

Fakultät Informatik

Professur Softwaretechnologie

OOSE 07

OBJEKTORIENTIERTE ANALYSE

Dr.-Ing. Birgit Demuth
Sommersemester 2019

Wiederholung

Verhaltensgetriebene Modellierung versus Strukturgetriebene Modellierung

Statische Analyse (aUML)

- UML-Klassendiagramme
- UML Objektdiagramme

Verhaltensanalyse (aUML)

- UML-Sequenzdiagramme
- [UML-Zustandsdiagramme]

Klausurbeispielaufgaben

Wie gehen wir vor, ein Analyseklassendiagramm zu erstellen? Variante: Verhaltensgetriebene Modellierung

Schritte (Wiederholung, CRC-Kartenmethode, vgl. OOSE_05):

1. Textanalyse → Klassen(kandidaten) → „leere“ CRC-Karten
2. Analyse von **Szenarien (Rollenspiel)** und Identifizierung von **Verantwortlichkeiten/Operationen**
 - Schrittweises Ausfüllen der CRC-Karten
 - Protokollierung jedes Szenariums in einem UML-Sequenzdiagrammen
3. Abbildung der CRC-Karten in einem ersten UML-Analyseklassendiagramm
4. Anreicherung des UML-Analyseklassendiagramms durch detaillierte Beschreibung der Klassenbeziehungen

Wie gehen wir vor, ein Analyseklassendiagramm zu erstellen? Variante: Strukturgetriebene Modellierung

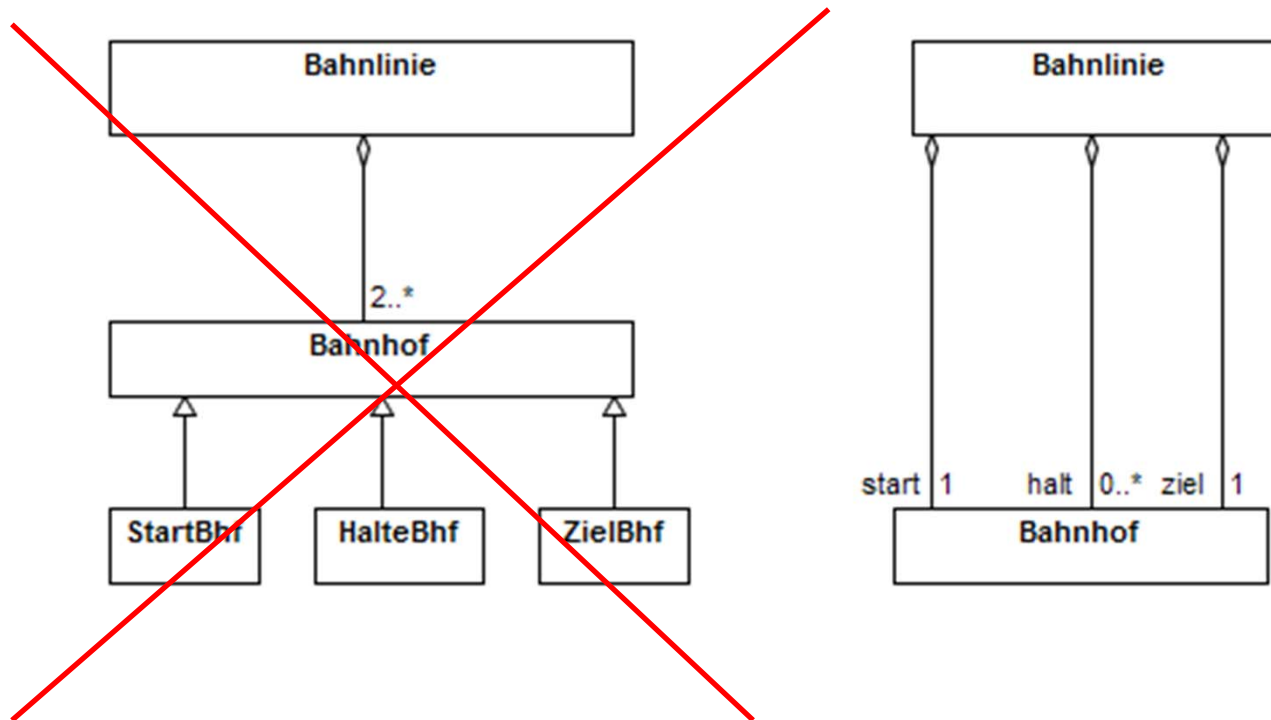
Textanalyse und schrittweise Anreicherung des Analyseklassendiagramms

Schritte:

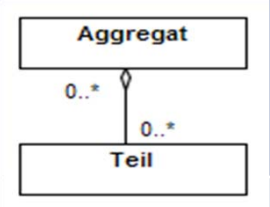
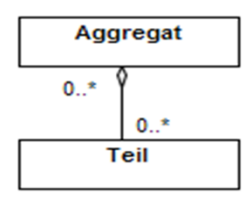
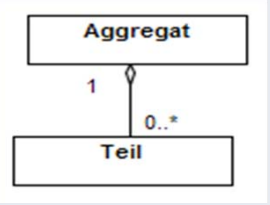
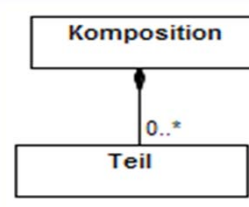
1. Klassen(-kandidaten) identifizieren
2. **Attribute** identifizieren
3. Enumerationen identifizieren
4. Klassenbeziehungen identifizieren
 - Vererbungsbeziehungen (Generalisierung/Spezialisierung)
 - Assoziation/Aggregation/Komposition/Assoziationsklassen/n-äre Beziehungen
5. Verfeinerte Beschreibung der Klassenbeziehungen (Multiplizitäten, Rollen, Assoziationsnamen mit Leserichtung)
6. Operationen identifizieren

UML-Klassendiagramme (aUML)

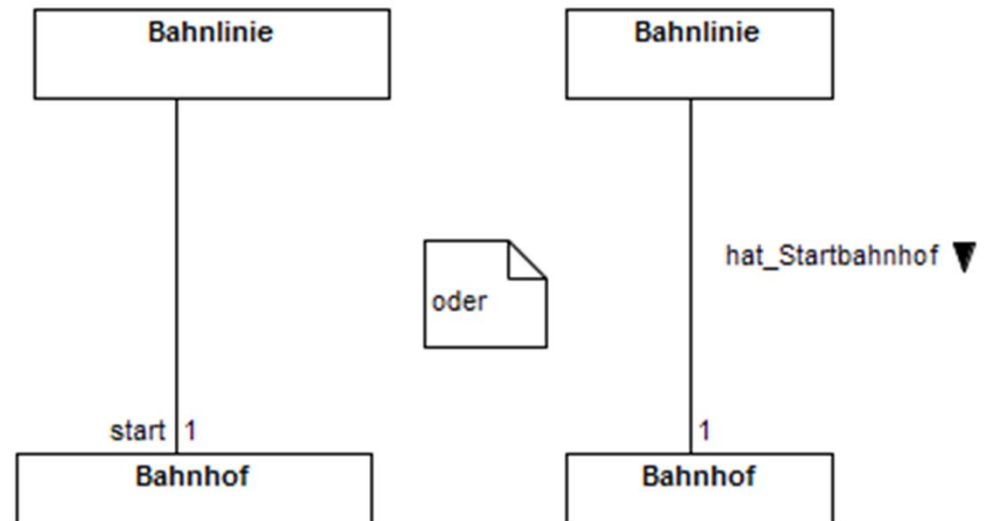
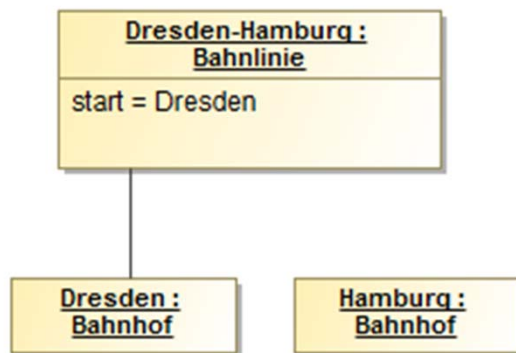
Vererbung oder Rollen?



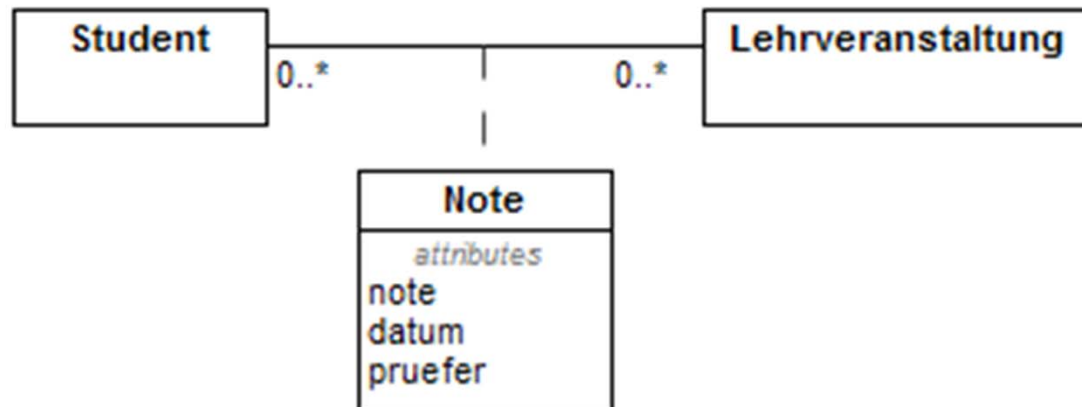
Teil-Ganzes-Beziehungen in UML

Eigenschaften	Unabhängiges Teil (independent part)	Abhängiges Teil (dependent part)
Geteiltes Teil (shared part)	<p>Aggregation</p> 	<p>Aggregation</p> 
Nicht geteiltes Teil (exclusively-owns-part)	<p>Aggregation</p> 	<p>Komposition</p> 

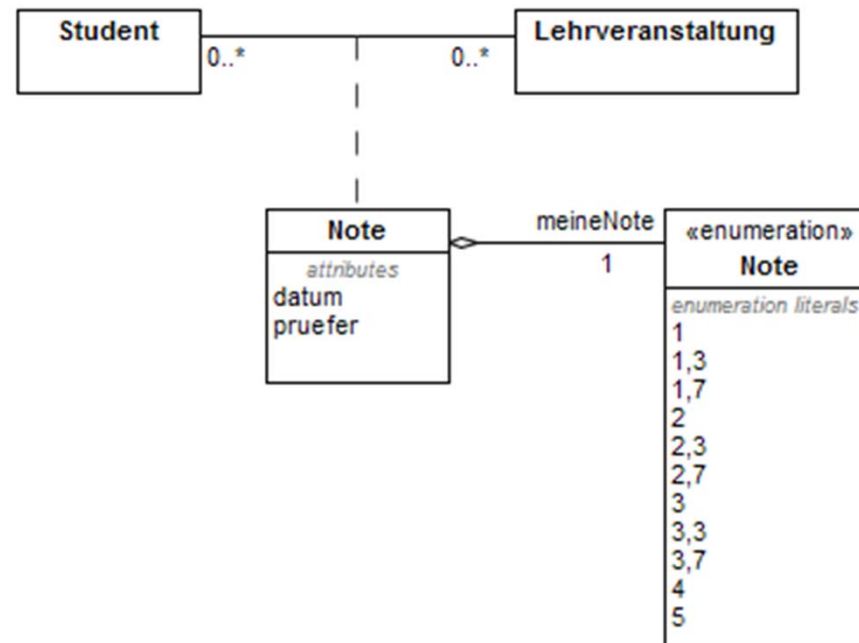
Assoziation mit Rolle oder Assoziationsnamen/Readerichtung



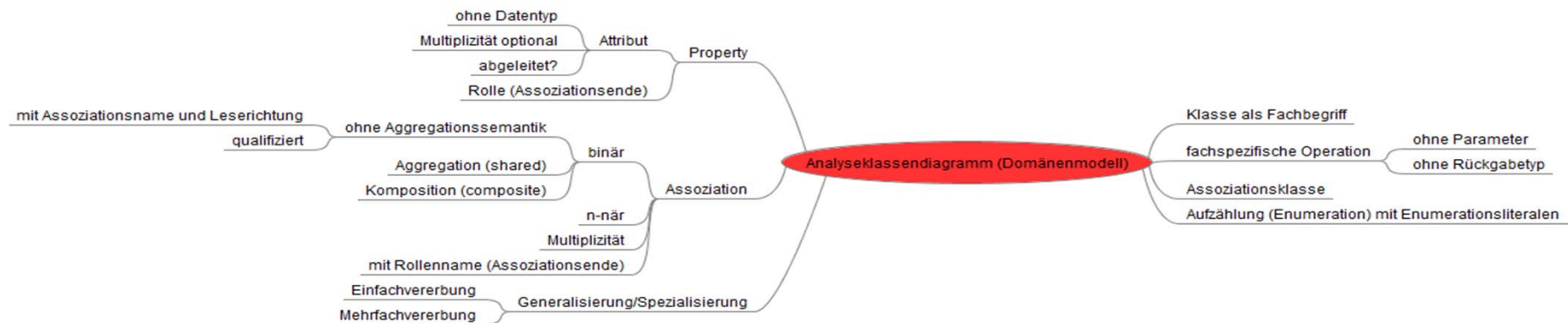
Assoziationsklasse



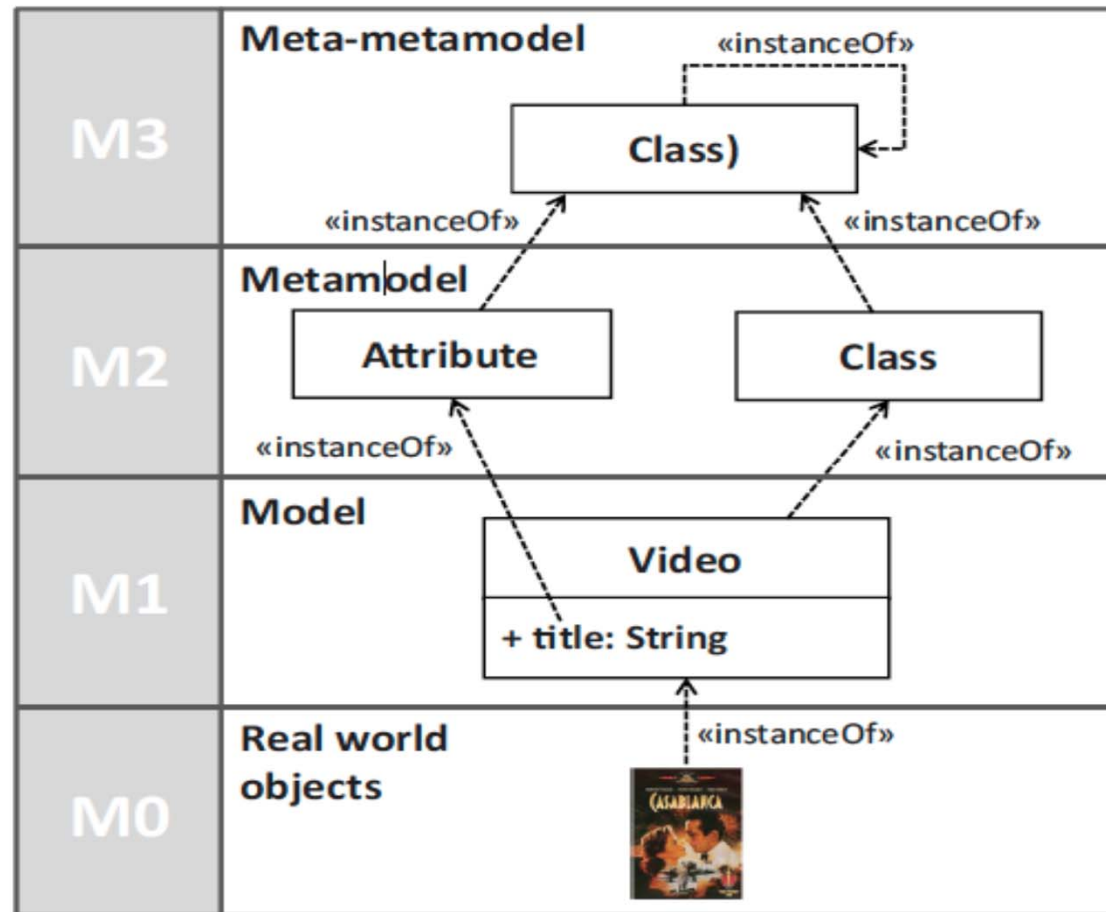
Enumeration



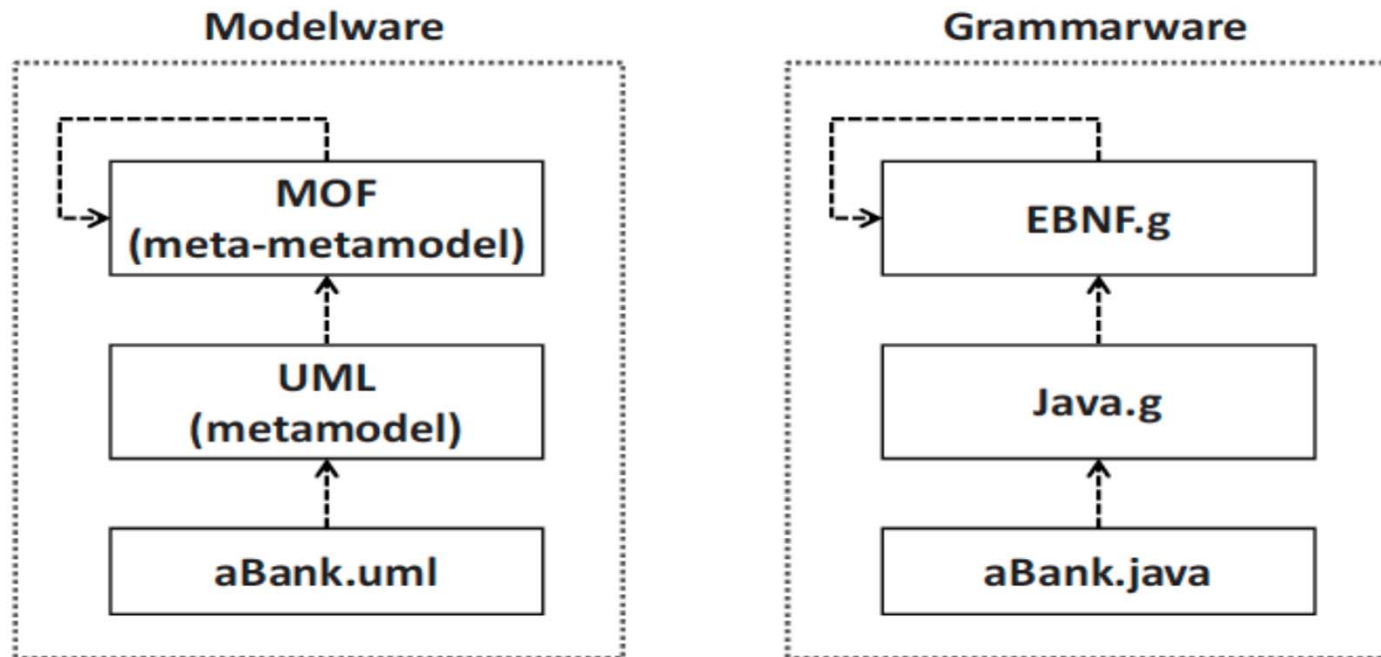
Welche Modellelemente enthält ein UML-Analyseklassendiagramm (Domänenmodell)? [2]



Metamodellierung [1]



Modelware vs. Grammarware [1]



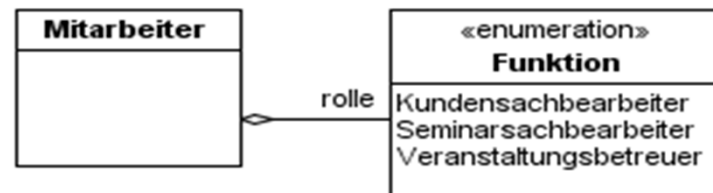
SemOrg (Klausur WS 2012/13)

Fallbeispiel SemOrg

Textanalyse
auf Basis der Informationen

- aus Aufgabe I-2 und I-3
- des Lastenheftes

gegeben:



→ Abstraktion der Fachbegriffe

→ **Nutze das Glossar!**

Glossar SemOrg [3]

Dozent: Führt als freier Mitarbeiter eine oder mehrere angebotene →Veranstaltungen durch. Ist fachlich in der Lage, ein oder mehrere →Seminare abzuhalten.

Firma: Mitarbeiter einer Firma (Ansprechpartner), der für die Aus- und Weiterbildung von Mitarbeitern zuständig ist und sich über Dienstleistungen informiert oder Mitarbeiter zu öffentlichen →Veranstaltungen schickt oder firmeninterne Veranstaltungen bucht.

Interessant: →Kunde, der sich für Dienstleistungen, z. B. Seminar-katalog, interessiert, aber noch an keiner →Veranstaltung teilgenommen hat.

Kunde: Mitarbeiter einer Firma oder Privatperson, der bzw. die an Dienstleistungen interessiert ist, oder ein Seminar bucht und besucht (→Teilnehmer, →Interessant).

Kundensachbearbeiter: Verantwortlich für die Kommunikation mit →Kunden und →Firmen einschließlich der Auskunftserteilung und Buchung.

Seminar: →Seminartyp.

Seminarsachbearbeiter: Verantwortlich für die Planung und Terminierung von →Seminaren und →Veranstaltungen. Zuständig für die Kommunikation und Akquirierung von →Dozenten.

Seminartyp: Beschreibt die Gemeinsamkeiten, die eine Menge von →Veranstaltungen besitzen wie Titel, Zielsetzung, Inhalt, Voraussetzungen.

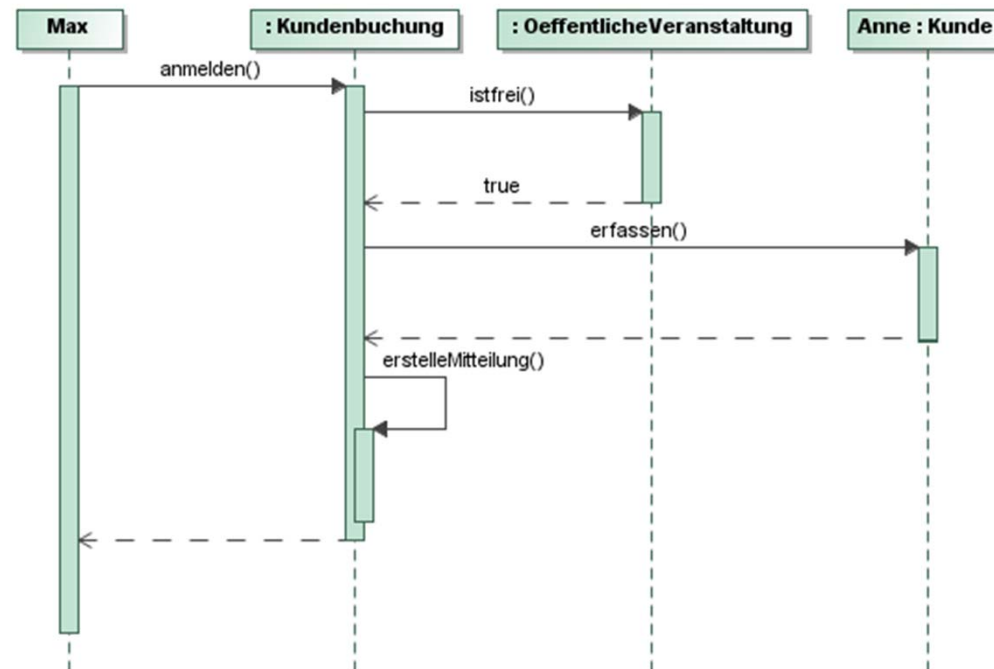
Seminarveranstaltung: →Veranstaltung.

Teilnehmer: →Kunde, der an einer →Veranstaltung teilnimmt bzw. teilgenommen hat.

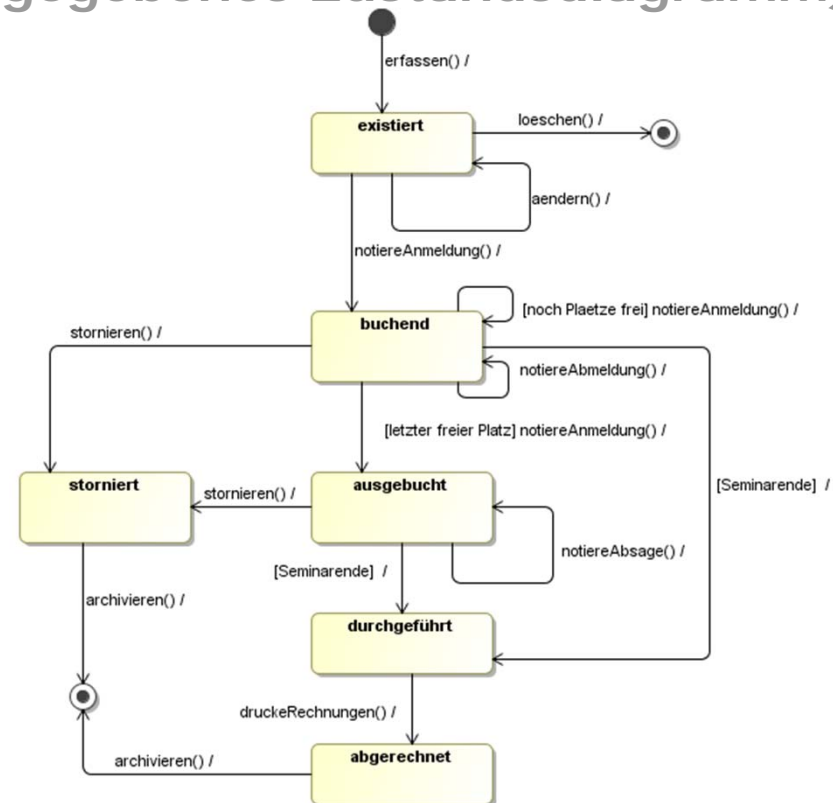
Veranstaltung: →Seminar, das zu einem festgelegten Zeitpunkt, an einem festgelegten Ort von einem oder mehreren →Dozenten durchgeführt wird.

Veranstaltungsbetreuer: Betreut die →Teilnehmer und →Dozenten einer →Veranstaltung.

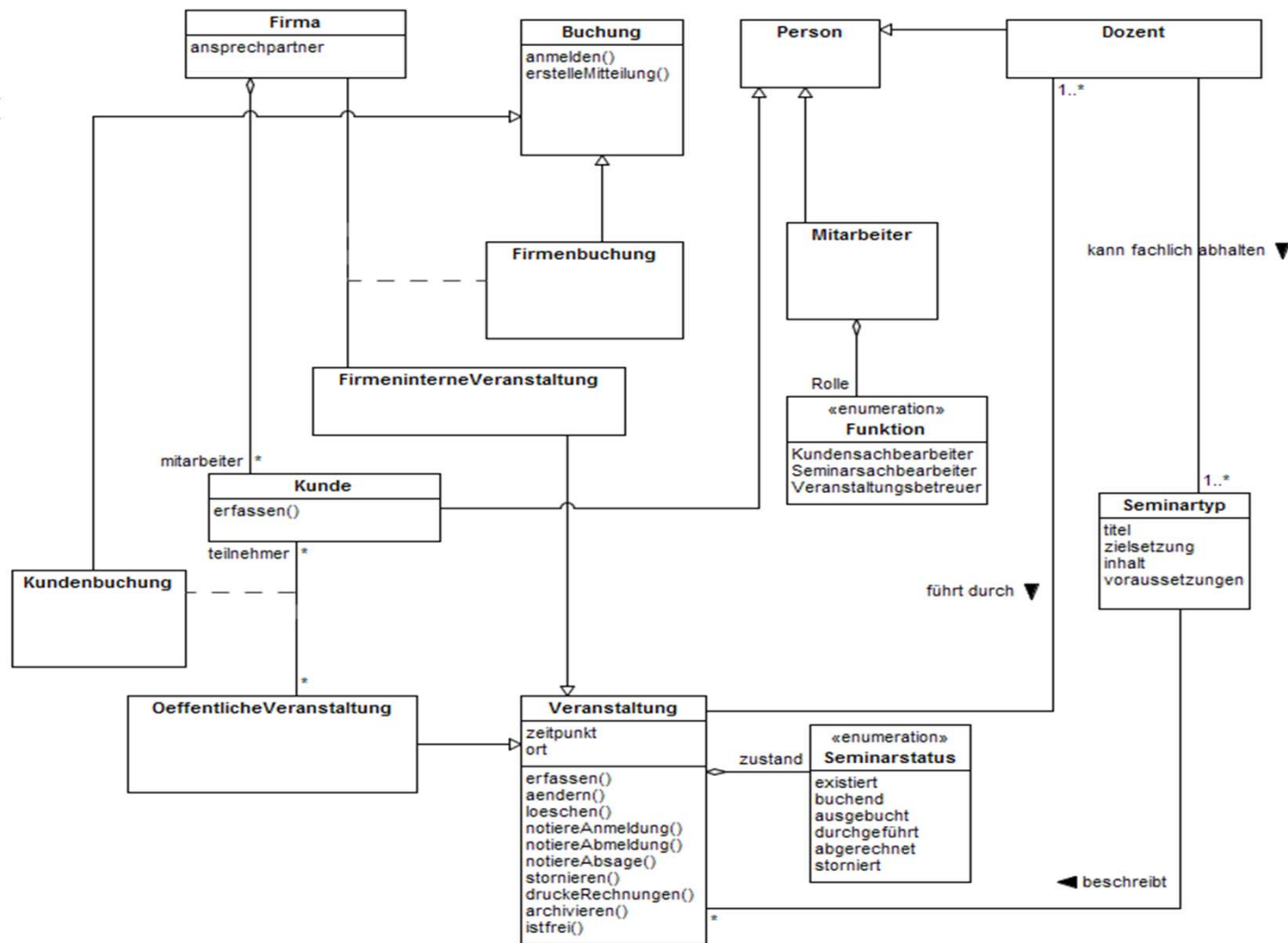
Fallbeispiel SemOrg (Aufgabe I-2: gegebenes Sequenzdiagramm)



Fallbeispiel SemOrg (Aufgabe 1-3: gegebenes Zustandsdiagramm)



Fallbeispiel SemOrg



Campus Management System (Klausur WS 2013/14)

Review eines Analyseklassendiagramms Fallbeispiel Campus Management System

Analyseklassendiagramm für den Teil **Lehrevaluation** gegeben

Gesucht

- Modellierungsfehler (F)
- Verletzung von Modellierungskonventionen in der Analyse (K)
- Vorschlag für eine Verbesserung (V)

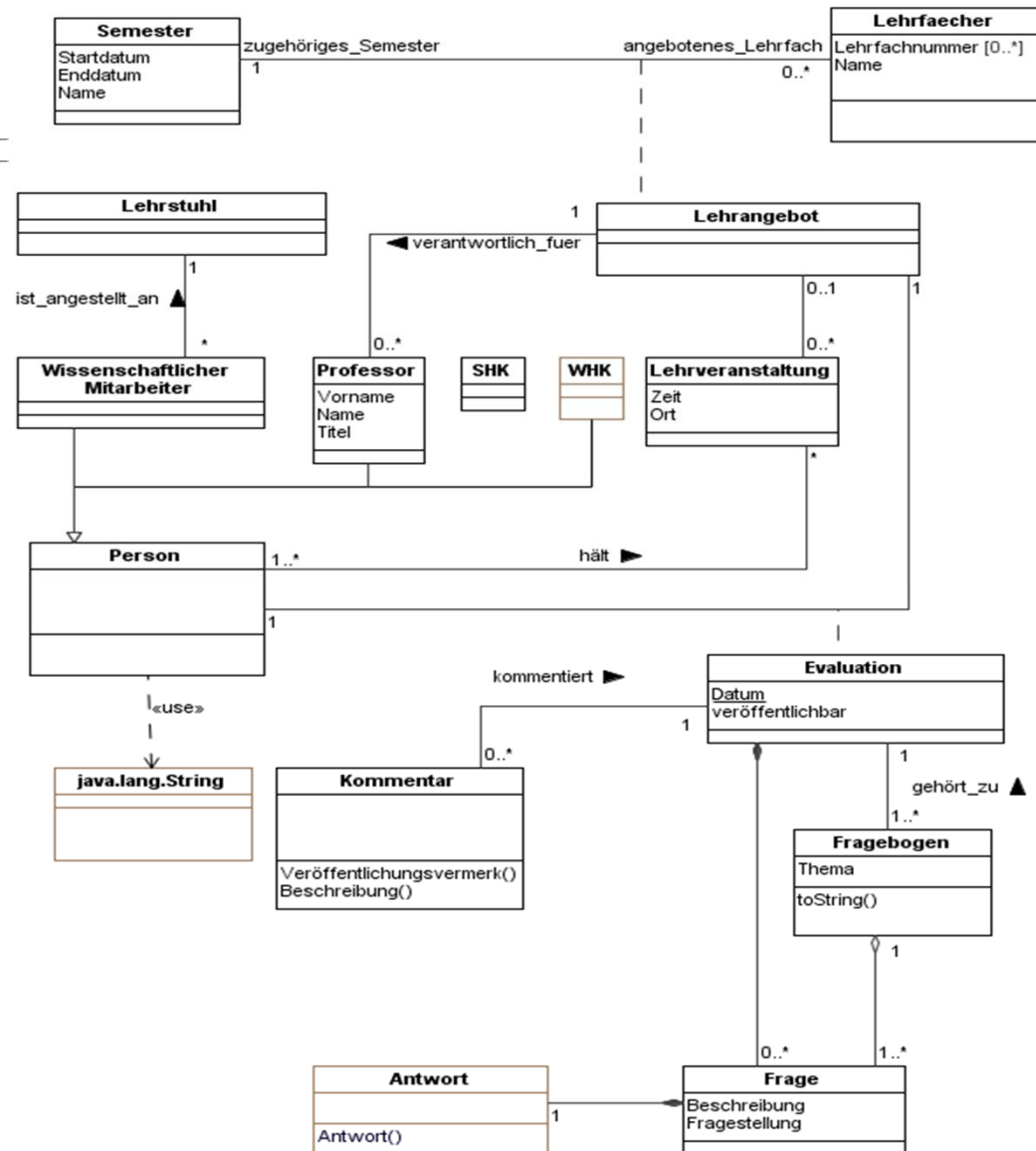
Vorgehen: Lesen der textuellen Spezifikation und mit dem Modell vergleichen!

Textuelle Spezifikation

Alle Angebote der Hochschule werden als Lehrfächer erfasst, die eine eindeutige Lehrfachnummer, einen Namen und eine Beschreibung besitzen. Lehrfächer werden allerdings nicht jedes Semester gehalten. Ein Lehrangebot beschreibt deshalb, in welchem Semester ein Lehrfach angeboten wurde und welcher Professor für dieses Lehrangebot zu diesem Semester verantwortlich ist. Ein Semester besitzt ein Start- und ein Enddatum sowie einen Namen. Lehrveranstaltungen werden jeweils durch Ort und Zeit beschrieben. Eine Lehrveranstaltung ist genau einem Lehrangebot zugeordnet. Zusätzlich ist wichtig, dass jede Lehrveranstaltung von einer Person gehalten wird. Personen können studentische Hilfskräfte, wissenschaftliche Mitarbeiter (die an einem Lehrstuhl angestellt sind) und Professoren sein. Jeder Lehrstuhl hat genau einen Professor als Lehrstuhlinhaber. Eine Person wird durch einen Vor- und Nachnamen sowie einen Titel gekennzeichnet. Ein Professor unterscheidet sich von einem wissenschaftlichen Mitarbeiter dadurch, dass er u. a., wie oben erklärt, für Lehrangebote verantwortlich ist. Dabei kann es sein, dass er in einem Semester keine Lehrangebote verantworten muss.

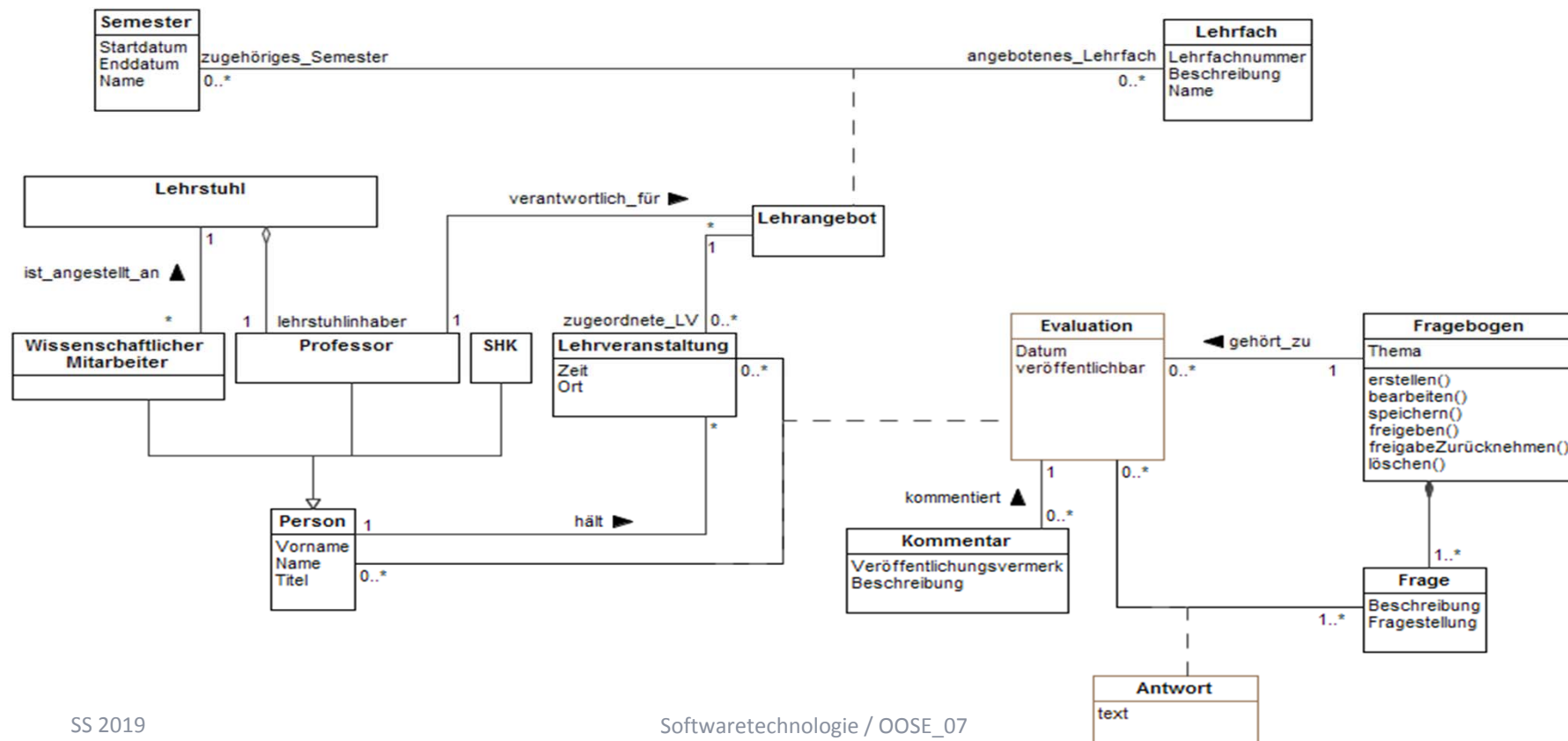
Eine Evaluation wird für eine bestimmte Person und eine bestimmte Lehrveranstaltung durchgeführt. Dabei ist wichtig festzuhalten, wann die Evaluation durchgeführt wurde und ob sie veröffentlicht werden kann. Jeder Evaluation liegt ein Fragebogen(Formular) zu Grunde, der jeweils ein Thema besitzt. Ein Fragebogen besteht aus einzelnen Fragen, die eine textuelle Fragestellung und eine Beschreibung beinhalten. Eine Evaluation berücksichtigt alle Antworten zu den Fragen eines Fragebogens. Eine Antwort ist dabei immer textuell beschrieben. Zusätzlich können zu jeder Evaluation Kommentare verfasst werden, die eine Beschreibung und einen Veröffentlichungsvermerk besitzen.

Analyse- klassen- diagramm mit Fehlern



Nr	Änderung betrifft ...	(optional) Kommentar	Art
1	Lehrfaecher	Singular verwenden	K
2	Lehrfachnummer[0..*]	Eindeutig bedeutet [1]	F
3	Lehrfaecher	Attribut Beschreibung fehlt	F
4	Assoziationsklasse Lehrangebot	Multiplizität Semester falsch	F
5	Leserichtung „verantwortlich_fuer“	umdrehen	F
6	Multiplizitäten Professor - Lehrangebot	umdrehen	F
7	Assoziation Lehrangebot-Lehrveranstaltung - Multiplizität von Lehrangebot	- 1 statt 0..1	F
8	- Nähere Beschreibung der Assoziation	- Z.B. mit Rollennamen	K
9	Assoziation „hält“ – Multiplizität von Person	1 statt 1..*	F
10	SHK: Vererbung fehlt	Zu Person	F
11	WHK	Nicht erklärt und damit falsch	F
12	Assoziation Lehrstuhl - Professor	fehlt	F
13	Professor: Attribute	Gehören zu Person	F
14	Assoziationsklasse Evaluation: Multiplizitäten und	0..*/0..* statt 1/1	F+F
14a	Beteiligte Klassen: Lehrangebot und Person	Lehrveranstaltung statt Lehrangebot	F
15	Datum der Evaluation	Statische Eigenschaft gehört nicht in die OOA	K
16	toString()	Gehört nicht in die OOA	K
17	Evaluation – Fragebogen: Multiplizitäten	0..* und 1 statt 1 und 1..*	F+F
18	Antwort	Attribut Text fehlt	F
19	Antwort()	Konstruktor zu technisch, gehört nicht in die OOA	K
20	Kommentar - Methoden	Sind Attribute	F+F
21	Evaluation – Frage - Antwort	Antwort ist Assoziationsklasse	V
22	java.lang.String()	gehört nicht in das Analyseklassendiagramm	K

Fallbeispiel Campus Management System Refaktorisiertes Analyseklassendiagramm



UML-Zustandsdiagramme (aUML)

Klassifikation von UML-Zustandsmodellen

nach Abstraktionsebene

- Analyseebene → **Analysezustandsmodell (ASM)**
- Entwurfsebene → **Entwurfzustandsmodell (DSM)**

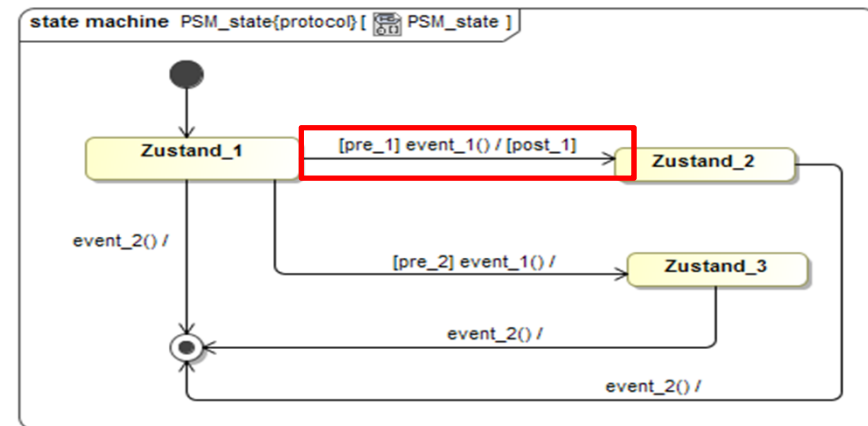
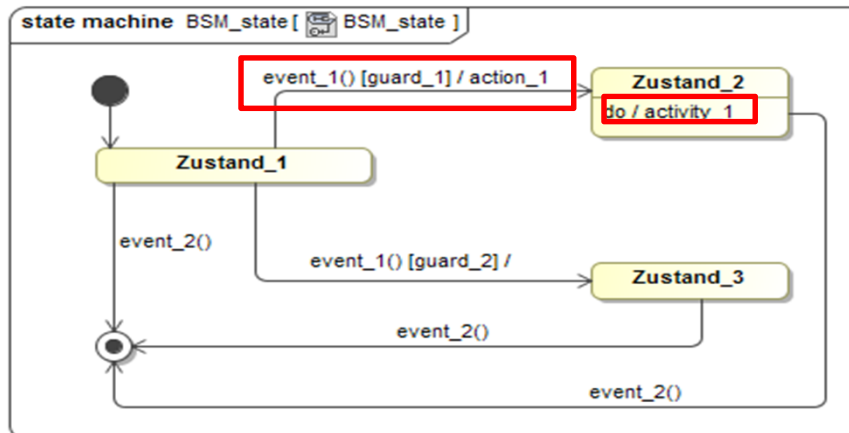
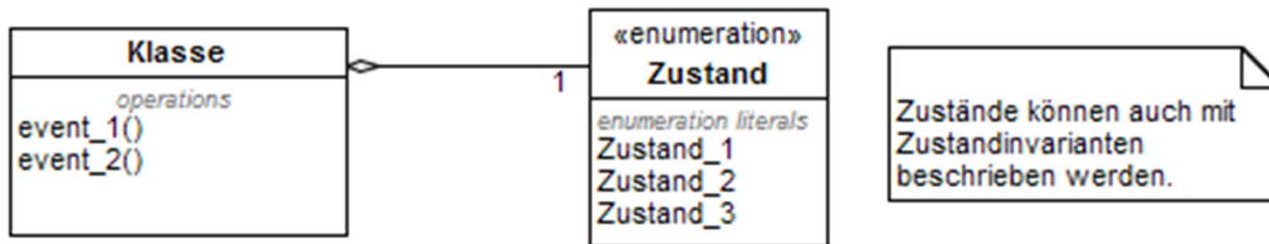
danach, worauf sich das Zustandsmodell bezieht

- Objekt → **Objektlebenszyklus (OLC)**
- Anwendungsfall → **Anwendungsfallebenszyklus (ULC)**
- Beschreibung eines technischen Gerätes/Systemes
→ **(technische) Steuerungsmachine (CSM)**

entsprechend der UML2.0-Spezifikation

- Spezifikation des Verhaltens
→ **Verhaltens(zustands)maschine (BSM)**
(Behavioral state machine)
- Spezifikation der zulässigen Reihenfolge von Operationen
→ **Protokollmaschine (PSM)**
(Protocol State Machine)

UML 2: Verhaltenszustandsmaschine (BSM) versus Protokollmaschine (PSM) [OLC]



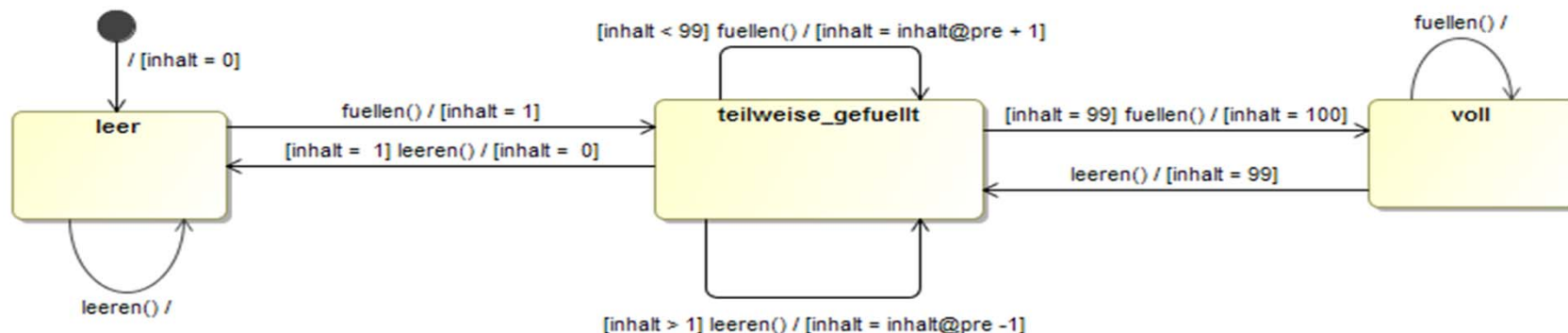
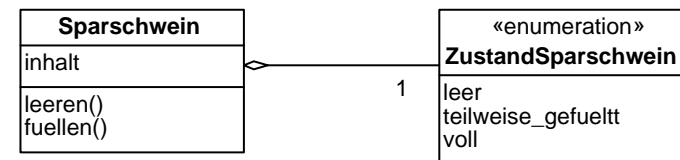
Objektlebenszyklus (PSM) [OLC] Lebenszyklus eines Sparschweins (Klausur SS 2013)

Ein Sparschwein hat einen Inhalt (inhalt), ist aber zu Beginn leer und kann solange mit einzelnen Eurostücken aufgefüllt werden (fuellen()), bis es voll ist.

Es passen genau 100 Eurostücke in das Sparschwein.

Solange das Sparschwein nicht leer ist, können jederzeit wieder einzelne Euros entnommen werden (leeren()).

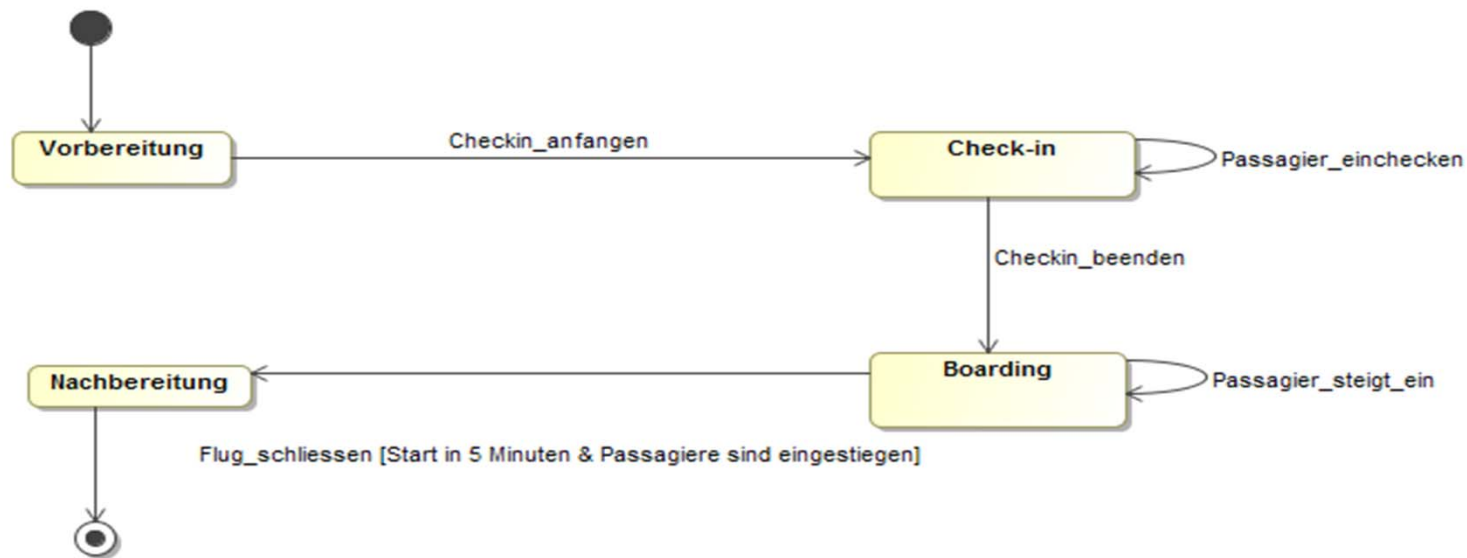
Solange das Sparschwein nicht voll ist, kann jederzeit wieder ein Euro eingezahlt werden.



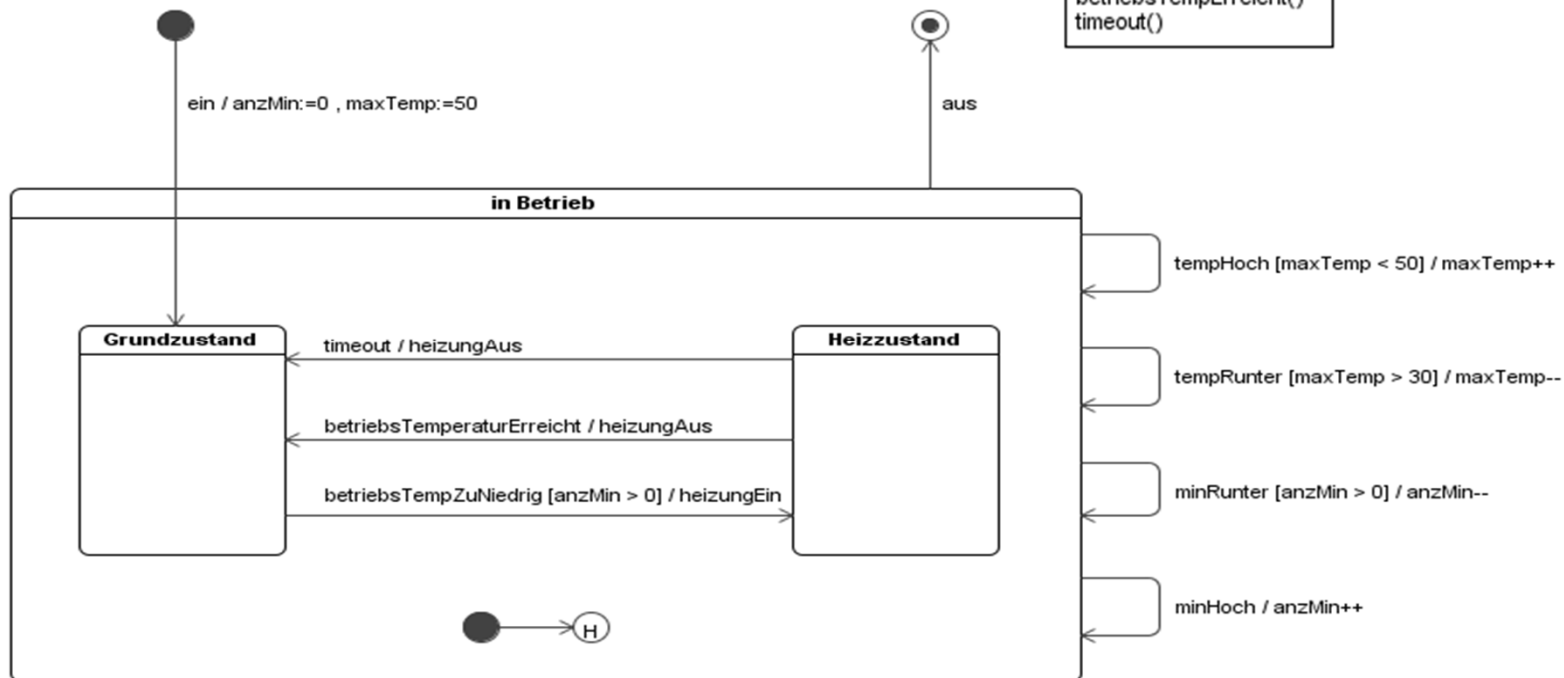
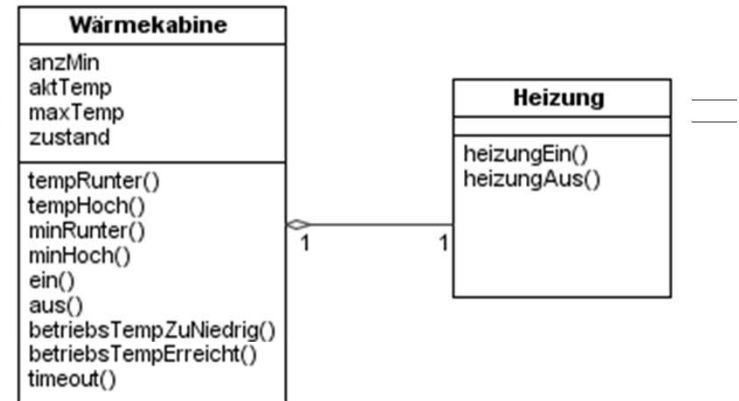
Habe ich mein Zustandsmodell (BSM/PSM) richtig konstruiert? (nach [4])

- (1) Habe ich das Zustandsdiagramm für einen **nicht-trivialen Lebenszyklus** erstellt?
- (2) Welche **Zustände** enthält das Diagramm?
- (3) Existiert ein **Startzustand**?
- (4) Welche **Ereignisse/Aufrufe** sind zu modellieren?
- (5) Welche **Operationen** besitzt das Objekt? (nur BSM)
- (6) Sind Operationen als **Aktivitäten** oder **Aktionen** zu modellieren? (nur BSM)
- (7) Hängen Zustandsübergänge von Bedingungen ab?
- (8) Habe ich geeignete **Zustandsnamen** gewählt?
- (9) Ist das Zustandsmodell **konsistent** mit dem zugehörigen Klassendiagramm?
- (10) Sind alle **Zustandsübergänge** korrekt eingetragen?

Anwendungsfallebenszyklus (BSM) Flug abwickeln (2)



Steuerungsmaschine (BSM) Infrarot-Wärmekabine (Klausur WS 2006/07)



Literatur

- (1) Marco Brambilla, Jordi Cabot, Manuel Wimmer: Model-Driven Software Engineering in Practice. Morgan & Claypool, 2012
- (2) Birgit Demuth (Hrsg.): Softwaretechnologie für Einsteiger. Pearson Studium, 2. geänderte Auflage, 2014
- (3) Helmut Balzert: Lehrbuch der Softwaretechnik. Basiskonzepte und Requirements Engineering. Spektrum Akademischer Verlag Heidelberg, 3. Auflage, 2009
- (4) Heide Balzert: UML kompakt mit Checklisten. Spektrum Akademischer Verlag, 2001

Klausuren und zugeordnete Dokumente liegen auf
<ftp://ftp.ifsr.de/klausuren/SWT/>

Ende