

**Methodische und Praktische
Grundlagen der Informatik (MPGI 3)
WS 2008/09**

Softwaretechnik

Steffen Helke

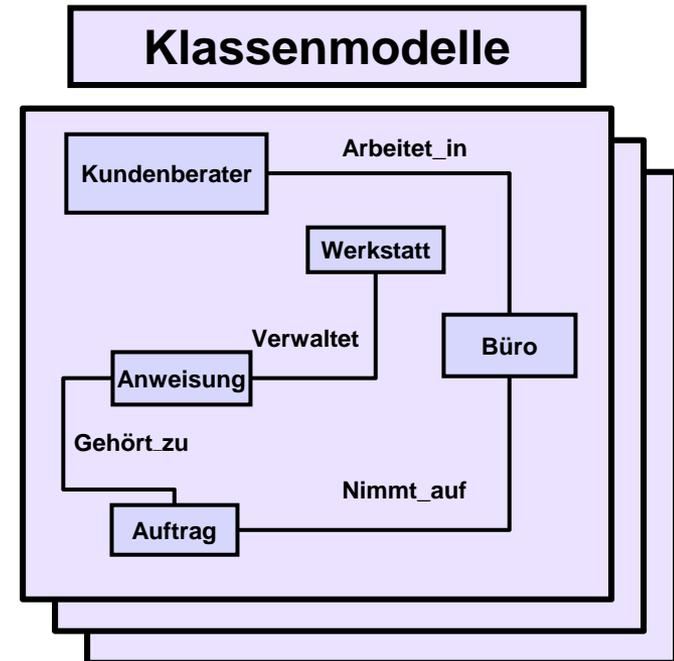
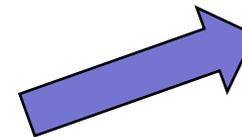
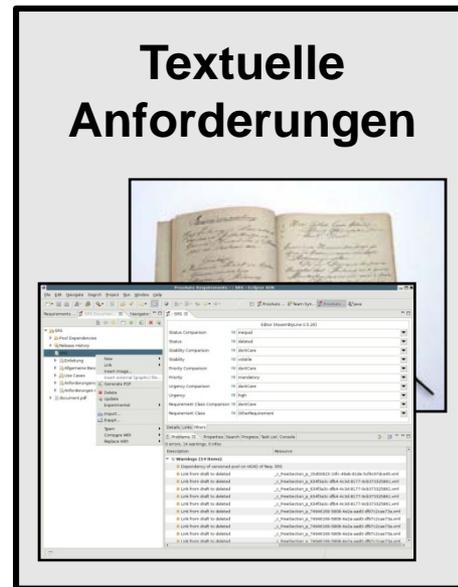
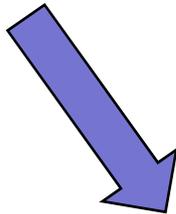
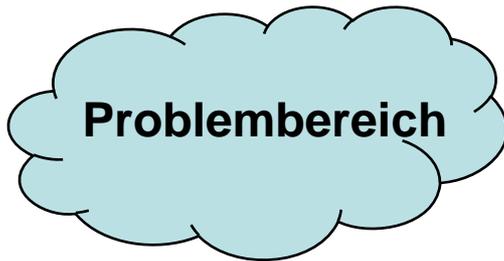
Andreas Mertgen (Organisation)

Rojahn Ahmadi, Georgy Dobrev, Daniel Gómez Esperón,
Simon Rauterberg, Jennifer Ulrich (Tutoren)

Was machen wir heute?

- **Werkzeug-Vorführung**
 - **IBM Rational Software Modeler**
- **Statische Modellierung**
 - UML-Klassendiagramme (Wiederholung)
- **Verhaltensmodelle in der UML**
 - Anwendungsfalldiagramme (**Use-Case-Modelle**)
 - **Sequenzdiagramme**

Was haben wir bisher erreicht?



Analyse, 1. Schritt: **Klassenmodell**

Anforderungsdefinition

Analyse

Schnittstellenmodell

Use-Case-Modell

Sequenzdiagramme

Aktivitätsdiagramme

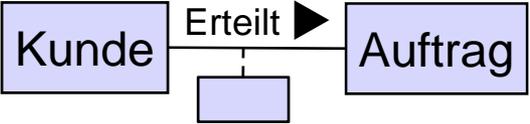
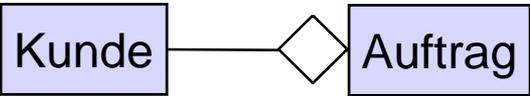
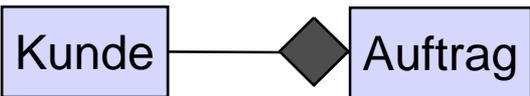
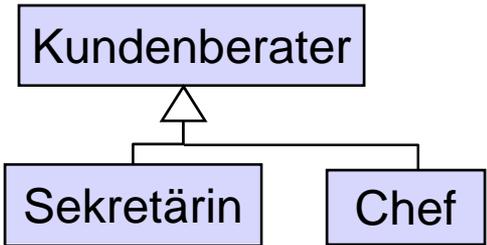
Analyse-Klassenmodell

Vor- und Nachbedingungen
von System-Operationen

Klassenmodell

Data Dictionary

Notationen für Klassenmodelle

	<p>Klassen: Namen, Attribute und Methoden</p>
	<p>Assoziationen: Namen, Leserichtung, Rollen Assoziationsklassen, mehrstellige Assoziationen</p>
	<p>Aggregationen: Ist-Teil-von-Beziehungen</p>
	<p>Kompositionen: exklusiv und existenzabhängig</p>
	<p>Multiplizitäten: Namen, Leserichtung und Rollen</p>
	<p>Generalisierung/Spezialisierung: Strukturierungsmittel auf Typebene, Schlüsselwörter: complete, incomplete, disjoint, overlapping</p>

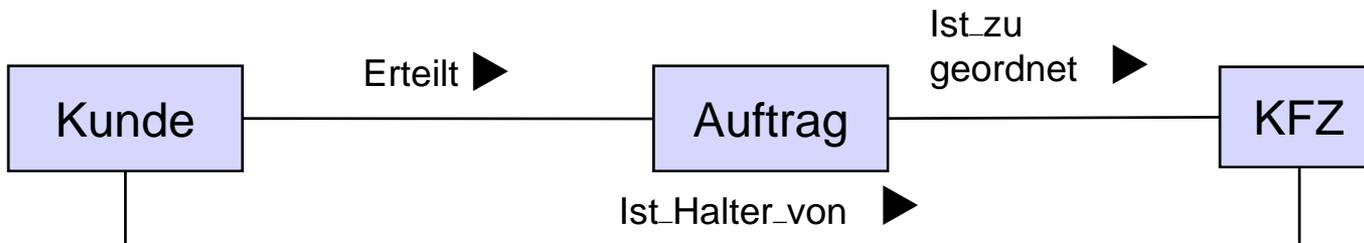
Textuelle Anforderungen an Software



- Ein Kunde erteilt einen Reparaturauftrag für sein Auto. Der Auftrag wird im Büro von einer Sekretärin oder einem Chefmechaniker aufgenommen.
- Für jeden Auftrag wird zunächst eine Reparaturanweisung erstellt. Ein freier Mechaniker druckt diese an einem Terminal in der Werkstatt aus und übernimmt damit die Reparatur.
- Nach erfolgter Reparatur ergänzt der Mechaniker die Anweisung am Terminal um die Schadensbehebung (Arbeitsstunden und verbrauchtes Material). Zusätzlich festgestellte Schäden oder fehlendes Material werden in der Anweisung vermerkt und die weitere Vorgehensweise wird ans Büro verwiesen.

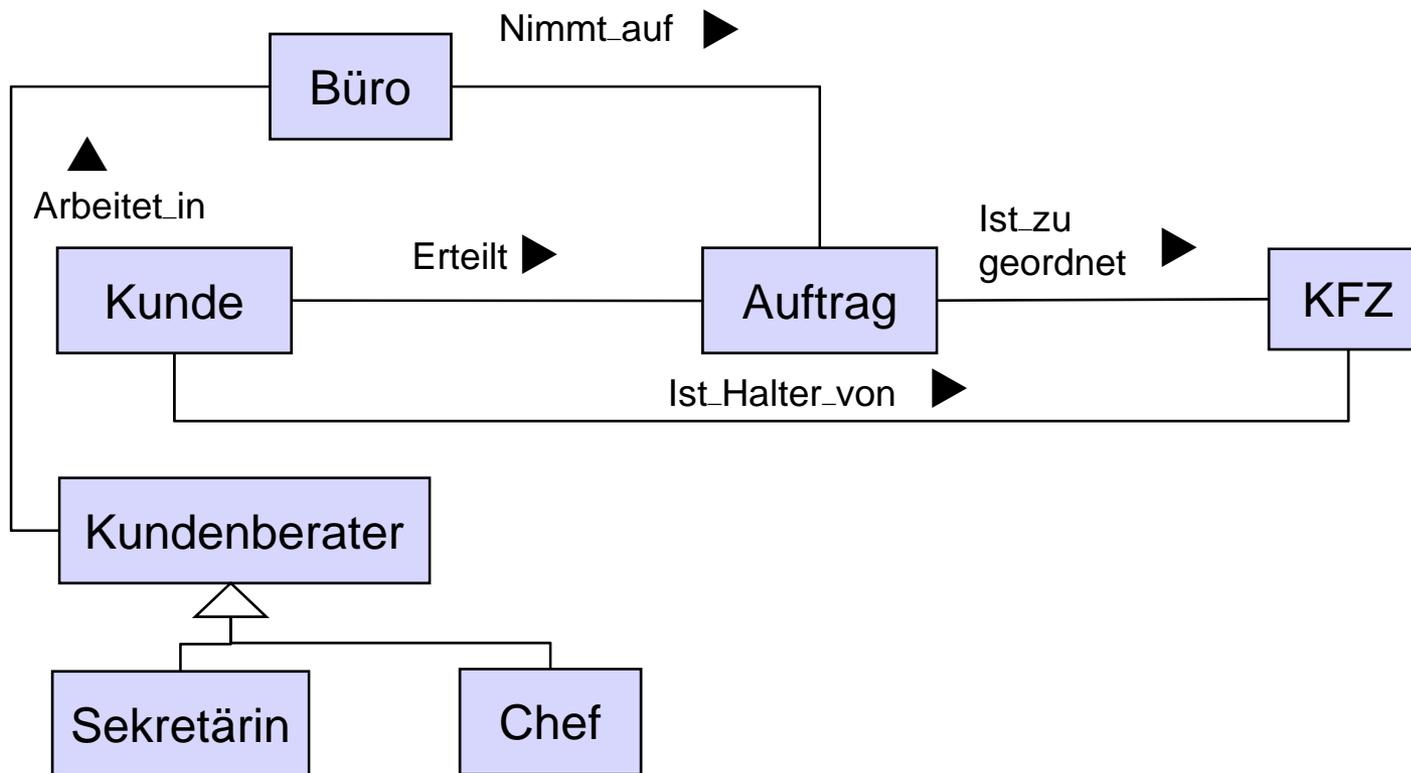
Softwaresystem für eine KFZ-Werkstatt

Ein Kunde erteilt einen Reparaturauftrag für sein Auto.



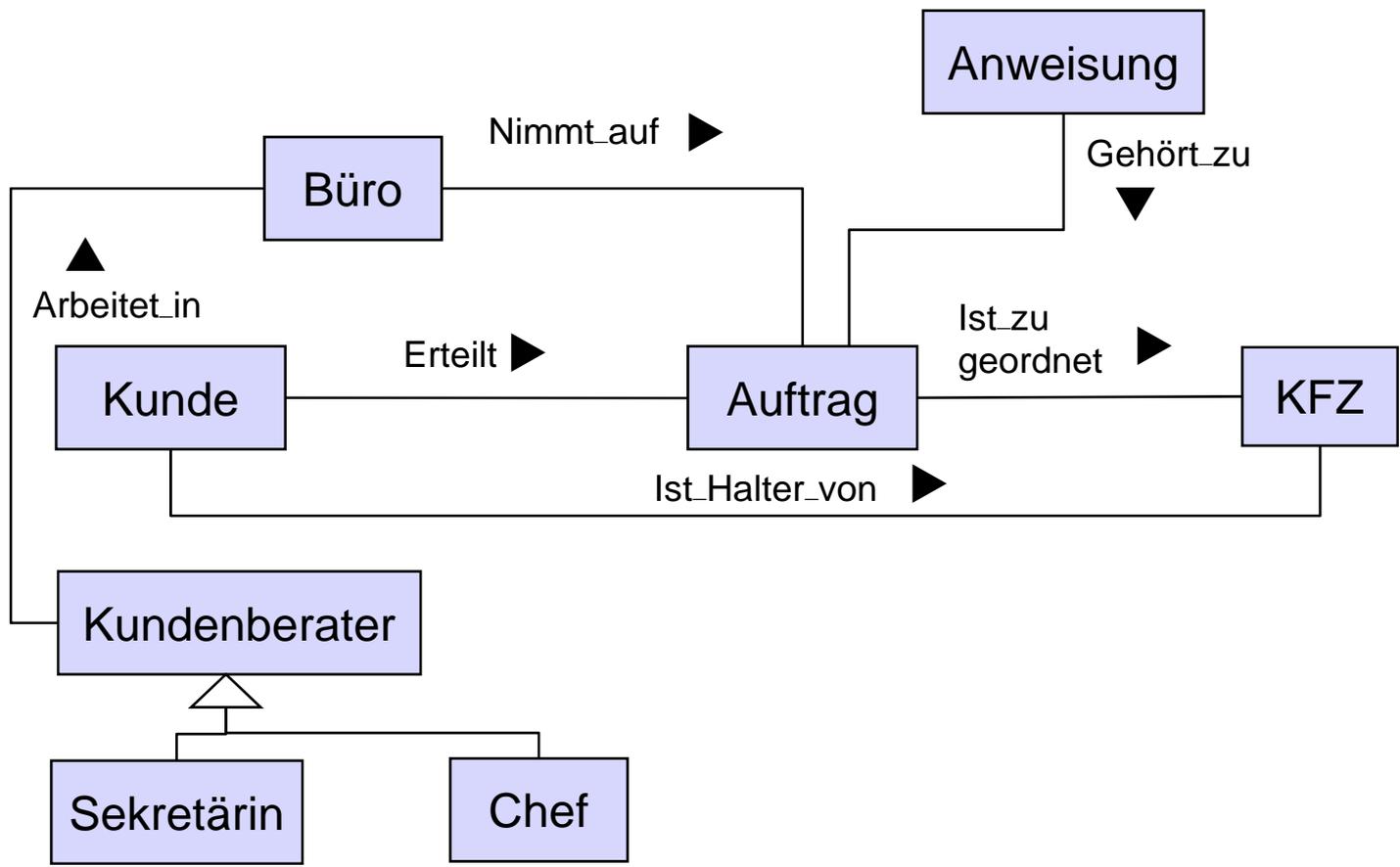
Softwaresystem für eine KFZ-Werkstatt

Ein Kunde erteilt einen Reparaturauftrag für sein Auto. Der Auftrag wird im Büro von einer Sekretärin oder einem Chefmechaniker aufgenommen.



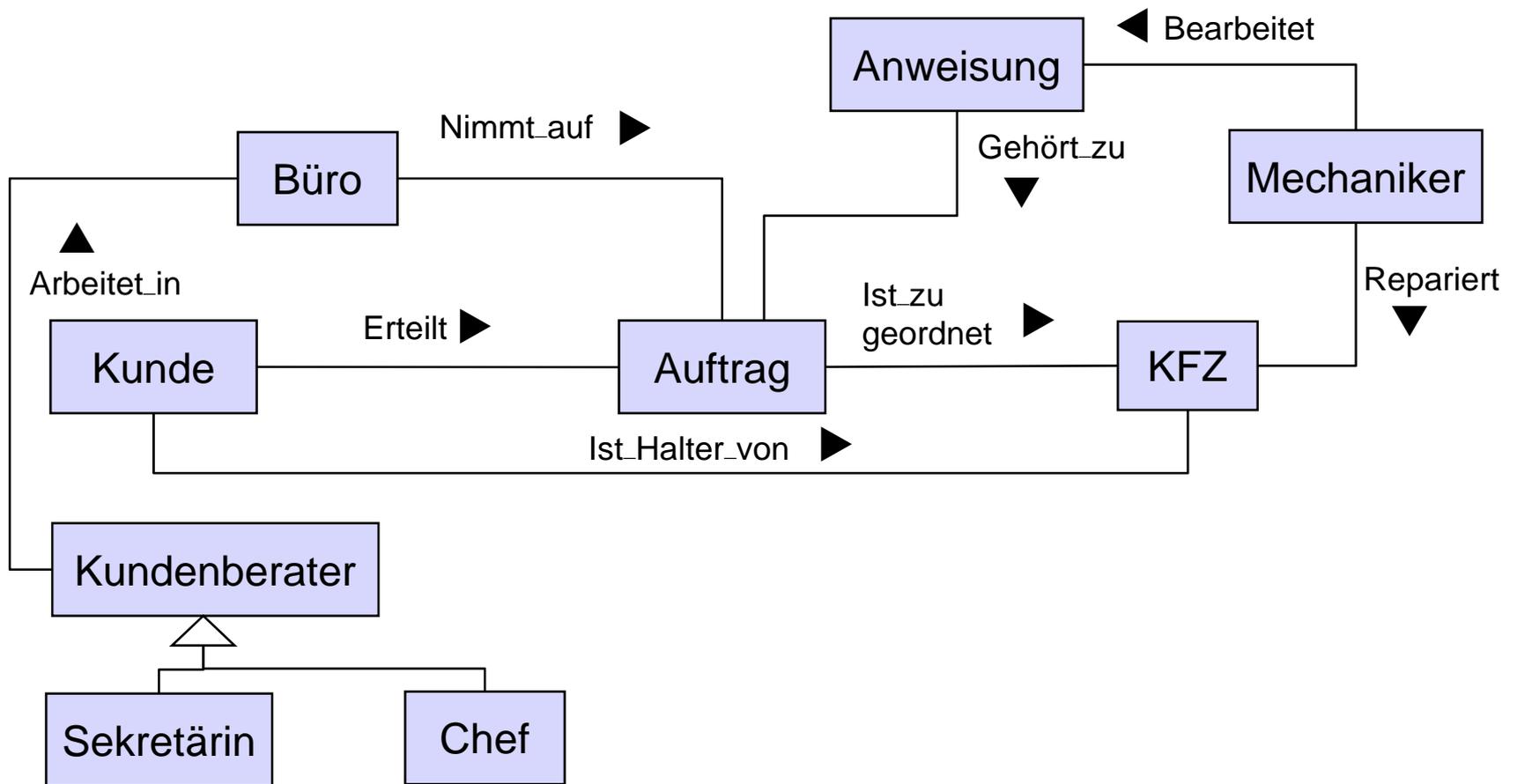
Softwaresystem für eine KFZ-Werkstatt

Für jeden Auftrag wird zunächst eine Reparaturanweisung erstellt.



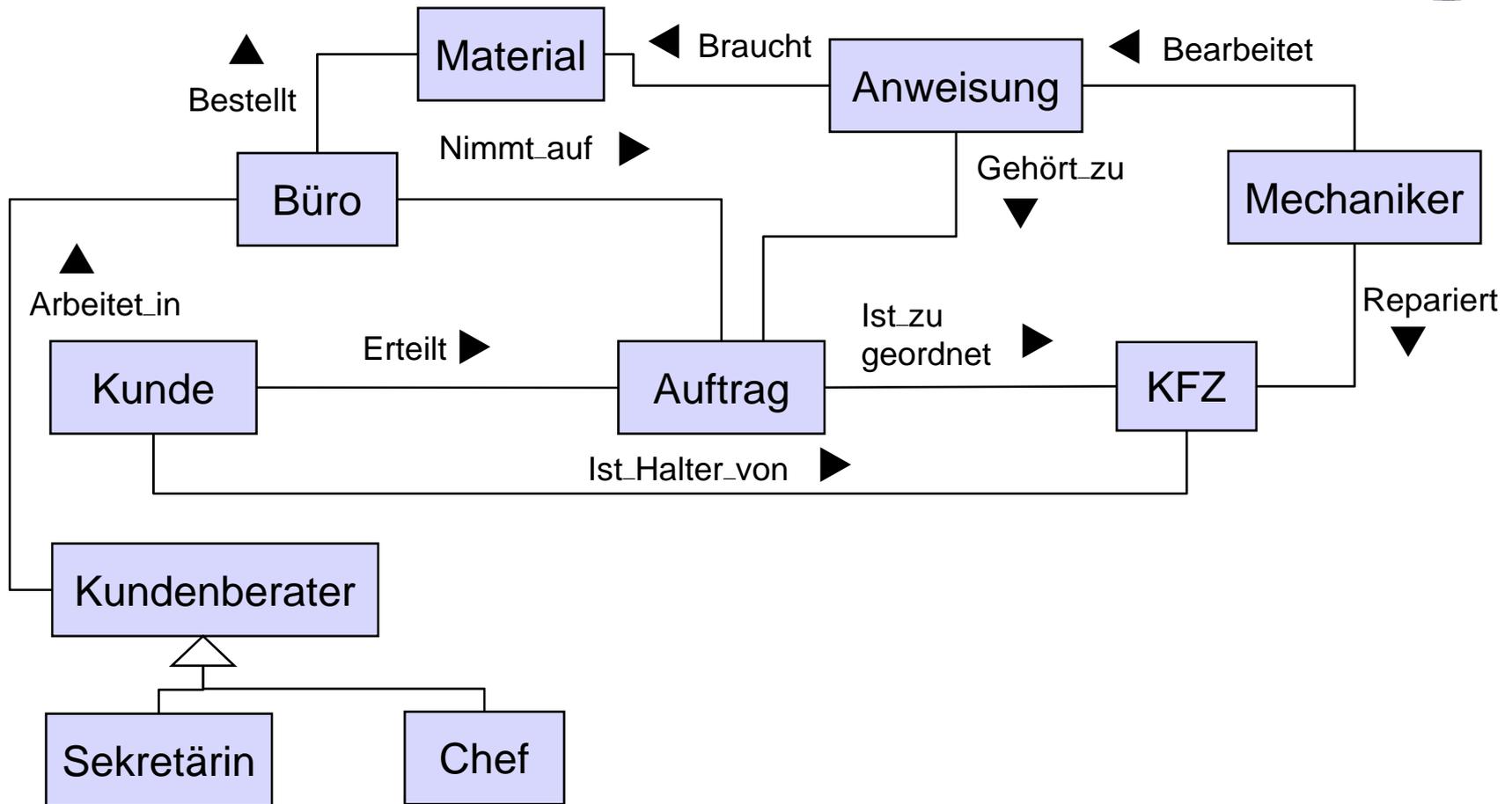
Softwaresystem für eine KFZ-Werkstatt

Ein Mechaniker druckt in der Werkstatt Anweisungen aus und übernimmt Reparatur.



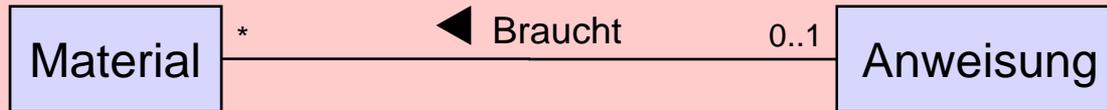
Softwaresystem für eine KFZ-Werkstatt

Fehlendes Material wird in einer Anweisung vermerkt und zur Bearbeitung ans Büro verwiesen.

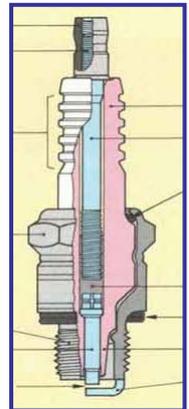
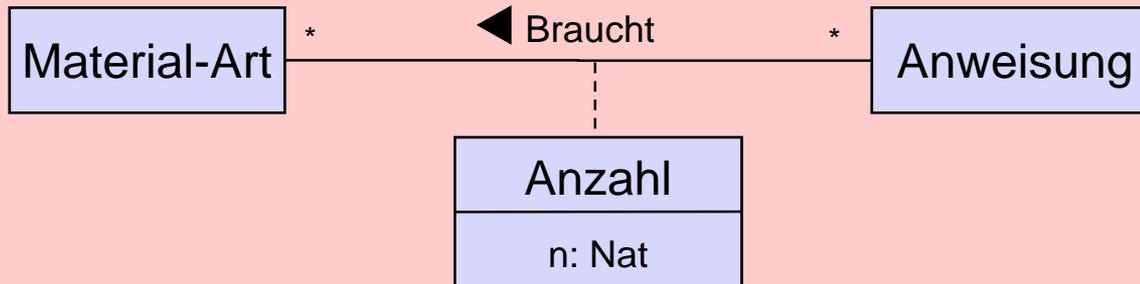


Welche Objekte modellieren wir mit einer Klasse?

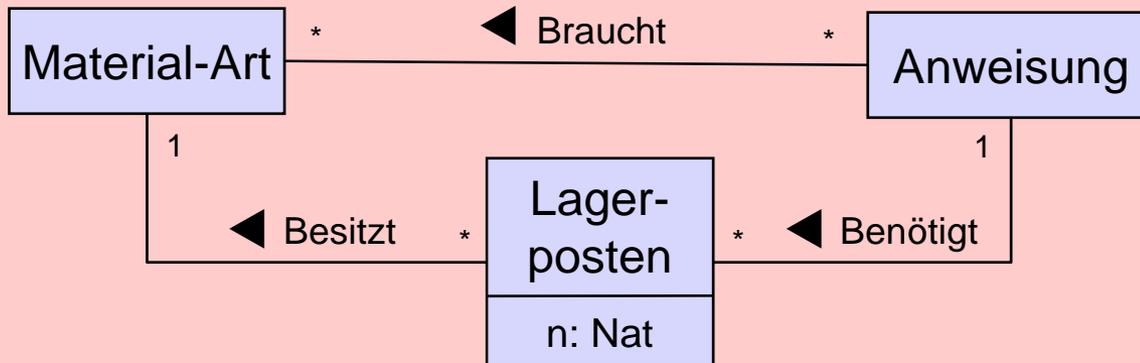
Variante 1: Objekt beschreibt „echtes“ Material

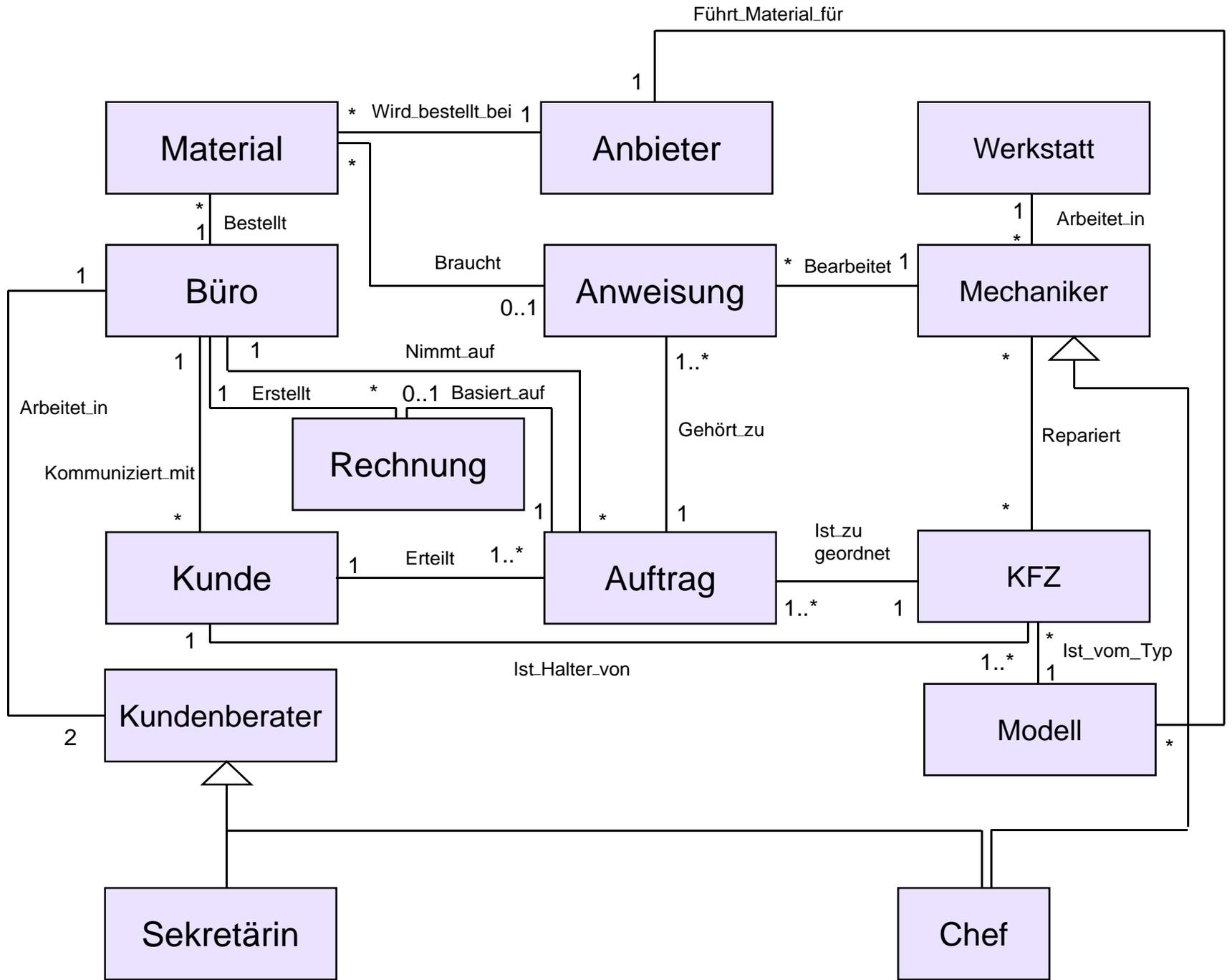


Variante 2: Objekt beschreibt die Material-Art



Variante 3: Objekt beschreibt einen Material-Lagerposten





Konventionen zur besseren Lesbarkeit

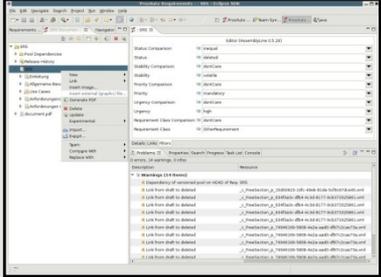
- **Klassen:**
 - Großer Anfangsbuchstabe,
 - Substantiv
- **Assoziationen:**
 - Großer Anfangsbuchstabe,
 - Zusammengesetzte Bezeichner durch „_“ trennen,
 - Verben im Singular verwenden

z.B. Überprüfen der Regel

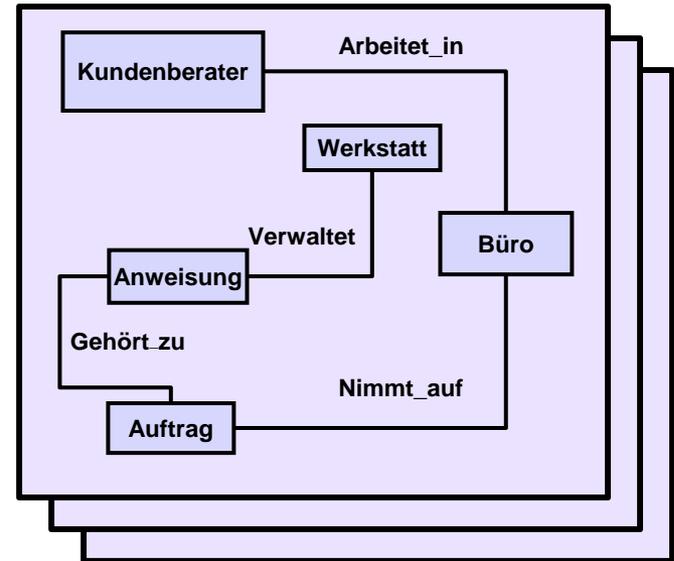
Klasse & Assoziation & Klasse = Satz

Use-Case-Modelle und Sequenzdiagramme

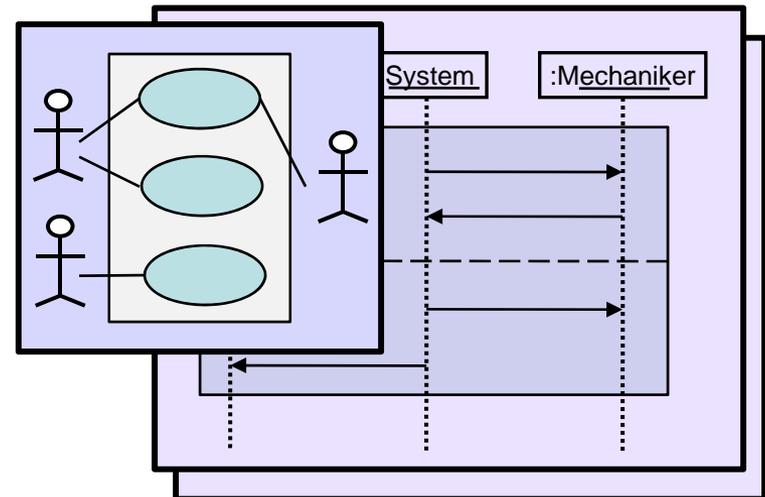
Textuelle Anforderungen



Klassenmodelle



Verhaltensmodelle



Analyse, 2. Schritt: Use-Case-Modelle

Anforderungsdefinition

Analyse

Schnittstellenmodell

Use-Case-Modell

Sequenzdiagramme

Klassenmodell

Akteure (*actors*)

Aktionsfolgen

Beziehungen:

include

extend

generalize

Diagramme

UML-Klassenmodell

Abhängigkeiten

Vererbungen

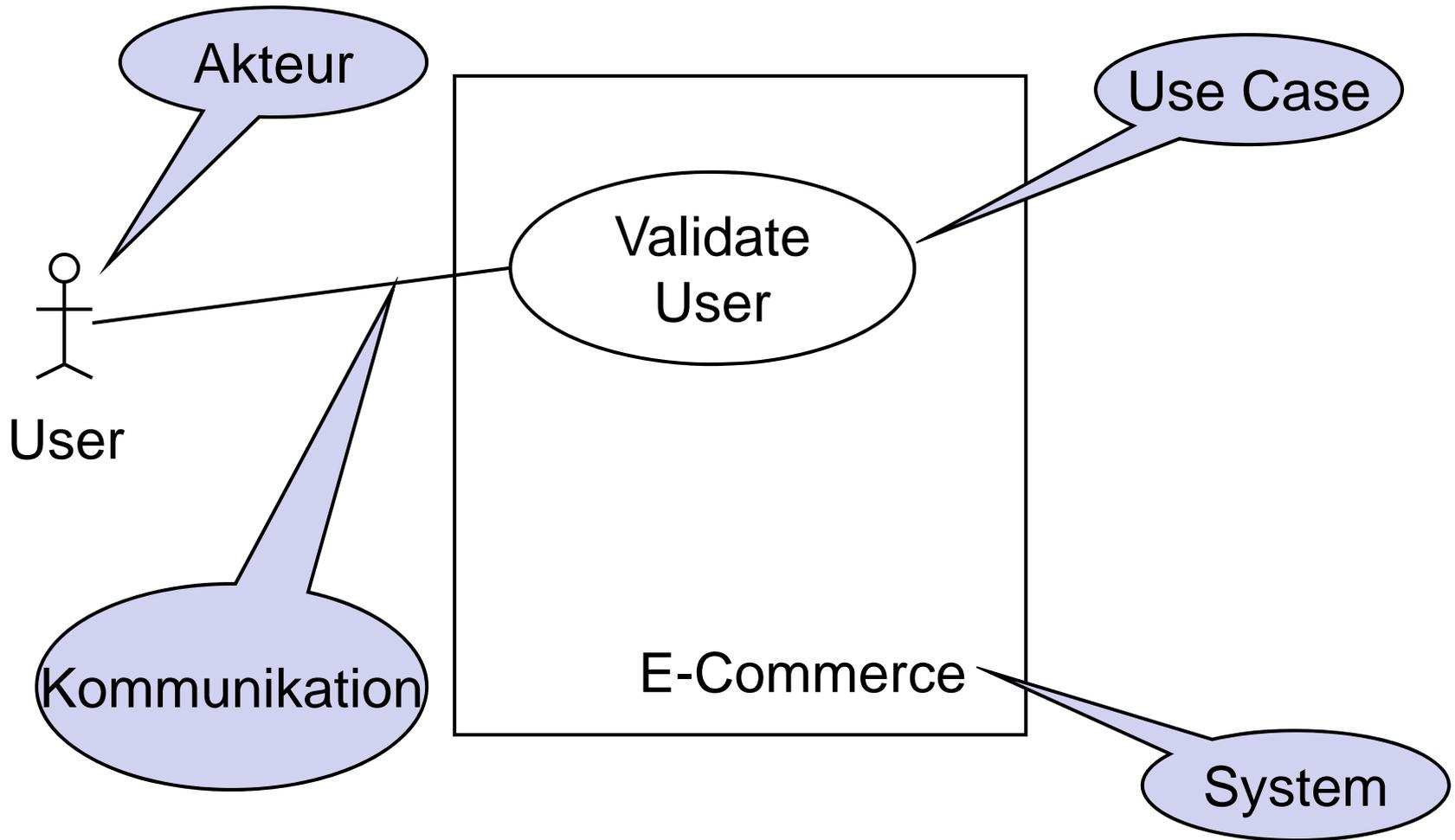
Data Dictionary

Use-Case-Modell

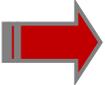
Erster Schritt, um das mögliche Verhalten des Systems zu beschreiben (Anwendungsfälle)

- **Akteure (*actors*)**
 - sind außerhalb des Systems
 - repräsentieren verschiedene Benutzerrollen
- **Anwendungsfälle (Use-Case)**
 - Menge von Aktionsfolgen
 - jede Folge beschreibt mögliche Interaktion der Akteure mit dem System (Szenario)
 - von Akteur angestoßen
 - System liefert beobachtbares Ereignis „von Wert“

UML-Notation: Use-Case-Modell



Beschreibung von Use-Cases

- Use-Cases kapseln Verhalten
 - Normalfälle
 - Ausnahmefälle
 - Konkretisierung von Use-Cases
 - mit Szenarien, die als Sequenzdiagramme dargestellt werden.
-  Sequenzdiagramme sind nächster Schritt in der Analyse!

Verbale Beschreibung von Use-Cases

Use-Case „Validate User“

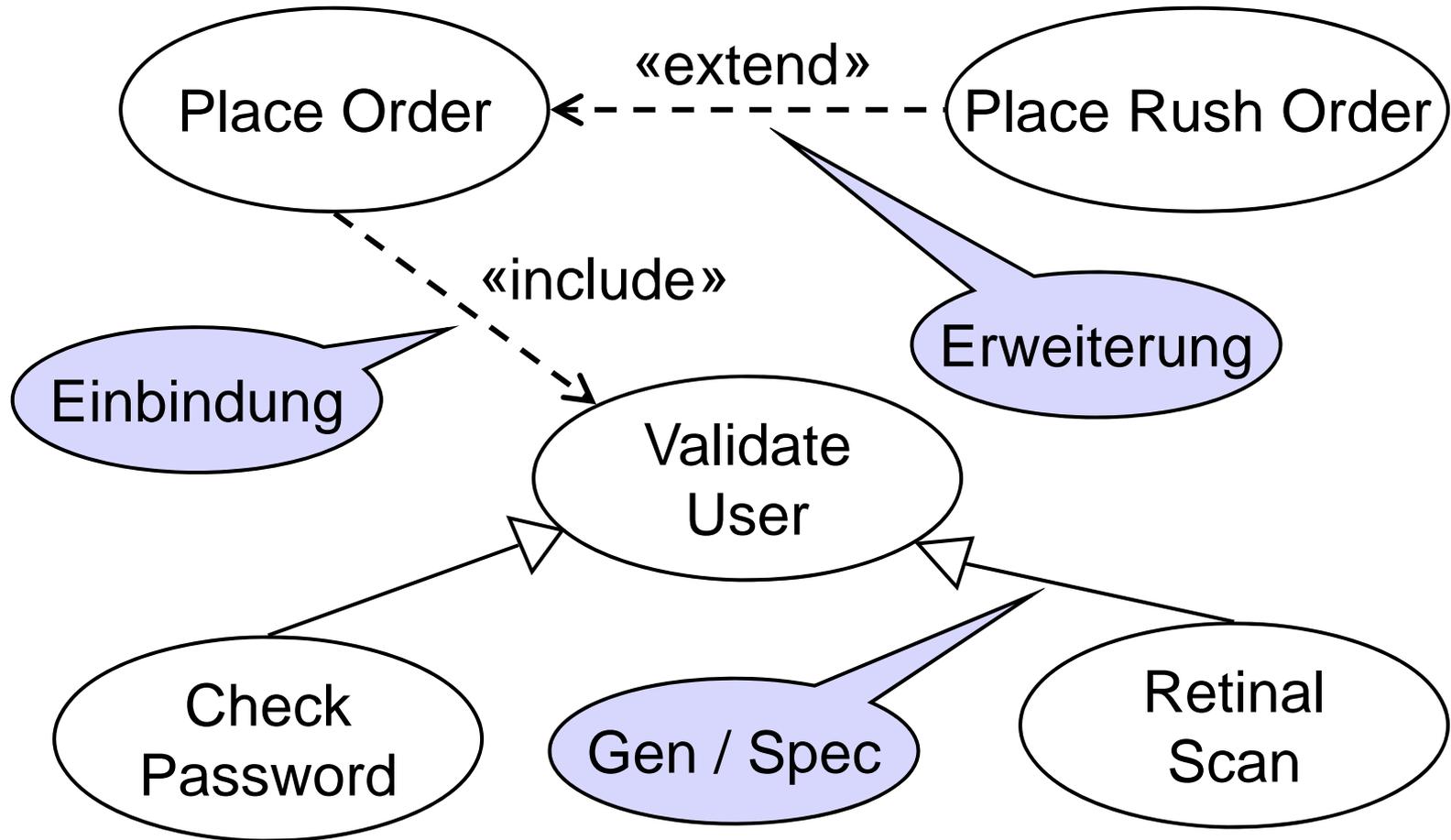
– Normalfall

- System fragt Kunden nach PIN;
- Kunde tippt PIN ein;
- System prüft Richtigkeit der PIN

– Ausnahmefälle

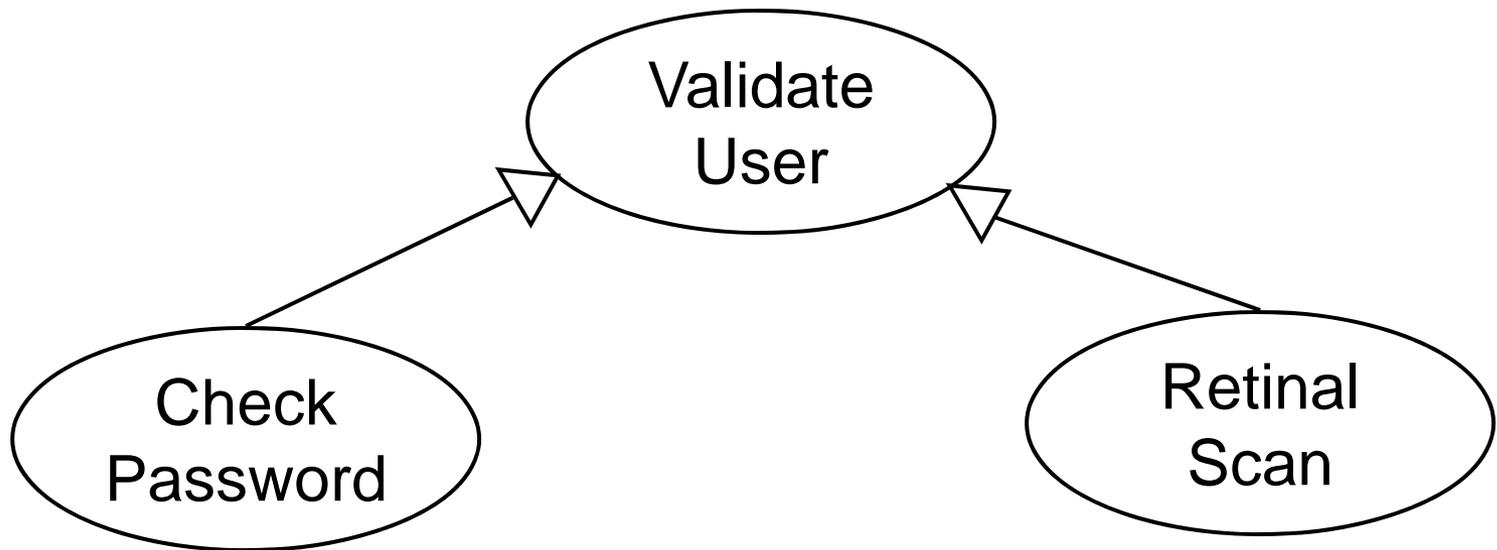
- Benutzer bricht Transaktion ab;
- Benutzer hat sich vertippt, drückt Taste CLEAR;
- Benutzer gibt dreimal falsche PIN ein

Beziehungen zwischen Use-Cases

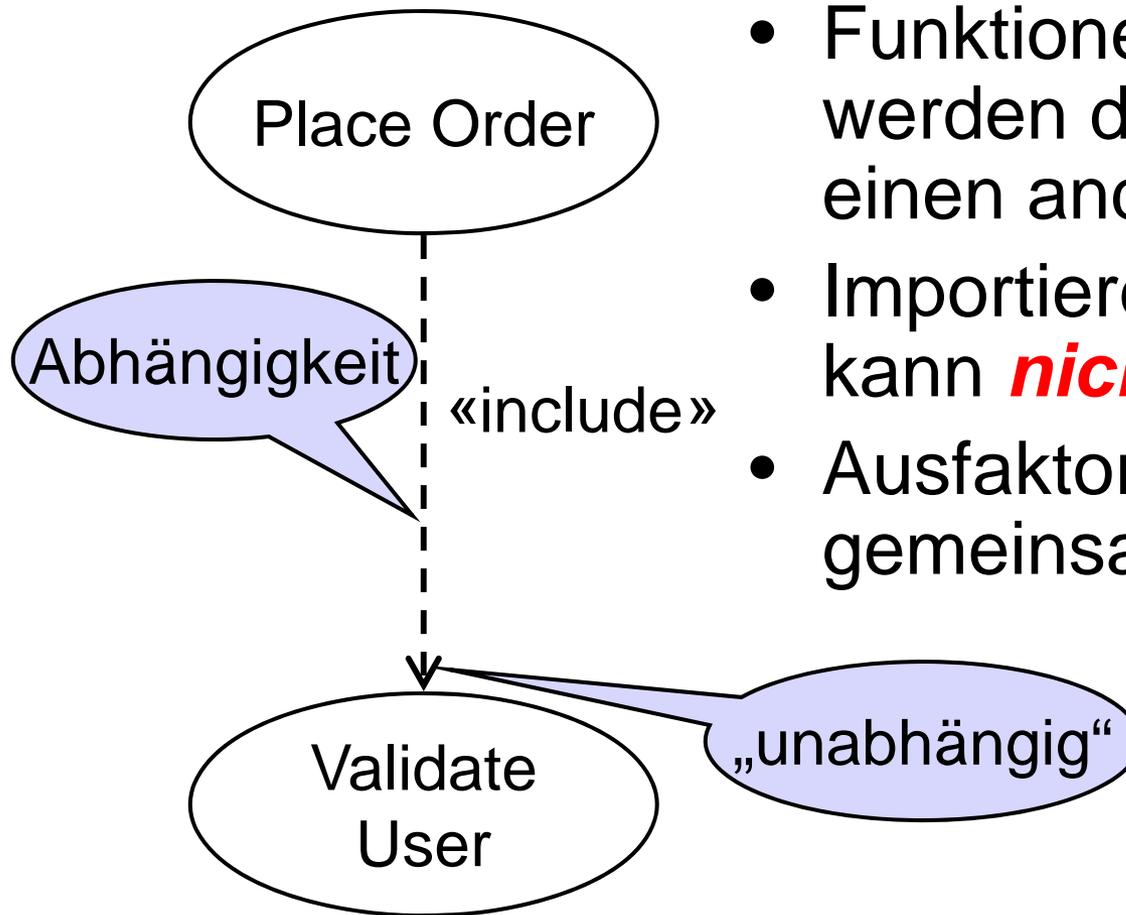


Generalisierung / Spezialisierung von Use-Cases

- Verschiedene Use Cases „realisieren“ das Verhalten der Funktionsgruppen eines allgemeineren auf unterschiedliche Art.

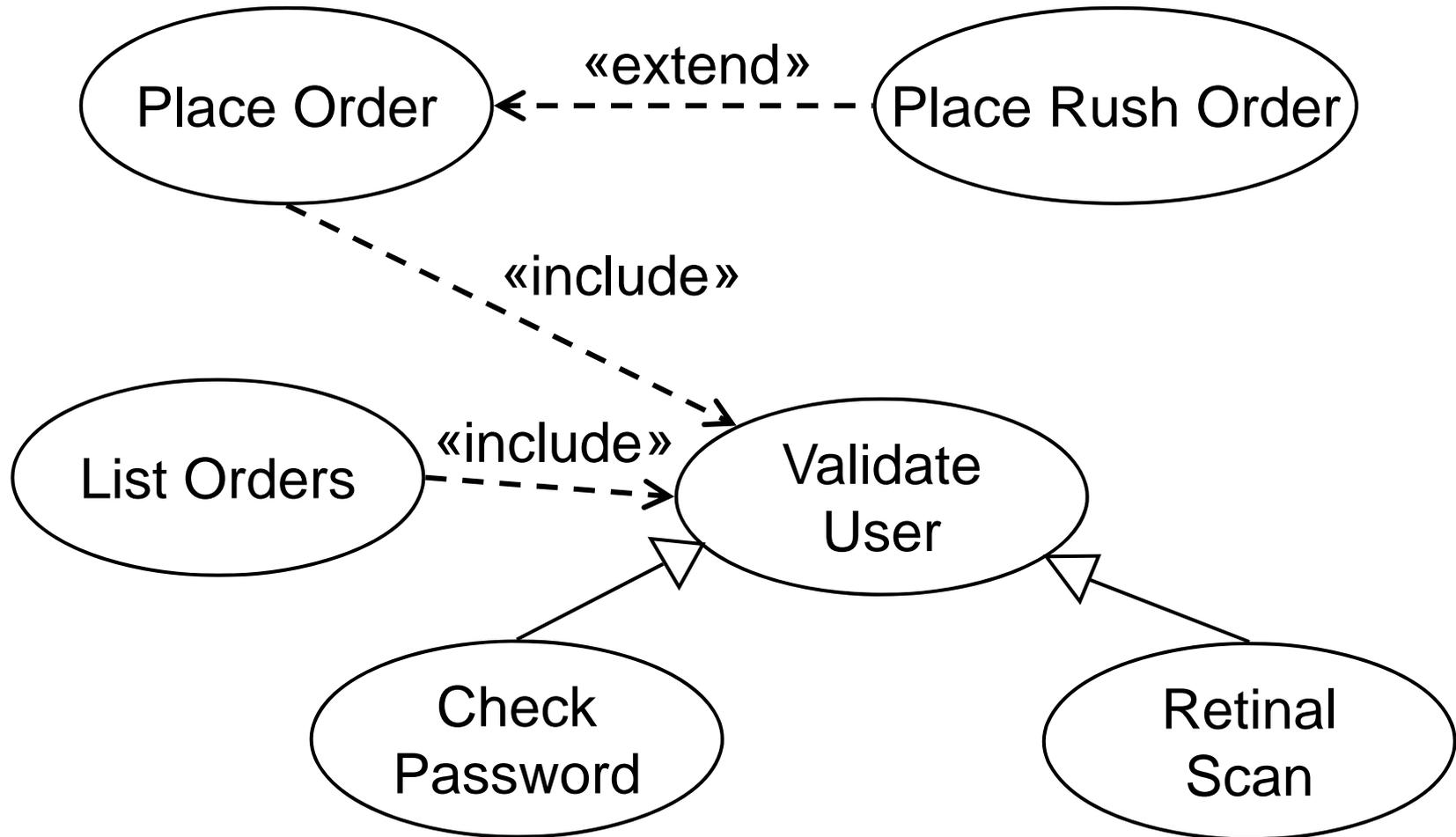


Einbindung eines Use-Case



- Funktionen eines Use Case werden durch Bezug auf einen anderen beschrieben
- Importierender Use Case kann **nicht** allein vorkommen
- Ausfaktorisierung gemeinsamen Verhaltens

«include» für gemeinsames Verhalten

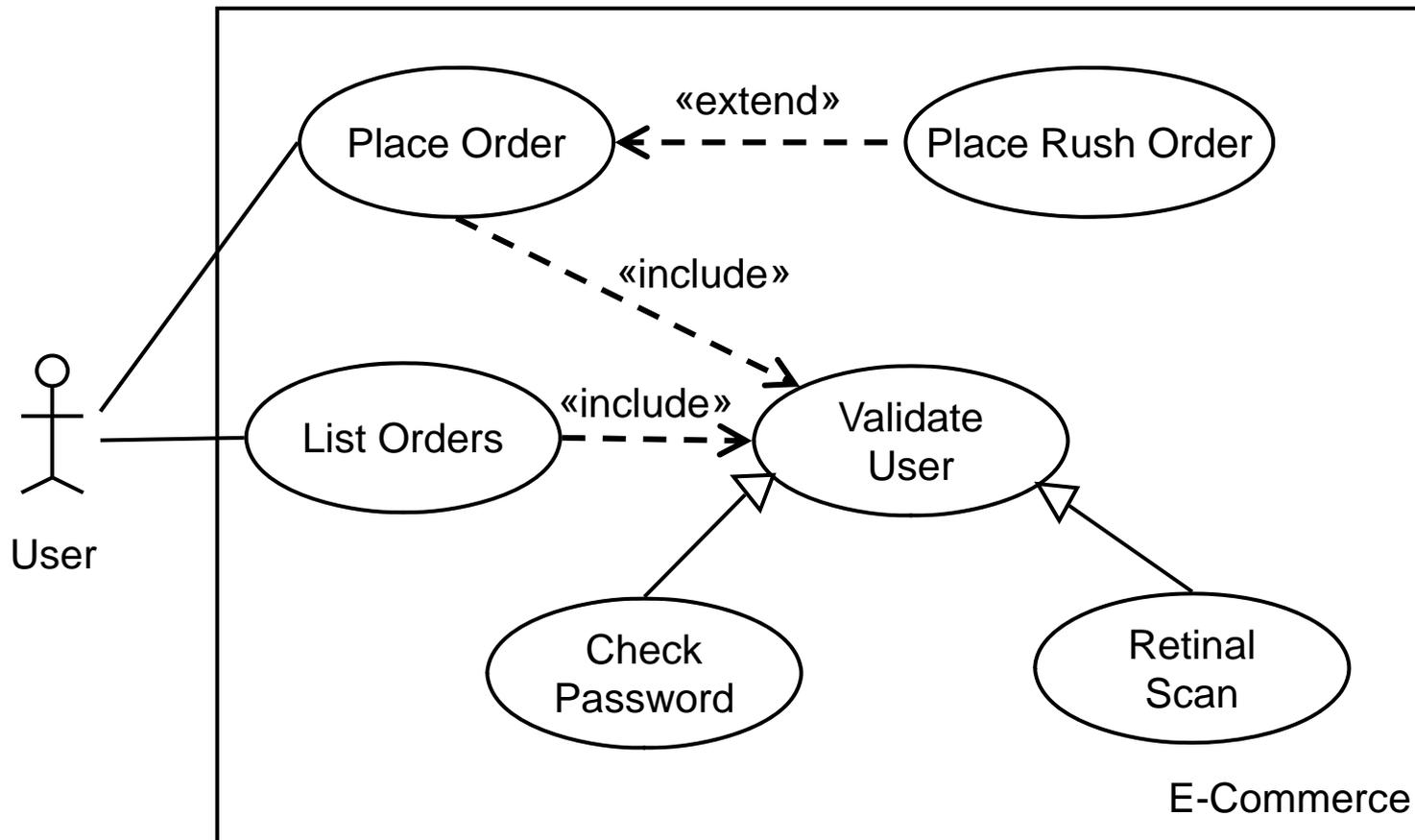


Erweiterung eines Use-Case



- Erweiternder Use Case beschreibt optionales Verhalten
- Einbindung in Aktionsfolgen des Basis-Use-Case über **extension points**
- Basis-Use-Case **kann allein** vorkommen

Akteur-Assoziationen und Use-Case-Beziehungen



Konventionen

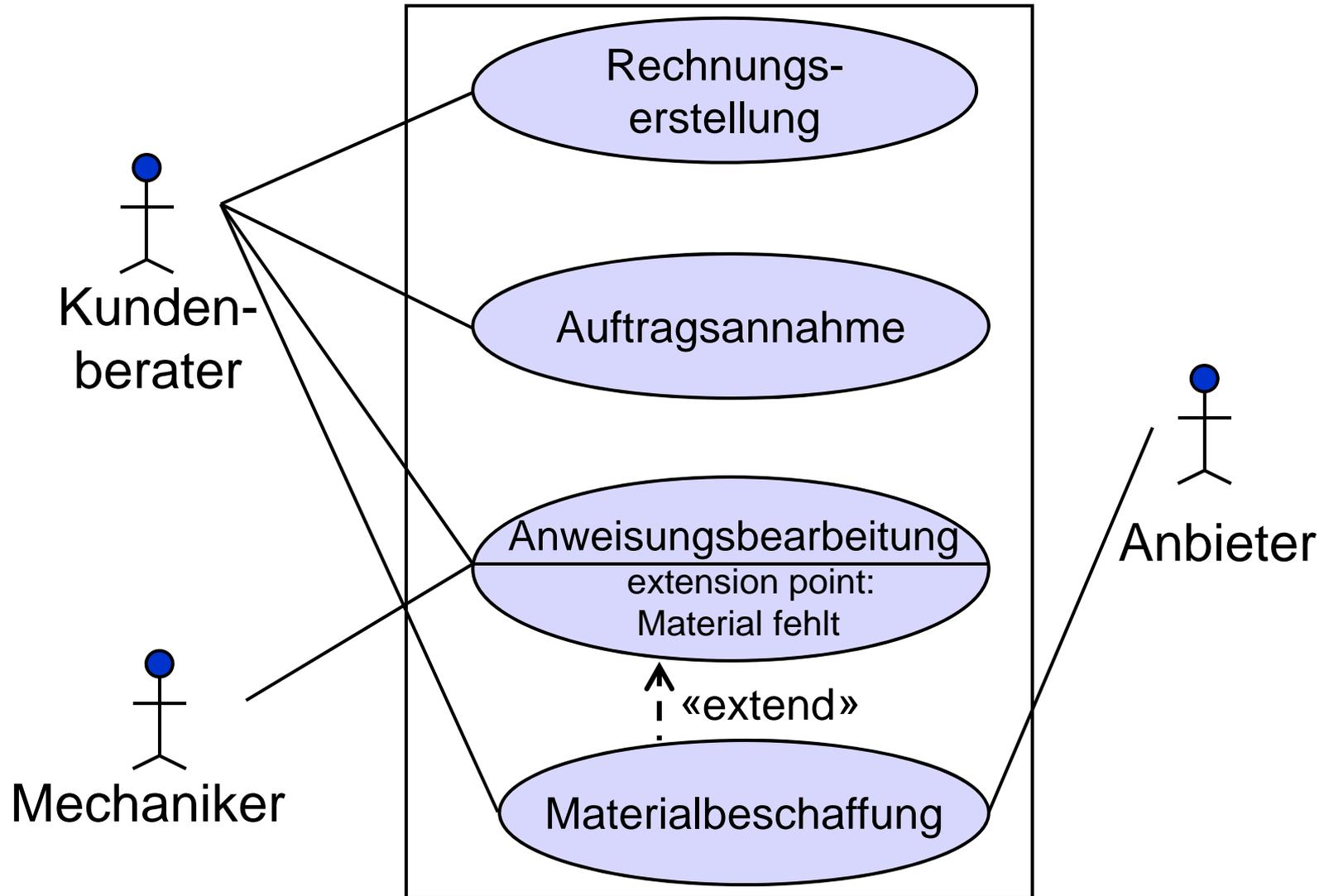
Namen für Use-Cases

- Bezeichner, die eine Tätigkeit beschreiben
- Beispiele:
 - als Substantiv (z.B. „Materialbeschaffung“)
 - oder als Verb (z.B. „Material beschaffen“)

KFZ-Werkstatt : Anforderungen (1)

- Ein Kunde erteilt einen Reparaturauftrag für sein Auto. Der Auftrag wird im Büro von einer Sekretärin oder einem Chefmechaniker aufgenommen.
- Für jeden Auftrag wird zunächst eine Reparaturanweisung erstellt. Ein freier Mechaniker druckt diese an einem Terminal in der Werkstatt aus und übernimmt damit die Reparatur.
- Nach erfolgter Reparatur ergänzt der Mechaniker die Anweisung am Terminal um die Schadensbehebung (Arbeitsstunden und verbrauchtes Material). Zusätzlich festgestellte Schäden oder fehlendes Material werden in der Anweisung vermerkt und die weitere Vorgehensweise wird an das Büro verwiesen.

Use-Case-Modell: KFZ-Werkstatt



Analyse, 3. Schritt: Sequenzdiagramme

Anforderungsdefinition

Analyse

Schnittstellenmodell

Use-Case-Modell

Sequenzdiagramme

Aktivitätsdiagramme

Klassenmodell

Analyse-Klassenmodell

Nachbedingungen

Operationen

Data Dictionary

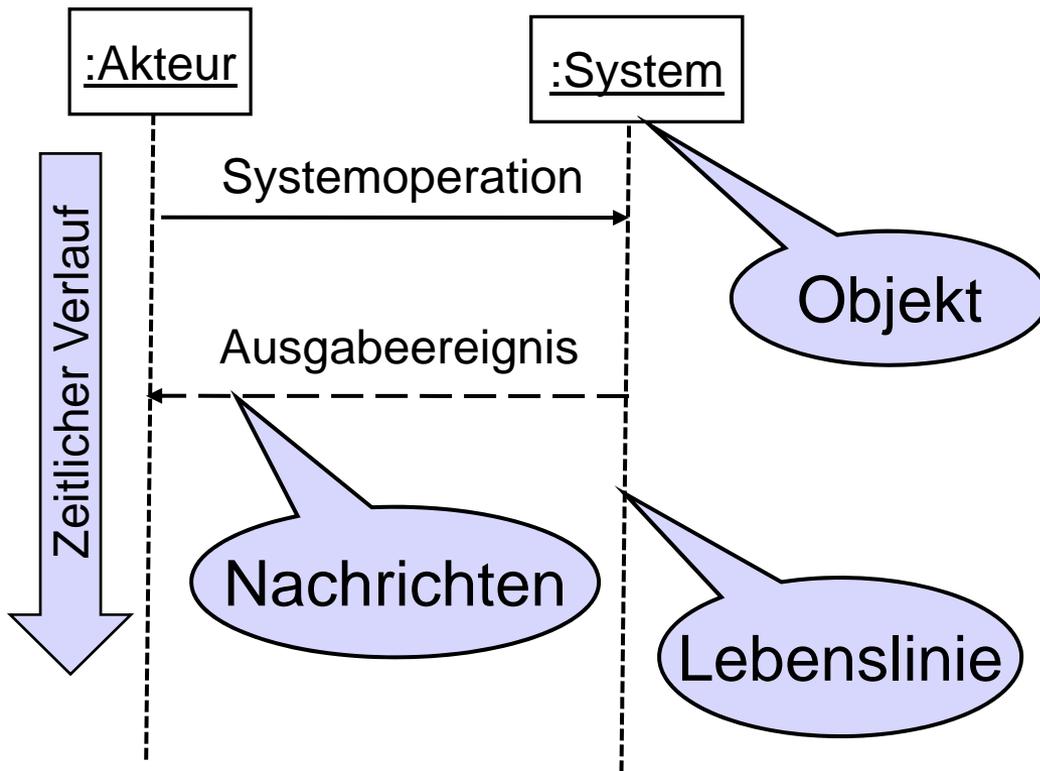
Szenarien für Use Cases:

sequentielle Abläufe

- Systemoperationen
- Ausgabeereignisse

Notationen für Sequenzdiagramme

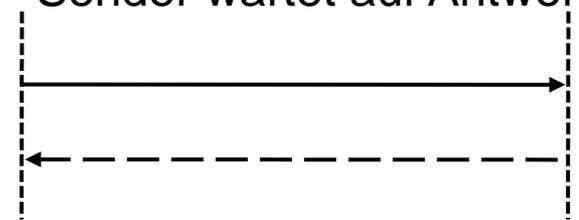
Aufbau



Kommunikation

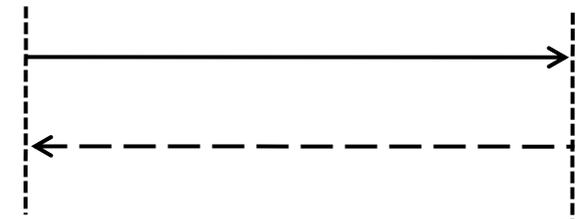
Synchron:

- Sender wartet auf Antwort



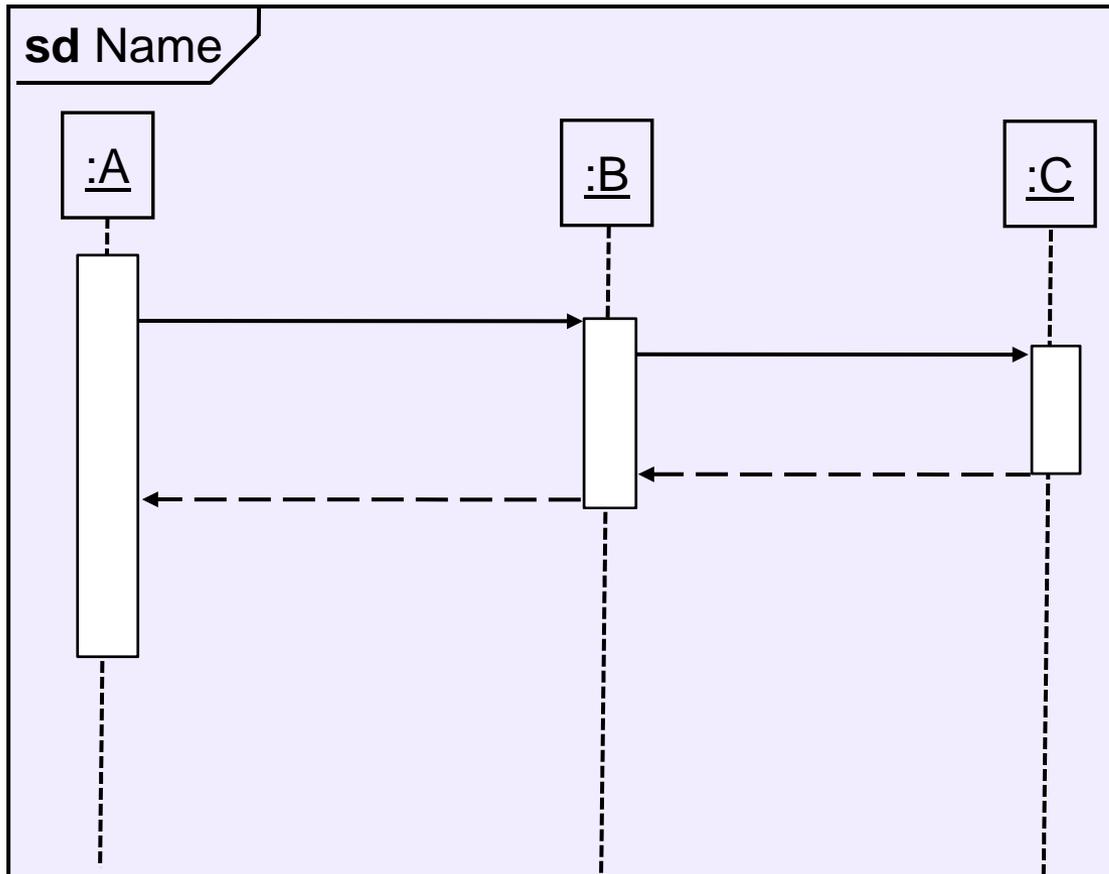
Asynchron

- Sender arbeitet weiter



Notationen für Sequenzdiagramme

Aktivitätszonen



Zweck:

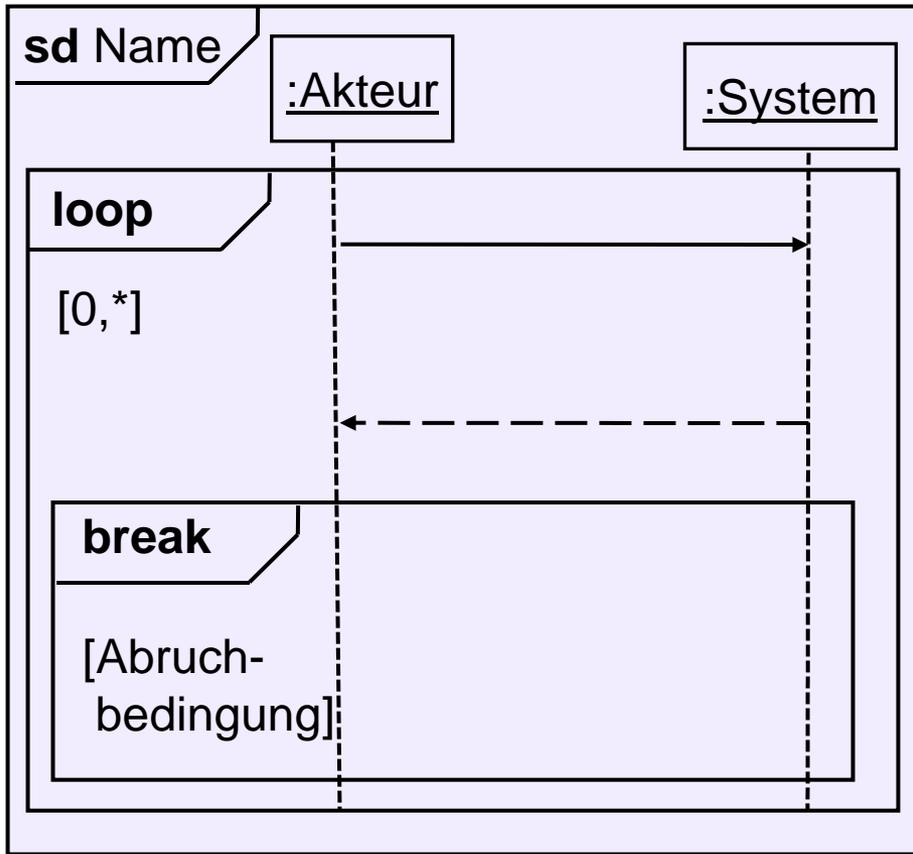
- Beschreibung von sequentiellen und geschachtelten Kontrollflüssen möglich

Hinweis:

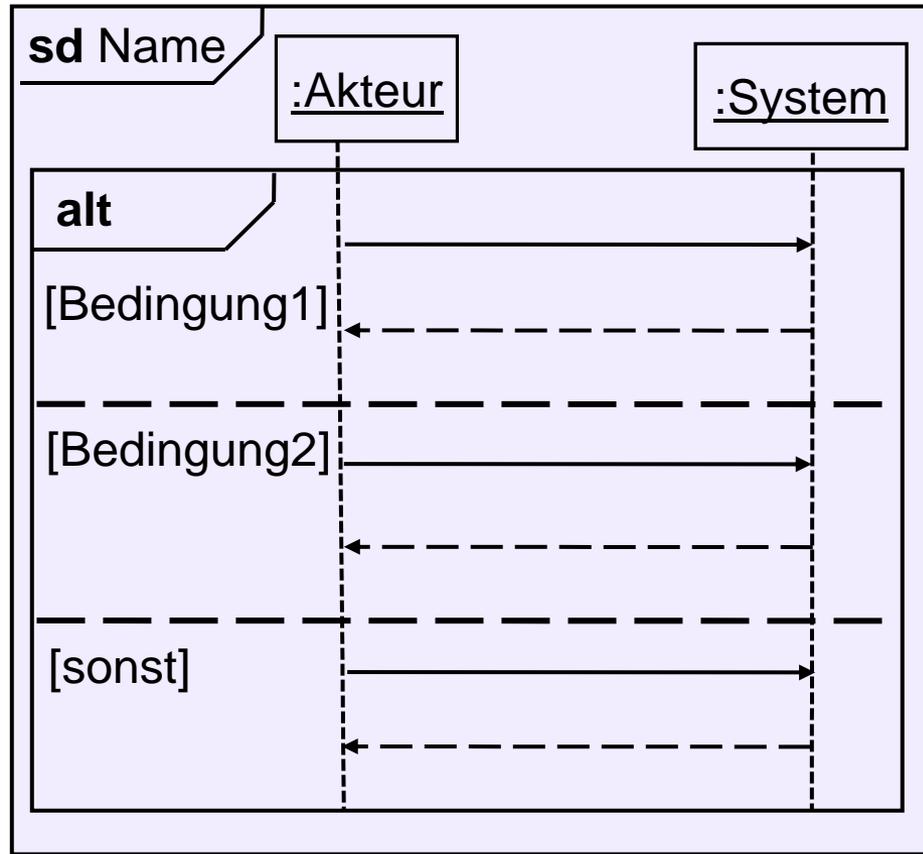
- Spielt in der Analyse zunächst keine Rolle
- wird im Entwurf aufgegriffen

Notationen für Sequenzdiagramme

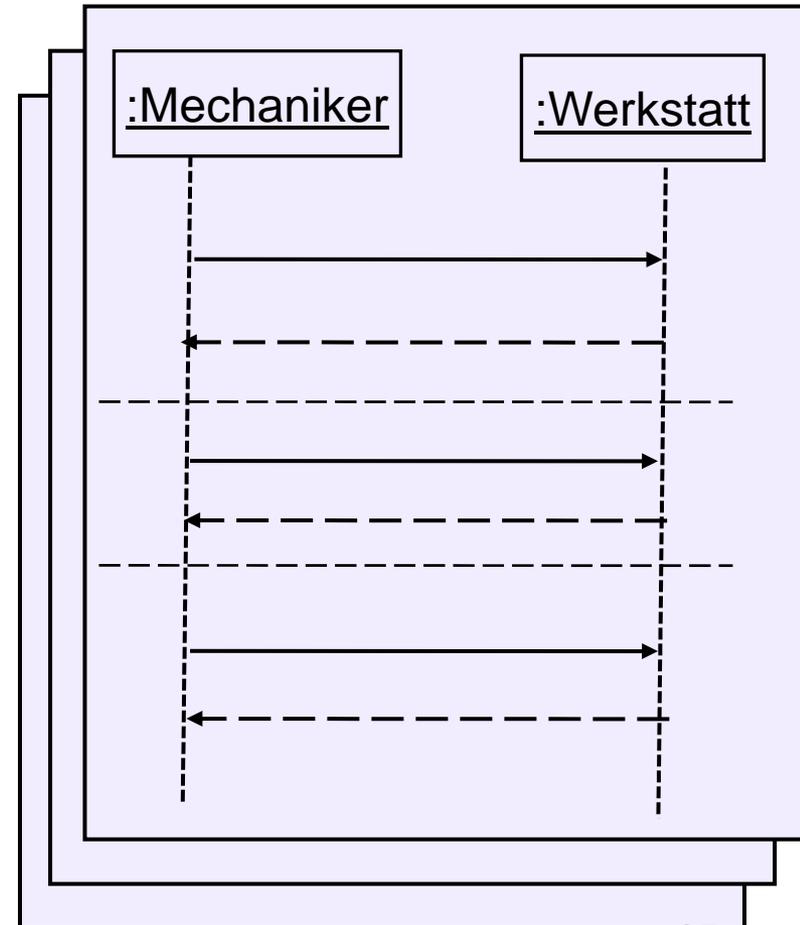
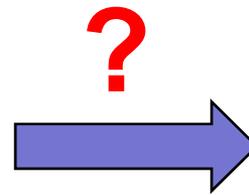
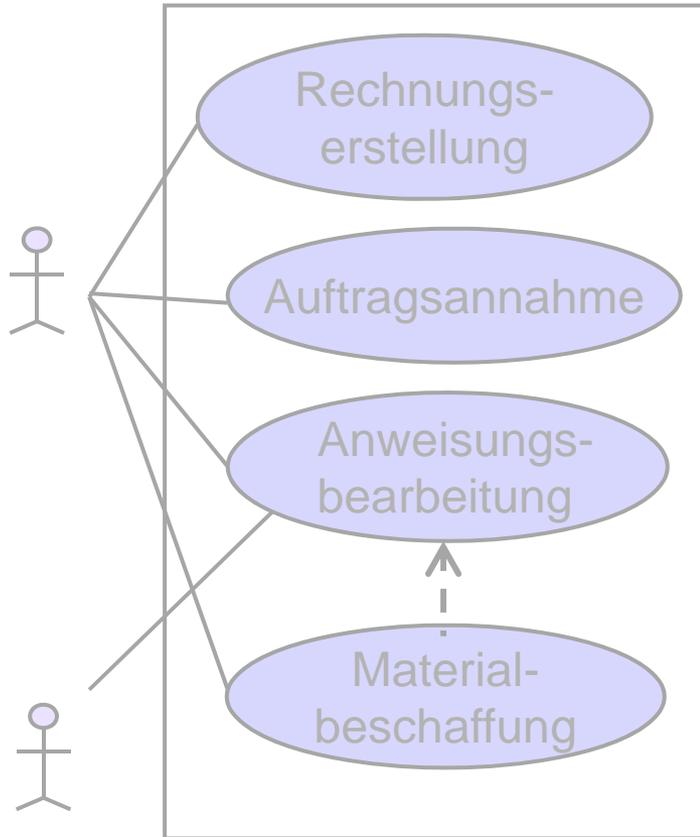
Schleifen



Alternativen



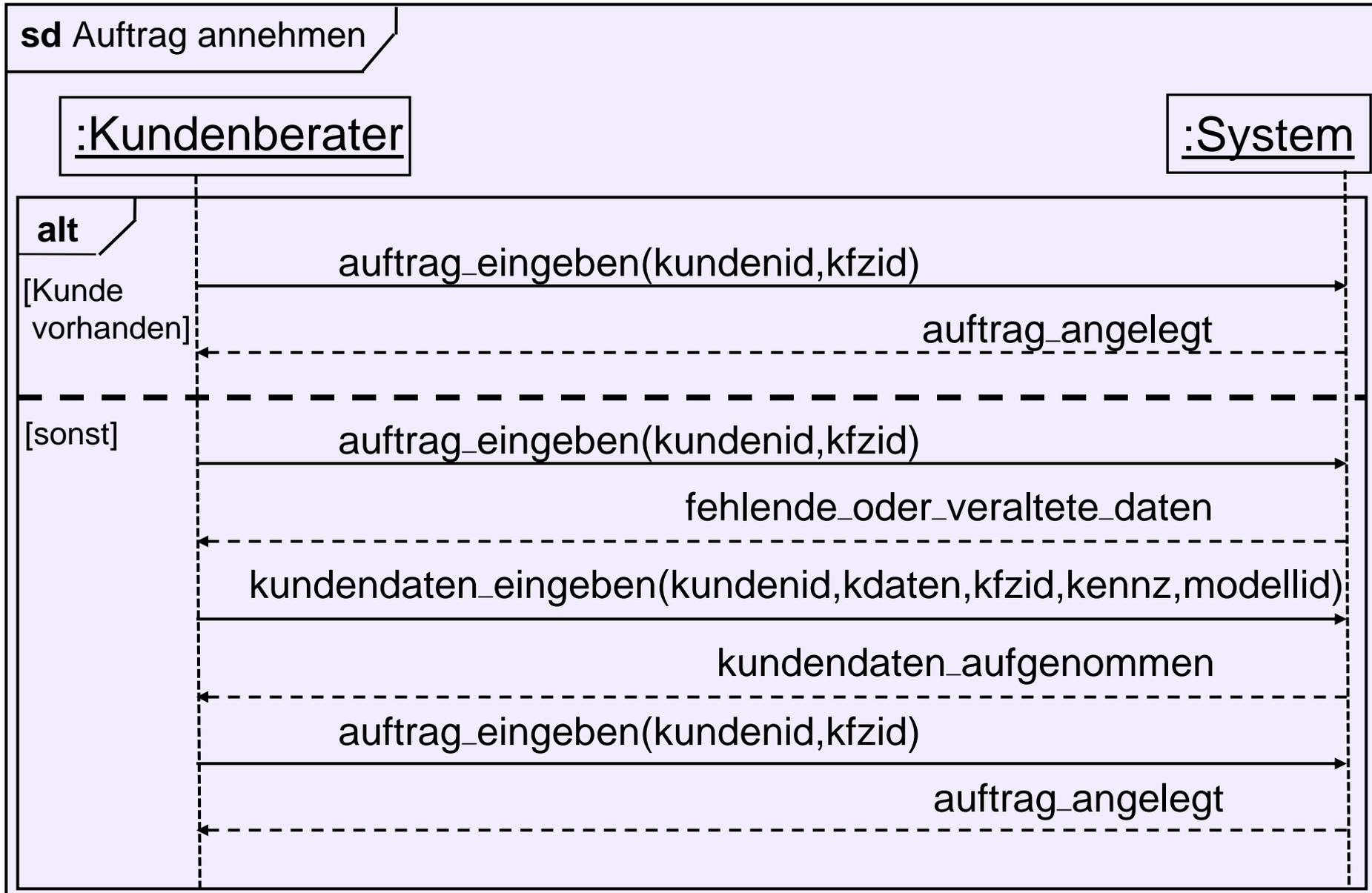
Wie kommen wir zu Sequenzdiagrammen für die Werkstatt?



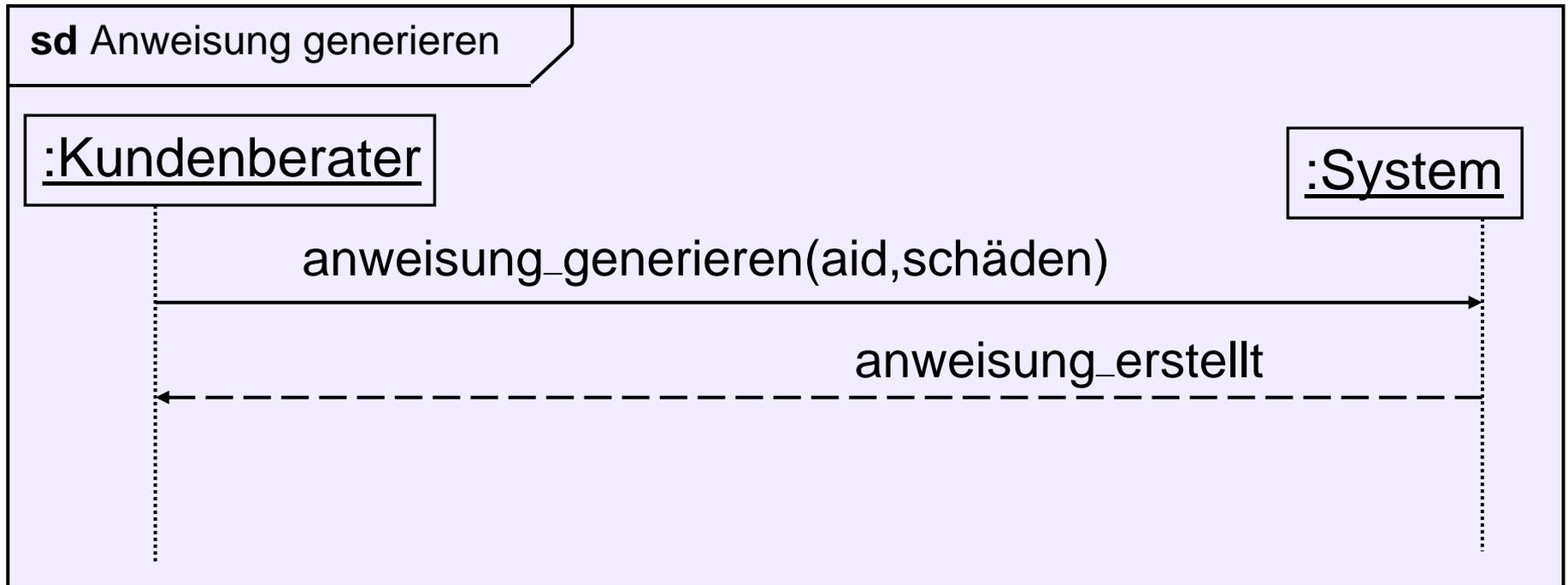
Beispiel: Szenario für Use-Case „Anweisungsbearbeitung“



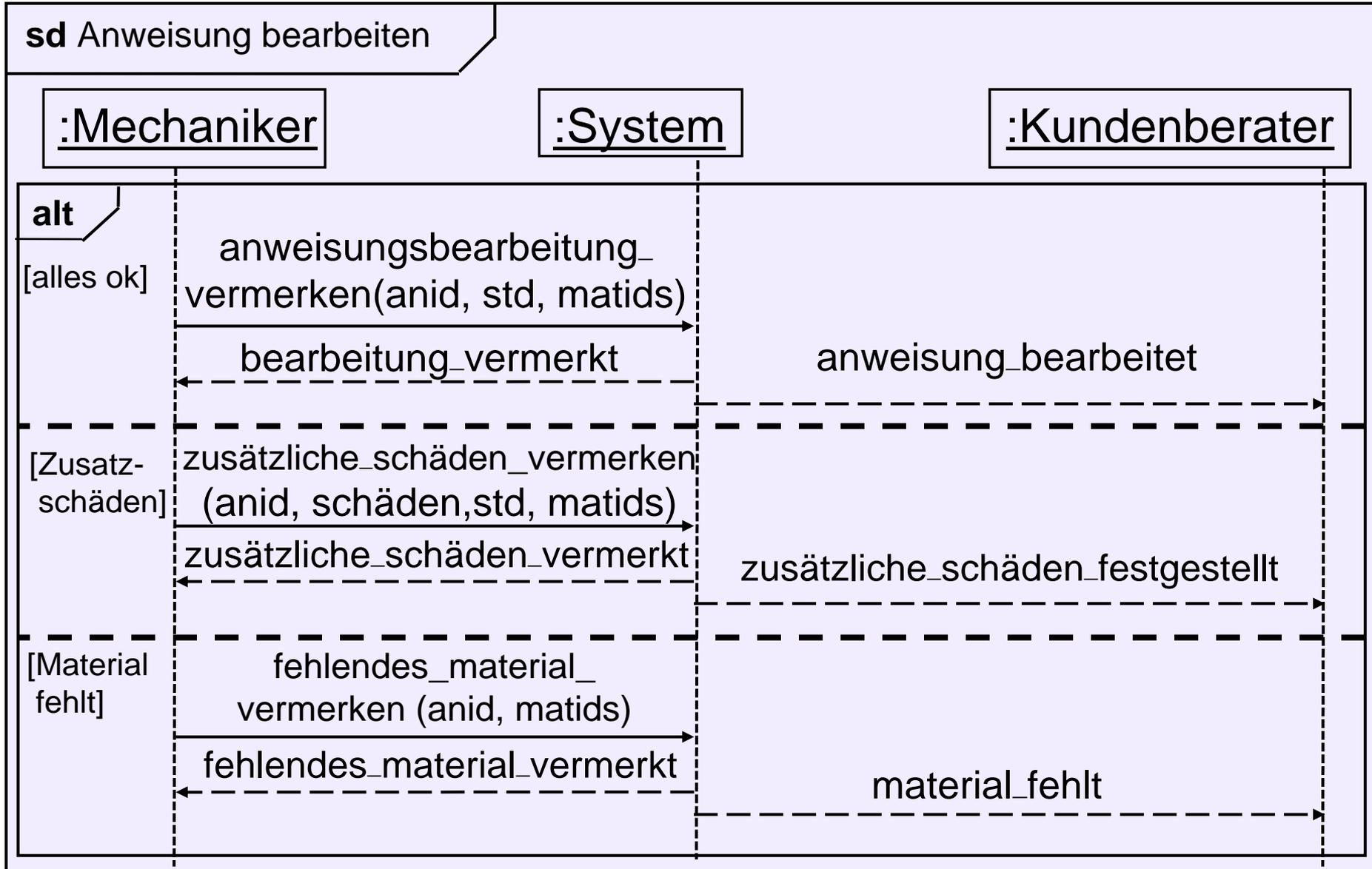
Use-Case „Auftragsannahme“



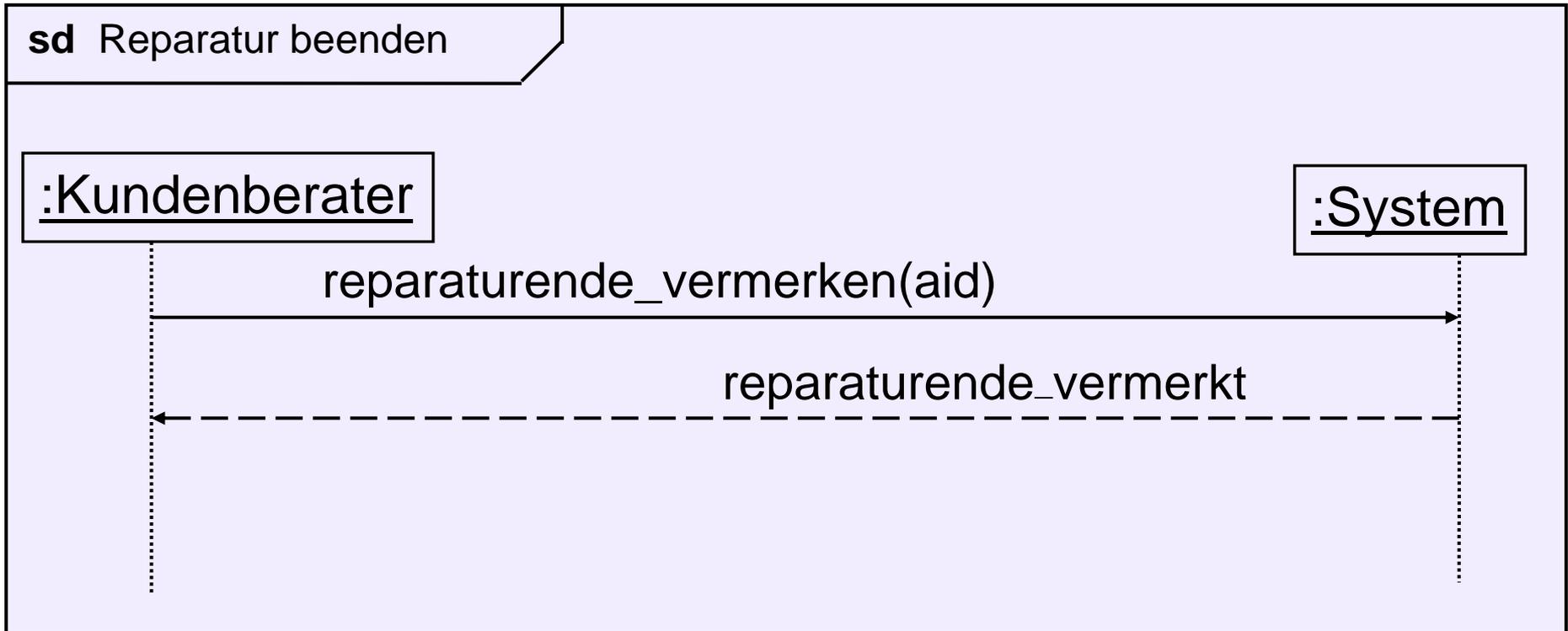
Use-Case „Auftragsannahme“



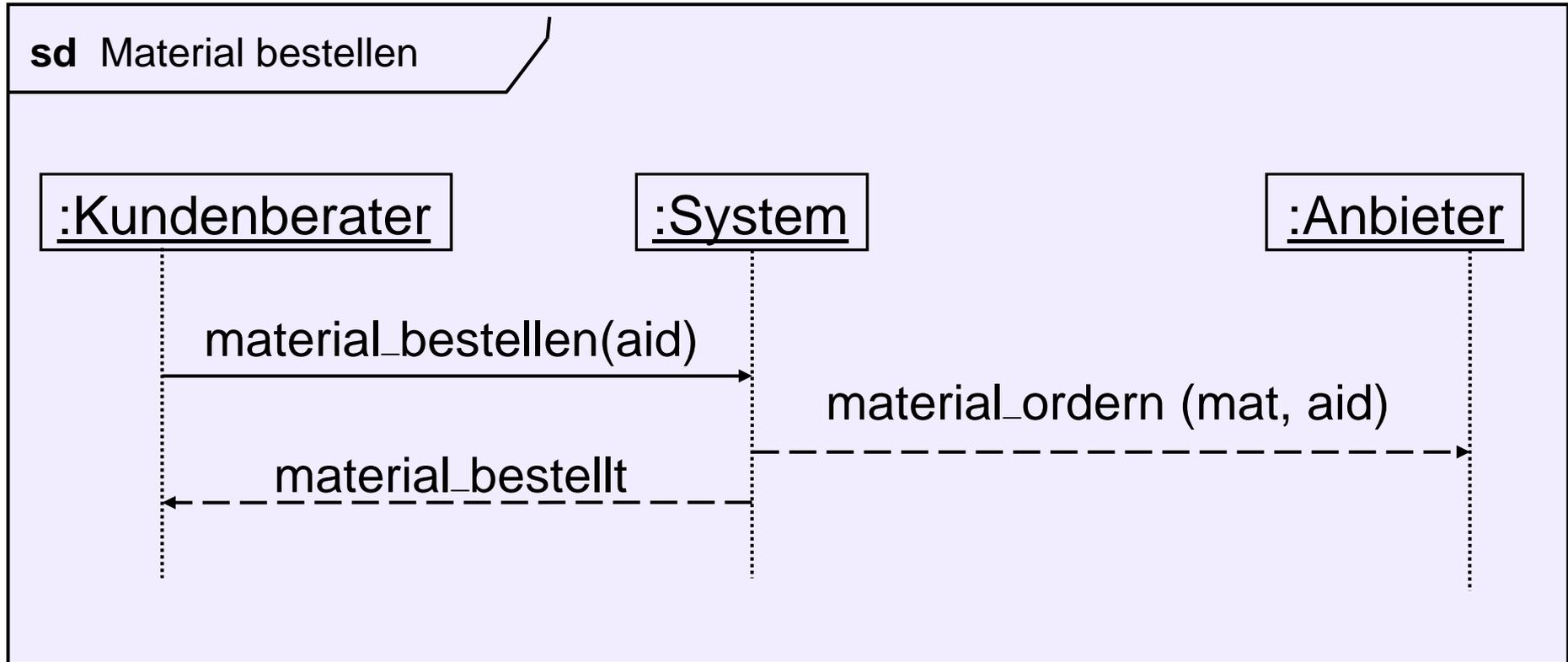
Use-Case „Anweisungsbearbeitung“



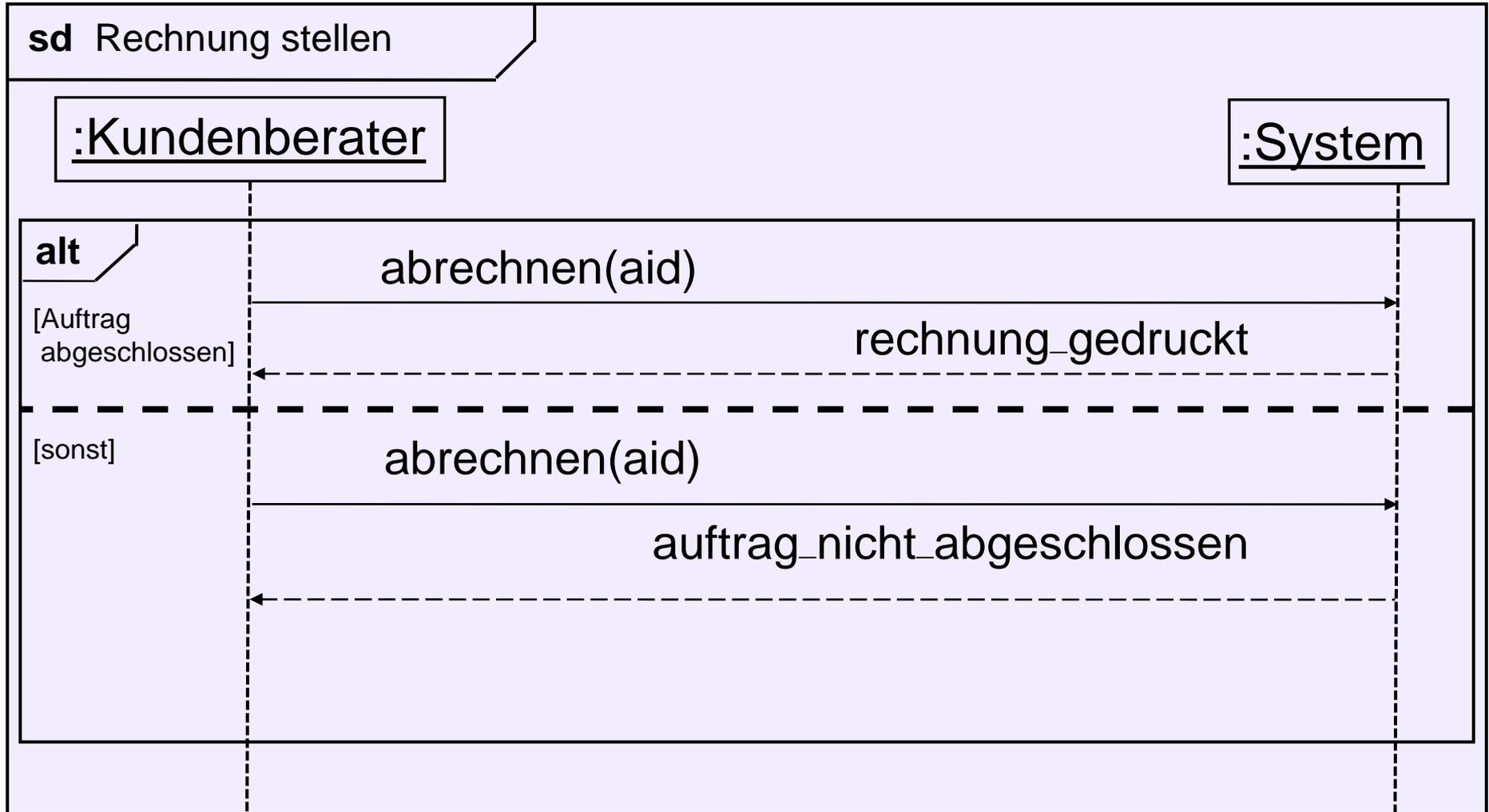
Use-Case „Anweisungsbearbeitung“



Use-Case „Materialbeschaffung“



Use-Case „Rechnungserstellung“



Was haben wir bis jetzt erreicht?

Anforderungsdefinition

Analyse

Schnittstellenmodell

Use-Case-Modell

Sequenzdiagramme

Aktivitätsdiagramme

Analyse-Klassenmodell

Vor- und Nachbedingungen
von System-Operationen

Klassenmodell

Data Dictionary