

Vorlesung Software Engineering I

7 Unified Modeling Language: Anwendungsfalldiagramme

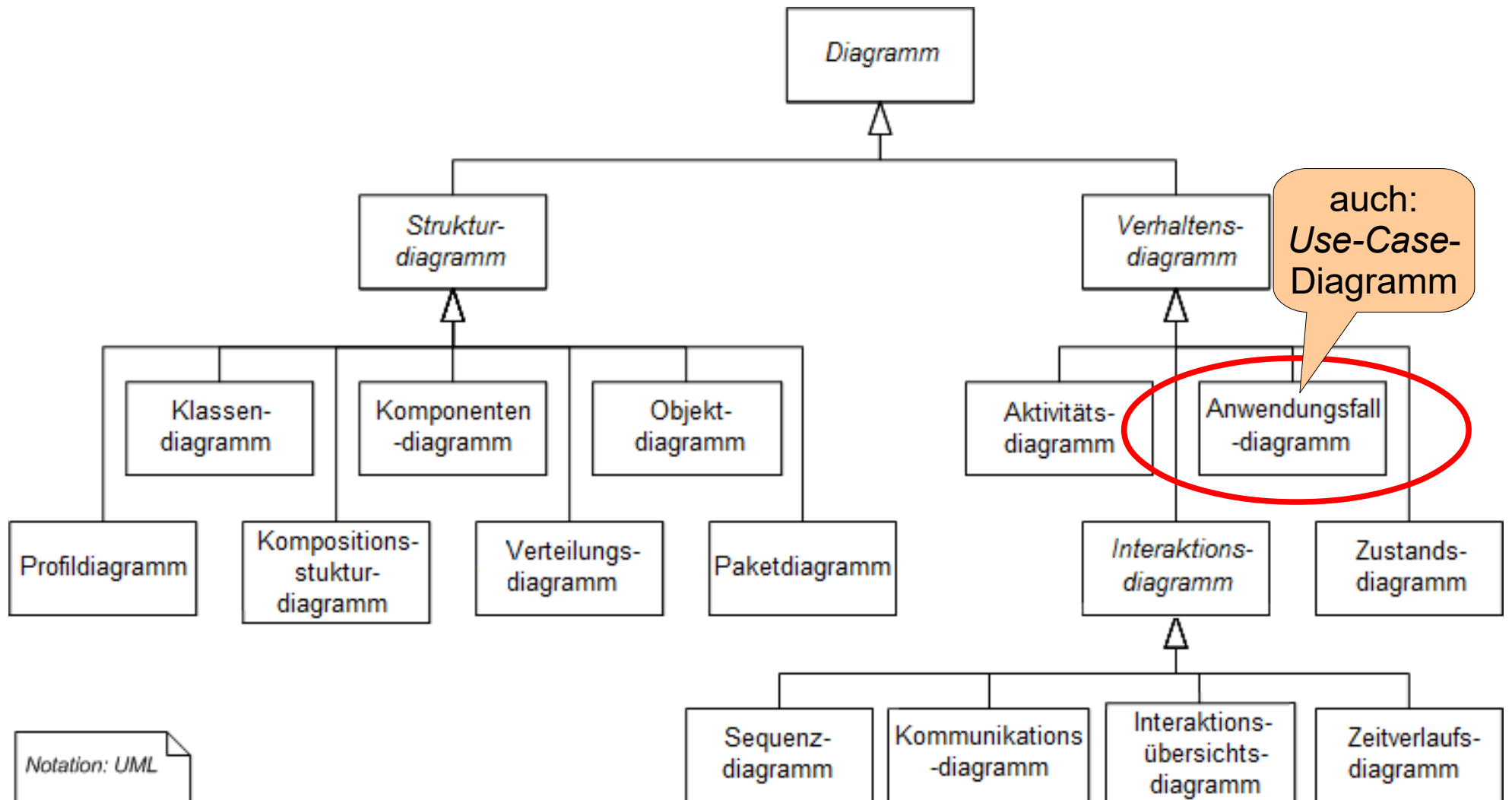
WS 2019/20

Prof. Dr. Dirk Müller

Übersicht

- Einführung
- Software-Entwicklungsprozesse
- Anforderungsanalyse
- Prozessanalyse und -modellierung
- Objekt-orientierte Analyse
- UML
 - Ziele, Grenzen, Geschichte, Metamodellierung, Bewertung
 - [Anwendungsfalldiagramme](#)
 - Klassendiagramme
 - Aktivitätsdiagramme
 - Zustandsdiagramme

Diagrammarten in der UML 2.4.1



Quelle: <http://upload.wikimedia.org/wikipedia/de/5/53/UML-Diagrammhierarchie.png>
Download am 27.05.2014

Motivation

- Erstellung eines Anwendungsfalldiagramms als einer der **ersten Analyse-Schritte**
 - verhaltensorientierte OOA: 1. Schritt
 - datenorientierte OOA: 2. Schritt
- **Nutzersicht** auf ein System von außen: *Black Box*
- **Zerlegung** in Anwendungsfälle
- **Schnittstellen** und **Kommunikationspartner** explizieren
- Anwendungsfälle können als **Schritte** bei einer **inkrementellen** Entwicklung dienen
- **Was** leistet mein System für seine **Umwelt?**
 - hohes Abstraktionsniveau
 - einfache Notation

Grundelemente

- **Anwendungsfall**

- direkte Ableitung aus **funktionalen** Anforderungen
- eigentlich ein Anwendungsfall**typ** (mehrere Instanzen möglich)
- **Name** Pflicht: „Substantiv Infinitiv“ oder substantiviertes Verb

Buch ausleihen

- **System**

- Festlegung des **Betrachtungsgegenstandes**
- Verhalten muss nach außen **ganzheitlich** angeboten werden

Bibliothek

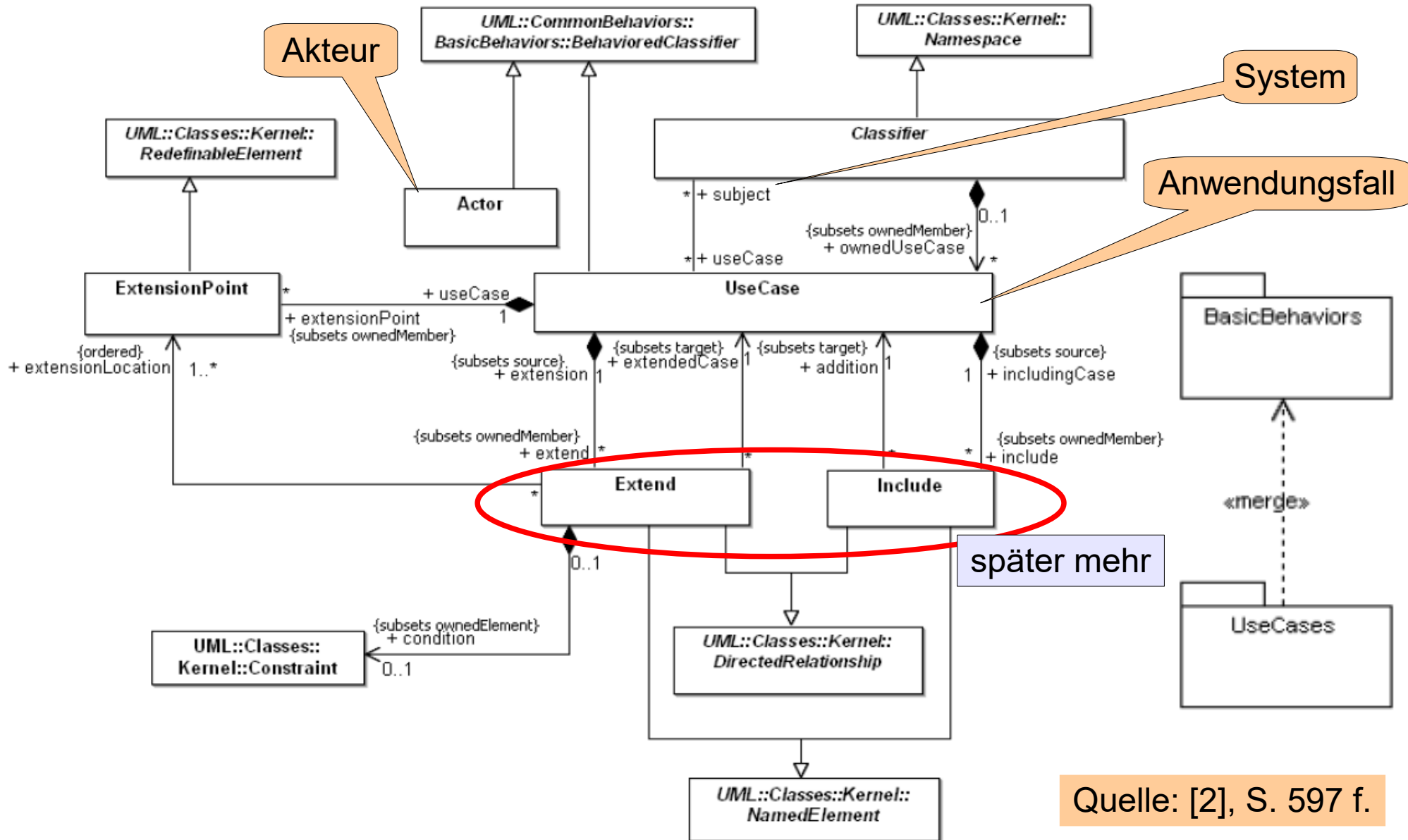
- **Akteur**

- immer **außerhalb** des Systems
- Nutzer oder andere Systeme
- nicht verfeinerbar
- internes Verhalten nicht modelliert
- muss einen **Namen** haben



Weihnachtsmann

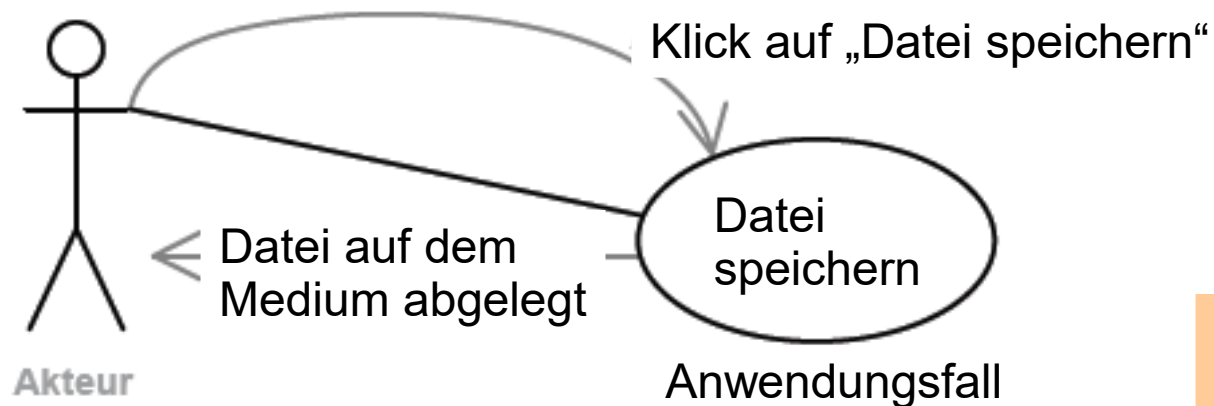
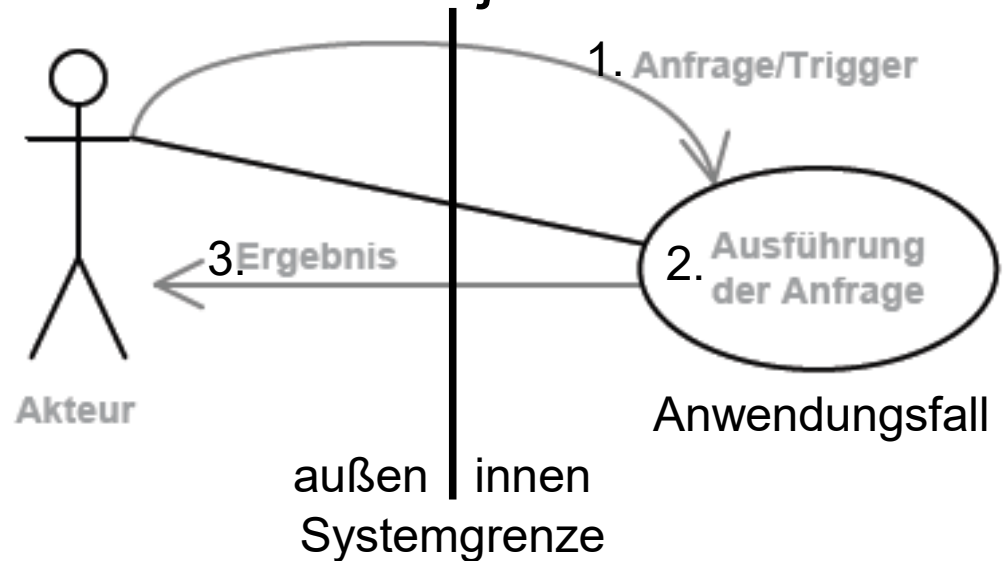
Abstrakte Syntax



Quelle: [2], S. 597 f.

Assoziation Akteur-Anwendungsfall

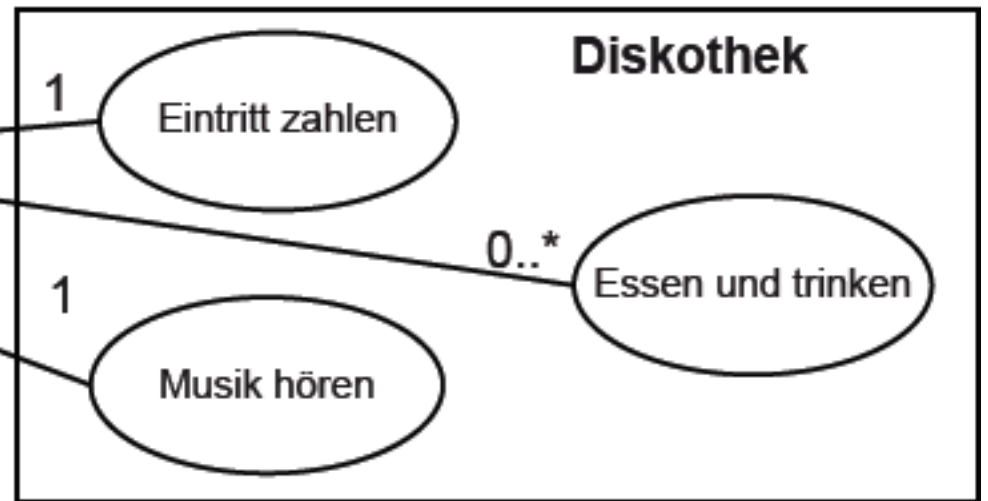
- muss **binär** sein (in einer Assoziation nur jeweils 1 Akteur und 1 Anwendungsfall)
- Akteur kommuniziert als **Black Box** mit System
- Normalfall: bidirektionaler Datenfluss, sonst durch **gerichtete** Assoziation verfeinert



basiert auf:
[3], S. 247

Assoziation Akteur-Anwendungsfall: Rollen, Richtungen, Multiplizitäten

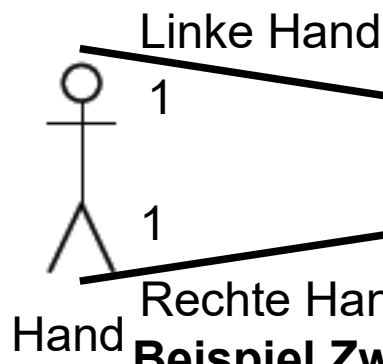
- Jeder Besucher ist Zahler und zahlt genau 1x Eintritt.



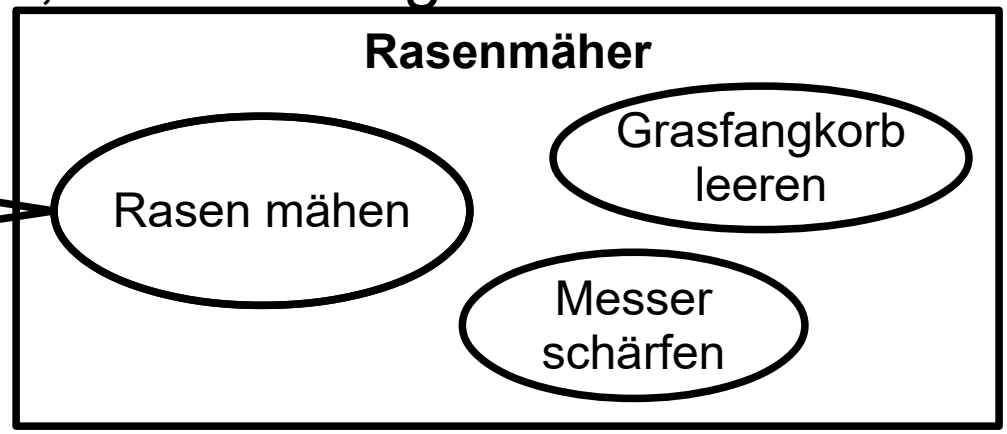
- Essen und trinken beliebig oft (auch gar nicht) möglich

Quelle: [3], S. 253

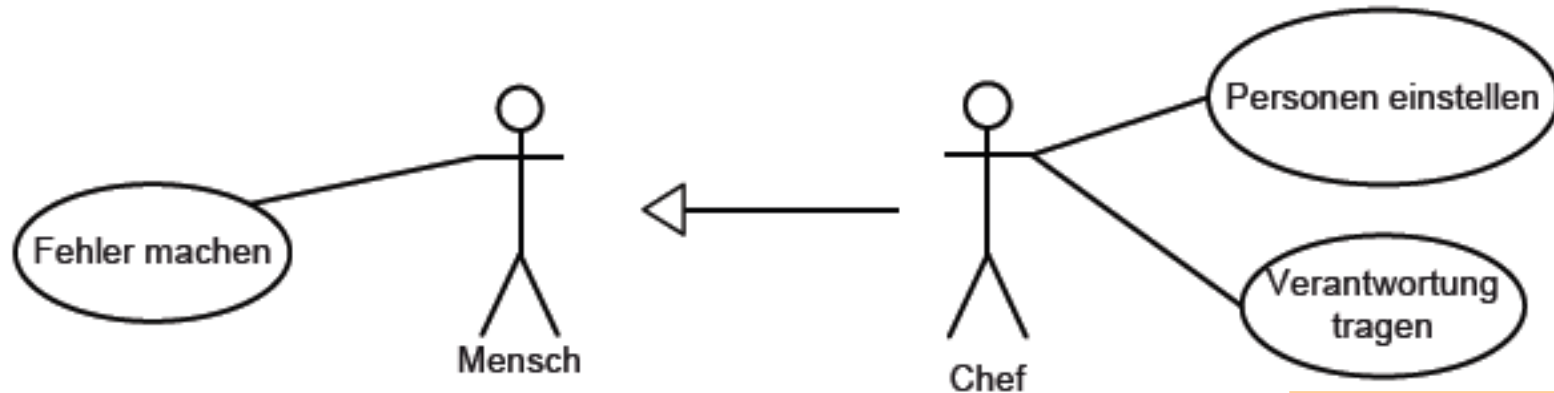
- kein Entkommen der Musik, die einseitig auf uns wirkt



Beispiel Zweihandsteuerung



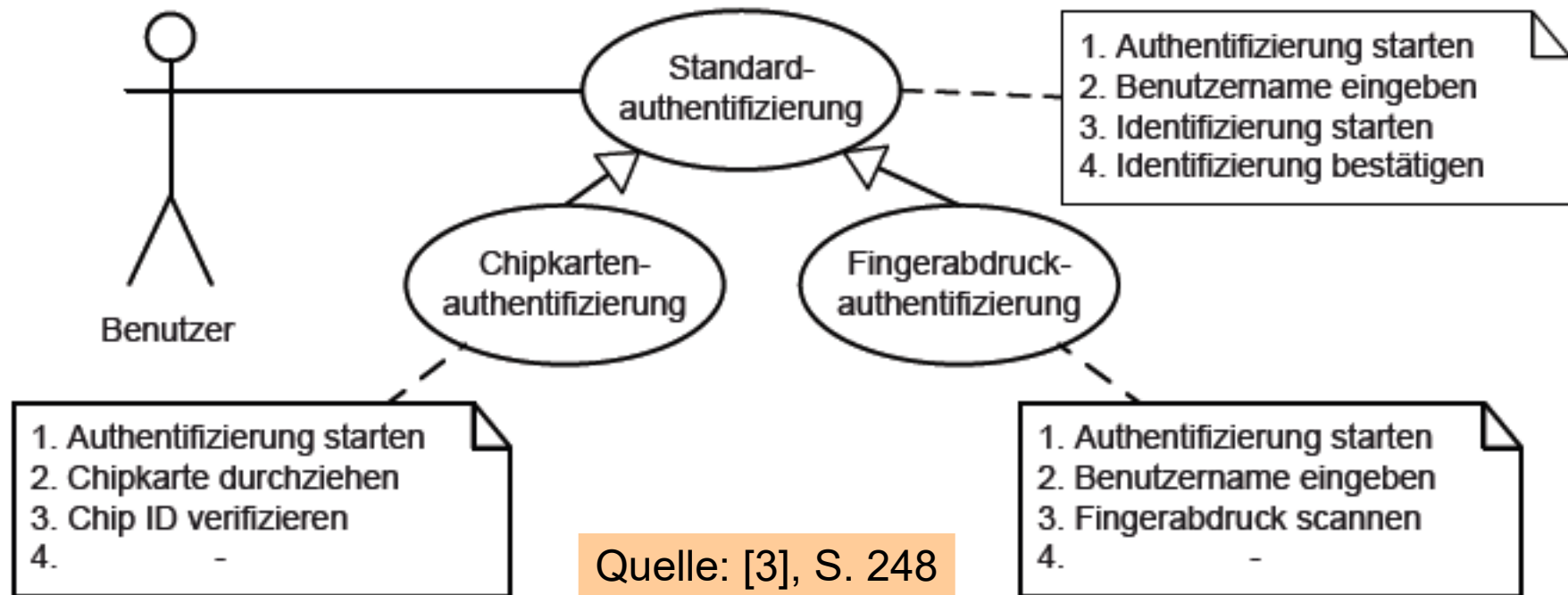
Generalisierung von Akteuren



Quelle: [3], S. 254

- spezialisierter Akteur **erbt** vom vererbenden Akteur Assoziationen, also insbesondere die **assoziierten Anwendungsfälle**
- „Auch ein Chef ist ein Mensch, der Fehler macht.“
- **Umkehrung** gilt **nicht**
 - Nicht jeder Mensch kann Personen einstellen.
 - „Verantwortung tragen“: laut Modell analog, evtl. Modell überarbeiten

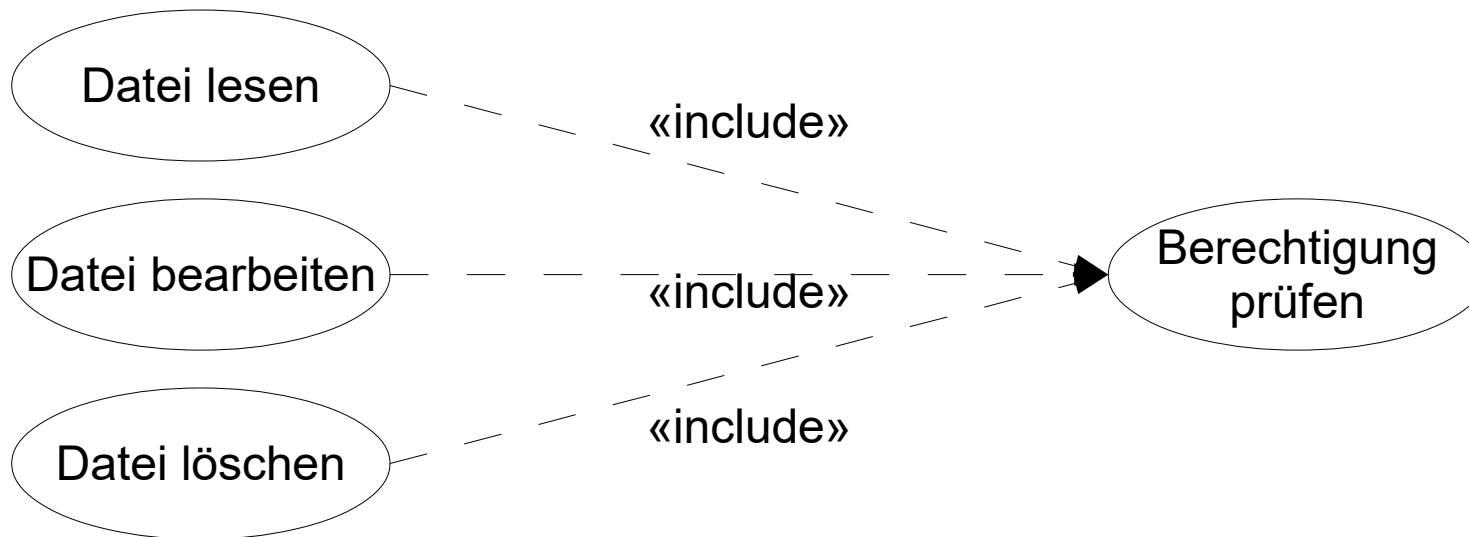
Generalisierung von Anwendungsfällen



- Beschreibung von **Varianten** eines Anwendungsfalls
 - Standardfall und Spezialfälle
- **Verhalten** und **Assoziationen zu Akteuren** werden vererbt
- im Beispiel: Standard mittels Wissen, Spezialfälle mittels Besitz bzw. körperlicher Merkmale (Biometrie)

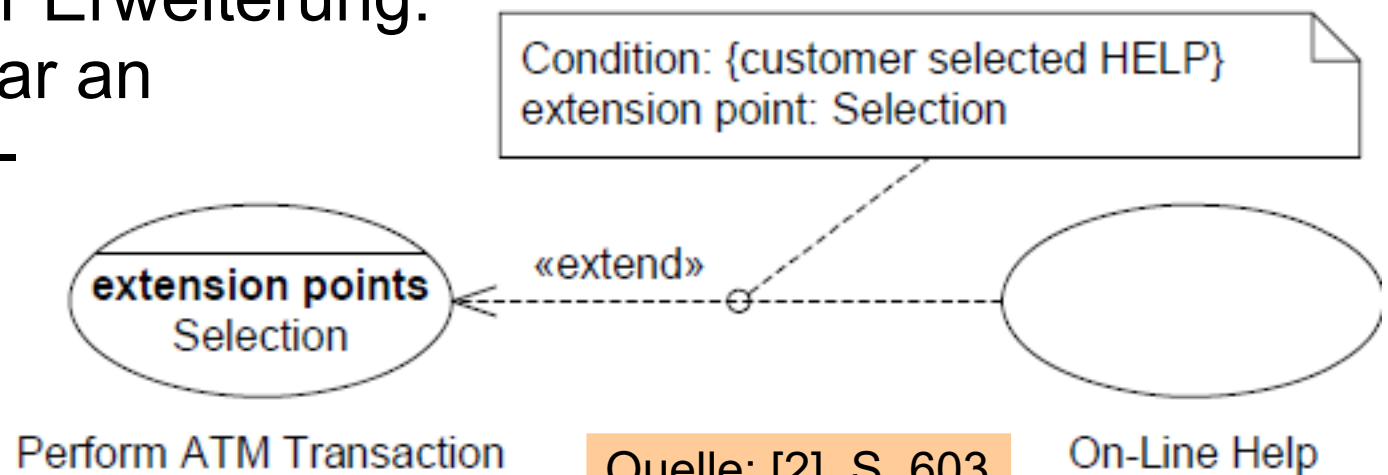
Include-Beziehung zwischen Anwendungsfällen

- Ziel: Redundanz reduzieren, übersichtlichere Modellierung
- **Wiederverwendung** gemeinsamer Teile von Abläufen
 - **obligatorische** Ausführung einer Unterroutine
- **gerichtete** 1-zu-1-Beziehung, einbindender Anwendungsfall hängt von eingebundenem ab, aber nicht umgekehrt
- **zyklische** Abhängigkeit (selbst oder Kreis) **verboten**

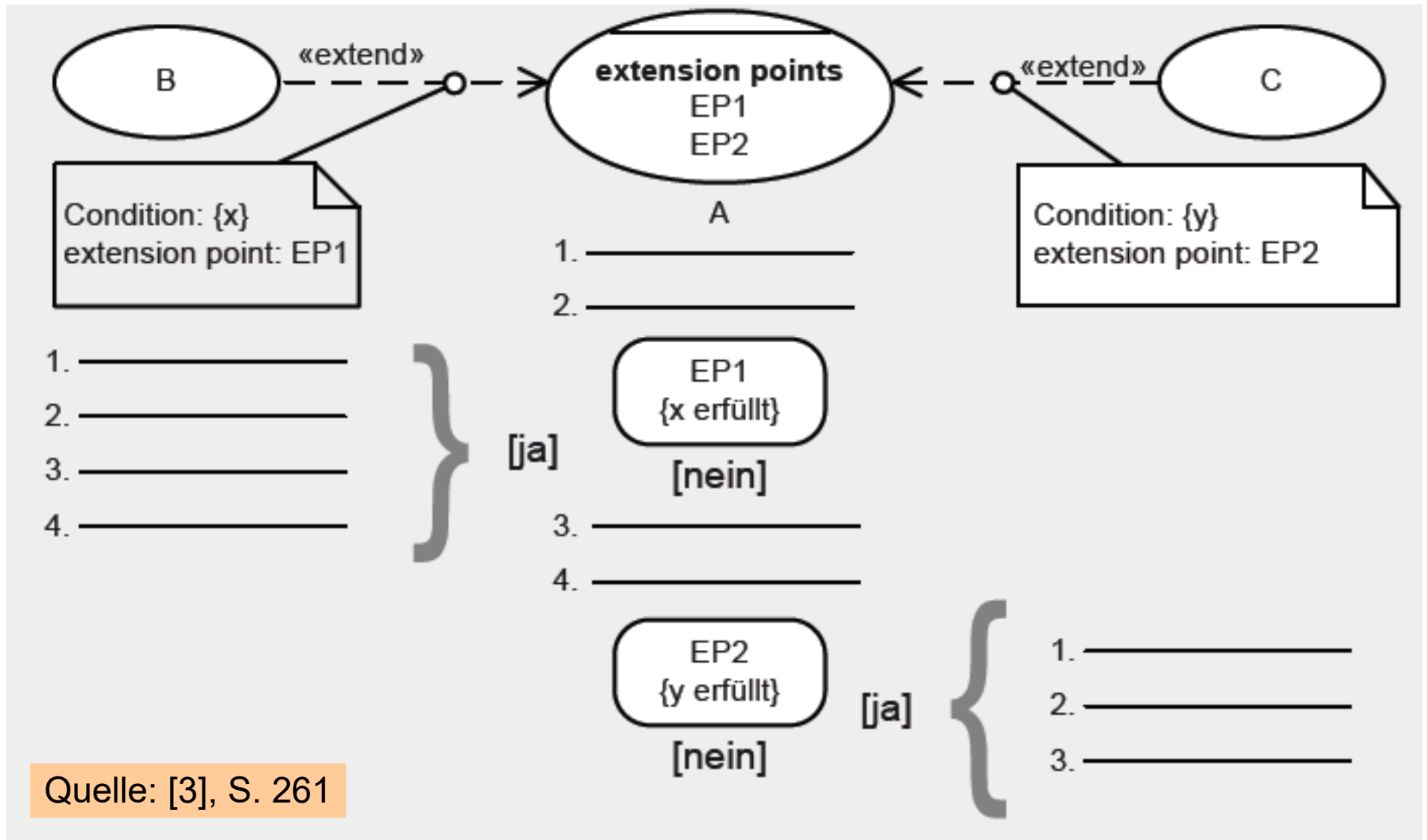


Extend-Beziehung zwischen Anwendungsfällen

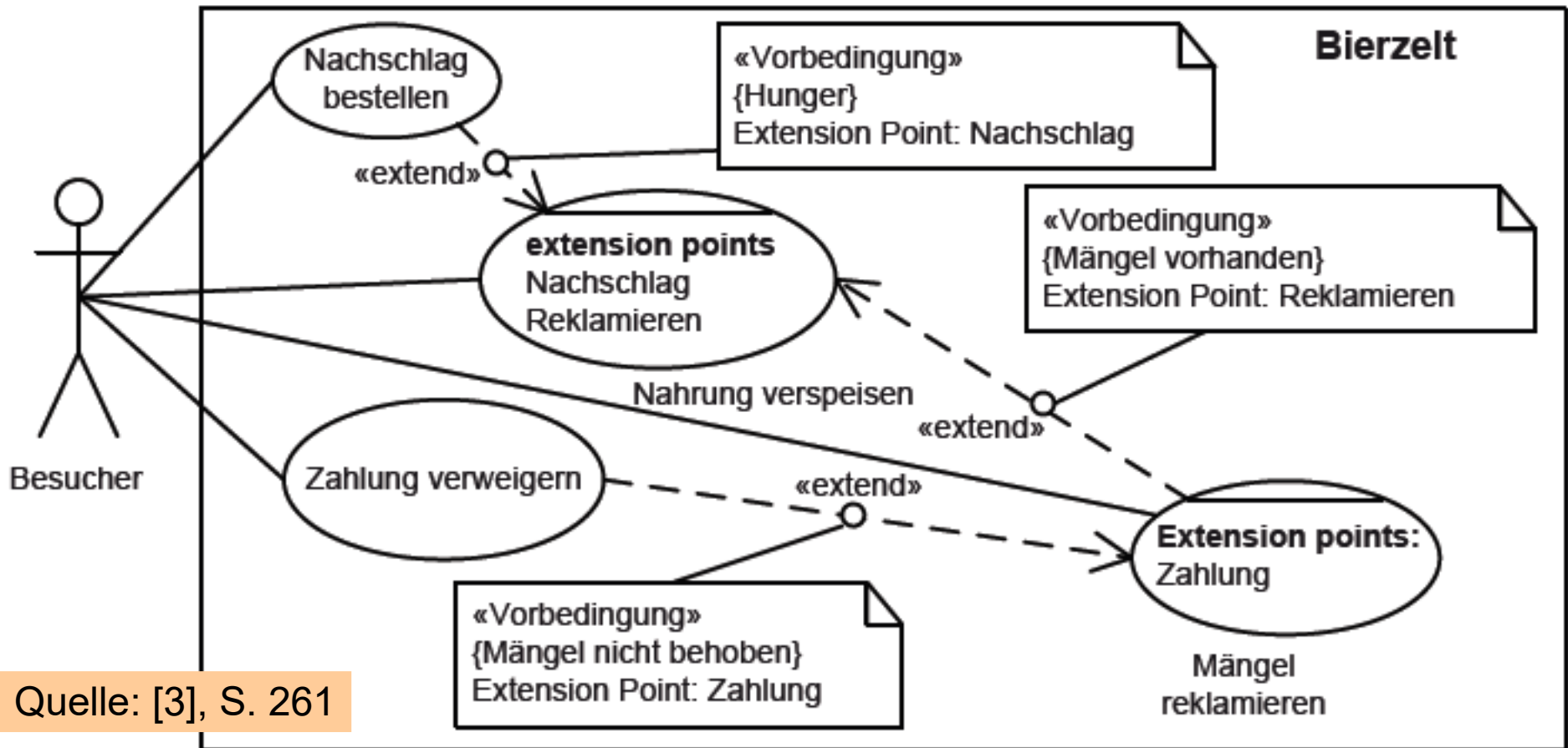
- **bedingte** Erweiterung eines in sich geschlossenen Anwendungsfalls zur Beschreibung von Sonderfällen
- gerichtete 1-zu-1-Beziehung vom erweiternden zum erweiterten Anwendungsfall
- **Erweiterungspunkt**: Verzweigungsstelle im Ablauf des erweiterten Anwendungsfalls
- **Bedingung** für Erweiterung: als Kommentar an Erweiterungsbeziehung, Standard: wahr



Ablauf einer «extend»-Beziehung

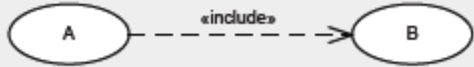



Mehrere Erweiterungspunkte



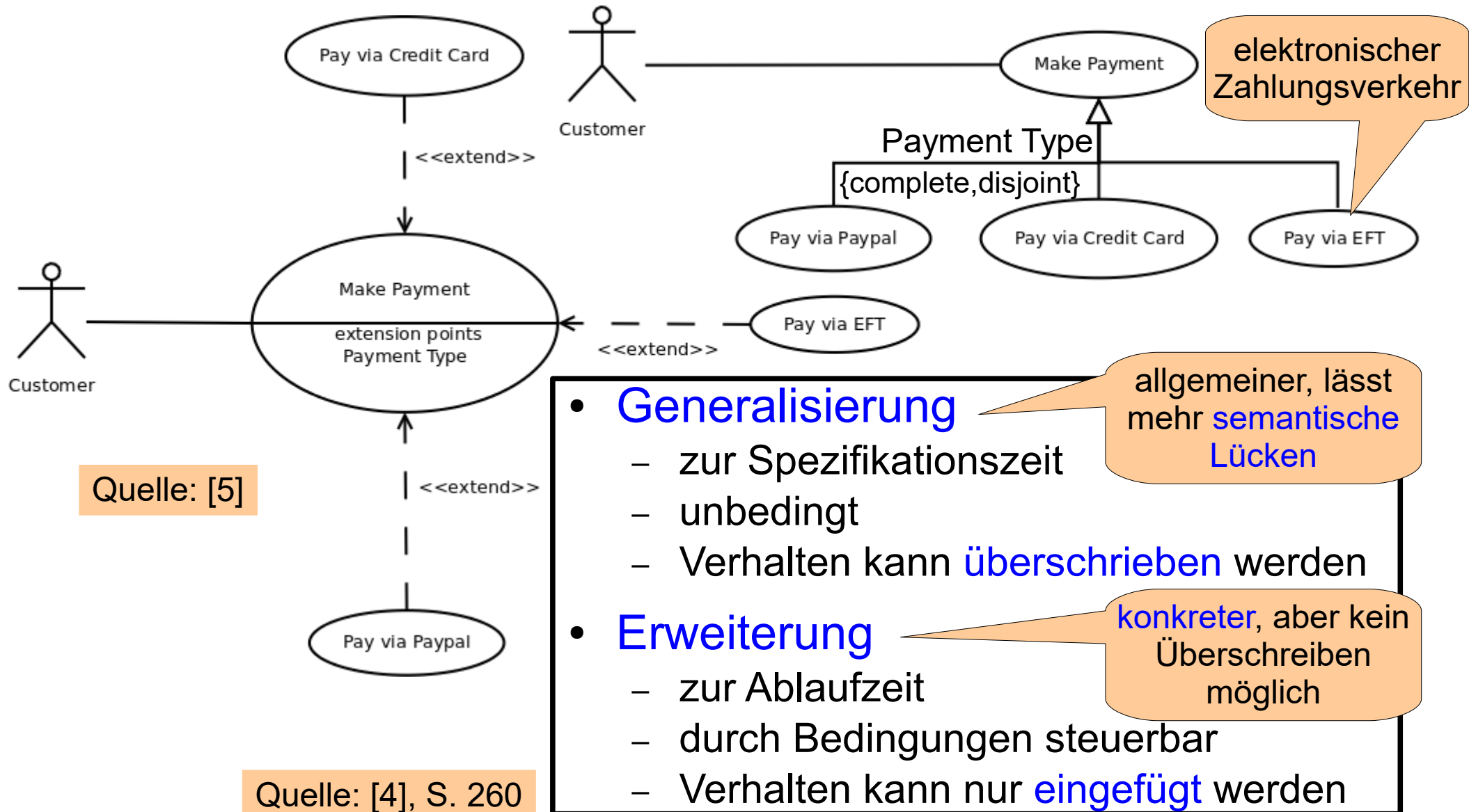
Semantik: mehrere Fragmente, deren Ausführung nur vom Wahrheitswert der Bedingung abhängt [2], S. 602; nur eine Verhaltensausführung, obwohl zwei Anwendungsfälle

Vergleich *Include* vs. *Extend*

	«include»-Beziehung	«extend»-Beziehung
Notation		
Bedeutung	Ablauf von A schließt immer Ablauf von B ein.	Ablauf von A kann, muss aber nicht durch Ablauf von B erweitert werden.
Wann wird die Beziehung genutzt?	Ablauf von B kann in verschiedenen Use-Cases genutzt werden.	A besitzt neben Normalverhalten auslagerbare Sonderfälle.
Bedeutung für die Modellierung	A ist meist unvollständig und wird erst durch Inklusion B zu einem vollständigen Ablauf.	A ist meist vollständig und kann durch B optional erweitert werden.
Abhängigkeiten	A muss B bei der Modellierung berücksichtigen. B wird unabhängig von A modelliert, um die Nutzung durch weitere Use-Cases sicherzustellen (Wiederverwendbarkeit), B muss in sich nicht vollständig sein („B weiß nicht, durch wen er inkludiert wird“).	A muss durch Angabe von Erweiterungspunkten auf die Erweiterung durch B vorbereitet werden. B wird in sich vollständig und unabhängig von A modelliert („B weiß nicht, wen er erweitert“).

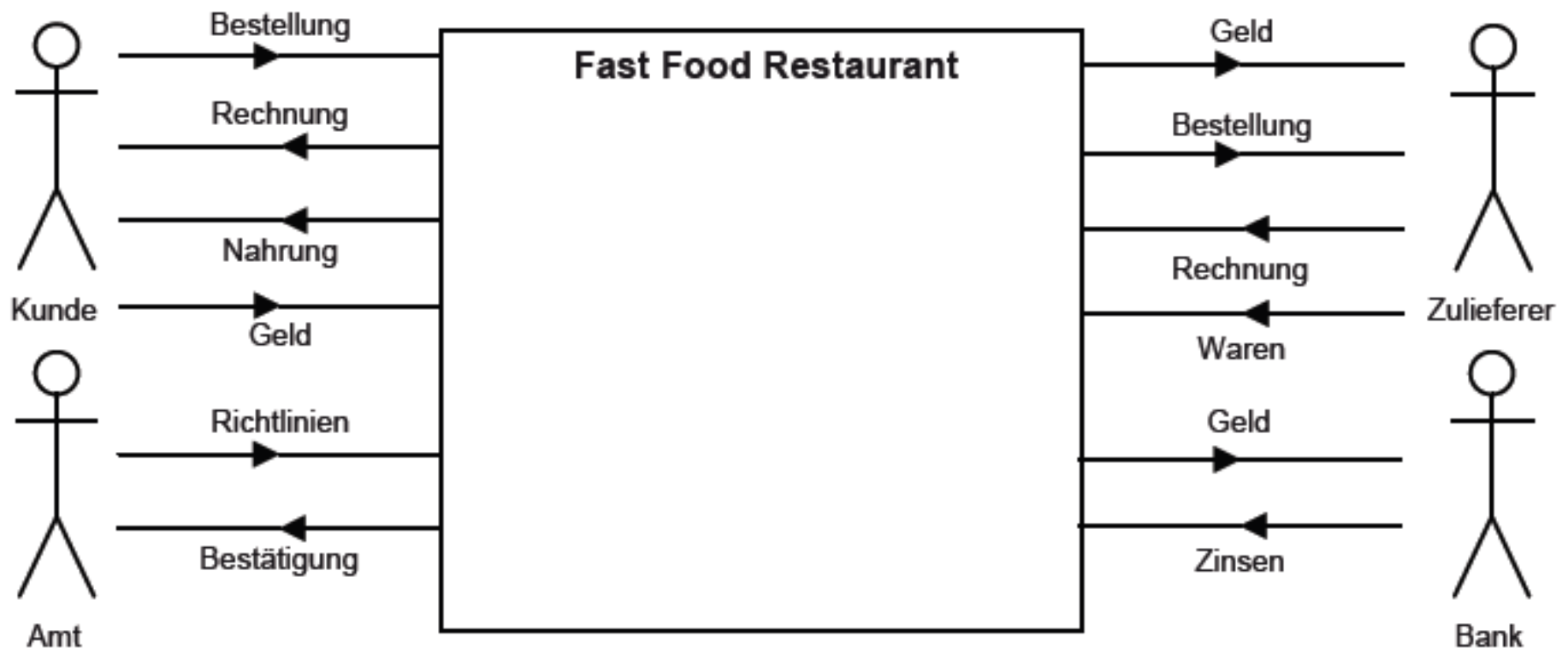
Quelle: [3], S. 262

Modellierungsvarianten: Generalisierung vs. Erweiterung von Anwendungsfällen



Kontextdiagramm

- Anwendungsfalldiagramm **ohne** Anwendungsfälle
- seit 1980ern in der **Strukturierten Analyse** [4] benutzt



Quelle: [3], S. 255

Zusammenfassung

- **Was** leistet mein System für seine Umwelt?
(Nachbarsysteme, Nutzer)
- **Anwendungsfall**
 - Teil des Systemverhaltens als *Black Box*
- **System**
 - Container für Anwendungsfälle
- **Akteur**
 - immer extern, nicht weiter verfeinert
 - Person oder anderes System
- **Include**: unbedingte **Einbindung** eines anderen Anwendungsfalls
- **Extend**: bedingte **Erweiterung** eines Anwendungsfalls

Literatur

- [1] OMG Unified Modeling Language™ (OMG UML), Infrastructure Version 2.4.1, August 2011, Download am 27.05.2014, <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>
- [2] OMG Unified Modeling Language™ (OMG UML), Superstructure Version 2.4.1, August 2011, Download am 27.05.2014, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>
- [3] Chris Rupp, Stefan Queins: „UML 2 glasklar“, Hanser Verlag, 2012, 4. Auflage
- [4] Tom DeMarco: „Structured Analysis and System Specification“, Prentice Hall, 1979
- [5] Stack Overflow: „Use case generalization versus extension“, Download am 3.6.2014
<http://stackoverflow.com/questions/15133595/use-case-generalization-versus-extension>