



Universität  
Zürich<sup>UZH</sup>

Institut für Informatik

Norbert Seyff

# Requirements Engineering I

## Szenarien und Anwendungsfälle

# 1 Grundlagen und Terminologie

---

- Idee: Die **Interaktion** zwischen **systemexternen Akteuren** und dem **System** ins Zentrum der Betrachtungen stellen
- Jede Interaktionssequenz wird durch ein **Szenario** (einen **Anwendungsfall**) beschrieben
- ⇒ **Benutzerorientierte Spezifikation** der **Funktionalität**

# Terminologie

---

**Szenario (scenario).** 1. Eine Beschreibung einer möglichen Folge von Ereignissen, die zu einem gewünschten (oder unerwünschten) Ergebnis führen. 2. Eine geordnete Menge von Interaktionen zwischen Partnern, in der Regel zwischen einem System und einer Menge systemexterner Akteure. Kann eine konkrete Interaktionsfolge (Beispielszenario) oder eine Menge möglicher Interaktionen (Typszenario, Anwendungsfall) sein. 3. In UML: Ein konkreter Durchlauf durch einen Anwendungsfall.

**Anwendungsfall (use case).** 1. Eine Beschreibung der möglichen Interaktionen zwischen einem Akteur und einem System, deren Ausführung einen Mehrwert erbringt. 2. Ein Typszenario.

**Akteur (actor).** 1. Eine **Person**, ein **System** oder ein **technisches Element** im **Kontext** eines Systems, der bzw. das mit dem System **interagiert**. 2. Ein Mensch, ein System oder ein technisches Element, der oder das **handelt** und Information verarbeitet, um gewisse **Ziele** zu erreichen.

# Bemerkungen zur Terminologie

---

- In der Terminologie von Jacobson und – in der Folge davon – in UML wird der Name Szenario ausschließlich für **Beispielszenarien** gebraucht.
- In diesem Skript wird Szenario in der **allgemeinen Bedeutung** gebraucht.
- Im Zusammenhang mit Szenarien und Anwendungsfällen wird **Akteur** in der Bedeutung (1) gebraucht.

## 2 Darstellung von Szenarien

---

Drei Teilprobleme:

- Modellierung eines **einzelnen Szenarios**
- **Übersichten** über alle Szenarien eines Systems
- **Zusammenhänge** zwischen Szenarien

## 2.1 Modellierung eines einzelnen Szenarios

---

- Freier Text
- Strukturierter Text
- Zustandsautomaten bzw. Statecharts
- UML-Aktivitätsdiagramme
- Interaktionsdiagramme
- Siehe auch Vorlesung Informatik IIa: Modellierung, Kapitel 10

# Szenariobeschreibung mit freiem Text

---

- Häufig zur Beschreibung konkreter Beispielszenarien verwendet

„Andreas Müller nimmt das Buch, das er ausleihen will, aus dem Regal und bringt es zum Ausleiheschalter. Dort werden seine Ausweiskarte sowie die Buchsignatur gelesen, das Buch als ausgeliehen registriert und das Diebstahlsicherungsetikett deaktiviert. Gleichzeitig wird ein Leihschein ausgedruckt. Anschließend nimmt er das Buch zusammen mit dem Leihschein und verlässt die Bibliothek durch die Diebstahlsicherungs-Schleuse.“

- + Flexibel und ausdrucksmächtig
- + Von Anwendungsexperten les- und schreibbar
- Unpräzise, Missverständnisse leicht möglich
- Fehler werden leicht übersehen

# Szenariobeschreibung mit strukturiertem Text

---

- Am häufigsten verwendete Form zur Beschreibung von Anwendungsfällen bzw. Typszenarien
- Angabe des Hauptakteurs und des Auslösers für die Ausführung des Szenarios
- Ablauf der Schritte im Normalfall
- Kennzeichnung der Reihenfolge durch Nummerierung der Schritte oder Interaktionsfolgen
- Angabe möglicher Ausnahmefälle



# Szenariobeschreibung mit strukturiertem Text – 2

---

Akteur(e): Benutzerin

Auslöser: Eine Benutzerin bringt ein Buch oder mehrere Bücher, das/die sie ausleihen möchte, zum Ausleiheschalter

Normalablauf:

1. Ausweiskarte der Benutzerin lesen und Angaben überprüfen
2. Signatur eines Buchs lesen und zugehörigen Katalogeintrag ermitteln
3. Ausleihe registrieren und Diebstahlsicherungsetikett deaktivieren
4. Wenn mehrere Bücher auszuleihen sind, mit den weiteren Büchern nach 2. und 3. verfahren
5. Leihschein drucken für alle ausgeliehenen Bücher
6. Der Benutzerin Bücher aushändigen, Vorgang abschließen

# Szenariobeschreibung mit strukturiertem Text – 3

---

## Alternative Abläufe:

- 1.1 Ausweiskarte nicht vorhanden oder gelesene Ausweiskarte ist ungültig: Vorgang abbrechen
- 2.1 Buch ist vorgemerkt für andere Person: Buch zur Seite legen, mit Schritt 4 fortfahren
- 2.2 Benutzerin hat mehr als ein überfälliges Buch nicht zurückgebracht: Vorgang abbrechen

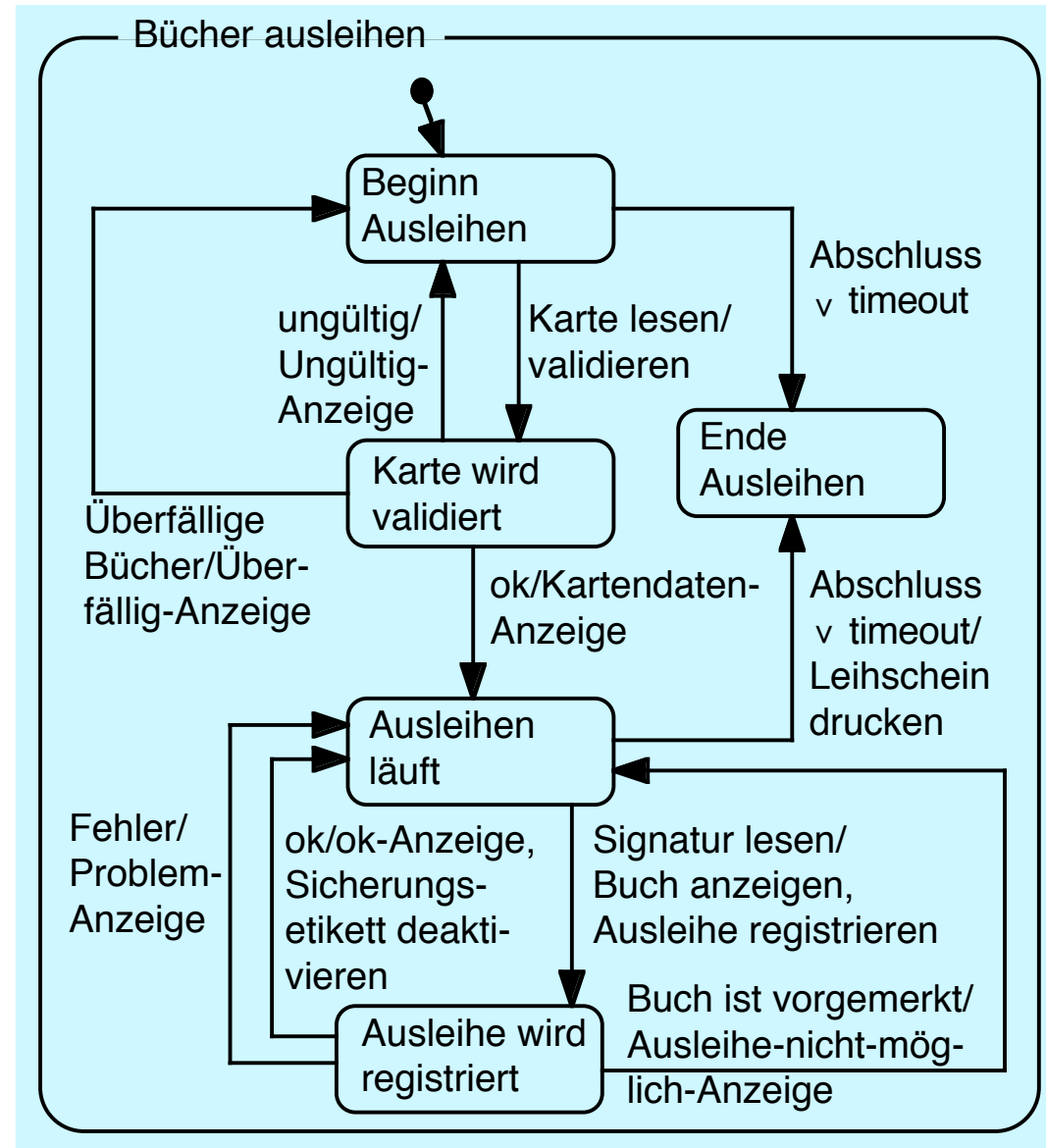
# Szenariobeschreibung mit strukturiertem Text – 4

---

- + Flexibel und ausdrucksmächtig
- + Von Anwendungsexperten les- und schreibbar
- + Präziser als freier Text, weniger Auslassungen und Fehler
- Aber nach wie vor oft zu unpräzise und fehlerträchtig
- Zusammenhänge mit anderen Szenarien/Anwendungsfällen werden nicht erfasst

# Szenariobeschreibung mit Statecharts

- Ereignis: **Anstoß** durch einen Akteur
- Aktion(en): **Reaktion** des Systems
- Zustand: **wann** ist **was** möglich

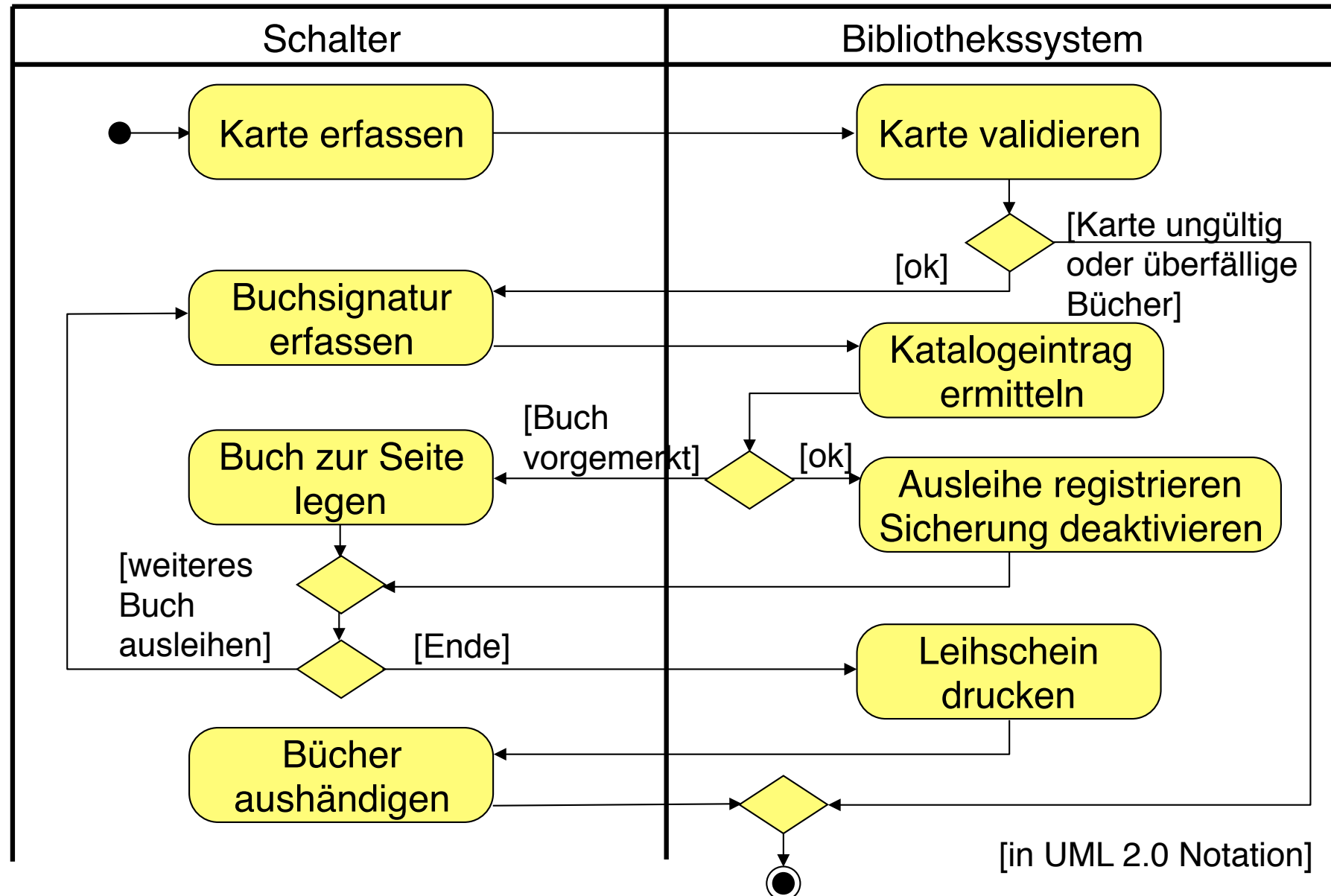


# Szenariobeschreibung mit Statecharts – 2

---

- + Aussagekräftig
- + Wählbarer Grad an Präzision (in der Beschreibung der Ereignisse und Aktionen)
- + Zusammenhänge mit anderen Szenarien/Anwendungsfällen sind modellierbar
- o Von Anwendungsexperten verstehbar (erfordert aber Unterstützung oder Ausbildung)
- Braucht Modellierungsexperten zur Erstellung

# Szenariobeschreibung mit UML-Aktivitätsdiagramm



# Szenariobeschreibung mit UML-Aktivitätsdiagramm 2

---

- Schritte als **Aktivitäten** modelliert
- **Trennung** von **Aktion** (durch Akteur) und **Reaktion** (des Systems) möglich (wird nicht immer gemacht)
- Modelliert Normal- und Fehlerfälle
- Bezüglich Vor- und Nachteilen **ähnlich wie Statecharts**

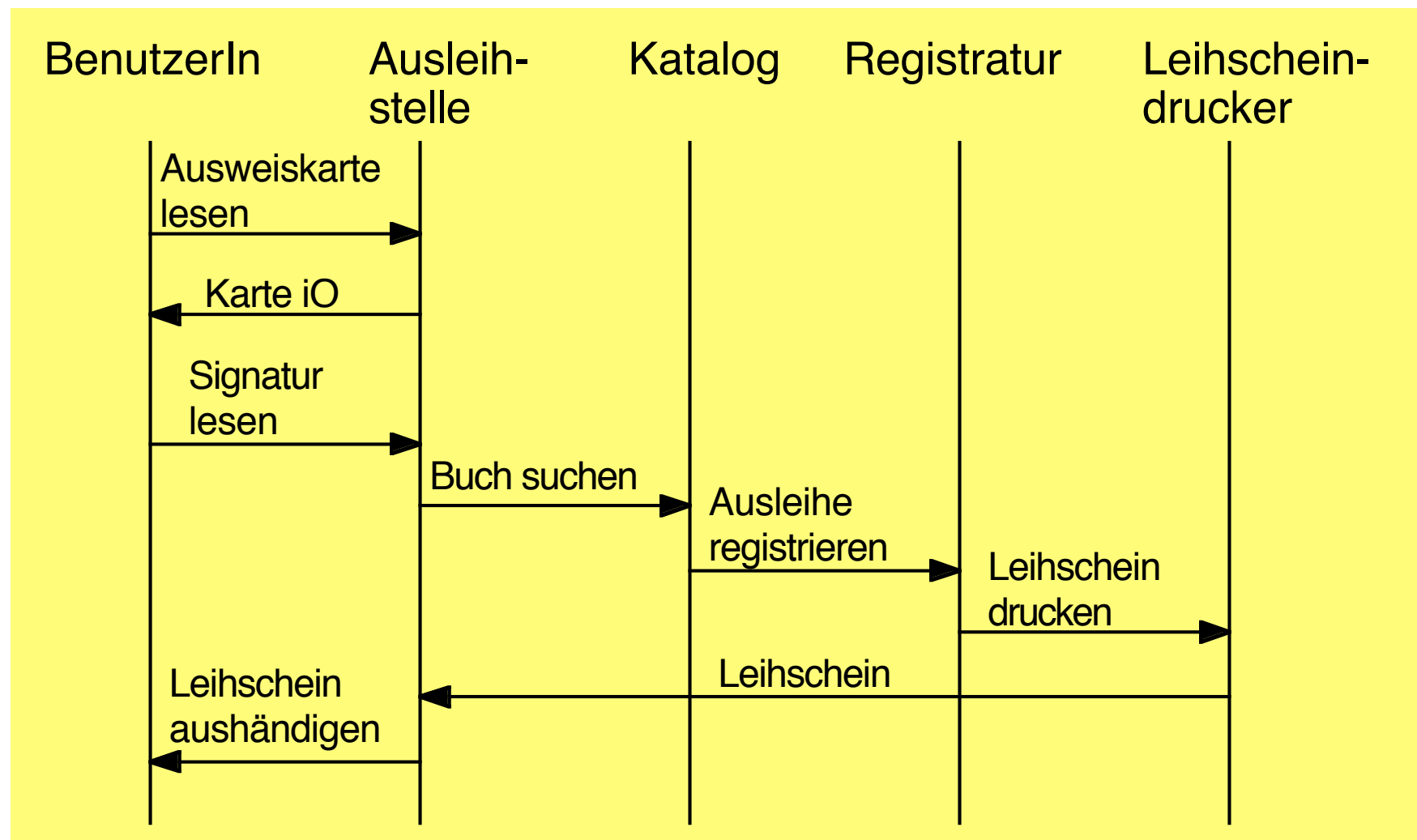
# Szenariobeschreibung mit Interaktionsdiagrammen

---

- Interaktionsdiagramme modellieren den **zeitlichen Ablauf** des **Austauschs von Ereignissen / Nachrichten** zwischen einer Menge von Partnern
  - Interaktionsdiagramme modellieren eine **konkrete Sequenz** von Interaktionen
    - ⇒ Modellierung von **Beispielszenarien**
  - Es gibt **verschiedene Ausprägungen** von Interaktionsdiagrammen
    - Sequenzdiagramme (sequence diagrams), Message sequence diagrams, MSCs
    - Heute überwiegend **UML Sequenzdiagramme**
- + Eingängige, intuitiv verständliche Notation**
- Nur für Beispielszenarien**



# Szenariobeschreibung mit Interaktionsdiagrammen – 2



## 2.2 Szenario-Übersichten

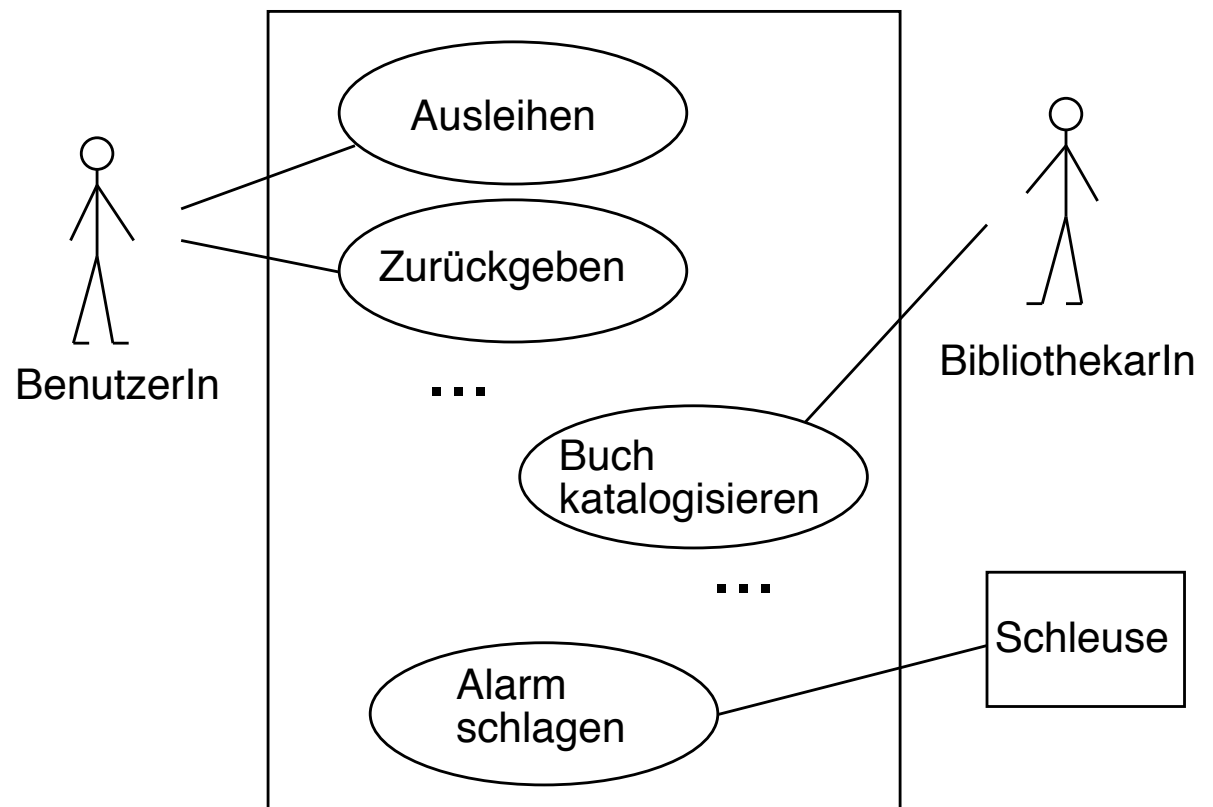
---

- UML Anwendungsfalldiagramm
- Einbettung in Objektmodell oder Subsystem-Modell

# UML Anwendungsfalldiagramm

---

- Überblick über alle Anwendungsfälle (Typszenarien) eines Systems
- Ist eine Art Kontextdiagramm



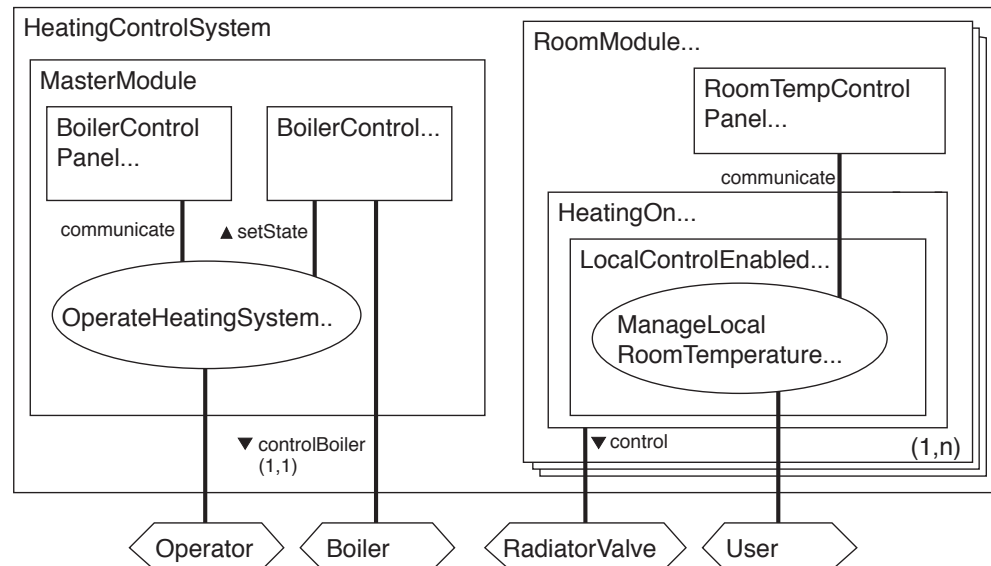
# UML Anwendungsfalldiagramm – 2

---

- + Zeigt, welche Akteure in welchen Anwendungsfällen mit dem System interagieren
- + Bei kleinen Modellen **übersichtlich**
- Modelliert Zusammenhänge zwischen Anwendungsfällen nur rudimentär
- Modelliert die Inhalte der Anwendungsfälle nicht
- Kennt **keine Zerlegung**

# Einbettung in Objektmodell

- Das System wird **hierarchisch** in Teilprobleme zerlegt
- Jedes Teilproblem wird durch ein **abstraktes Objekt** modelliert
- **Szenarien** werden in Teilproblemen **eingebettet**
- Beispiel (in ADORA):



- In UML ist eine Einbettung von Anwendungsfällen in Subsysteme (und damit indirekt eine gewisse Dekomposition) möglich

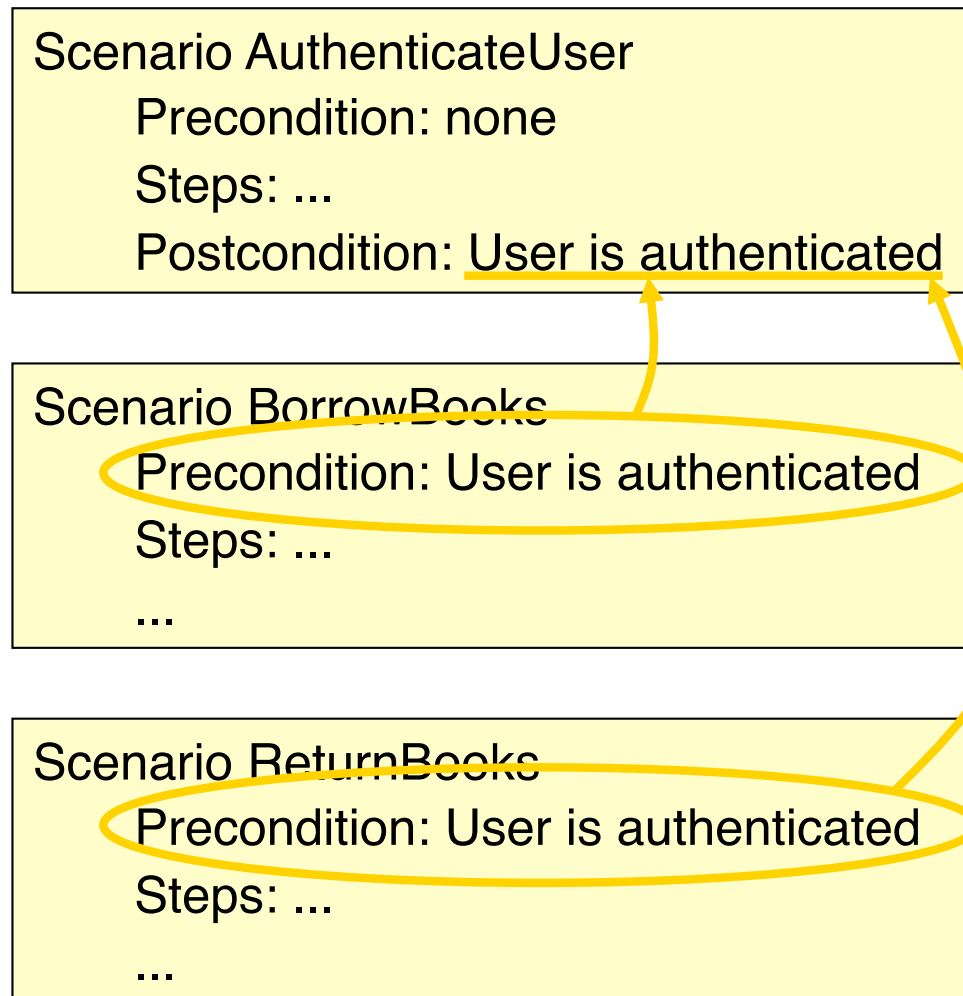
## 2.3 Zusammenhänge zwischen Szenarien

---

- Problem wird bisher weitgehend ignoriert (Glinz 2000b)
- Möglich beispielsweise mit
  - Voraussetzungen und Ergebniszusicherungen (Pre- / Postconditions) in strukturierten Szenariobeschreibungen
  - Statecharts (Glinz 1995, Glinz 2000a)
  - Erweiterten Jackson-Diagrammen (ADORA, Glinz et al. 2002)
  - speziellen Abhängigkeitsdiagrammen (Ryser und Glinz 2001)

# Zusammenhänge über Zusicherungen

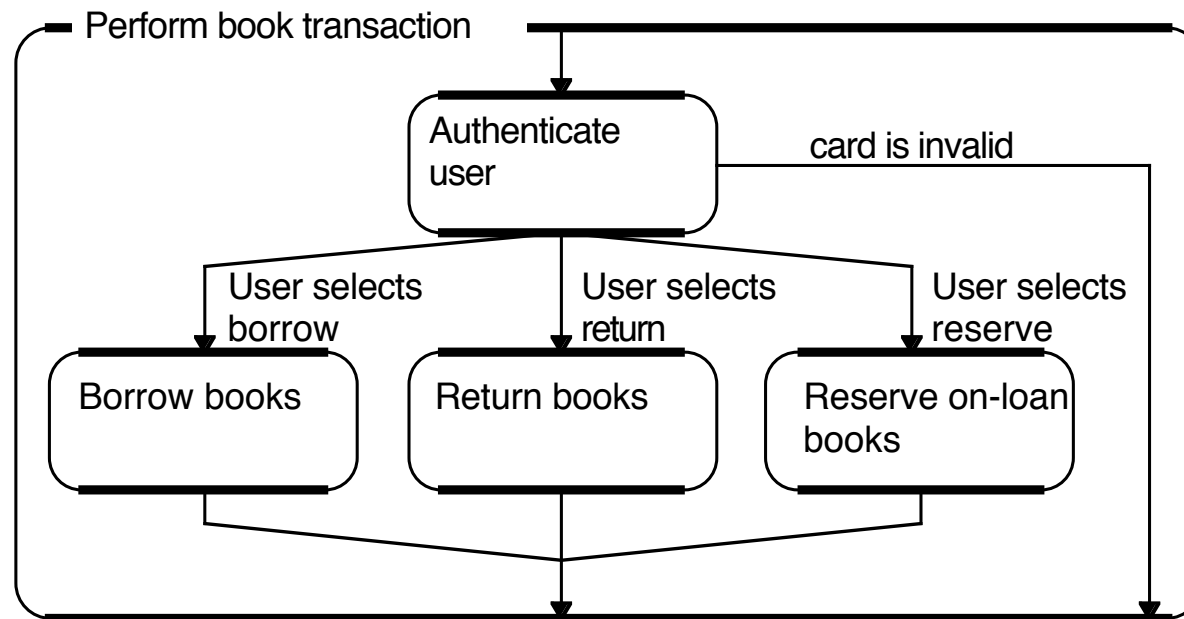
---



- Einfache Zusammenhänge der Art «B nach A» modellierbar
- Unübersichtlich
- Modellierung komplexer Zusammenhänge sehr aufwendig

# Zusammenhänge mit Statecharts

- Jedes Szenario wird als **Elementarzustand** oder als **Statechart** mit genau einem Anfangs- und Endzustand\* modelliert Glinz 2000a)
- Die klassischen Zusammenhänge (**Sequenz**, **Alternative**, **Iteration** und **Parallelität**) sind modellierbar
- **Forschungsergebnis**; in der Praxis (noch) nicht verwendet



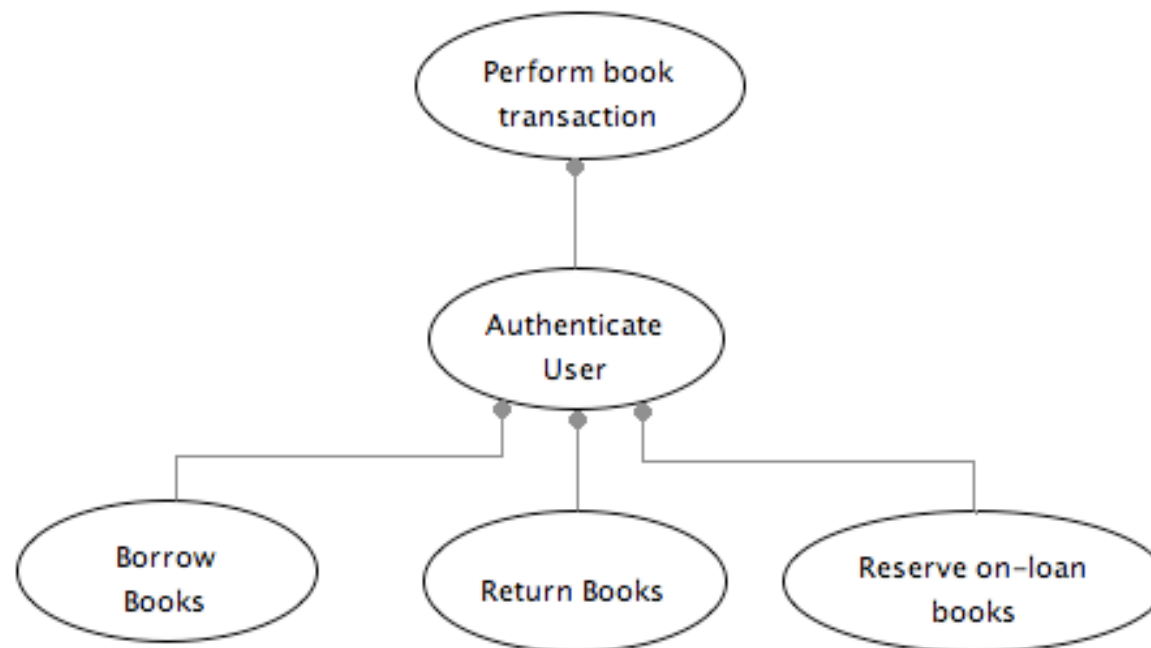
\*symbolisiert durch Balken oben und unten



# Erweiterte Jackson-Diagramme

---

- Wird in **ADORA** zur Modellierung von Szenario-Zusammenhängen verwendet





# 3 Methodik der Modellierung mit Szenarien

---

- Primär Szenarienanalyse
- Sekundär Ereignis/Reaktionsanalyse:
  - Bestimmung von Ereignis/Reaktionsketten, die das System aus einem Ruhezustand wieder in einen Ruhezustand bringen  
[Ein System ist (bezogen auf einen bestimmten Akteur) im Ruhezustand, wenn es keine pendenten Arbeiten für diesen Akteur zu erledigen hat.]
  - Nötigenfalls solche Folgen durch Beispielszenarien beschreiben
  - Zu Typszenarien/Anwendungsfällen abstrahieren
- Siehe auch Vorlesung Informatik IIa: Modellierung

# Methodik der Modellierung mit Szenarien – 2

---

- Jedes Typszenario / jeden Anwendungsfall **beschreiben** (mit strukturiertem Text oder präziser mit Statecharts bzw. Aktivitätsdiagrammen)
- Dabei **Normal- und Ausnahmeabläufe** beschreiben
- **Anwendungsfalldiagramm** als **Übersicht** modellieren
- Bei vielen, feingranular beschriebenen Typszenarien/Anwendungsfällen:  
Übersicht schaffen durch **Zusammenfassen** zusammengehöriger Szenarien/Anwendungsfälle zu einem übergeordneten, **vergrößerten** Szenario/Anwendungsfall

# Aufgabe: Szenarien bestimmen

---

Gegeben sei die Fallstudie «Institutsbibliothek».

- a) Bestimmen Sie die groben Typszenarien/Anwendungsfälle
- b) Betrachten Sie die Typszenarien/Anwendungsfälle für den Akteur *Bibliotheksbenutzer* im Detail

## 4 Verknüpfung von Szenarienmodellen mit Objekt- oder Klassenmodellen

---

- **Szenarien** modellieren die **Benutzersicht**; ignorieren Zusammenhänge, Daten und die zur Erzeugung der Resultate notwendigen Operationen weitgehend
- **Objekt- und Klassenmodelle** modellieren **Zusammenhänge**, **Daten**, und die zur Erzeugung der Resultate notwendigen **Operationen**; ignorieren aber die Benutzersicht weitgehend
- ⇒ **Beide** Sichten sind **notwendig**
- ⇒ Die Sichten müssen miteinander **verknüpft** werden

# Möglichkeiten der Verknüpfung

---

- **Integration** der Szenarien in ein Objektmodell
  - In ADORA gewählter Ansatz
  - In UML wegen der losen Kopplung der Teilmodelle nicht möglich
- **Querverweise**, vor allem aus den Szenarien ins Objekt- bzw. Klassenmodell
  - Auch in Sprachen mit loser Kopplung zwischen den Teilmodellen (z.B. UML) machbar
  - Beispiel:
    - 3 Benutzerin liest Buchcode ein
    - System identifiziert das Buch (Buch.Identifizieren), registriert die Ausleihe (Buch.Ausleihen), deaktiviert das Diebstahletikett
- In der Praxis fehlt diese Integration heute meistens