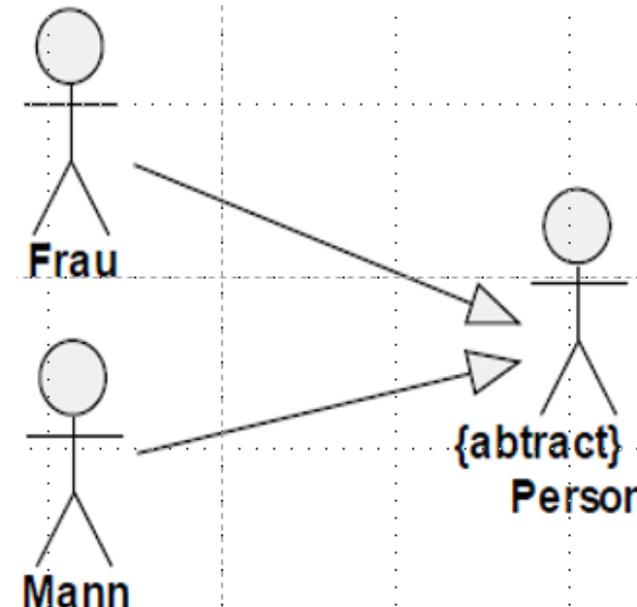
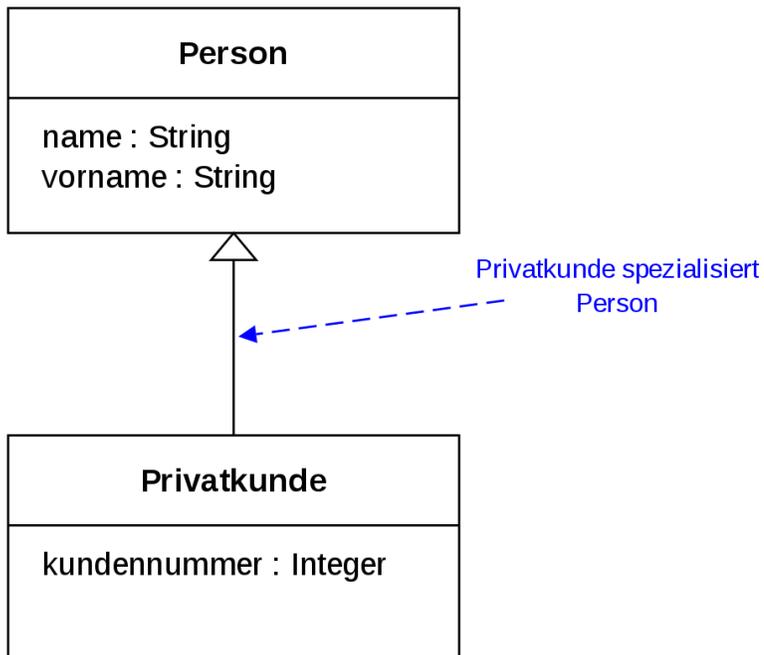


LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Generalisierung/Spezialisierung**

- gerichtete Beziehung
- zwischen genereller und spezieller Klasse/Objekt
- Spezialisierung erbt Eigenschaften der Generalisierung

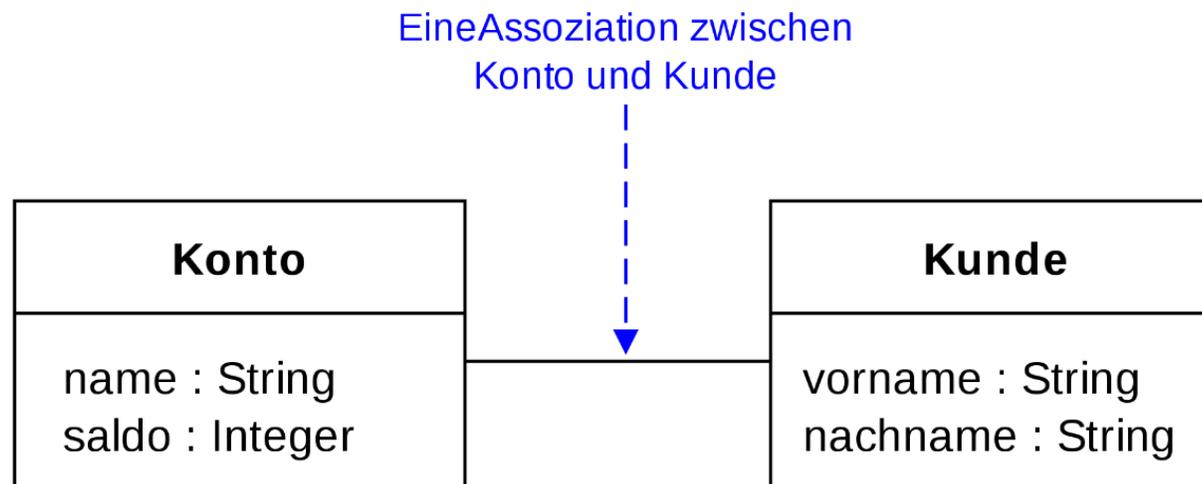


LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Assoziation**

- Definiert *Beziehung* zwischen 2 oder mehreren Klassen
- Optional: *Name*, Darstellung der *Rolle* und *Multiplizität* (Kardinalität im ER-Diagramm)



LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Multiplizitäten**

- beschreibt Beziehung zwischen 2 oder mehreren Klassen
- *Multiplizitäten* beschränken die Beziehung auf eine definierte Anzahl



Multiplizitäten definieren, wie viele Instanzen der Klasse am entsprechenden Assoziationsende an der Assoziation zu einer bestimmten Instanz der Klasse am anderen Ende der Assoziation teilnehmen können. Wobei $n \in \mathbb{N}$ und $m \in \mathbb{N}$. Einige Beispiele:

0 .. 1 (null oder einmal)

0 .. * (null oder mehrere Male)

1 .. * (ein oder mehrere Male)

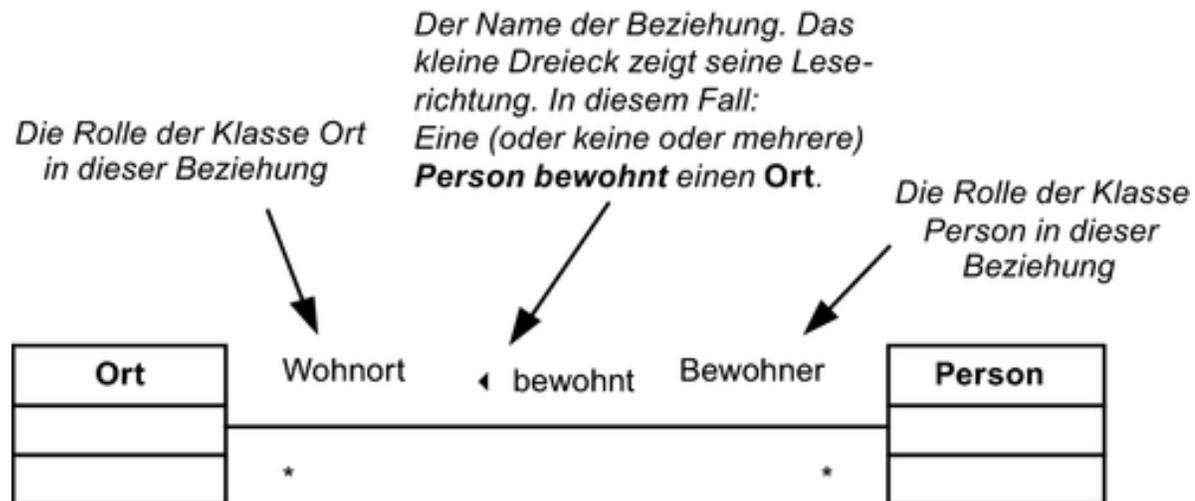
7 (genau sieben Mal)

1 (Standardwert, genau einmal)

LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Rolle**
 - beschreibt Beziehung zwischen 2 oder mehreren Klassen
 - definiert welche Rolle in der Klasse die Beziehung spielt



LE 3.4.3 Modellierung von Struktur und Daten

Notation UML-Klassendiagramme:

- **Aggregation und Komposition**

- Spezielle Formen der *Assoziation*
- Beziehung der Teile zu ihrem Ganzen

- **Aggregation**

- Teil des Ganzen
- Kann ohne Ganzes existieren (Multiplizität am Ganzen: **0..***)



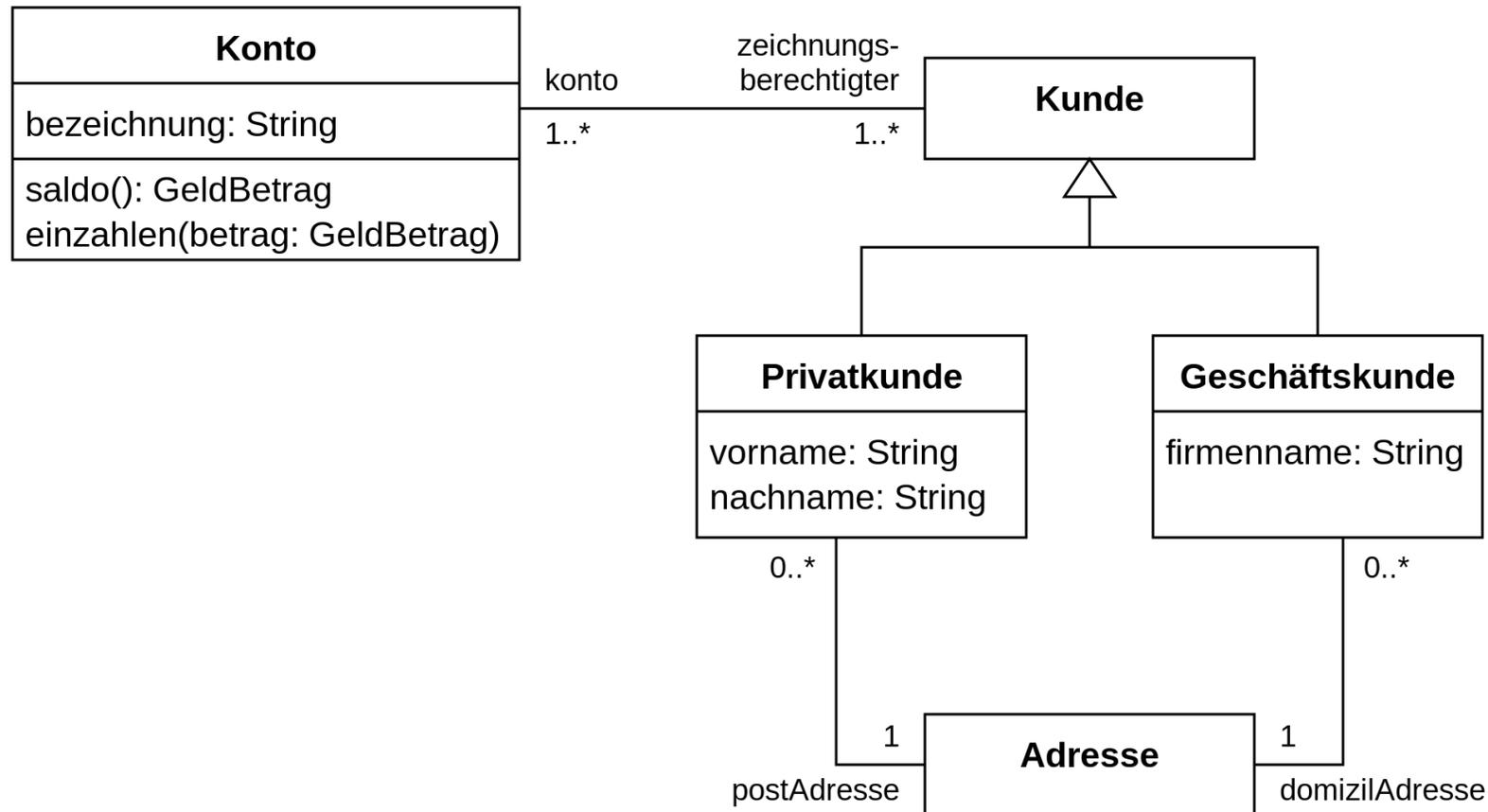
- **Komposition**

- Teil des Ganzen
- Kann ohne Ganzes **nicht** existieren (Multiplizität am Ganzen: **1**)



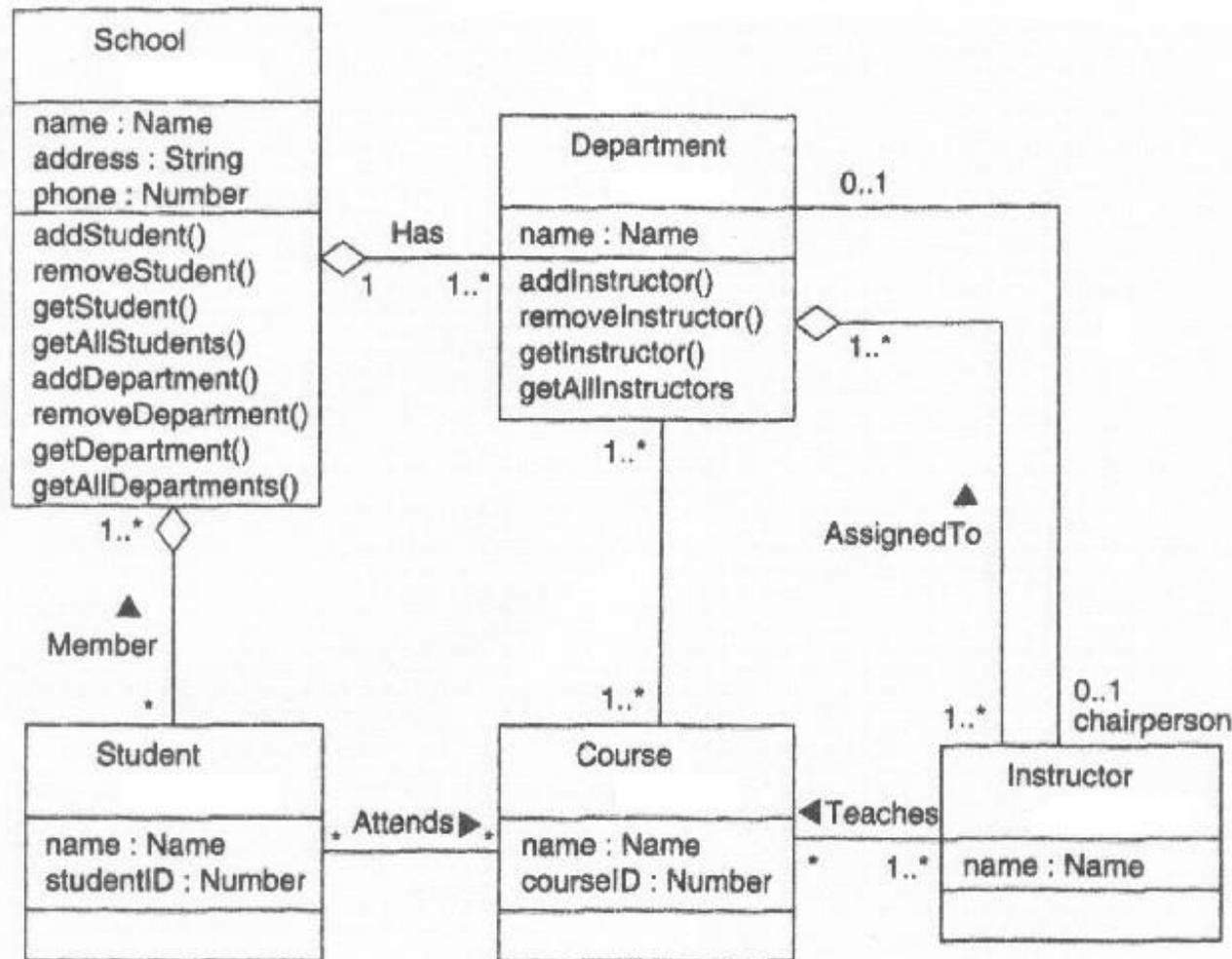
LE 3.4.3 Modellierung von Struktur und Daten

Beispiel UML-Klassendiagramm:



LE 3.4.3 Modellierung von Struktur und Daten

Beispiel UML-Klassendiagramm:



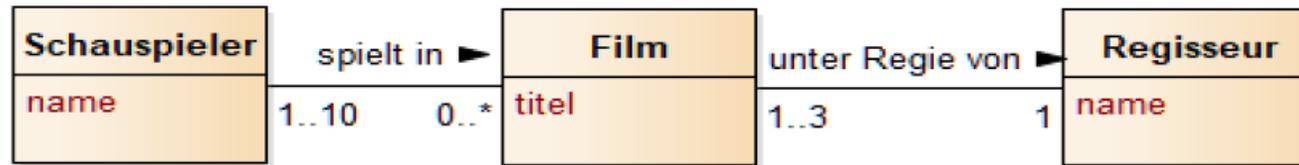
Grady Booch, James Rumbaugh, Ivar Jacobson:
The Unified Modeling Language User Guide,
amc press & Addison Wesley, 2001. Seite: 112

LE 3.4.3 Modellierung von Struktur und Daten

Beispiel Prüfung UML-Klassendiagramm:

6. Anforderungen modellbasiert dokumentieren

25. Zur Förderung junger Schauspieler und Regisseure wird ein Wettbewerb für Kurzfilme durchgeführt. Dabei werden die besten drei Filme ausgezeichnet. Die eingereichten Filme dürfen maximal 20 Minuten lang sein und müssen die im folgenden Diagramm beschriebenen Einschränkungen beachten. K0619
2 Punkte



Stimmen die folgenden Aussagen mit dem obigen Diagramm überein?

Stimmt überein	Stimmt nicht überein	
		A) Drei Regisseure können bei einem Film gemeinsam Regie führen.
		B) Ein Film kann einen einzigen Schauspieler haben.
		C) Ein Regisseur kann bei zwei eingereichten Filmen Regie führen.
		D) Ein Schauspieler kann in beliebig vielen Filmen auftreten.
		E) Ein Film muss zehn Schauspieler haben.

LE 3.4.3 Modellierung von Struktur und Daten

Übung UML-Klassendiagramm:

- Ziel: *Bibliothekssystem*
 - Verleiht Leihobjekte aus Bibliothek an Kunden/Mitarbeiter*innen

- Daten:

Personen	Leihobjekte	Aufbewahrungsort Leihobjekt
Kunden	Bücher	Raum
Mitarbeiter*innen	Audio CDs	Regal
	Video DVDs	

- **Aufgabe** (Gruppenarbeit max. 3 Teilnehmer)

- Erstellen eines **UML-Klassendiagramms**
 - Klassen + Attribute
 - Beziehungen (Generalisierung, Aggregation, Komposition)
 - Darstellung der Rolle und Multiplizitäten
- <https://app.diagrams.net/>
- Präsentation der Ergebnisse

LE 3.4.4

Modellierung von Funktion und Ablauf (Funktionsperspektive)

UML-Aktivitätsdiagramme

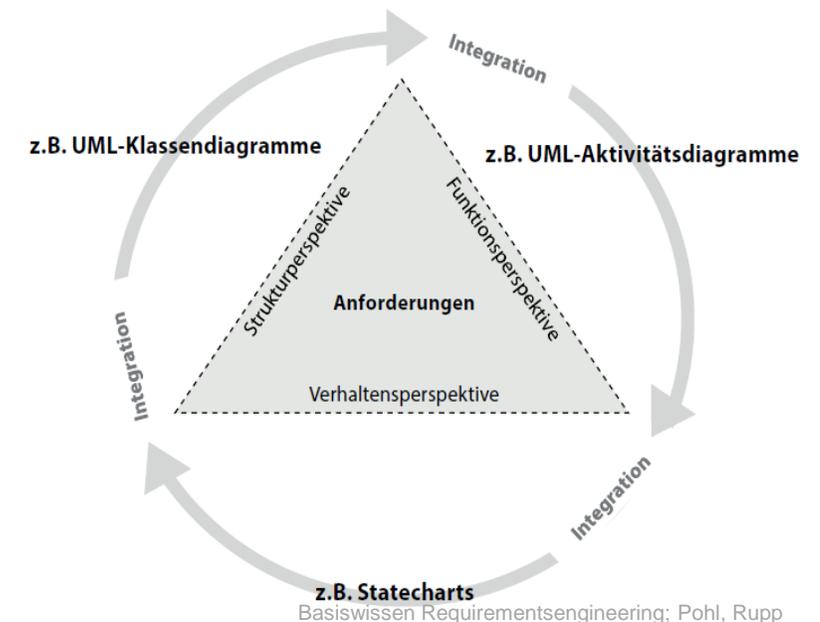
LE 3.4.4 Modellierung von Funktion und Ablauf

Modellierung aus der Funktionsperspektive:

- Fokus auf *Verarbeitung* von Ein- zu Ausgabedaten
- Beschreibung der Verarbeitung
- Beschreibung des Kontrollflusses
- *Funktionsmodelle* werden als *Flussdiagramme* dargestellt

Modellsprachen für die Funktionsperspektive:

- *UML-Aktivitätsdiagramme*
- UML-Sequenzdiagramme
- UML Use-Case Diagramme (siehe LE 3.4.2)
- Datenflussdiagramme (siehe LE 3.4.2)
- Business Process Modeling Notation (**BPMN**)
- Domänen-Story-Modelle (Domänenmodell)



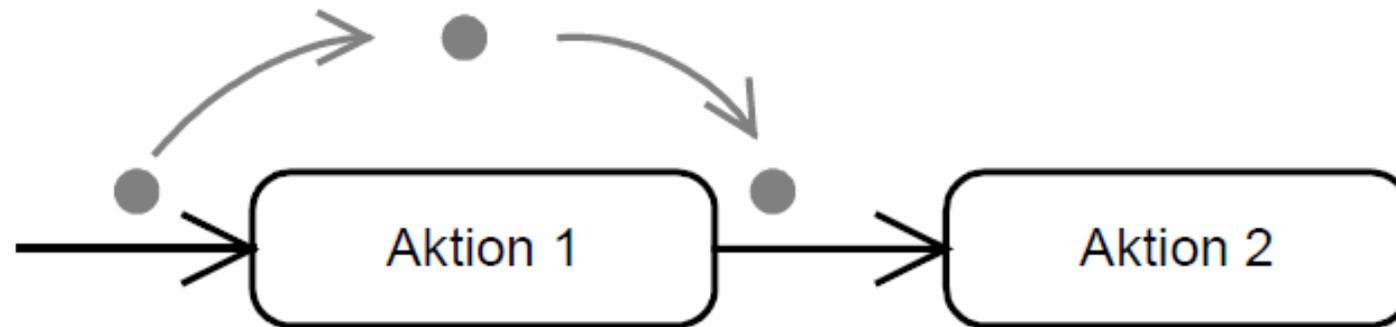
Basiswissen Requirementsengineering; Pohl, Rupp

LE 3.4.4 Modellierung von Funktion und Ablauf

Notation UML-Aktivitätsdiagramm:

- **Tokenkonzept**

- Der Ablauf in einer Aktivität wird durch den Tokenfluss gesteuert
- Ist ein Gedankenkonstrukt (keine explizite Modellierung)
- Ermöglicht die explizite Beschreibung des Verhaltens

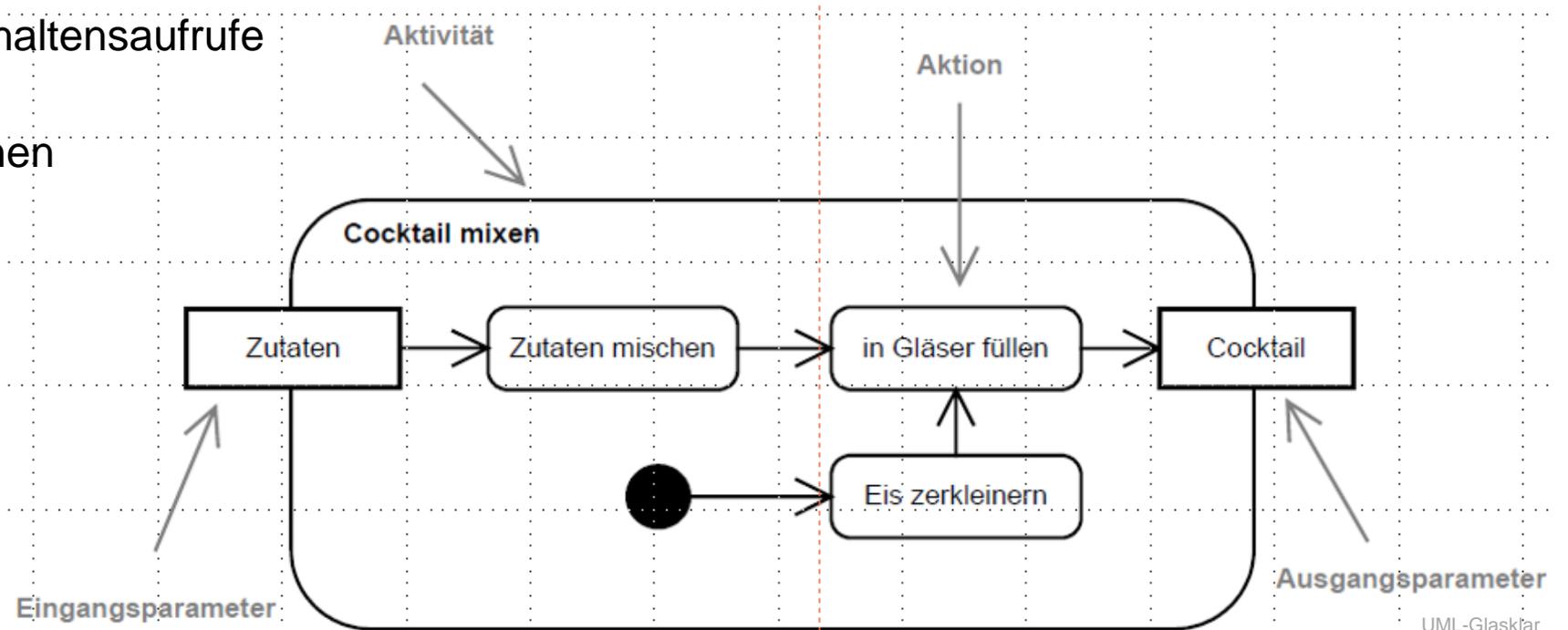


LE 3.4.4 Modellierung von Funktion und Ablauf

Notation UML-Aktivitätsdiagramm:

- **Aktivität**

- ist ein gerichteter Graph
- beschreibt das komplette Diagramm
- kann Ein- und Ausgangsparameter haben
- **Aktionen** sind Verhaltensauffufe (Funktionen)
- Summe der Aktionen realisiert Aktivität



LE 3.4.4 Modellierung von Funktion und Ablauf

Notation UML-Aktivitätsdiagramm:

- Aktionsknoten



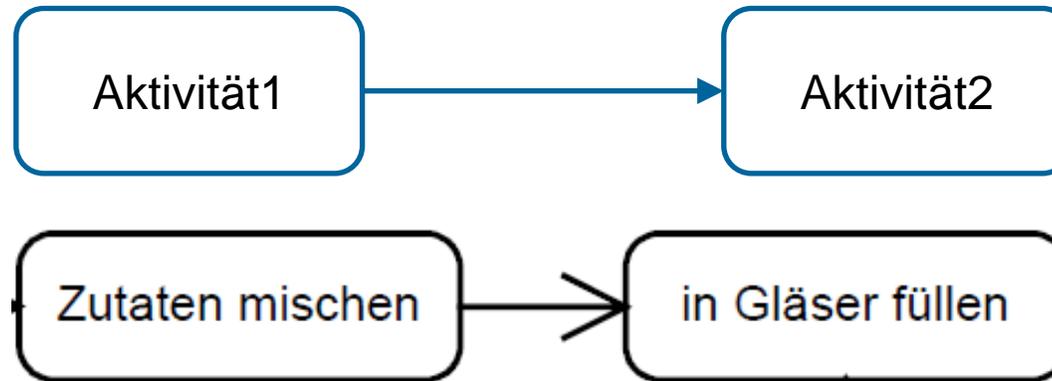
- Objektknoten



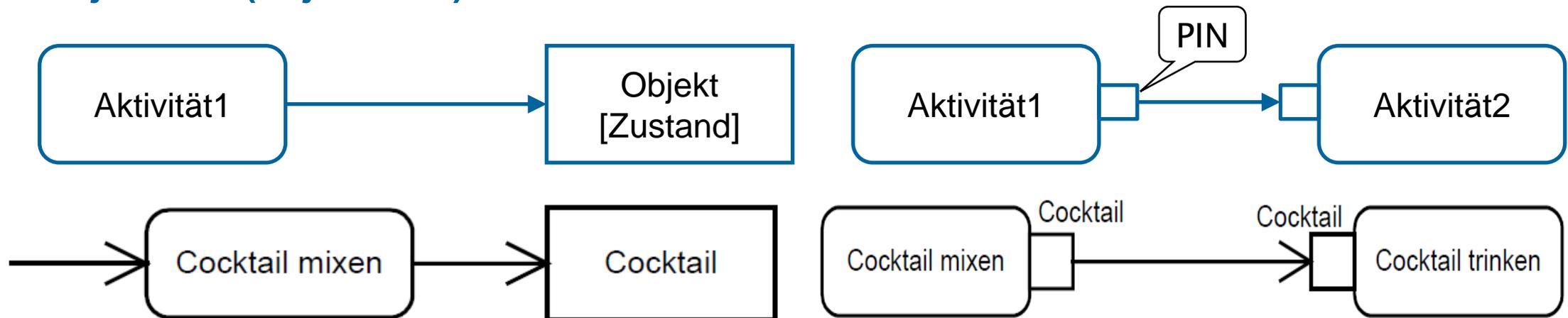
LE 3.4.4 Modellierung von Funktion und Ablauf

Notation UML-Aktivitätsdiagramm:

- **Kontrollfluss (control flow)**



- **Objektfluss(object flow)**



LE 3.4.4 Modellierung von Funktion und Ablauf

Notation UML-Aktivitätsdiagramm:

- **Kontrollelemente:**

- Startknoten (InitialNode)
Können mehrere sein (Parallelität)
- Endknoten (activity FinalNode)
Beendet sofort die gesamte Aktivität
- Kontrollfluss Endknoten (flow FinalNode)
Beendet nur die jeweilige Aktivität
- Kante (edge)
Flussrichtung zwischen Knoten

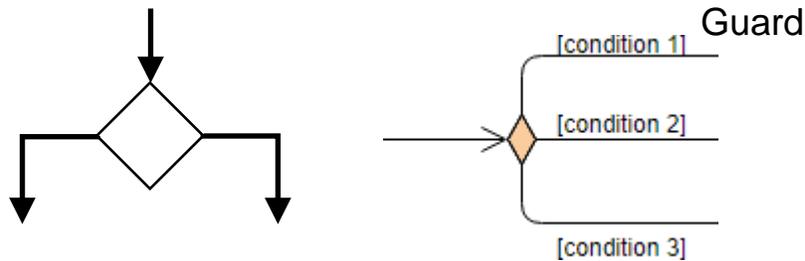


LE 3.4.4 Modellierung von Funktion und Ablauf

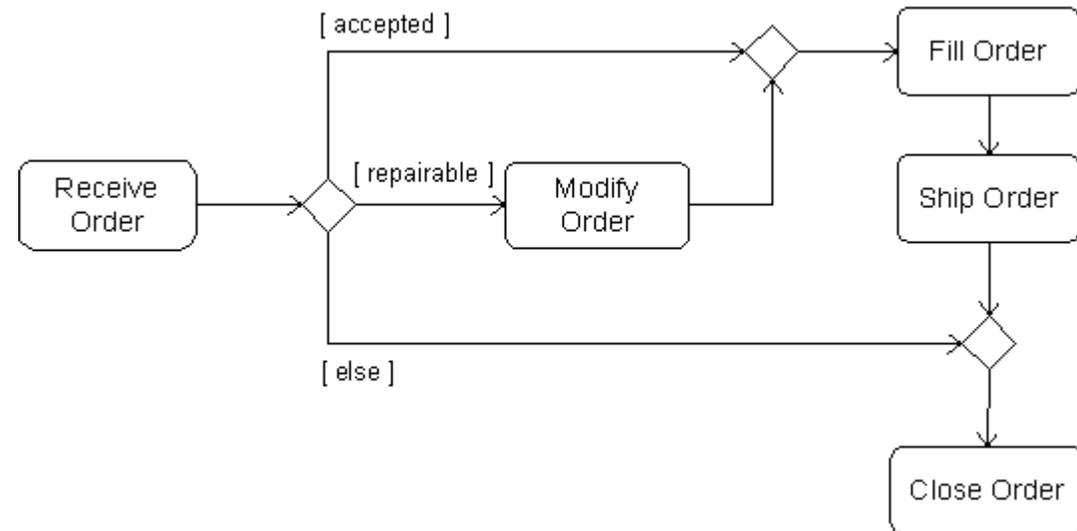
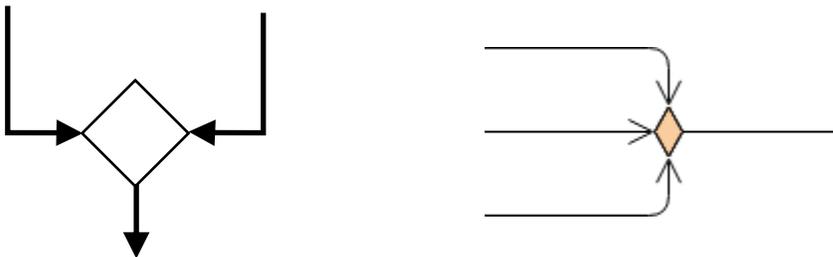
Notation UML-Aktivitätsdiagramm:

- Kontrollelemente:

- Verzweigungsknoten (DecisionNode)



- Verbindungsknoten (MergeNode)



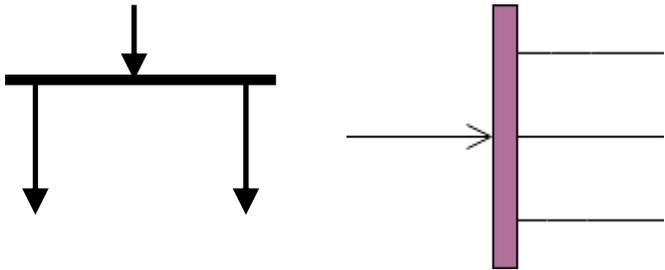
<https://online.visual-paradigm.com/diagrams/tutorials/activity-diagram-tutorial/>

LE 3.4.4 Modellierung von Funktion und Ablauf

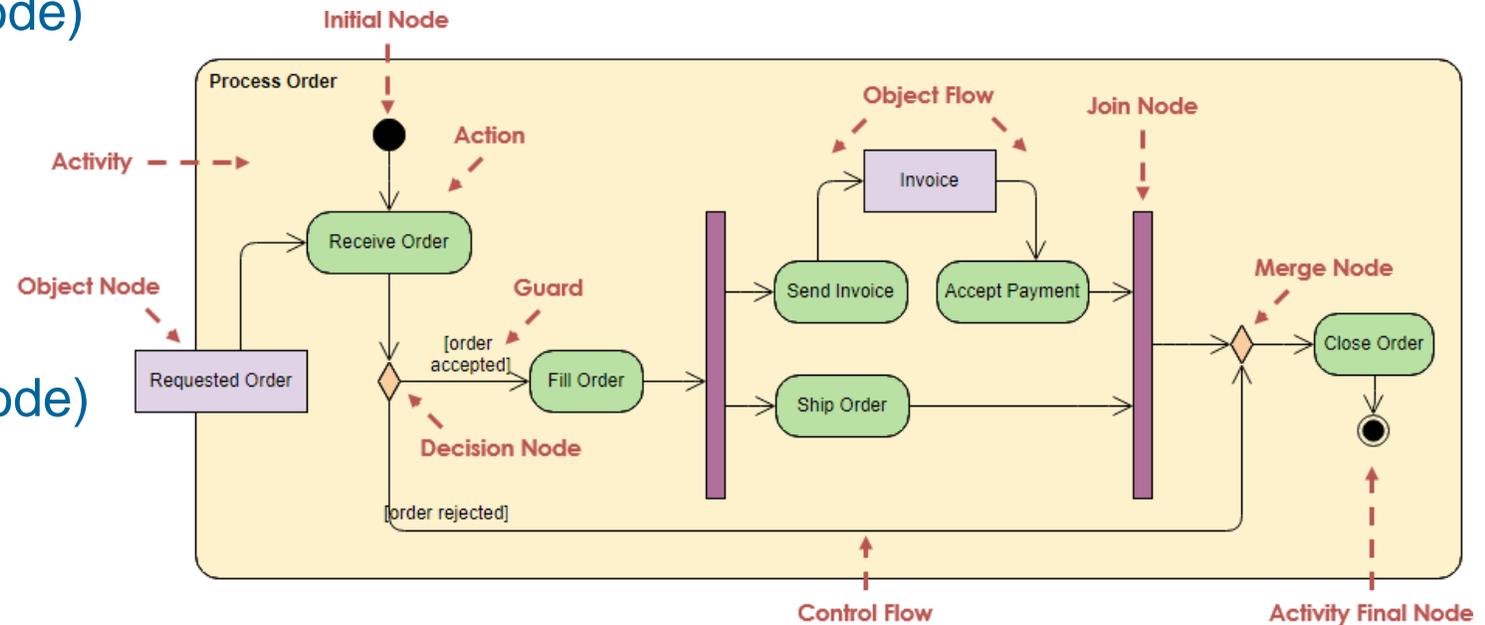
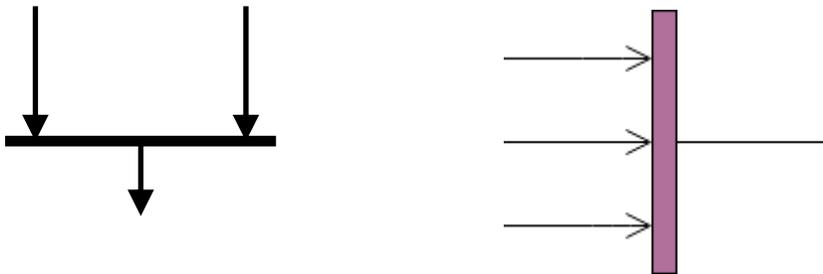
Notation UML-Aktivitätsdiagramm:

- Kontrollelemente:

- Parallelisierungsknoten (ForkNode)

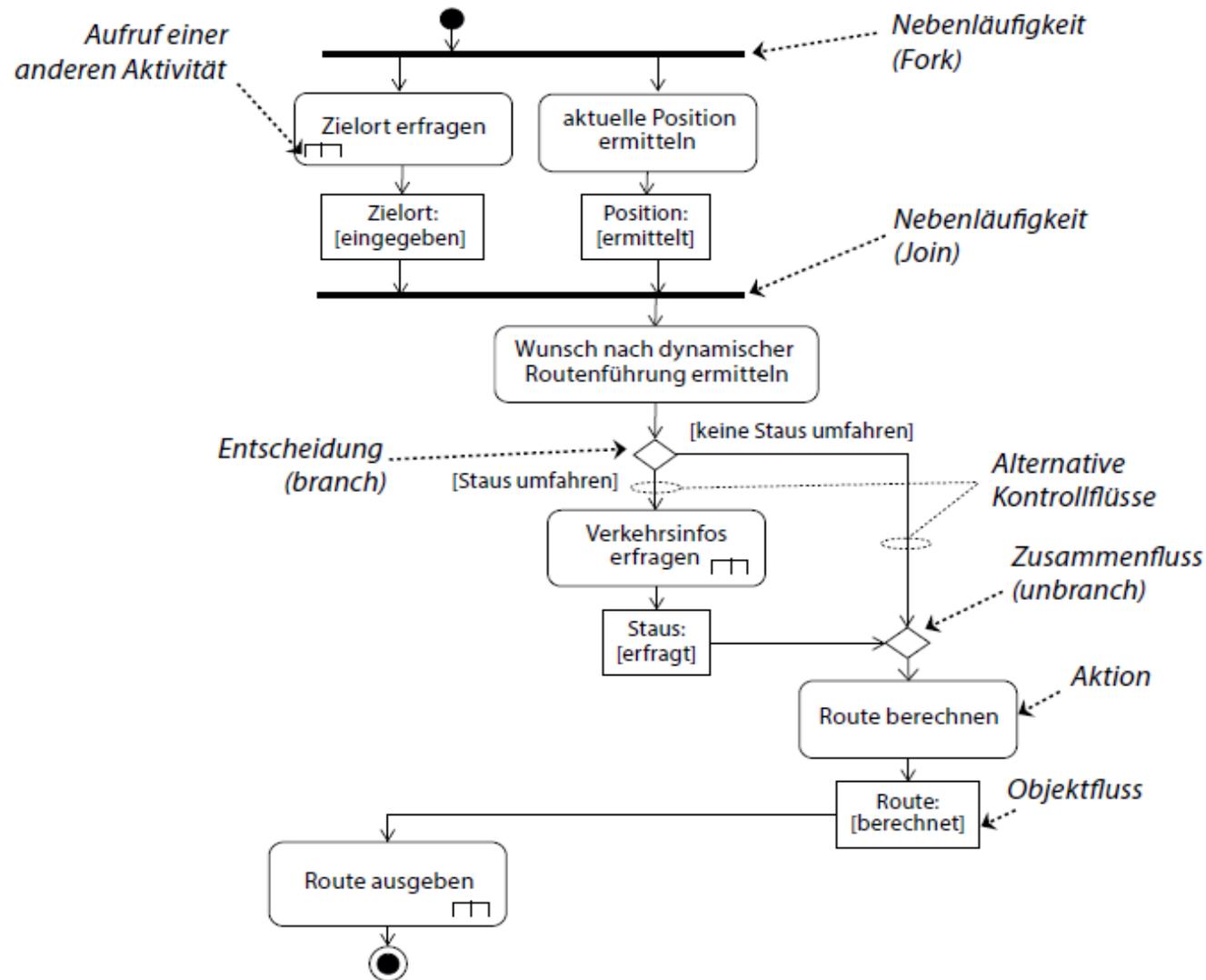


- Synchronisationsknoten (JoinNode)



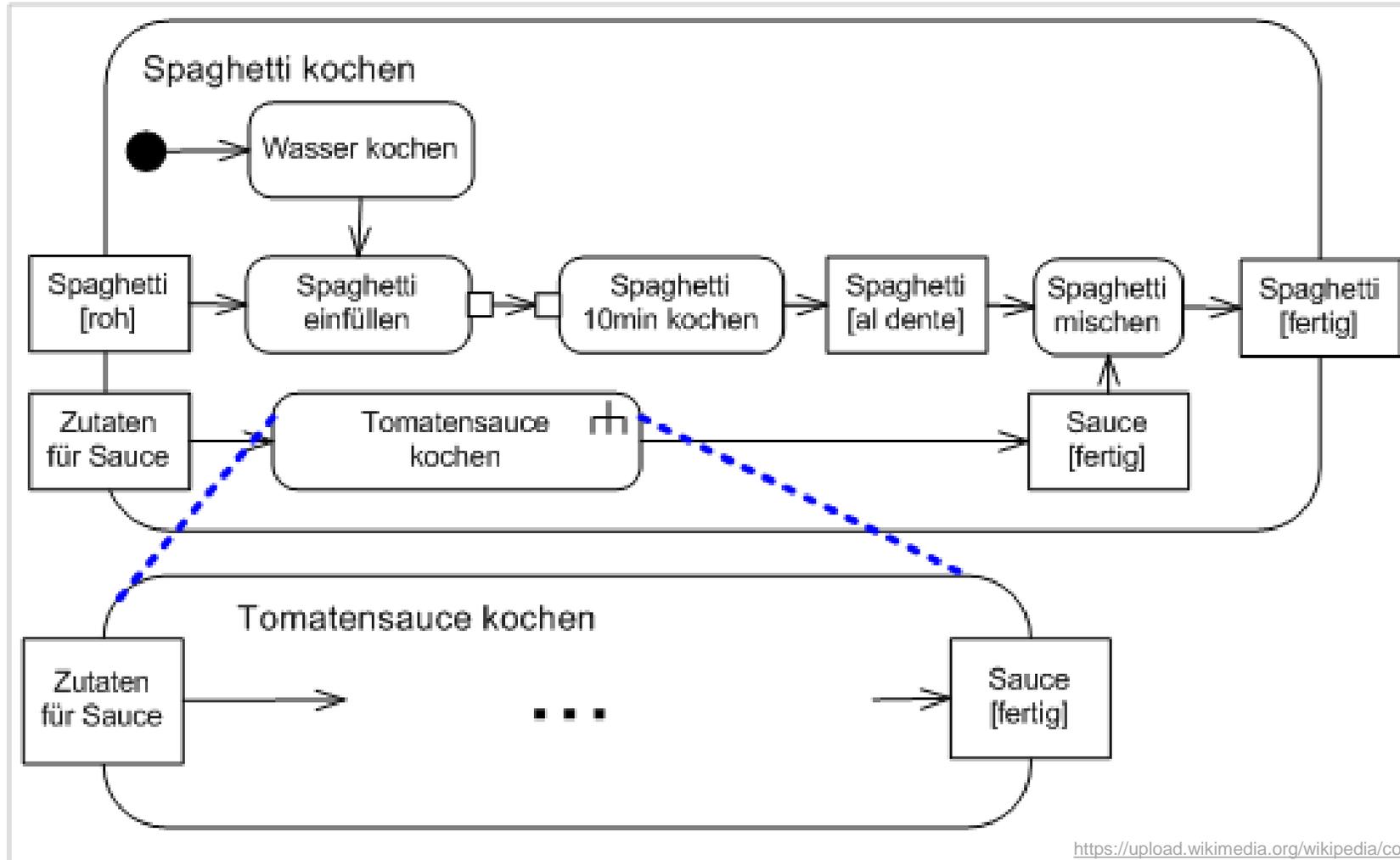
LE 3.4.4 Modellierung von Funktion und Ablauf

Beispiel UML-Aktivitätsdiagramm:



LE 3.4.4 Modellierung von Funktion und Ablauf

Beispiel UML-Aktivitätsdiagramm:

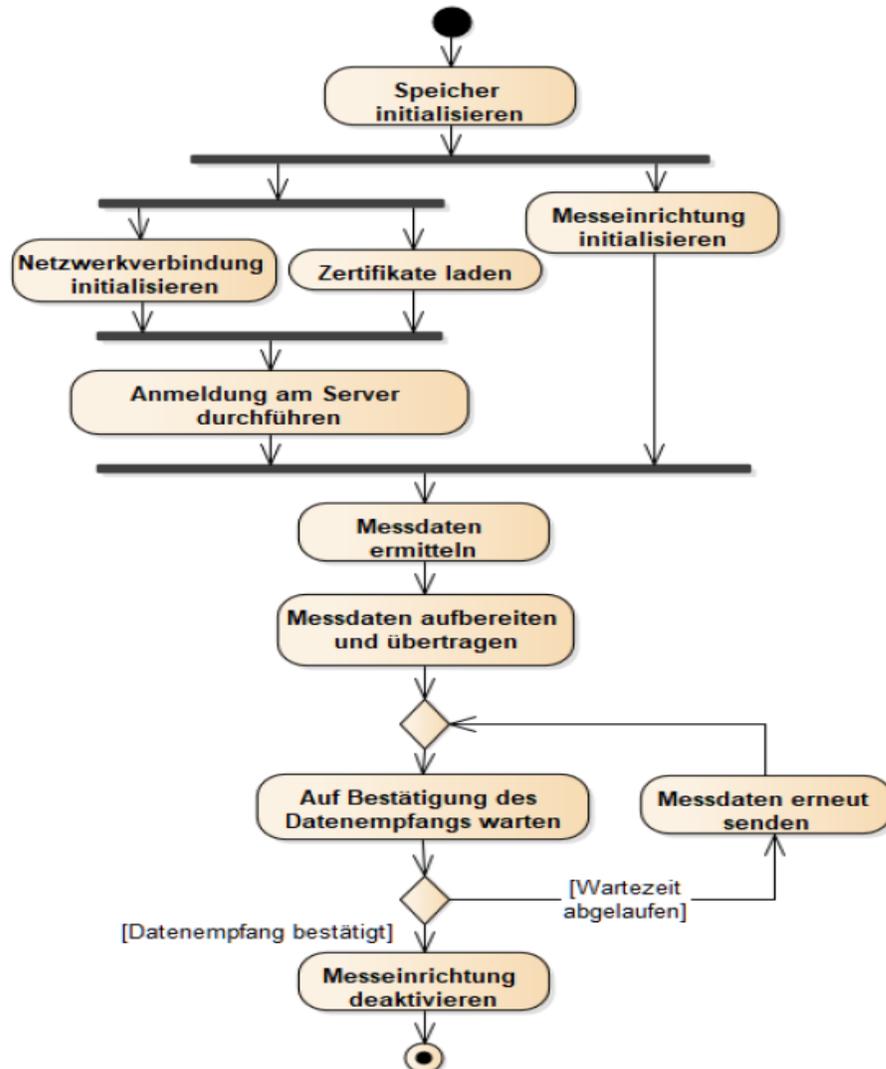


https://upload.wikimedia.org/wikipedia/commons/9/94/Activity_5.png

LE 3.4.4 Modellierung von Funktion und Ablauf

Beispiel Prüfung UML-Aktivitätsdiagramm:

Das folgende Aktivitätsdiagramm beschreibt die Durchführung einer Messung.



Stimmt überein	Stimmt nicht überein	
		A) <i>Messeinrichtung initialisieren</i> muss vor <i>Anmeldung am Server durchführen</i> erfolgen.
		B) <i>Anmeldung am Server durchführen</i> erfolgt sobald <i>Zertifikate laden</i> abgeschlossen ist.
		C) <i>Netzwerkverbindung initialisieren</i> und <i>Zertifikate laden</i> müssen zum gleichen Zeitpunkt abgeschlossen werden.
		D) <i>Auf Bestätigung des Datenempfangs warten</i> erfolgt immer direkt vor <i>Messeinrichtung deaktivieren</i> .



LE 3.4.4 Modellierung von Funktion und Ablauf

Übung UML-Aktivitätsdiagramm:

- Ziel: *Bibliothekssystem*
 - Verleiht Bücher an Kunden
- Funktionen:
 - Bücher verleihen
- Objekte:
 - Kunden (müssen registriert sein)
 - Bücher (müssen eingepflegt sein)
- Aufgabe
 - Erstellen eines UML-Aktivitätsdiagramms
 - Kunden registrieren
 - Bücher einpflegen
 - Bücher verleihen
 - Präsentation der Ergebnisse

LE 3.4.5

Zustands- und Verhaltensmodellierung (Verhaltensperspektive)

Statecharts – UML-Zustandsdiagramm

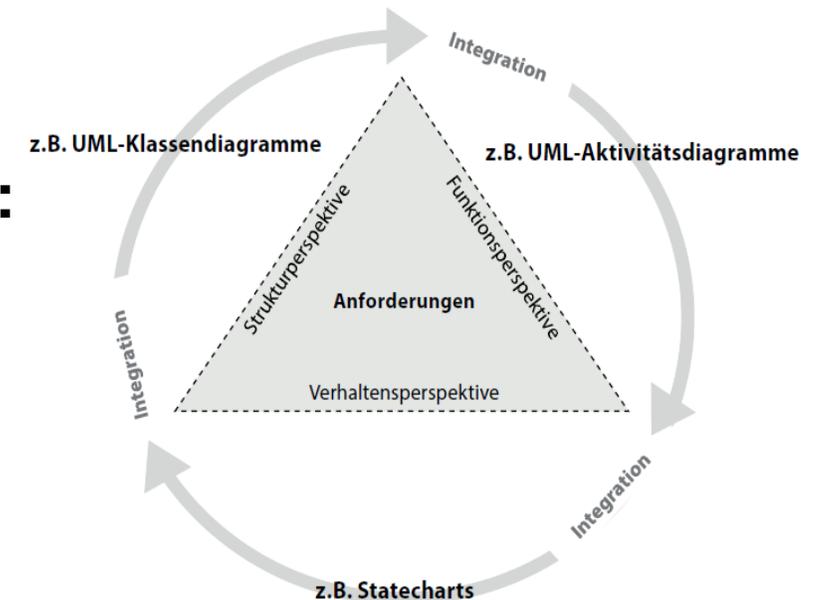
LE 3.4.5 Zustands- und Verhaltensmodellierung

Modellierung aus der Verhaltensperspektive:

- Modellierung des *dynamischen Verhaltens* des Systems
- Fokus auf *Systemzustände* und
- *Ereignisse* die zu Zustandswechsel führen

Modellsprachen für die Verhaltensperspektive:

- Statecharts (Zustandsdiagramme) [Hare1988]
- UML-Zustandsdiagramm [OMG2017]



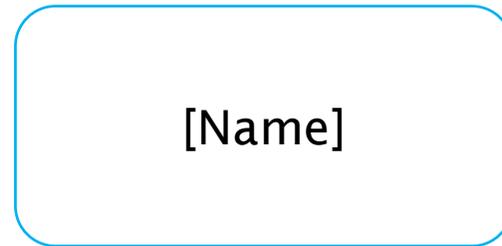
[Hare1988] David Harel. On Visual Formalisms. Communications of the ACM 1988, 31(5), 514–530.

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document, formal/2017-12-05. <https://www.omg.org/spec/UML/About-UML/>.

LE 3.4.5 Zustands- und Verhaltensmodellierung

Notation Statecharts:

- Zustand



- Startzustand



- Endzustand

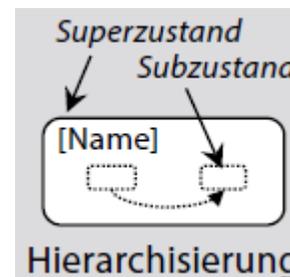
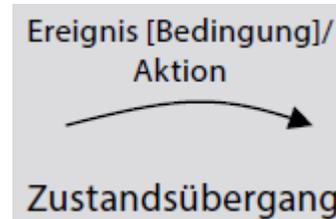


LE 3.4.5 Zustands- und Verhaltensmodellierung

Notation Statecharts:

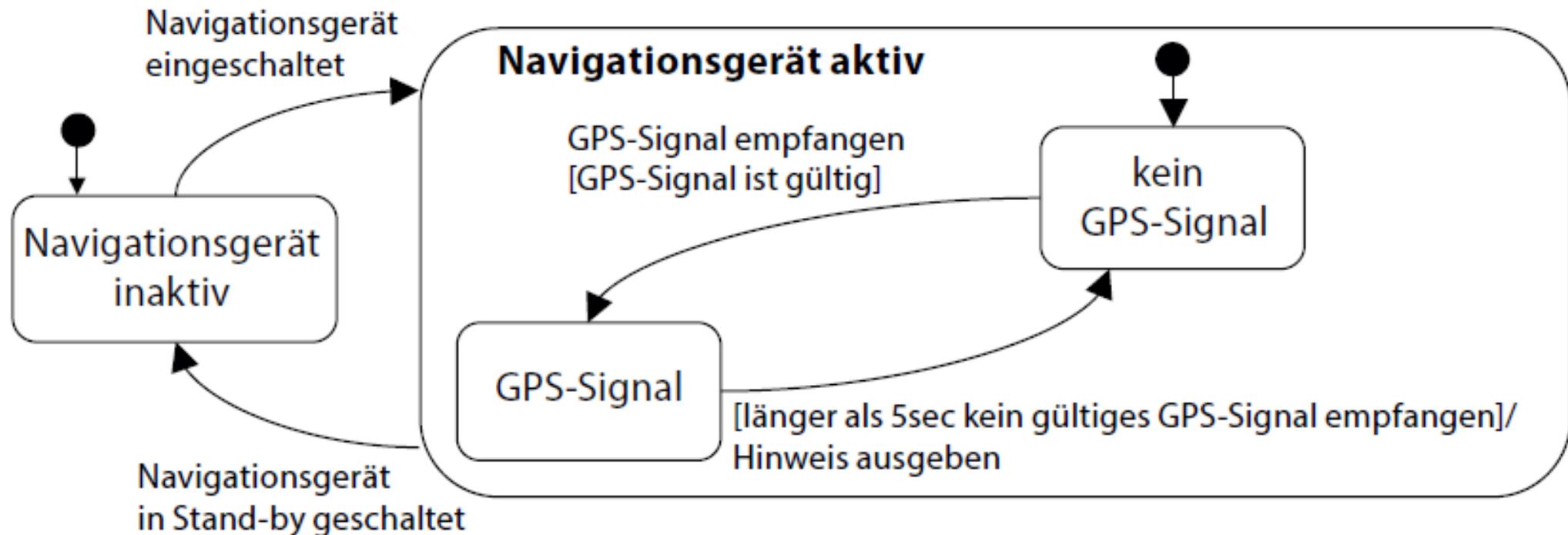
- Zustandsübergang
 - Ereignis (Bedingung)
 - Aktion

- Hierarchien



LE 3.4.5 Zustands- und Verhaltensmodellierung

Beispiel Statechart:

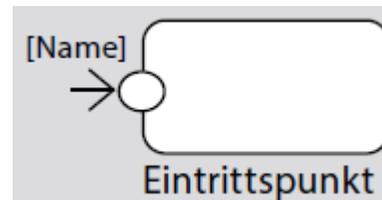


LE 3.4.5 Zustands- und Verhaltensmodellierung

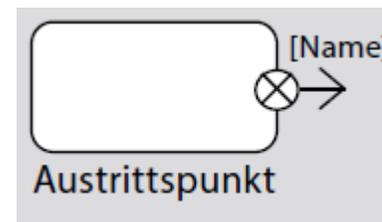
Notation UML-Zustandsdiagramme:

- basieren auf Statecharts
- zusätzlich explizite Eintritts- und Austrittspunkte bei hierarchischen Zuständen

- Eintrittspunkt

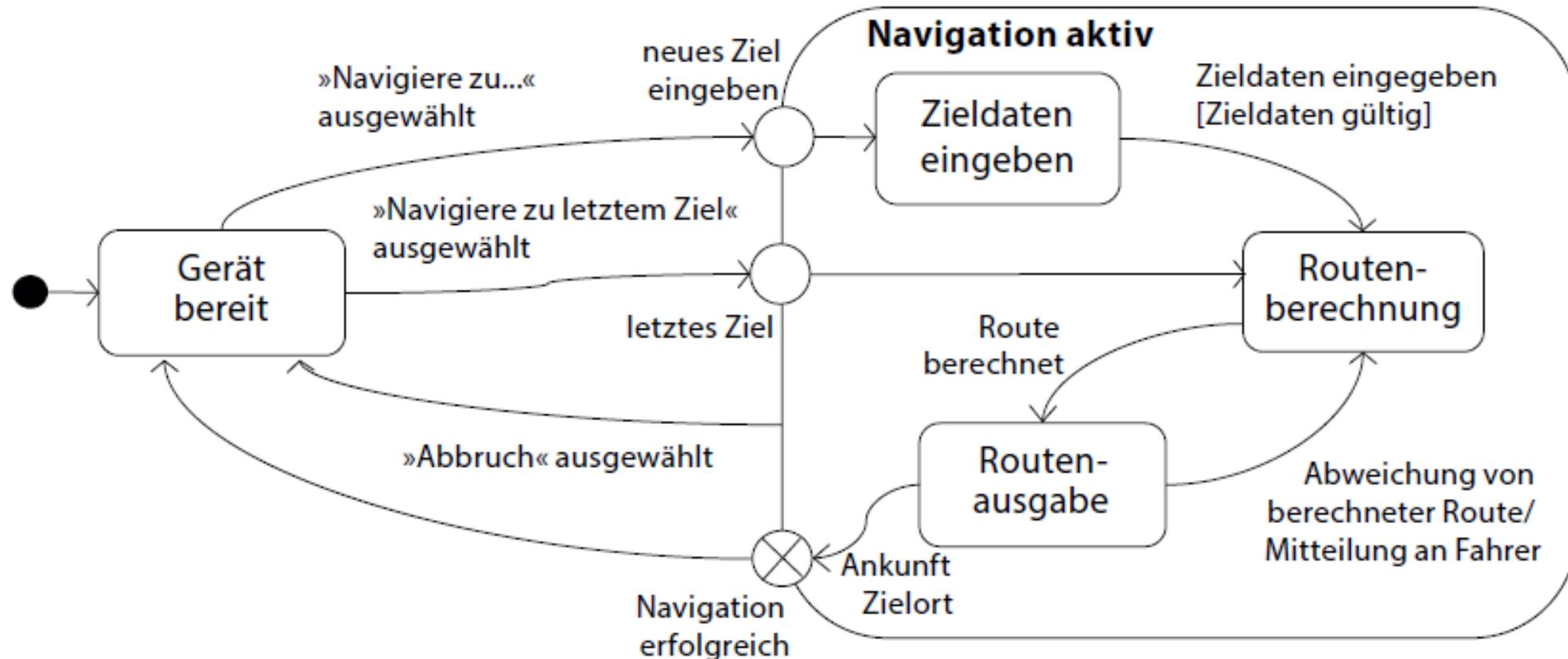


- Austrittspunkt



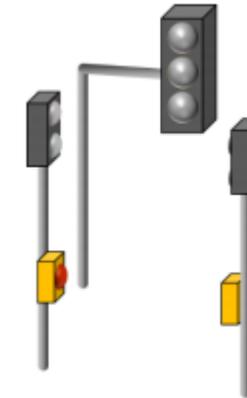
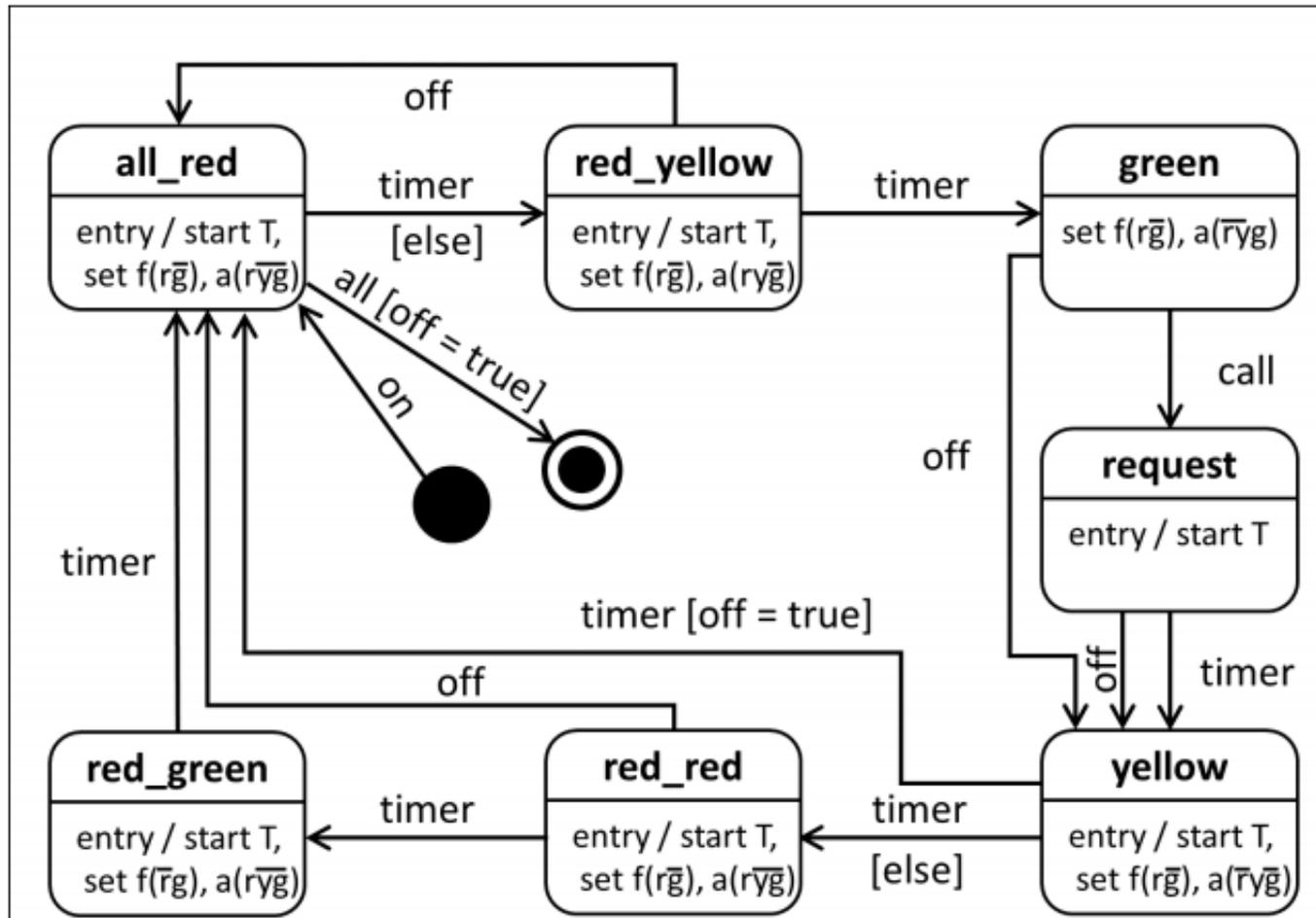
LE 3.4.5 Zustands- und Verhaltensmodellierung

Beispiel UML-Zustandsdiagramm:



LE 3.4.5 Zustands- und Verhaltensmodellierung

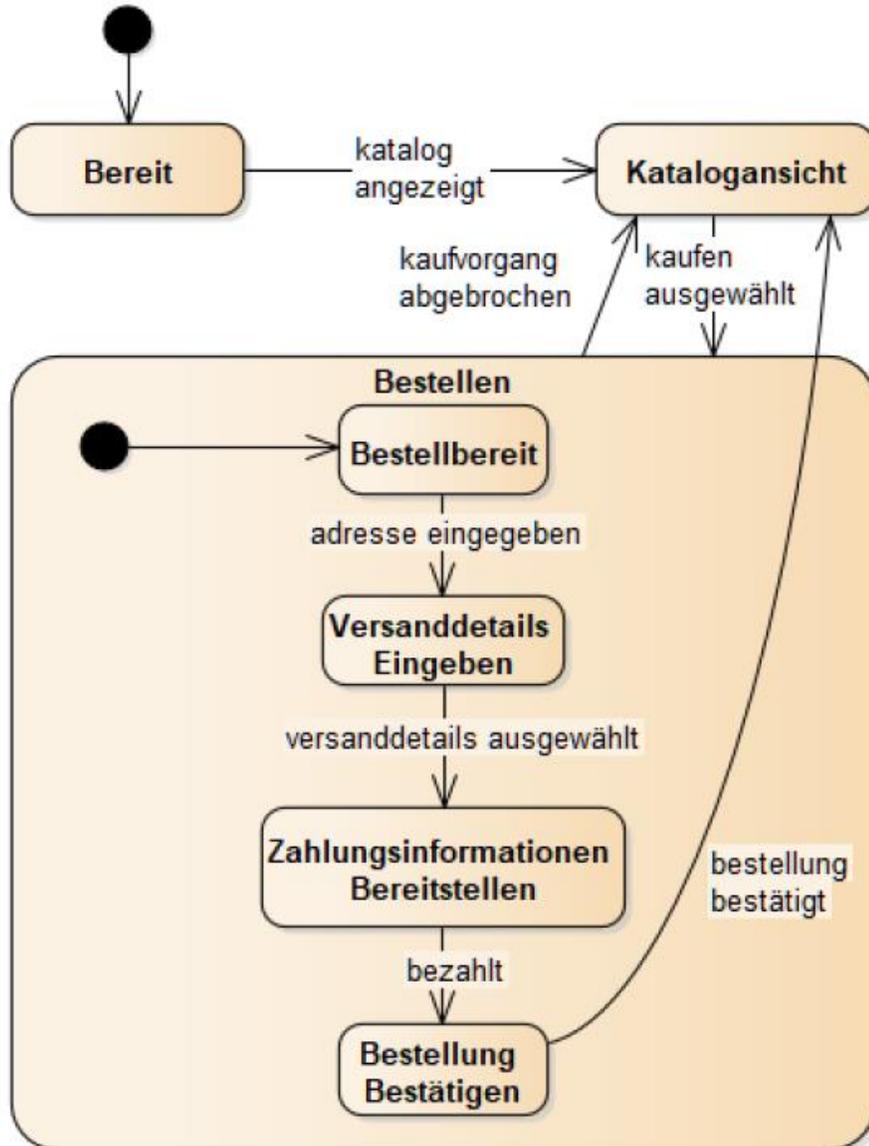
Beispiel UML-Zustandsdiagramm:



https://www.bs7-augsburg.de/aicher/files_codeconcert/11Prog/ANPR_UML_Zustandsdiagramm.pdf

LE 3.4.5 Zustands- und Verhaltensmodellierung

Beispiel Prüfung Statechart:



Stimmt überein
Stimmt nicht überein

	A) Ein direkter Übergang vom Zustand <i>Versanddetails Eingeben</i> zum Zustand <i>Katalogansicht</i> ist möglich.
	B) Ein direkter Übergang vom Zustand <i>Katalogansicht</i> zum Zustand <i>Bestellung Bestätigen</i> ist möglich.
	C) Das Ereignis <i>kaufen ausgewählt</i> muss eintreten, damit der Zustand <i>Bestellen</i> betreten wird.
	D) Um den Zustand <i>Bestellen</i> zu verlassen, muss das Ereignis <i>kaufvorgang abgebrochen</i> eintreten.

LE 3.4.5 Zustands- und Verhaltensmodellierung

Übung UML-Zustandsdiagramm:

- Ziel: *Getränkeautomat*
 - Verkauft Getränke an Kunden
- Aufgabe (Gruppenarbeit max. 3 Teilnehmer)
 - Modellieren eines Getränkeautomaten
 - Erstellen eines UML-Zustandsdiagramms
 - Getränkeausgabe
 - Retourgeldausgabe
 - <https://app.diagrams.net/>
 - Präsentation der Ergebnisse

LE 3.4.6

Modellierung von Zielen

UND/ODER-Bäume

LE 3.4.6 Modellierung von Zielen

Modellierung von Zielen:

- Geschäftsanforderungen beschreiben ein Geschäftsziel oder ein Bedürfnis.
- Ziele müssen daher den Fokus auf die Wertschöpfung legen
- Die Umsetzung der Anforderung liegt bei den wertschöpfenden Eigenschaften

Beschreibungsmodelle für Zieledefinitionen:

- Natürlichsprachige Dokumentation, vorlagenbasiert [Pich2010], [Pohl2010], [RoRo2012]
- Zielorientierte Anforderungssprache [GRL2020]
- **UND/ODER-Bäume** [AnPC1994]

[Pich2010] Roman Pichler: Agile Product Management with Scrum – Creating Products that Customers Love, Boston: Addison-Wesley, 2010.[Hare1988] David Harel. On Visual

Klaus Pohl: Requirements Engineering: Fundamentals, Principles, and Techniques. Berlin-Heidelberg: Springer, 2010.

[RoRo2012] Suzanne Robertson und James Robertson: Mastering the Requirements Process: Getting Requirements Right. 3. Auflage. Boston: Addison-Wesley, 2012.

[GRL2020] Goal oriented Requirement Language. University of Toronto, Canada <https://www.cs.toronto.edu/km/GRL>.

[AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.

LE 3.4.6 Modellierung von Zielen

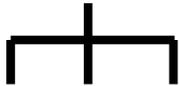
UND/ODER-Bäume:

Zweck:

- Beschreibung der **Intention** eines **Stakeholders**
- Beschreibung **charakteristischer Merkmale** eines zu entwickelnden Systems

Modellelemente:

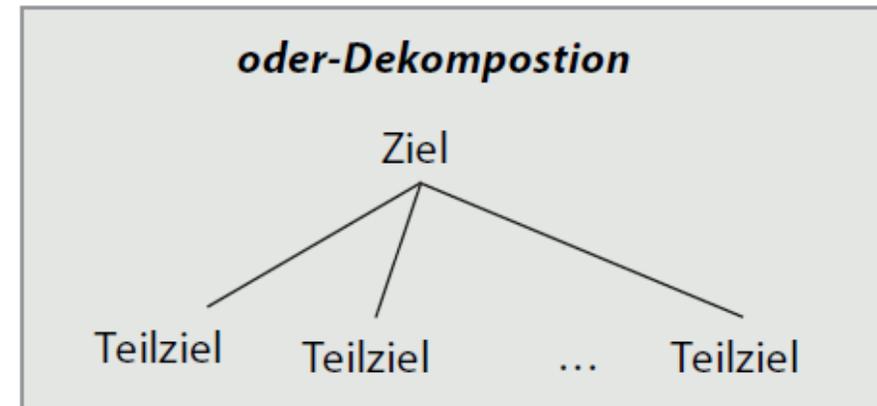
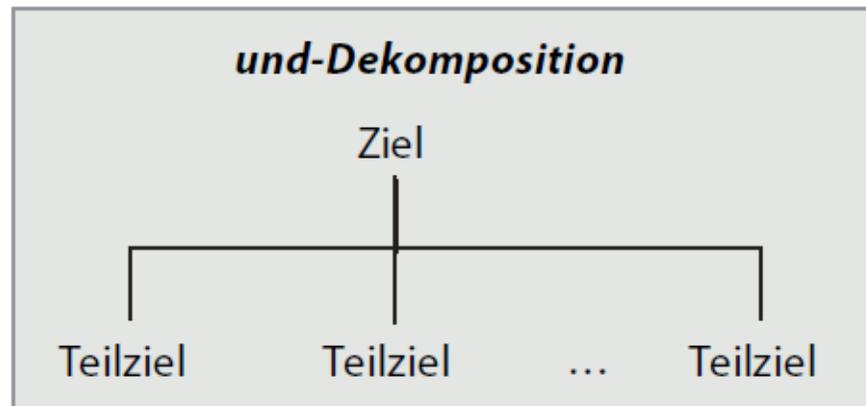
- **Verfeinerungsbeziehungen** (Dekompositionsbeziehungen) zwischen einem übergeordneten und untergeordneten Ziel:

- *Und-Dekomposition* 
(alle Teilziele müssen erfüllt sein, um das übergeordnete Ziel zu erfüllen)

- *Oder-Dekomposition* 
(mindestens ein Teilziel muss erfüllt sein, um das übergeordnete Ziel zu erfüllen)

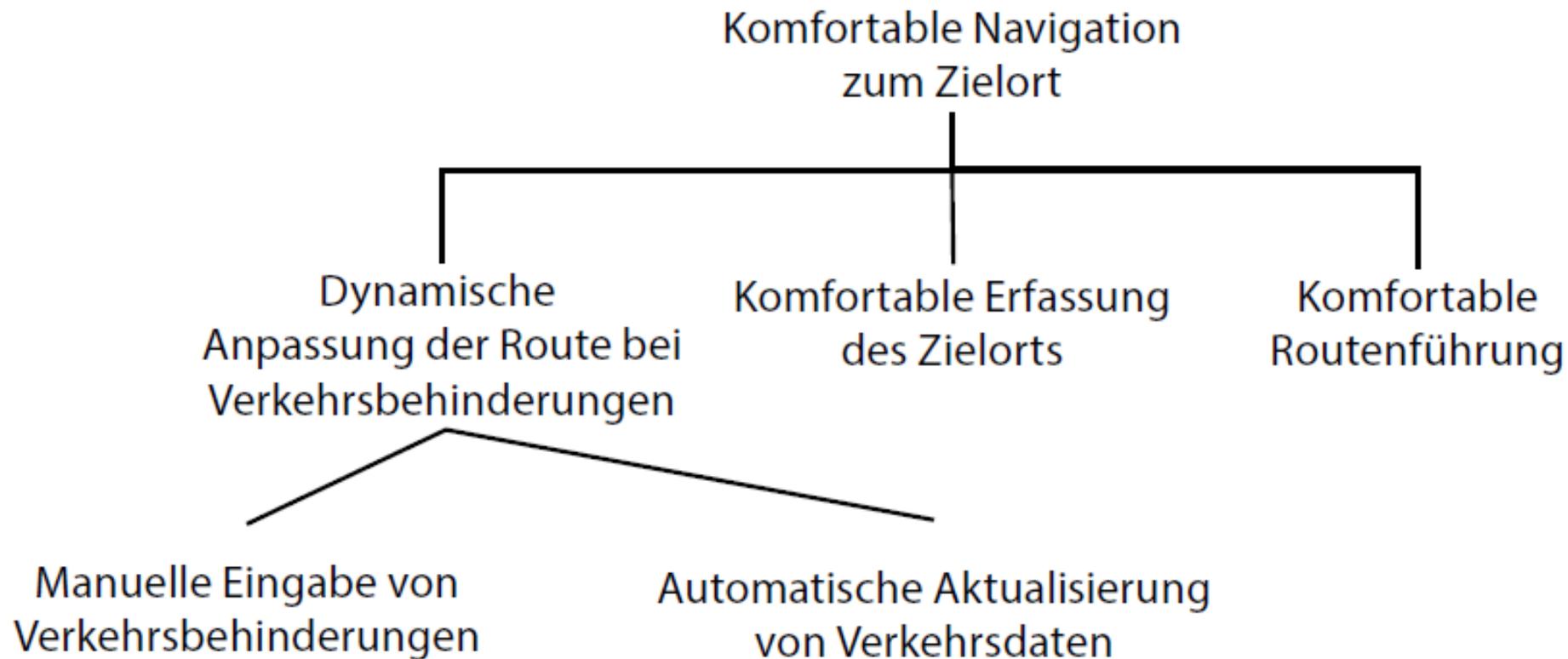
LE 3.4.6 Modellierung von Zielen

UND/ODER-Bäume Dekompositionsbeziehungen:



LE 3.4.6 Modellierung von Zielen

Beispiel UND/ODER-Baum:



Literatur

- Basiswissen Requirements Engineering:
Aus- und Weiterbildung nach IREB-Standard zum
Certified Professional for Requirements Engineering Foundation Level
Chris Rupp & Klaus Pohl; dpunkt.verlag GmbH; 2015
- Handbuch für das CPRE Foundation Level nach dem IREB-Standard
Aus- und Weiterbildung zum Certified Professional for Requirements Engineering (CPRE)
Foundation Level
Martin Glinz, Hans van Loenhoud, Stefan Staal, Stan Bühne
- https://www.ireb.org/content/downloads/4-cpre-foundation-level-syllabus-3-0/cpre_foundationlevel_syllabus_de_v.3.0.1.pdf
- Requirements-Engineering und -Management:
Aus der Praxis von klassisch bis agil
Christine Rupp, die SOPHISTen
Carl Hanser Verlag GmbH & Co. KG; 2020
- UML 2 glasklar: Praxiswissen für die UML-Modellierung
Christine Rupp, Stefan Queins, die SOPHISTen
Carl Hanser Verlag GmbH & Co. KG

Urheberrechtshinweis

Die Urheberrechte dieser Unterlage liegen beim Autor dieses Werkes.

Die Nutzung der bereitgestellten Unterlagen ist ausschließlich zur eigenen Verwendung im Rahmen des Kurses gestattet.

Eine Weitergabe an andere, nicht für den Kurs eingetragene Personen, wird ausdrücklich untersagt.

© Robert Jonas