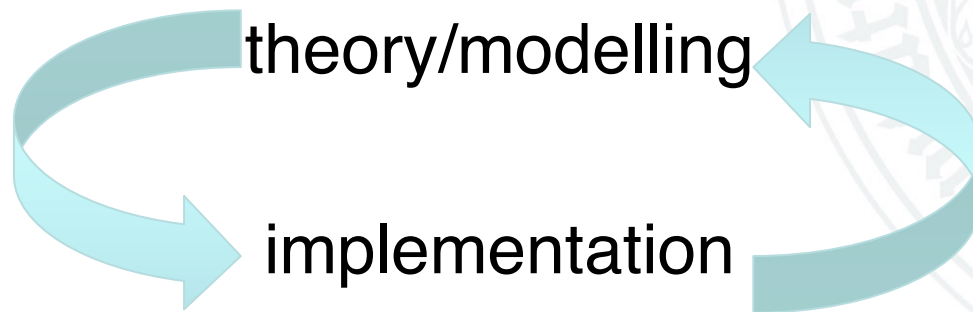


# Historisch- Kulturwissenschaftliche Informationsverarbeitung Woche 4

## UML Beschreibung und Praxis



# Development and modelling



Theoretical model  
*model for*

Empirical model  
*model of*

comparison  
calibration  
verification



*Deduction?*



*induction*



Generating  
knowledge

comparison  
evaluation



*created object*



*modelled object*

Thanks to Oliver Nakoinz for inspiration



# II 0. UML 2.0 (2003 / 04 ff.)

UML ist eine Sammlung von "*graphischen Sprachen*", d.h. Regelsystemen für die Konstruktion graphischer Schemata, die:

- ❖ unterschiedliche Perspektiven von Anforderungen an Systeme und Entwürfen von Systemteilen, sowie deren Zusammenwirken darstellen,
- ❖ einander dabei überlappen können und
- ❖ unabhängig voneinander verwendet werden können.

Am wichtigsten:

- ❖ Klassenmodelle beschreiben den strukturellen Aufbau eines Systems,
- ❖ Anwendungsfallmodelle (Use Cases) beschreiben die Interaktion mit dem System aus Benutzersicht.



# The role of UML

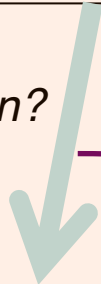
Theoretical model  
*model for*

Empirical model  
*model of*

*comparison  
calibration  
verification*



*Deduction?*



*induction*



*created object*

Generating  
knowledge



*modelled object*

*comparison  
evaluation*



# II 1. Klassendiagramme

Die Klassendiagramm stellt

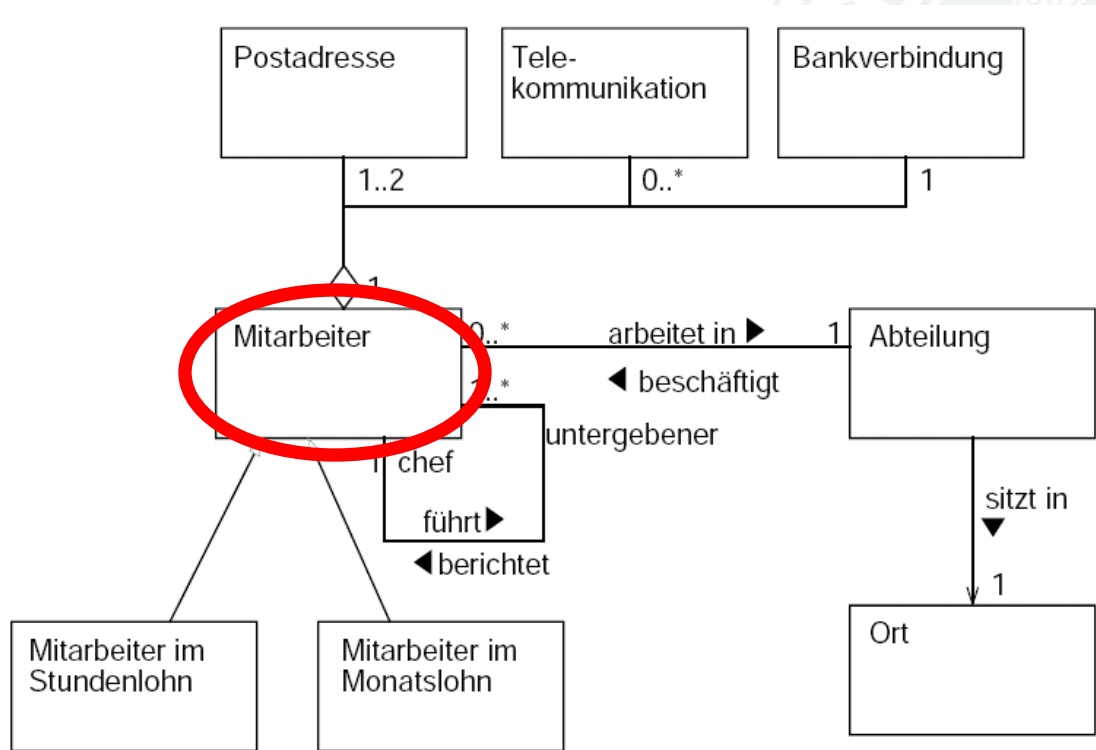
- die Klassen
- die Generalisierung
- die Assoziationen zwischen Klassen

dar

Beschreibt das statische Model



# II 1. Klassendiagramme

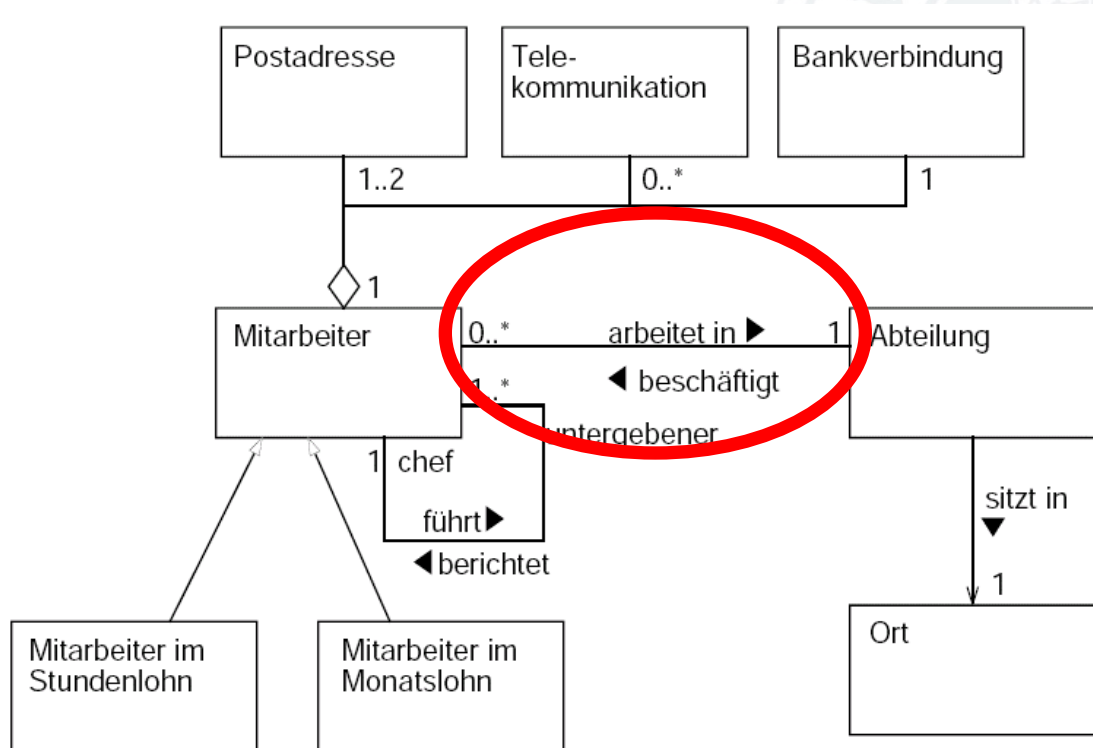


*Objekt* „Mitarbeiter“

(kann Attribute und Methoden haben) → Programmierung



# II 1. Klassendiagramme

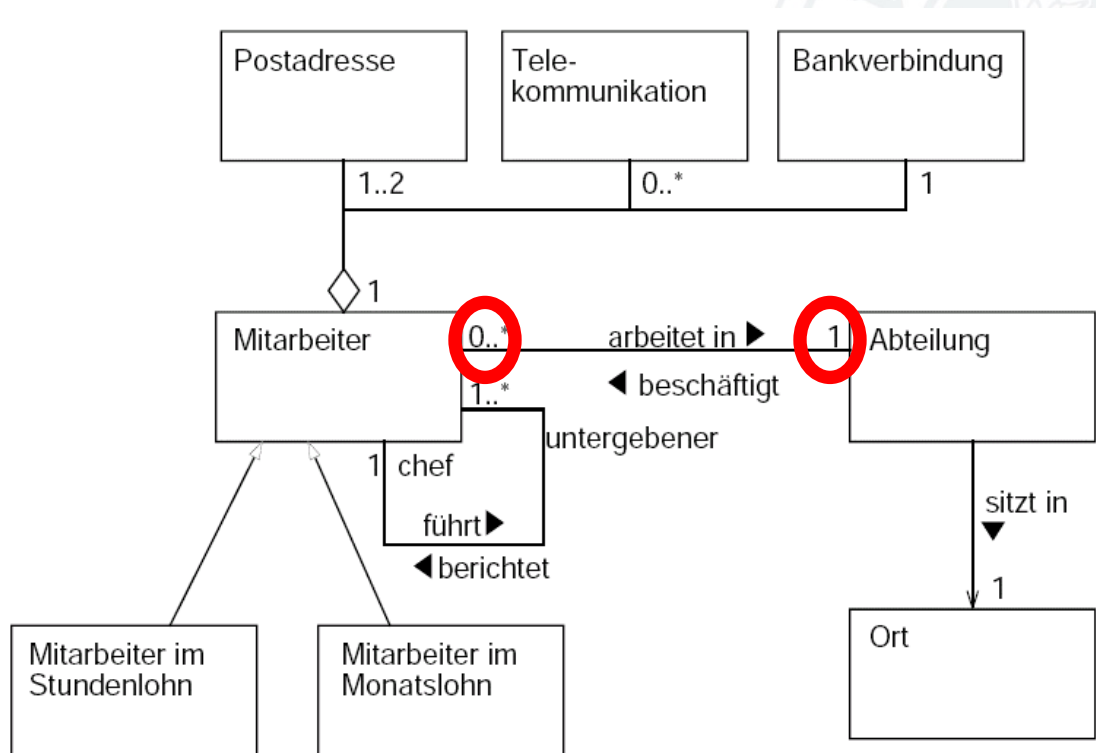


*Binäre Assoziation* beschreibt die Beziehungen zwischen Klassen





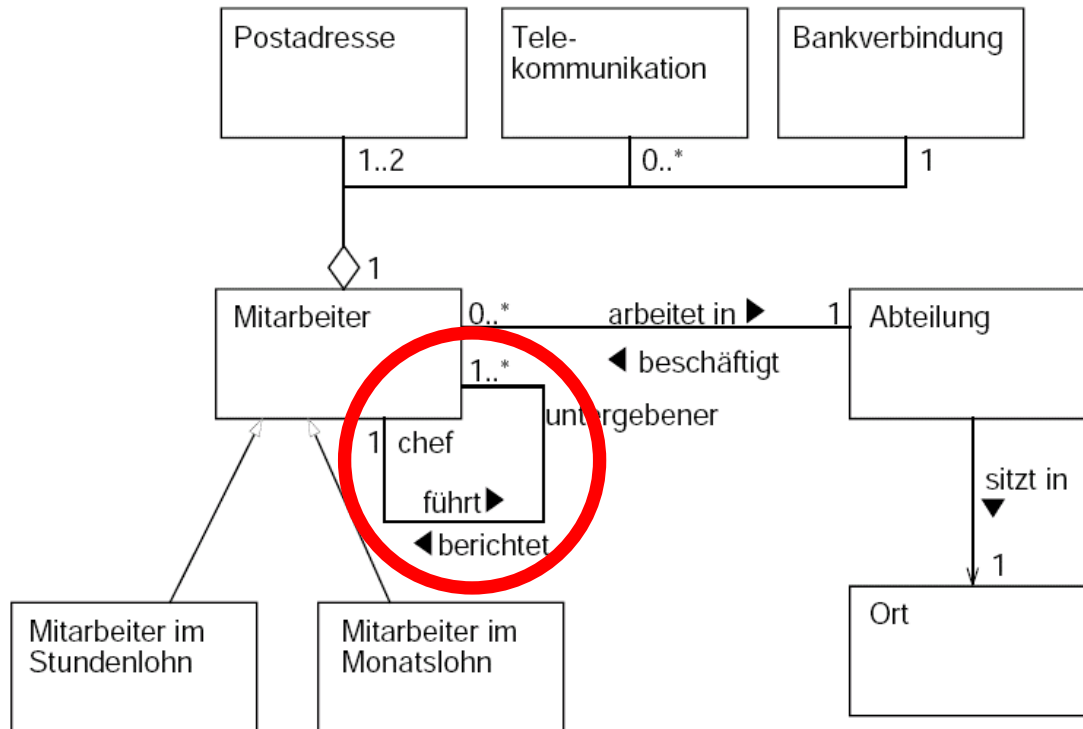
# II 1. Klassendiagramme



*Multiplizität* gibt an, wie viele Objekte an einer Assoziation beteiligt sein können.



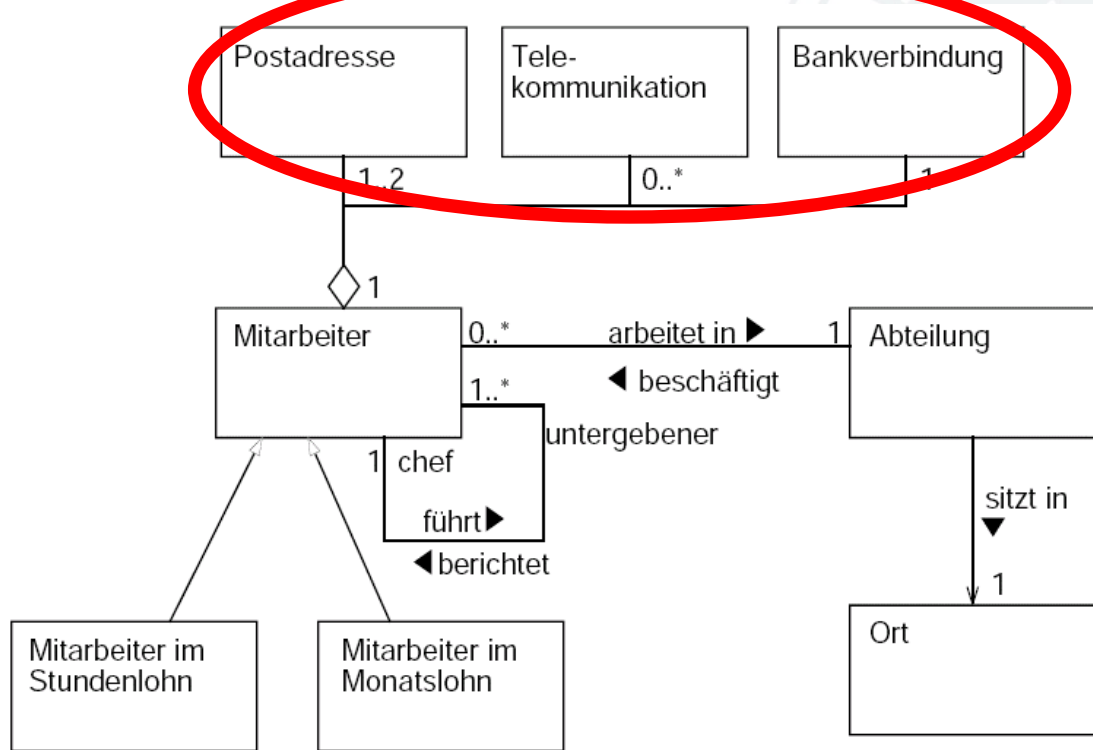
# II 1. Klassendiagramme



*Reflexive Assoziation* verbindet Objekte einer Klasse miteinander.



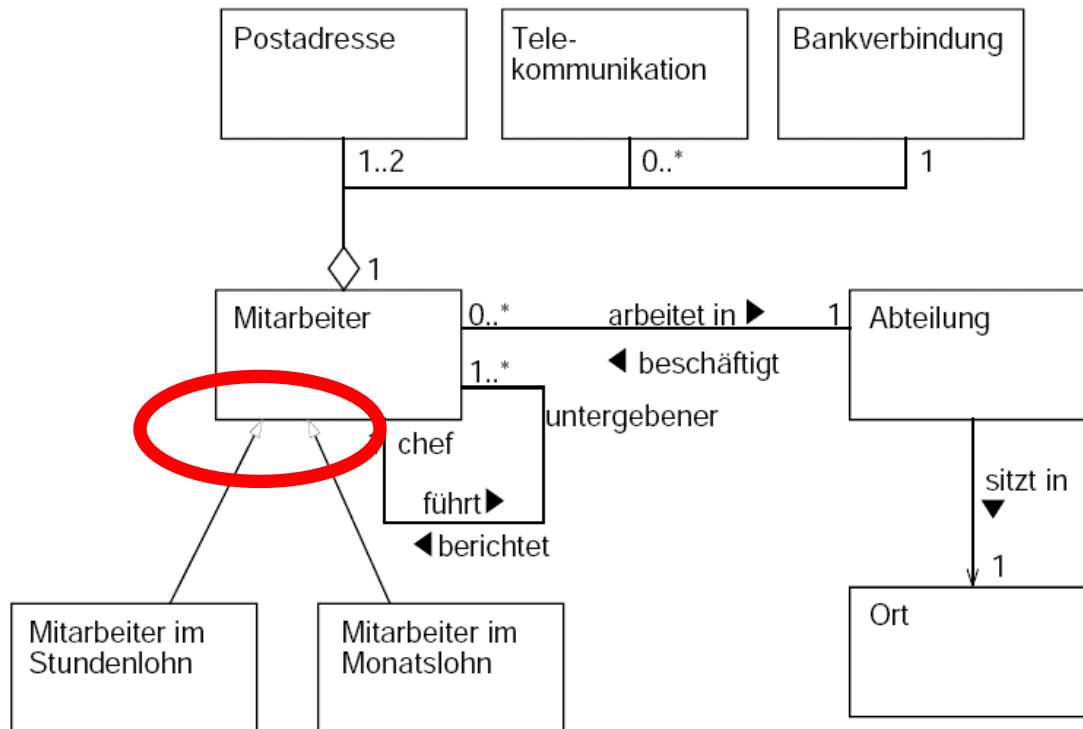
# II 1. Klassendiagramme



*Aggregation* verbindet beliebig viele Klassen zu einer übergeordneten.



# II 1. Klassendiagramme



*Generalisierungsbeziehung zwischen Superklasse und Subklasse.*



# II 2. Anwendungsfalldiagramme

Das Verhalten eines Systems kann als Sammlung von *Anwendungsfällen* (= *use cases*) beschrieben werden.

Ein Anwendungsfall beschreibt eine Klasse möglicher Interaktionen.

Konkrete Anwendungsfälle heißen auch *Szenarien*.  
( → scenario based design.)

Anwendungsfälle werden in strukturiertem Text beschrieben.

Alle möglichen Anwendungsfälle - oder ein für ein bestimmtes Teilsystem relevanter Teil - werden als *Anwendungsfalldiagramm* realisiert.



# II 2. Anwendungsfalldiagramme

Anwendungsfall als strukturierter Text (auch als Aktivitäts – oder Zustandsdiagramme)

Beispiel: "Buch an einem Selbstausleiheautomaten ausleihen"

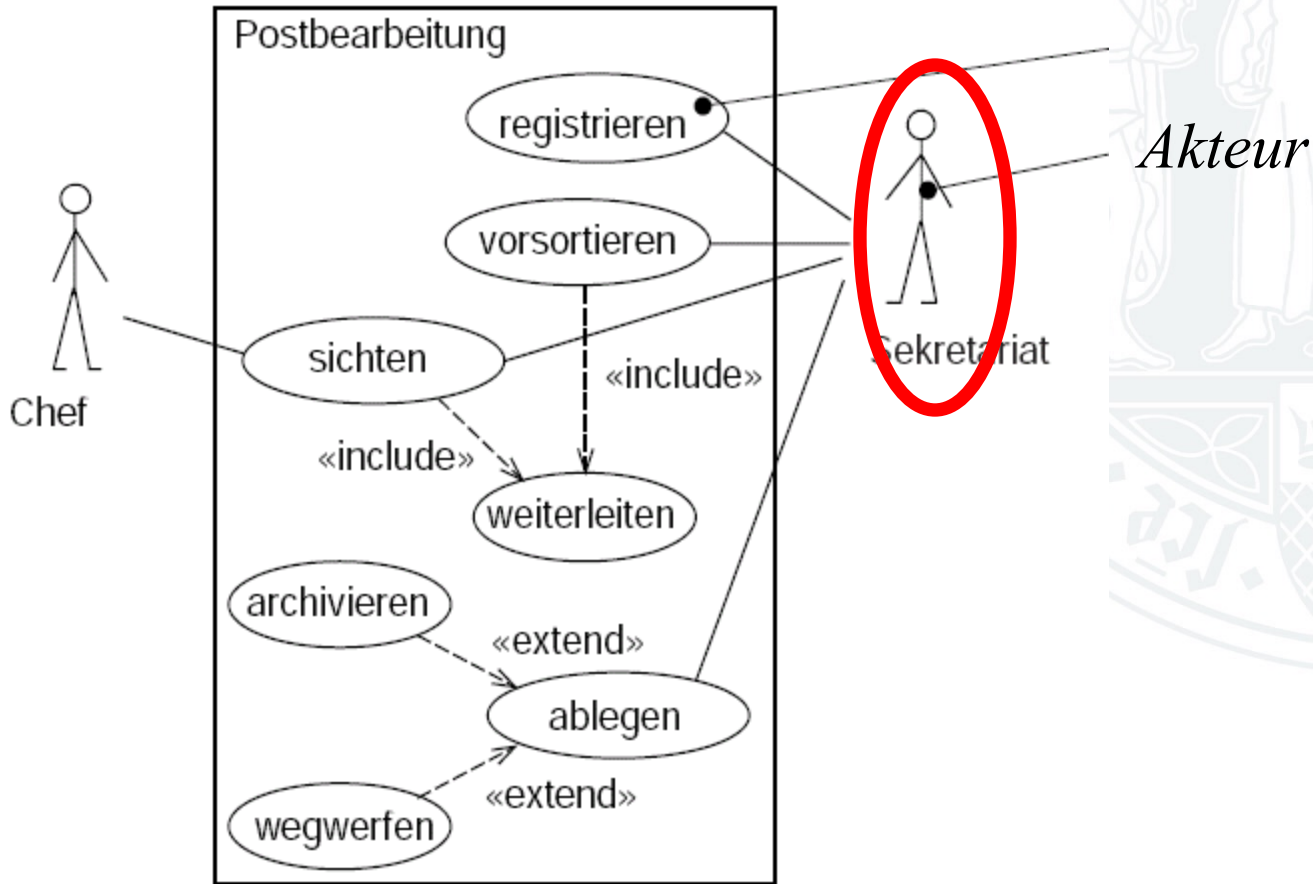
Normallfall:

1. BenutzerIn liest Ausweis in System ein; System validiert Ausweis.
2. BenutzerIn wählt "Ausleihen"; System aktiviert Ausleihfunktion.
3. BenutzerIn liest Buchcode ein; System identifiziert das Buch, registriert Ausleihe, deaktiviert das Diebstahletikett.

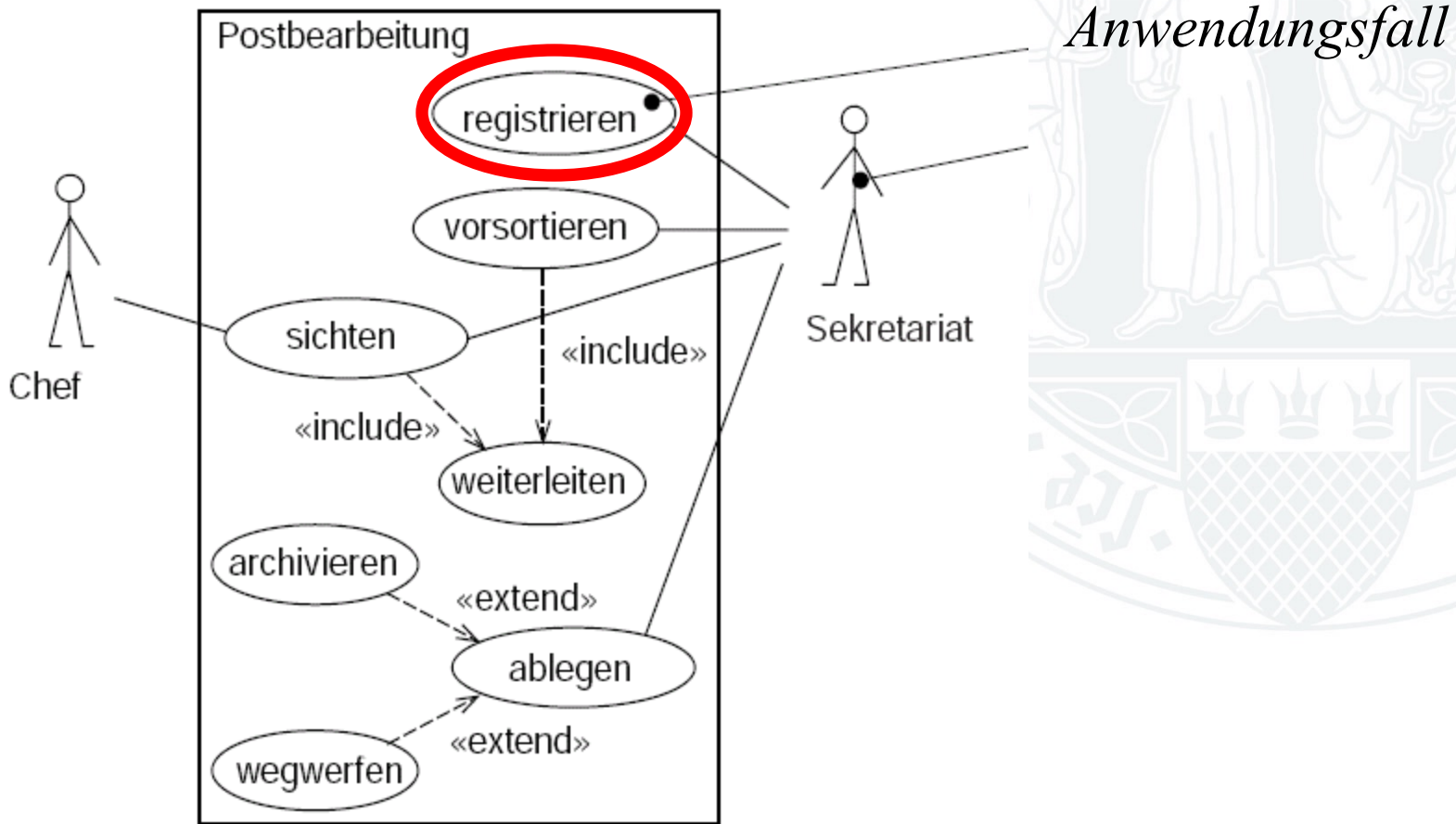
Auch Sonderfälle



# II 2. Anwendungsfalldiagramme

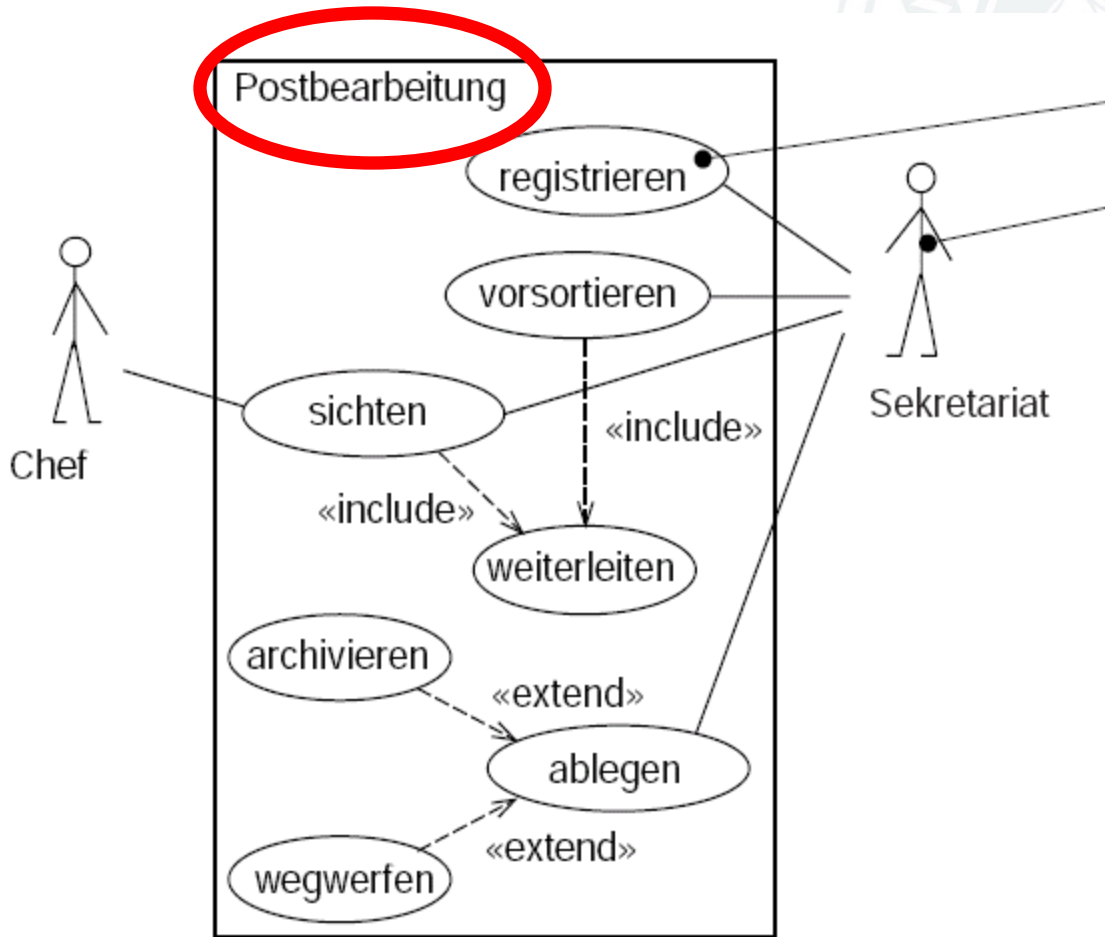


# II 2. Anwendungsfalldiagramme



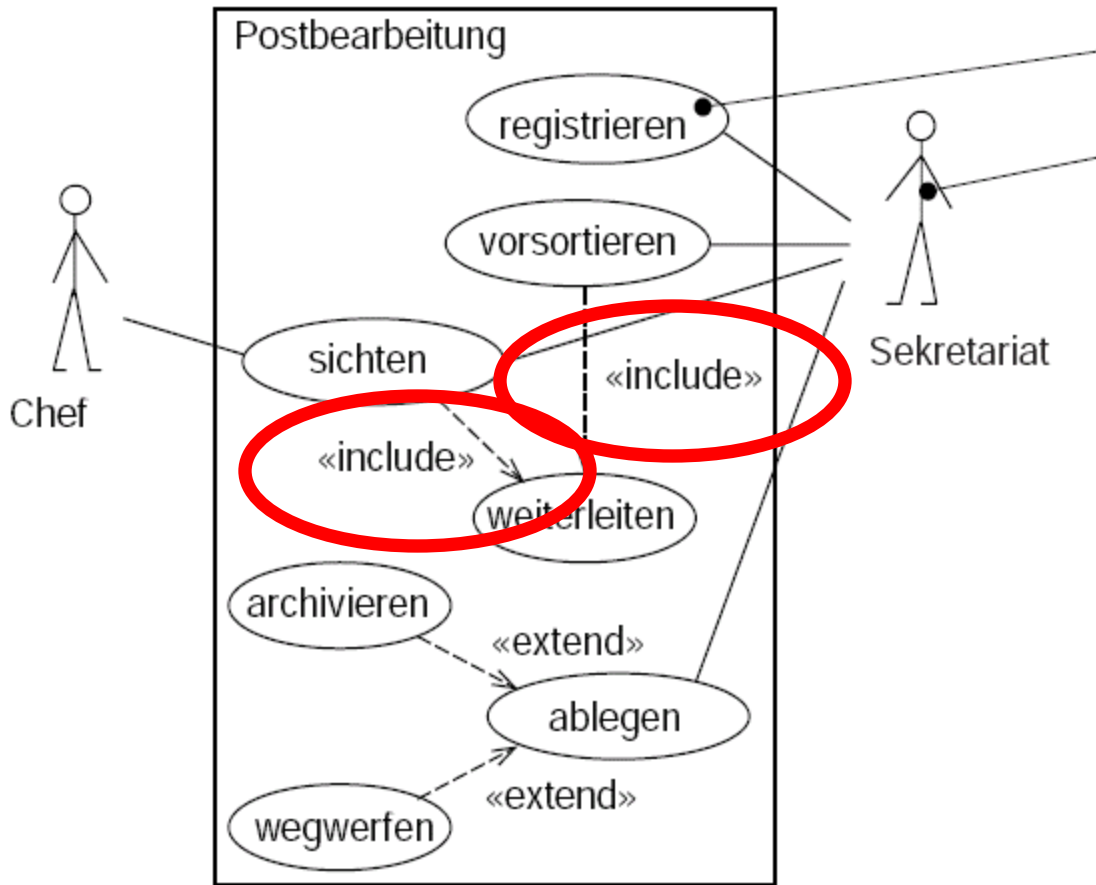


# II 2. Anwendungsfalldiagramme



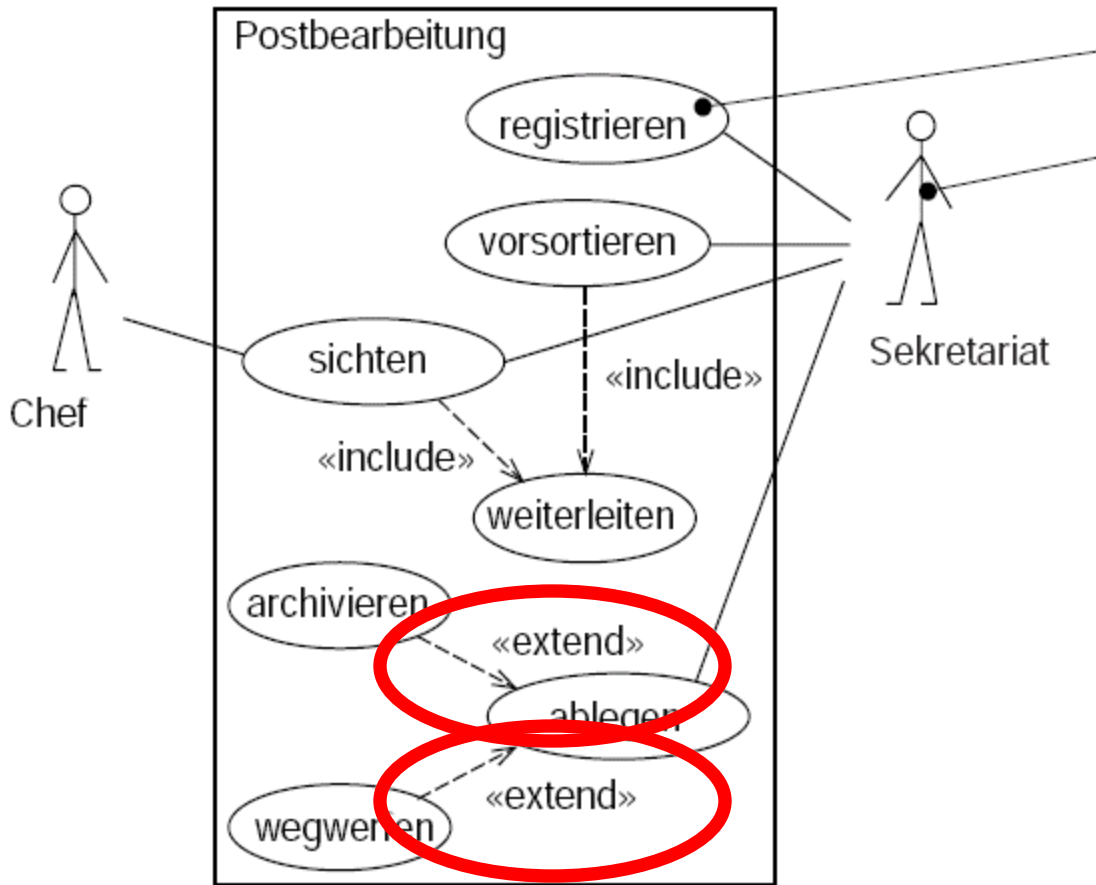
*Anwendungsfalldiagramm*

# II 2. Anwendungsfalldiagramme



*Include:*  
Bindet anderen Anwendungsfall ein, der an mehreren Stellen genutzt werden kann.

# II 2. Anwendungsfalldiagramme



*Extend:*  
Modelliert Varianten, die einen Basisanwendungsfall abwandeln.

# II 3. Zustandsdiagramme

*Zustandsdiagramme* modellieren das dynamische zeitliche Verhalten eines Systems.

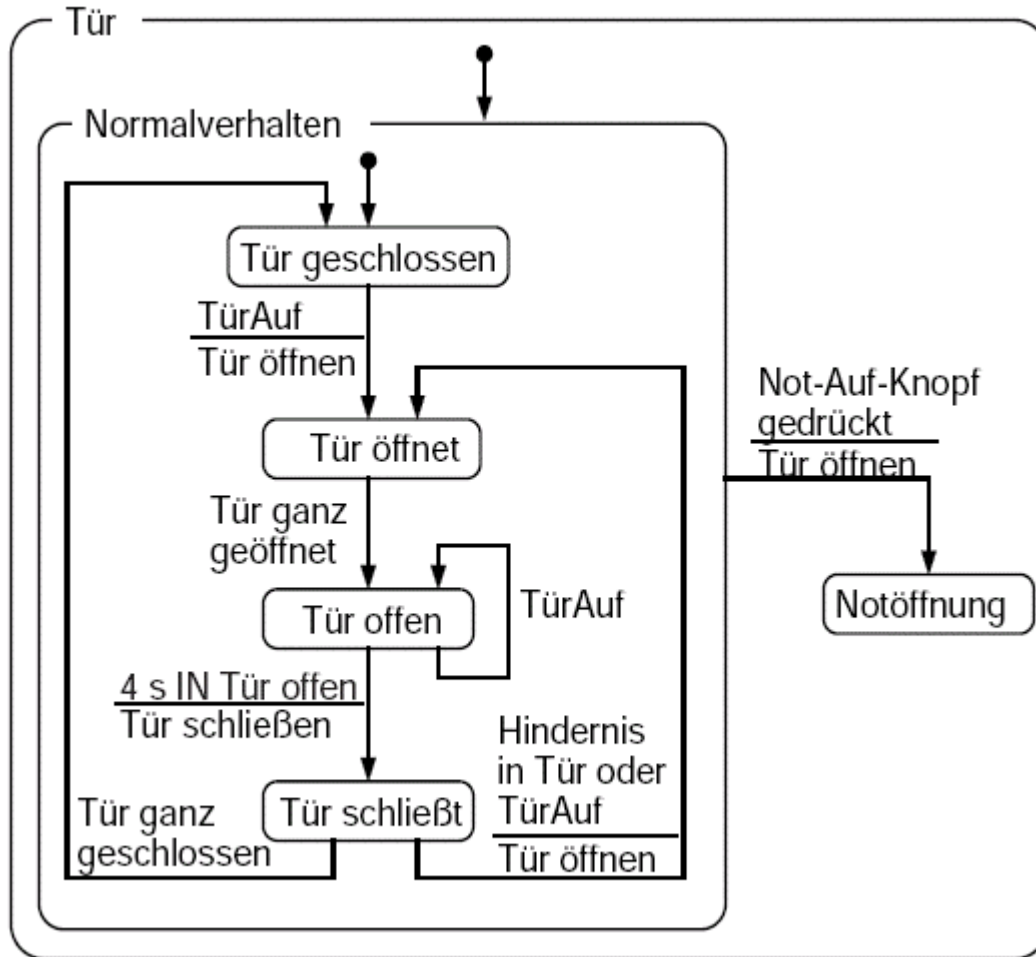
Auch *state machine* → *state diagram*

Mögliche Zustände der Objekte einer Klasse oder eines Teilsystems.

Dynamik des Systemverhaltens: Reaktionen auf äußere Ereignisse.



# II 3. Zustandsdiagramme



# II 4. Aktivitätsdiagramme

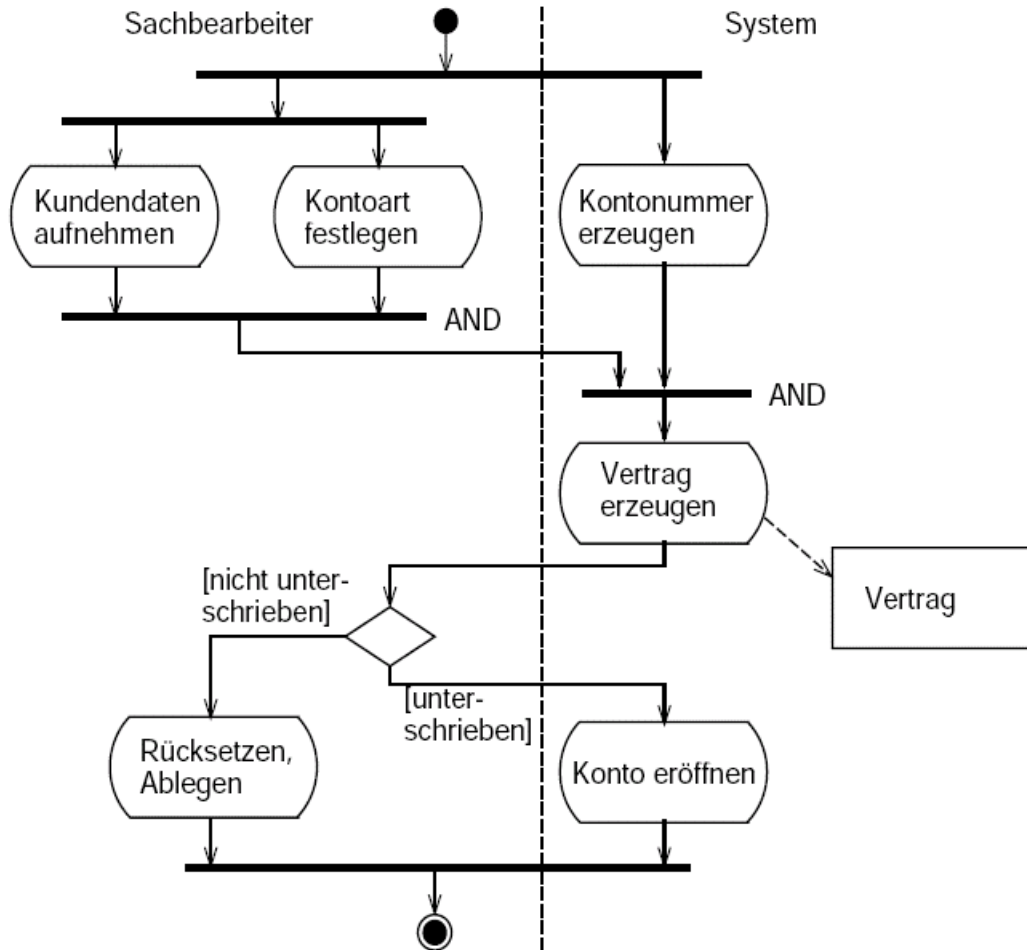
*Aktivitätsdiagramme* beschreiben Abläufe in einem System.

Verbinden *Aktivitäten*, einen *Steuerfluss* und *Objektzustände* miteinander.

Erinnern stark an traditionelle "*Flussdiagramme*" (und haben auch alle ihrer Nachteile).



# II 4. Aktivitätsdiagramme



# II 5. Interaktionssicht

Ziel: Darstellung der Interaktion ausgewählter Objekte in zeitlicher Folge.

Entweder als *Sequenzdiagramme*, die die Zeitachse in den Mittelpunkt rücken ...

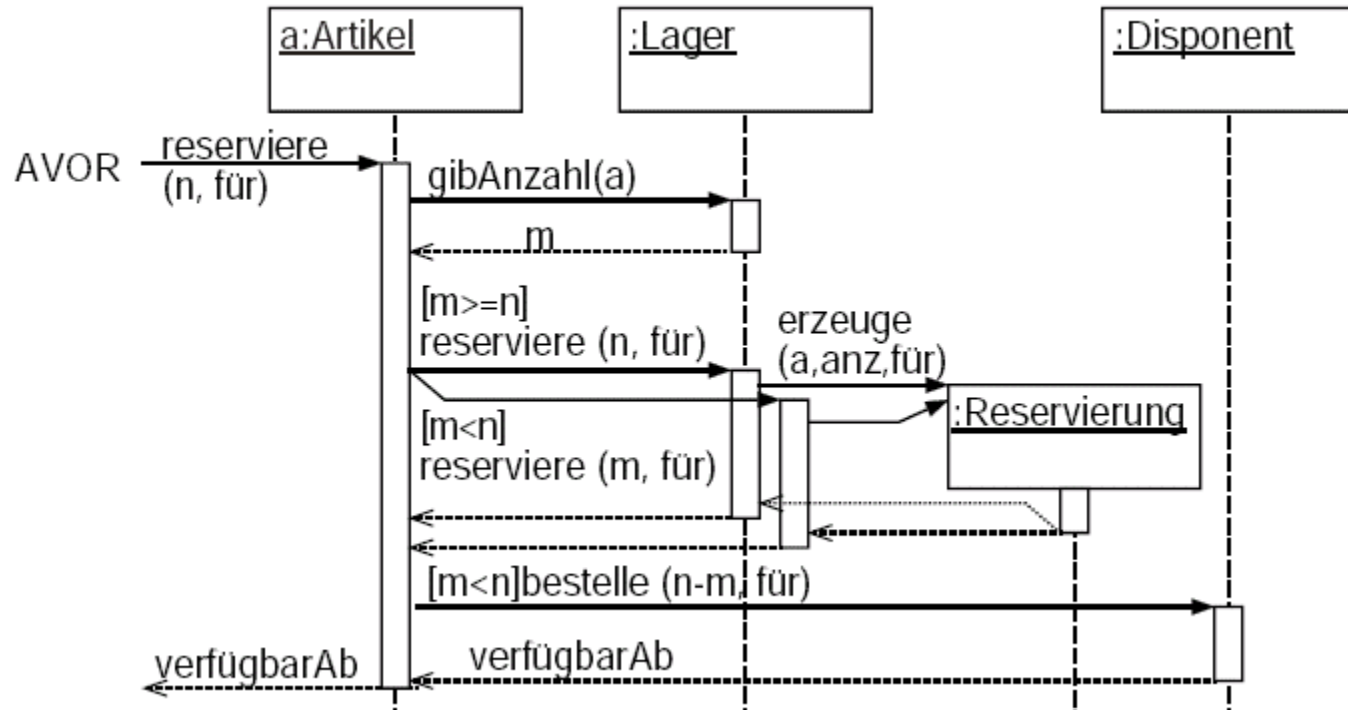
... oder als *Zusammenarbeitsdiagramme* die Objektstruktur und Aufrufe der Objekte in den Vordergrund rücken.

(Beide Diagrammtypen sind logisch äquivalent!)





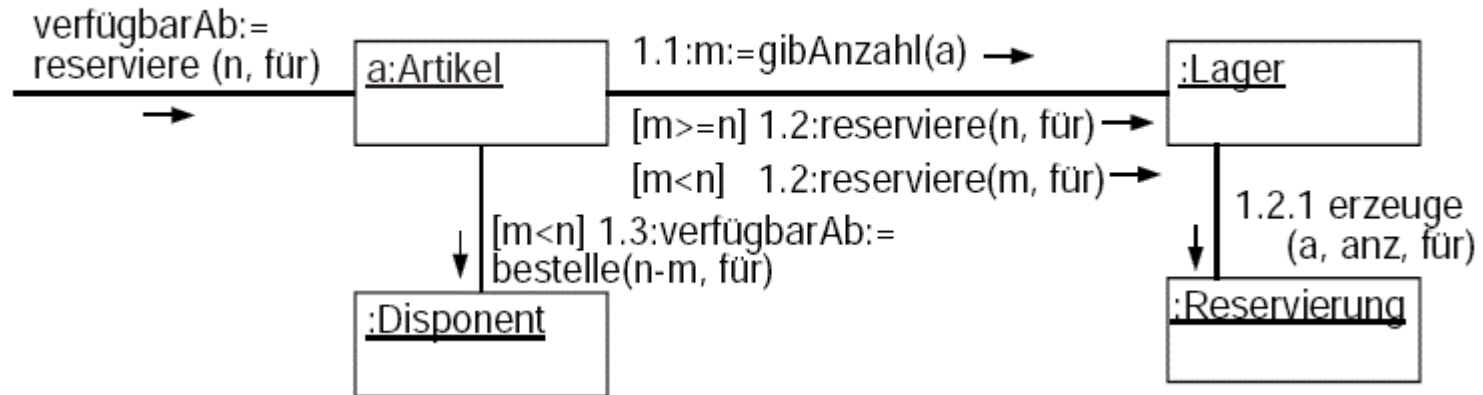
# II 5. Interaktionssicht



Als Sequenzdiagramm ...



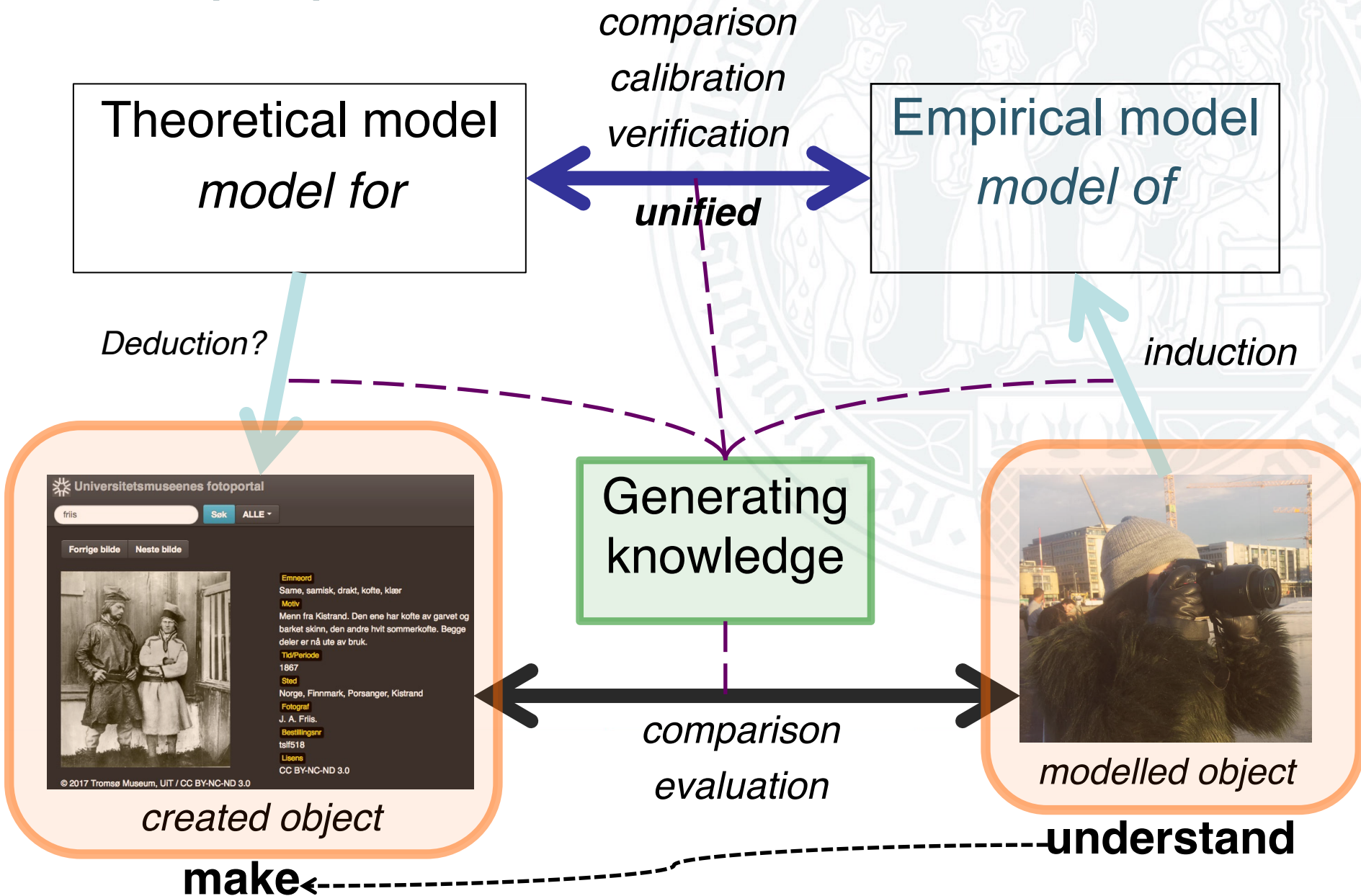
# II 5. Interaktionssicht



... und als Zusammenarbeitsdiagramm.



# The purpose of UML



# Modellierung – Aufgaben

## Hintergrund

Ein Kunstmuseum plant die Entwicklung eines Bildverwaltungssystems für Architektur, Monumente und andere Kunstgegenstände.

Durch eine Informationsveranstaltung mit der Museumsleiterin haben Sie eine kurze Einführung in die Problematik erhalten.



# Modellierung – Aufgaben

## Problembeschreibung

Das Museum wünscht ein System, mit Hilfe dessen sowohl Museumsangestellte, als auch interessierte Bürgerinnen und Bürger Bilder von Kunstwerken einreichen können.

Das System soll zum einen dazu dienen, digitale Bilder über eine einfache Schnittstelle einzureichen.

Zum anderen sollen Scan-Stationen für analoge Positiv- und Negativbilder in den Arbeitsprozess einbezogen werden.

Die Inhaltsanbieterinnen und -bieter sollen Zugang zu einem Computerwerkzeug für die Erstellung von Meta-daten haben.

Diese Metadaten werden später von Museumsangestellten kontrolliert und mit einer Datenbank verknüpft, bevor die Bilder online veröffentlicht werden.

Ebenfalls notwendig ist ein integriertes System für die Speicherung der anfallenden Daten.

# Modellierung – Aufgaben

## Ihre Aufgabe

Formulieren Sie einen ersten Modellversuch für diesen Problemkomplex:

Erstellen Sie ein vorläufiges Model und beschreiben Sie den Modellierungsprozess.

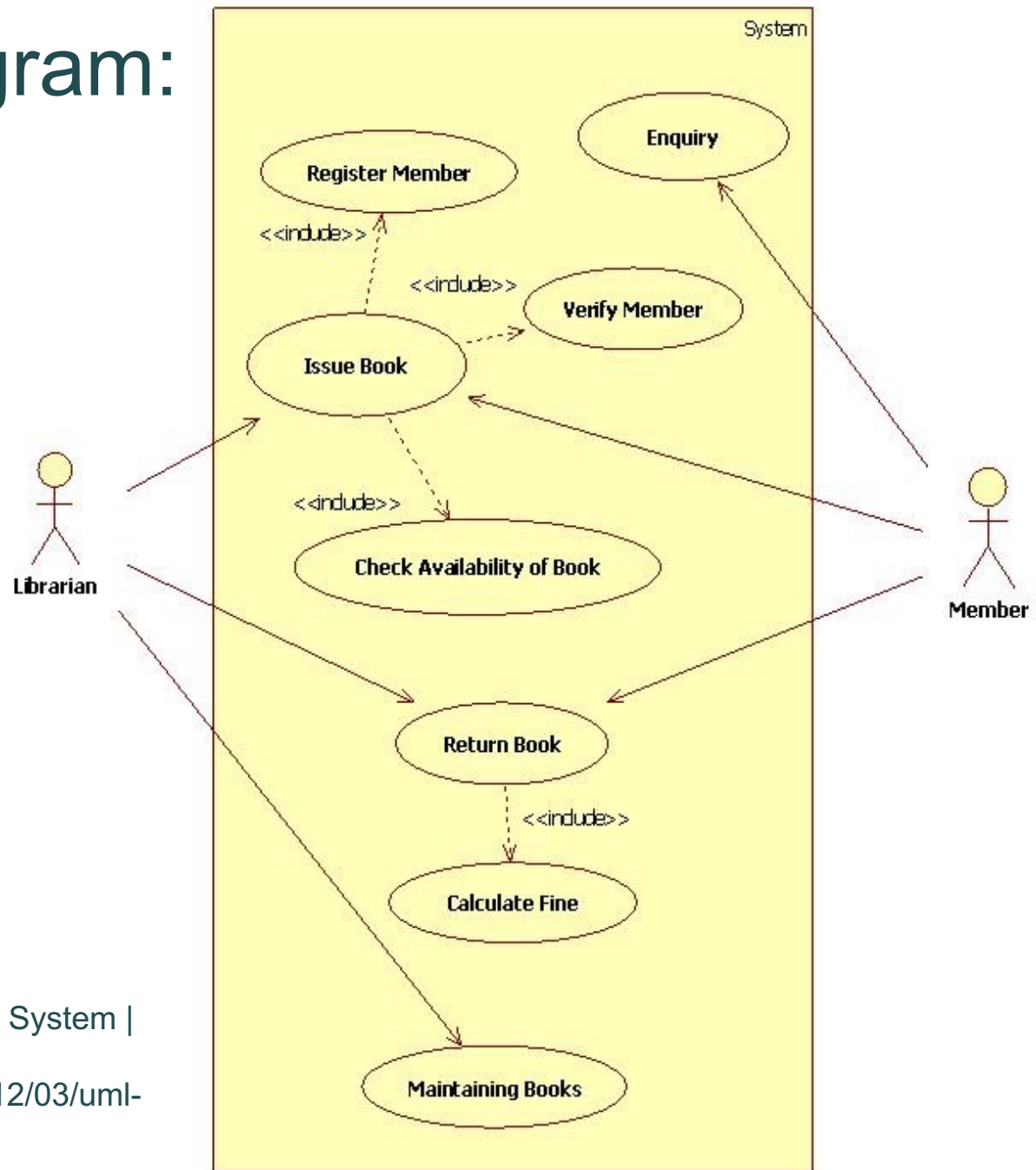
Text und UML-Diagramme zusammen mit textuelle Beschreibungen.

*Einreichung: 7. Mai 17:00*

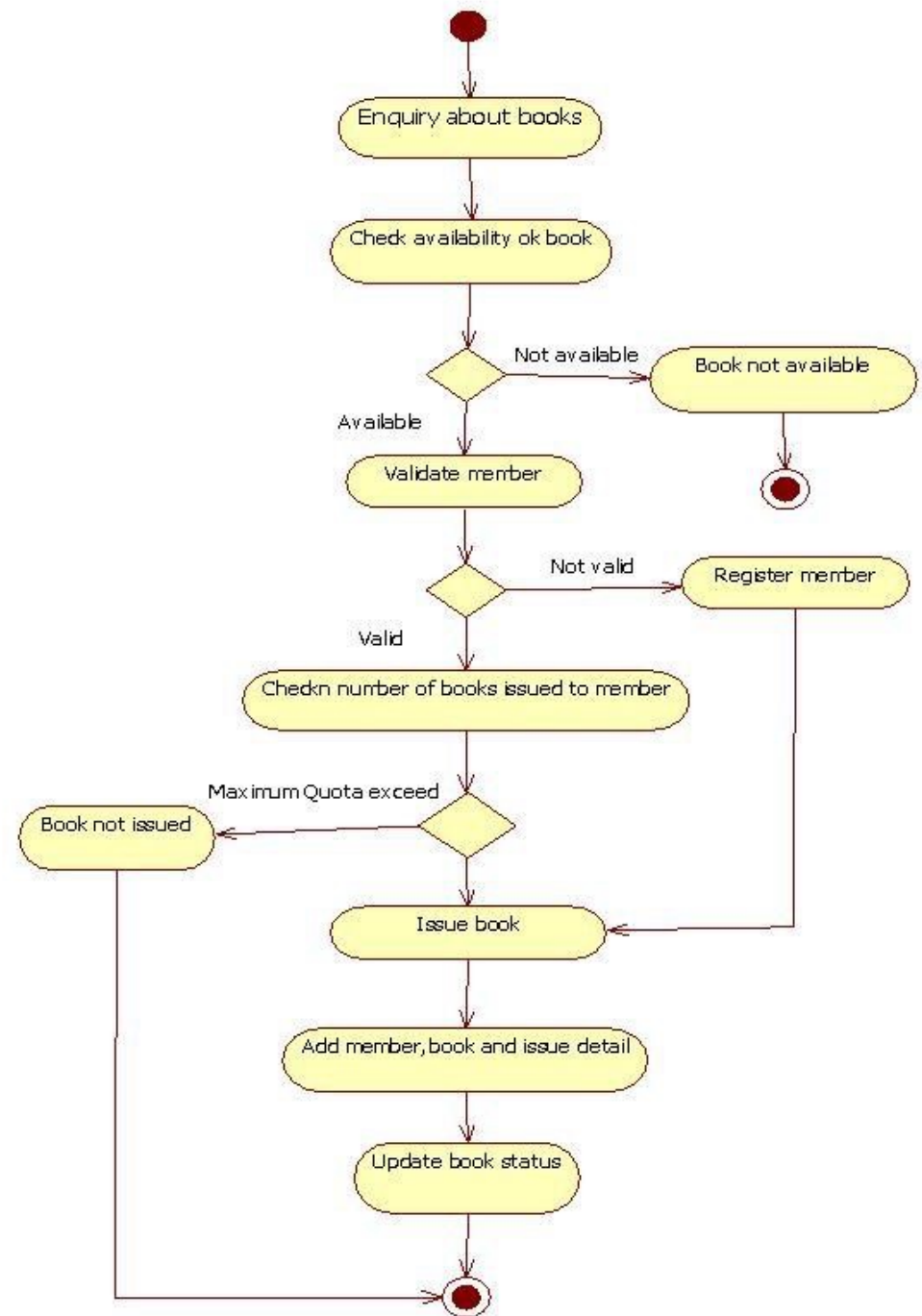
*Kommentaren in der Vorlesung 9. Mai*



# Use Case Diagram: Library Management System

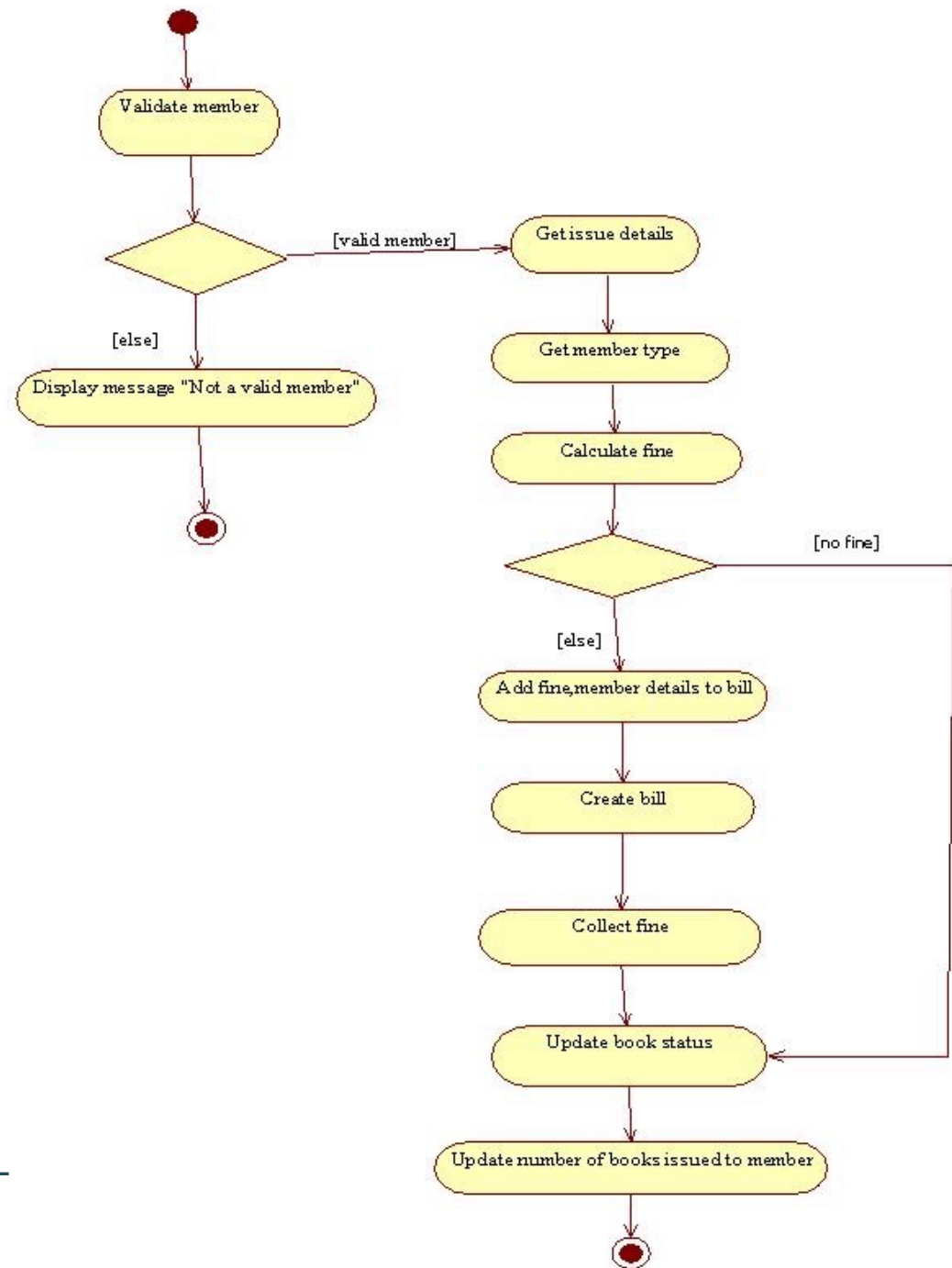


# Activity Diagram for Issue Book in Library

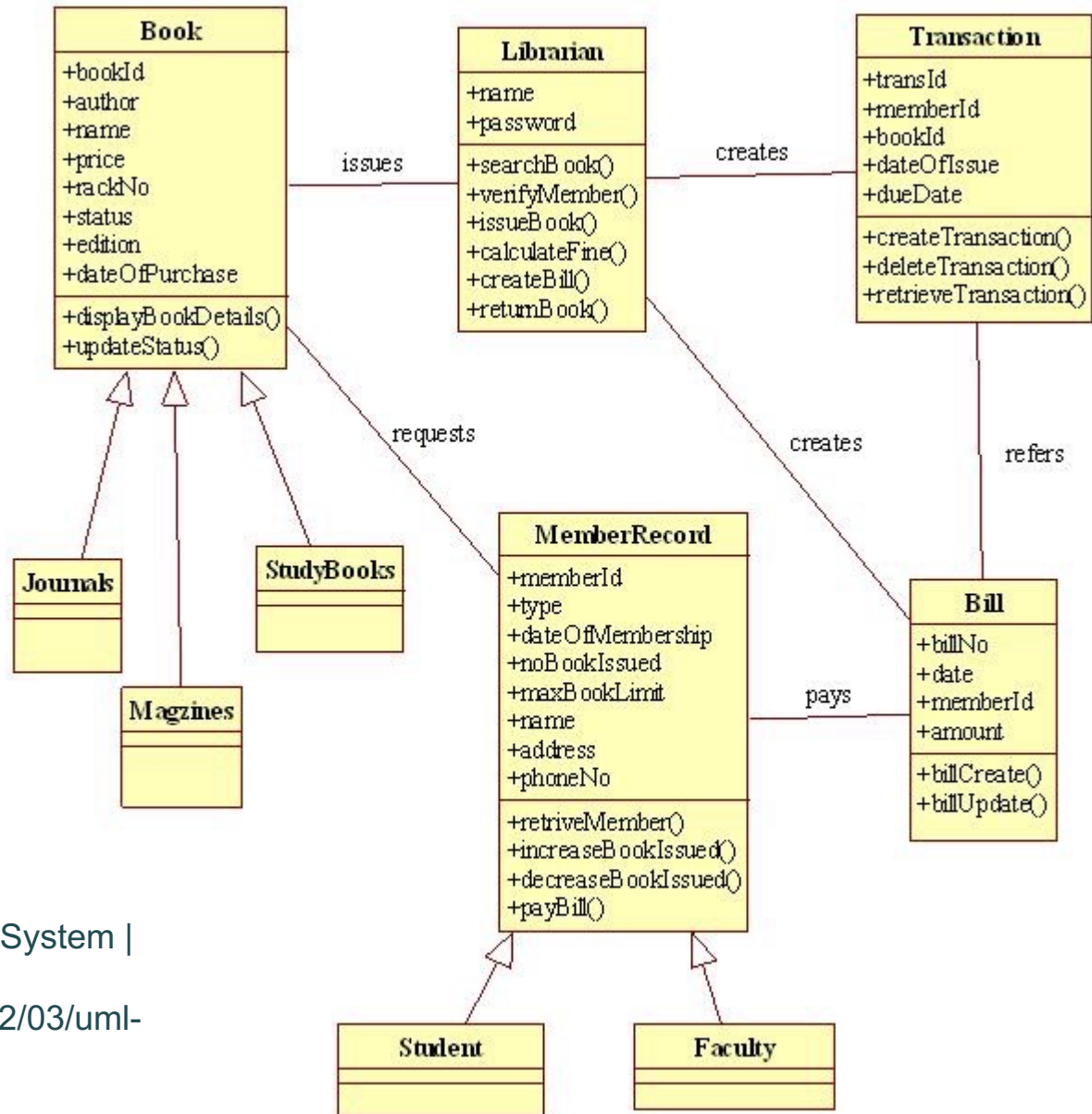




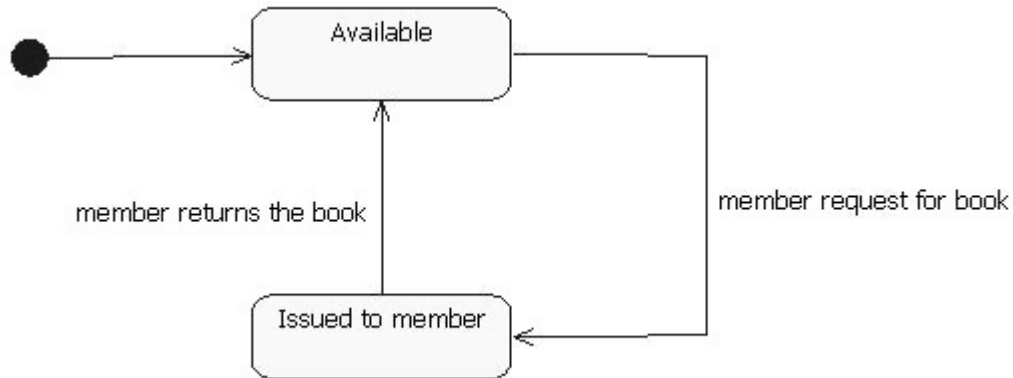
# Activity Diagram for Return Book in Library



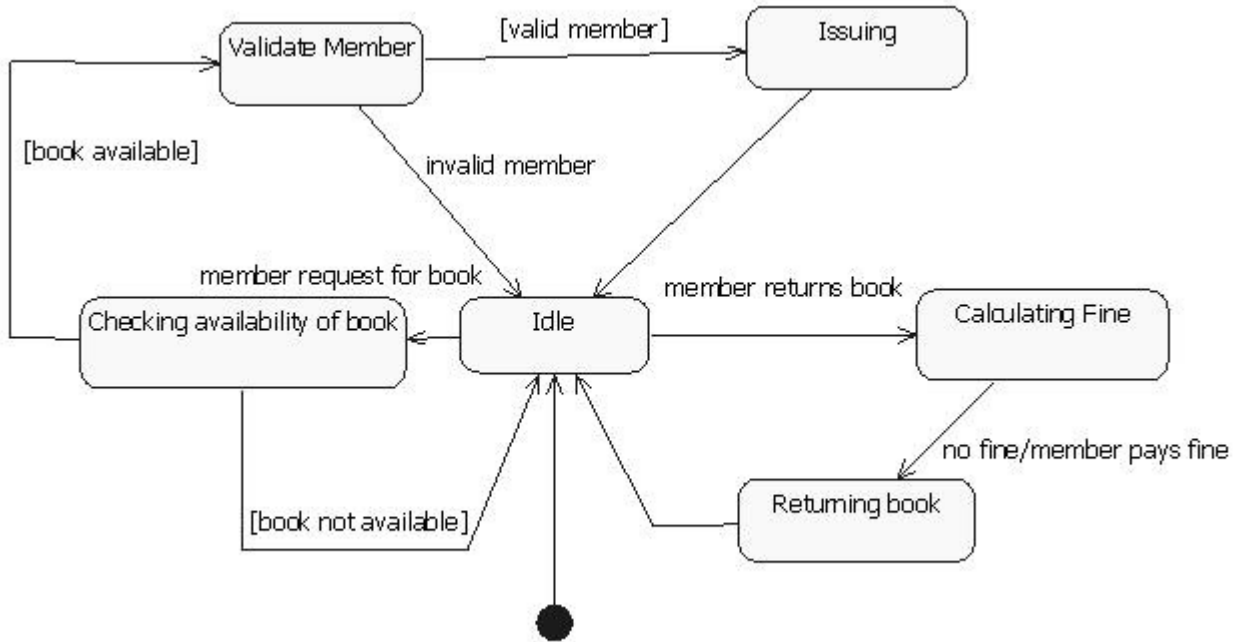
# Class Diagram for Library System



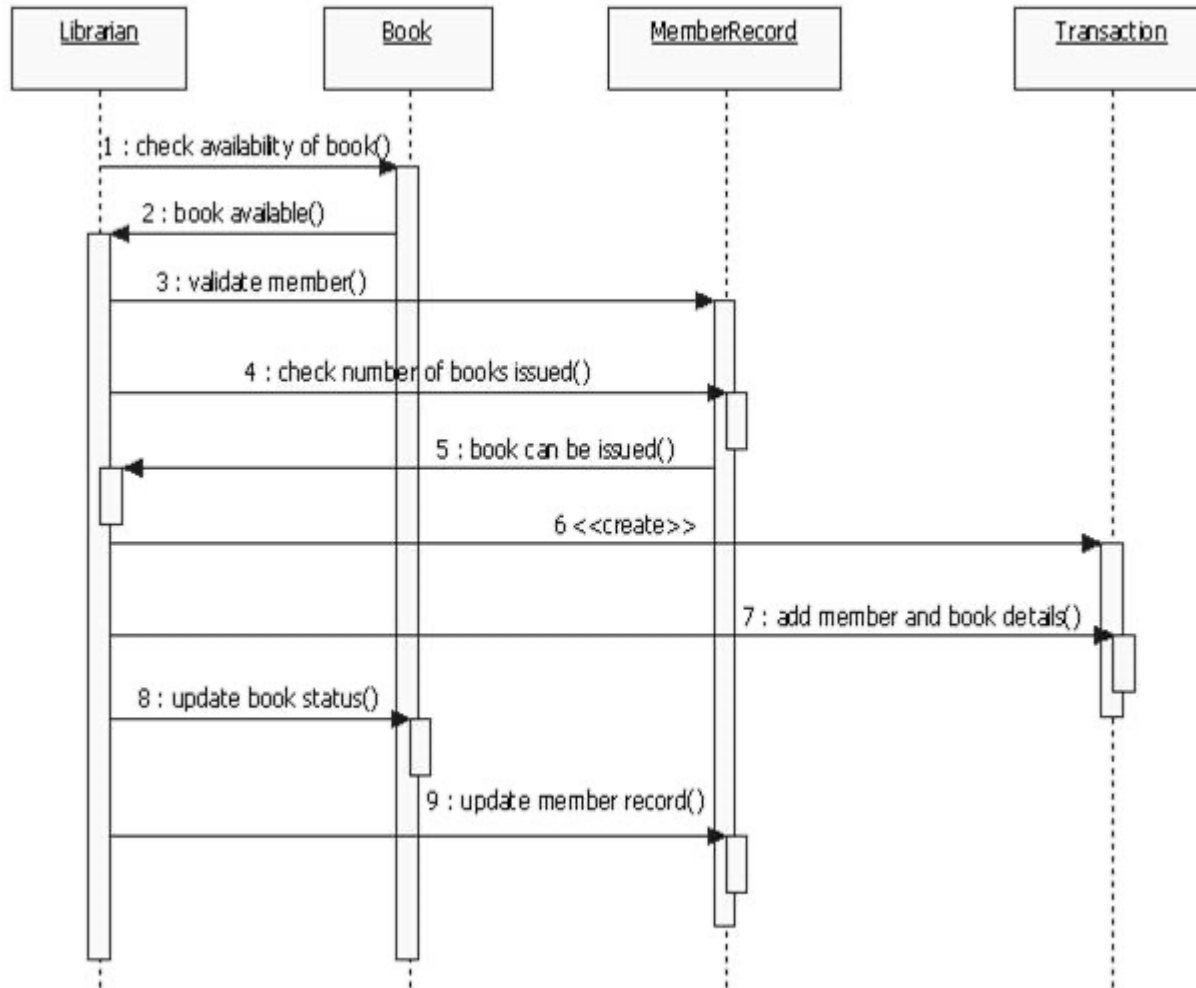
# State diagram for Book



# State diagram for Librarian



# Sequence diagram for issuing book



# Sequence diagram for returning book

