

Martin Glinz

Requirements Engineering I

Kapitel 8

Szenarien und Anwendungsfälle



Universität Zürich
Institut für Informatik

8.1 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). 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 (abstraktes Szenario oder **Typszenario**) sein.

Anwendungsfall (use case). 1. Ein Typszenario. 2. Eine durch genau einen Akteur angestoßene Folge von Systemereignissen, welche für den Akteur ein Ergebnis produziert und an welchem weitere Akteure teilnehmen können (Jacobson et al. 1992).

Akteur (actor). 1. Ein **Mensch**, der **Ziele** hat und zu deren Erreichung **handelt** oder ein **Element des Anwendungsbereichs**, das zur Erreichung bestimmter Ziele dient und hierzu handeln und/oder Informationen verarbeiten kann. 2. Eine **Rolle**, welche ein **externes System** oder ein **Mensch** gegenüber dem System einnehmen kann.

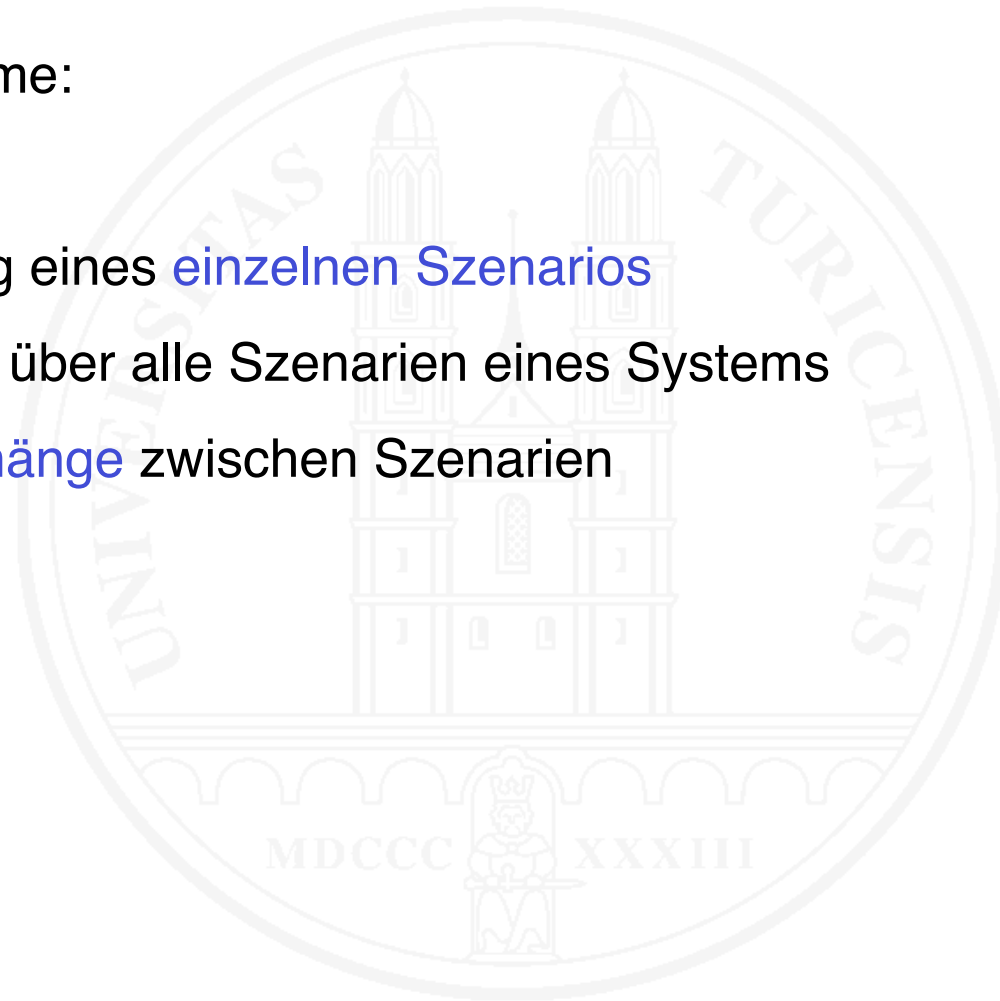
Bemerkungen zur Terminologie

- 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 2. gebraucht.

8.2 Darstellung von Szenarien

Drei Teilprobleme:

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



8.2.1 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:

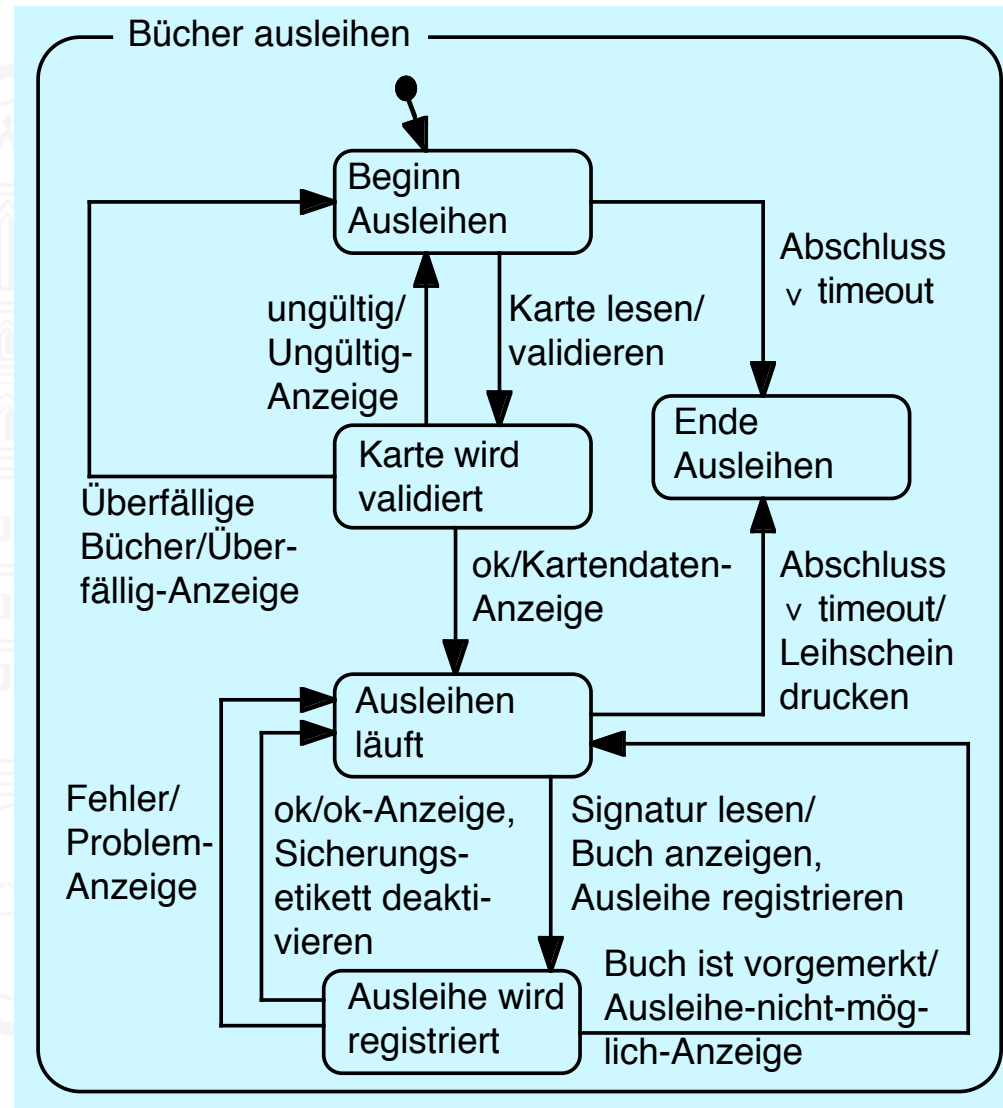
- 1.1 Ausweiskarte nicht vorhanden oder gelesene Ausweiskarte ist ungültig: Vorgang abbrechen
- 2.1 Buch ist vorgemerkt für andere Person: Buch zur Seite legen, mit Schritt 4 fortfahren
- 2.2 Benutzerin hat mehr als ein überfälliges Buch nicht zurückgebracht: Vorgang abbrechen

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

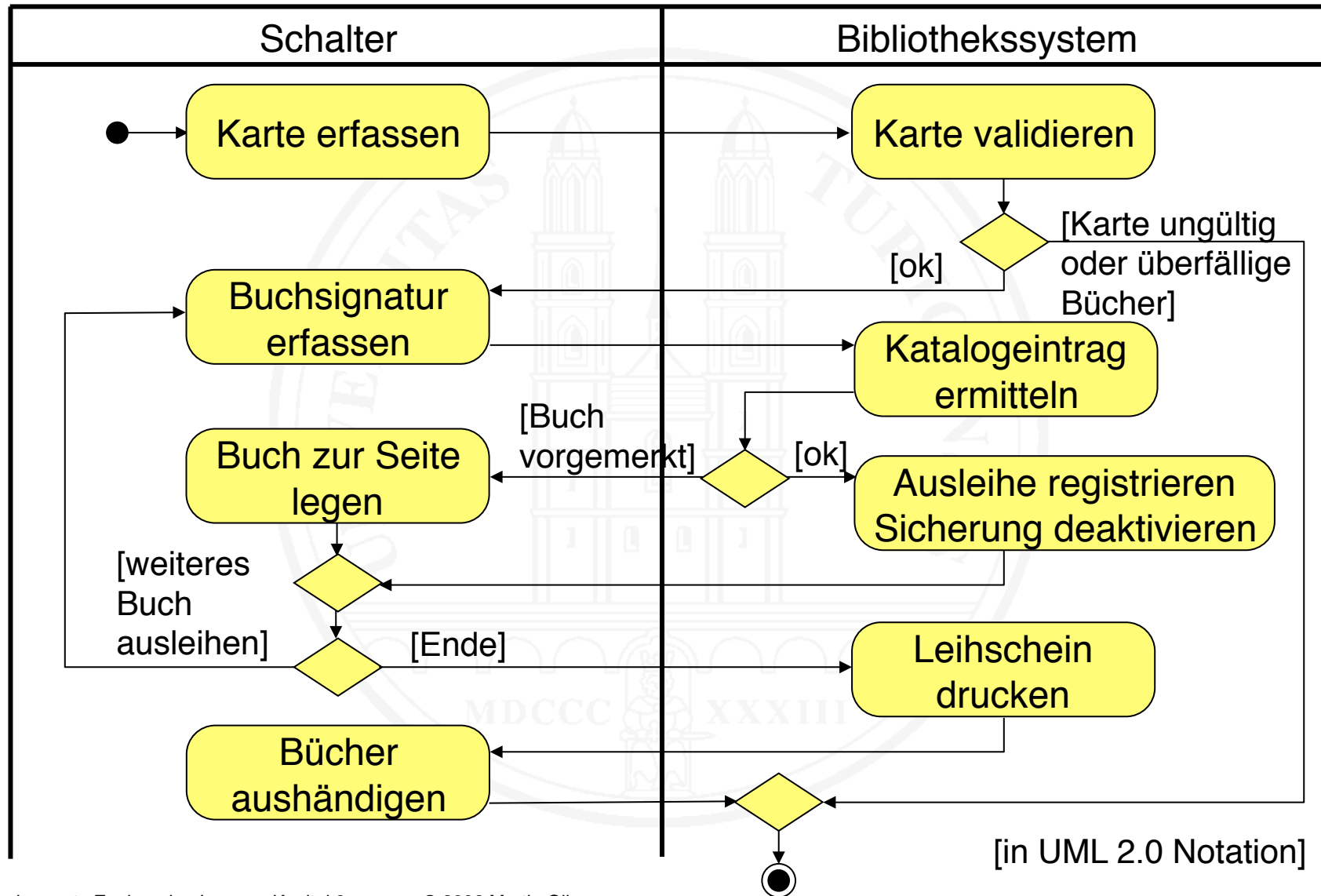
- Ereignis: **Anstoß** durch einen Akteur
- Aktion(en): **Reaktion** des Systems
- Zustand: **wann ist was möglich**



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



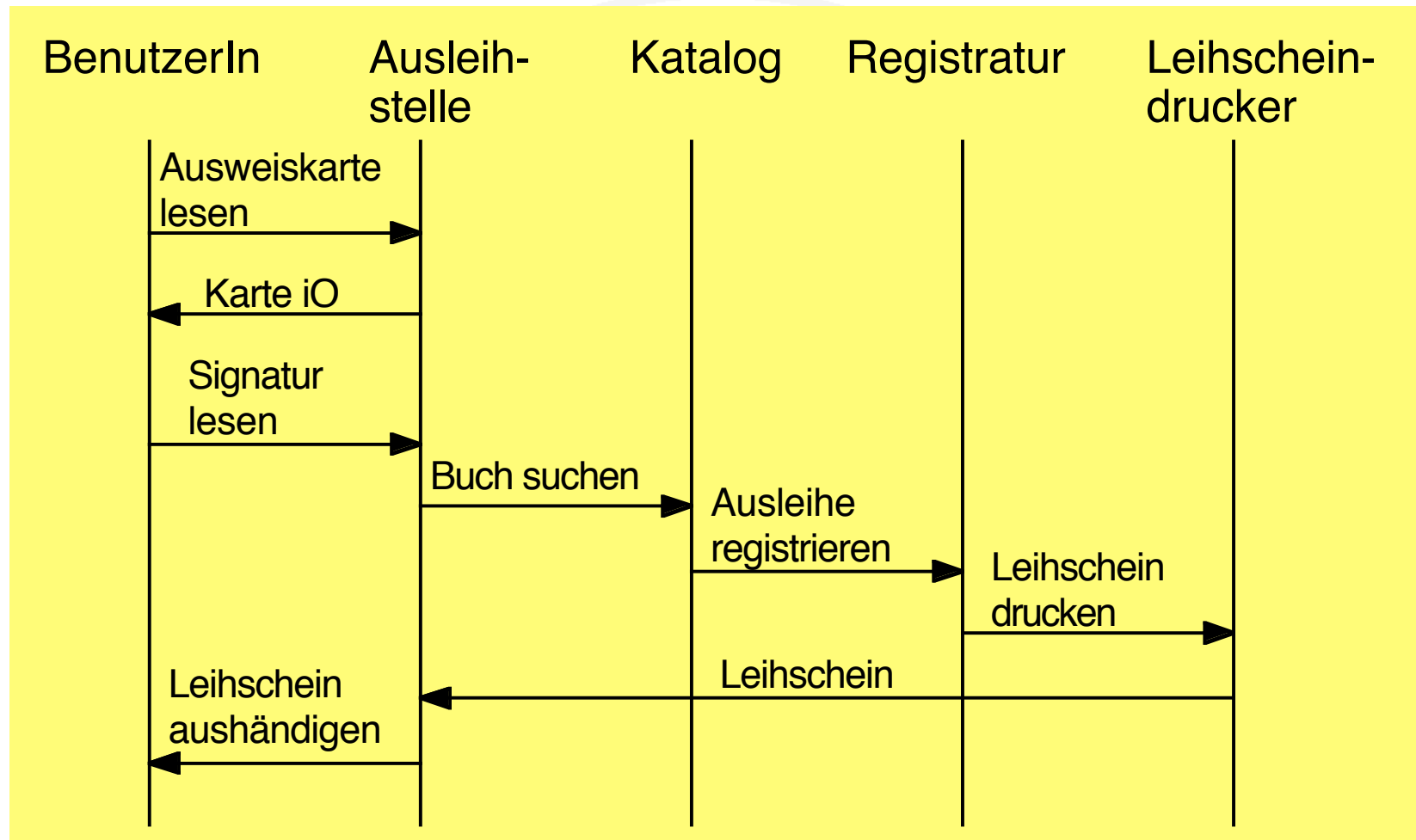
Szenariobeschreibung mit UML-Aktivitätsdiagramm 2

- 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



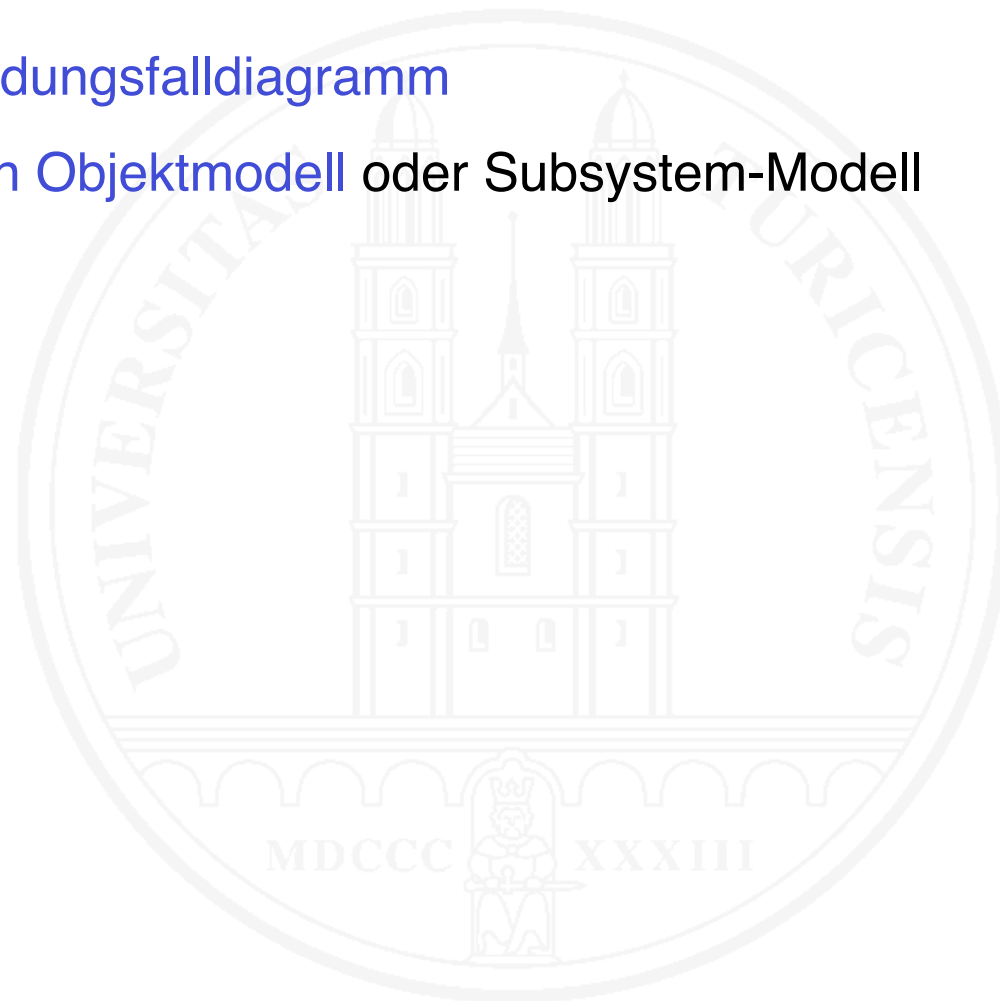
Aufgabe 8.1: Qualität von Szenarien

Vergleichen Sie die Beispiele der strukturierten Beschreibung mit Text und der Beschreibung mit UML Aktivitätsdiagramm. Wo ist letztere präziser?

Prüfen Sie die genannten Szenarien auf ihre Adäquatheit. Untersuchen Sie, ob alle Ausnahmefälle spezifiziert sind.

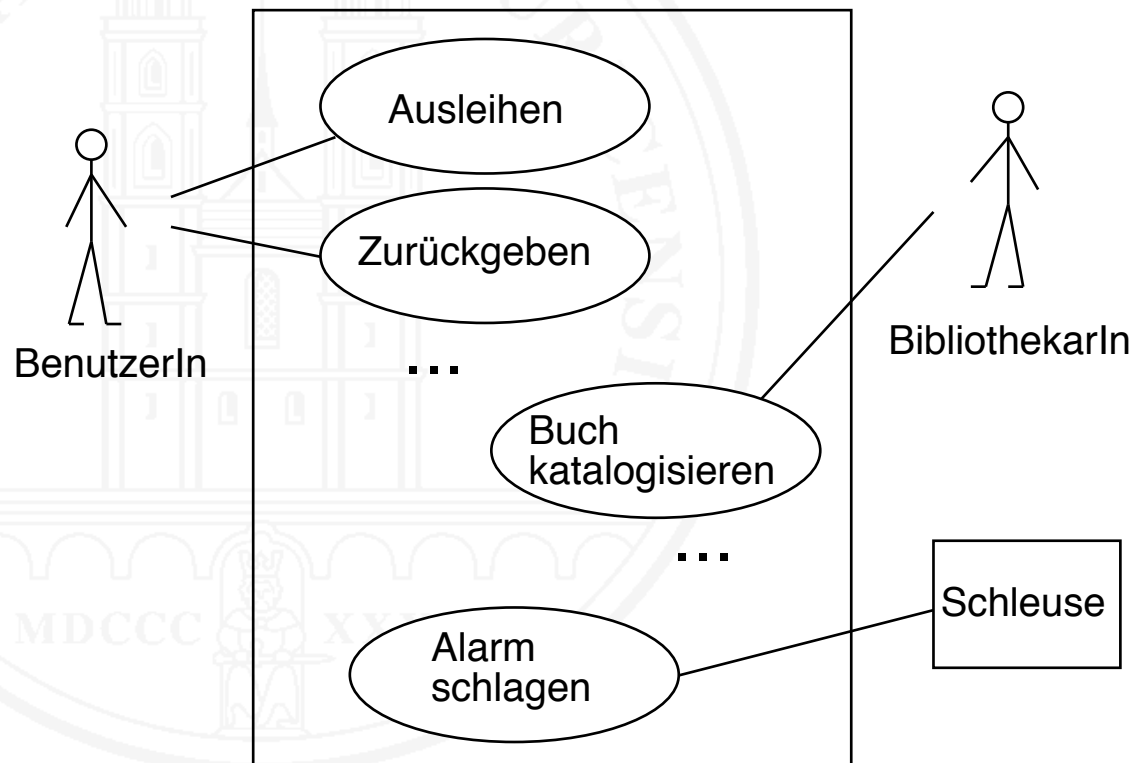
8.2.2 Szenario-Übersichten

- UML Anwendungsfalldiagramm
- Einbettung in Objektmodell oder Subsystem-Modell



UML Anwendungsfalldiagramm

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

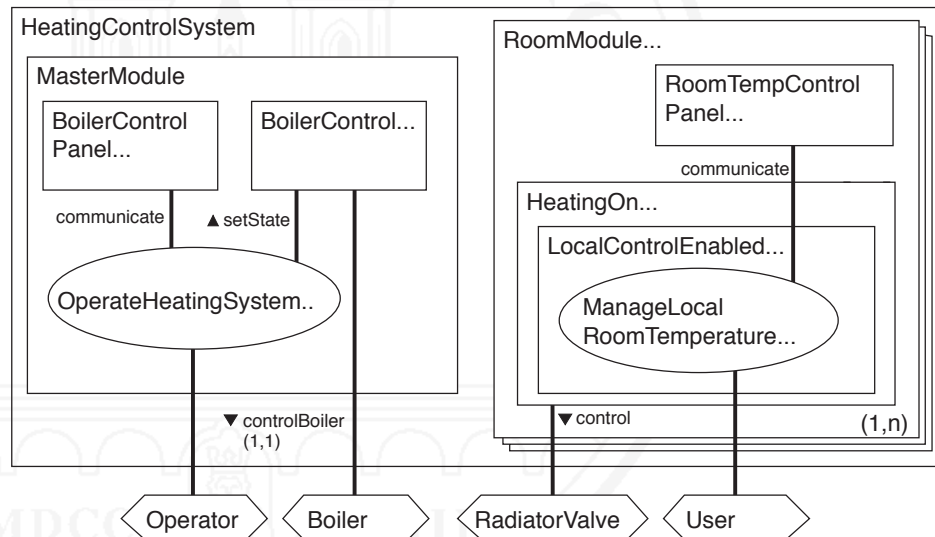


UML Anwendungsfalldiagramm – 2

- + 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

8.2.3 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

Scenario AuthenticateUser
Precondition: none
Steps: ...
Postcondition: User is authenticated

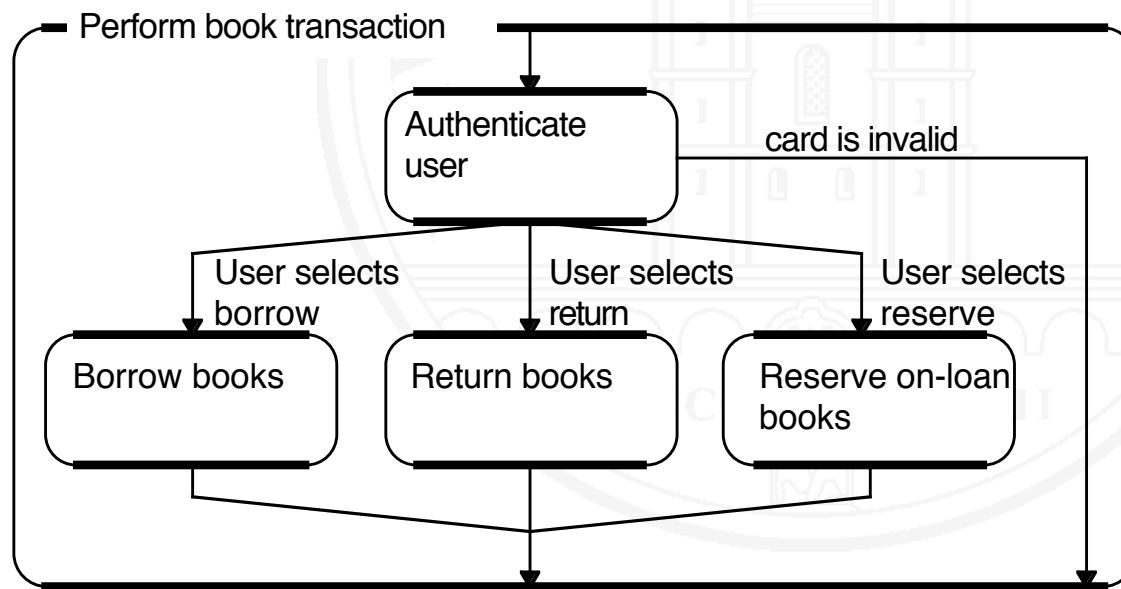
Scenario BorrowBooks
Precondition: User is authenticated
Steps: ...
...

Scenario ReturnBooks
Precondition: User is authenticated
Steps: ...
...

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

Zusammenhänge mit Statecharts

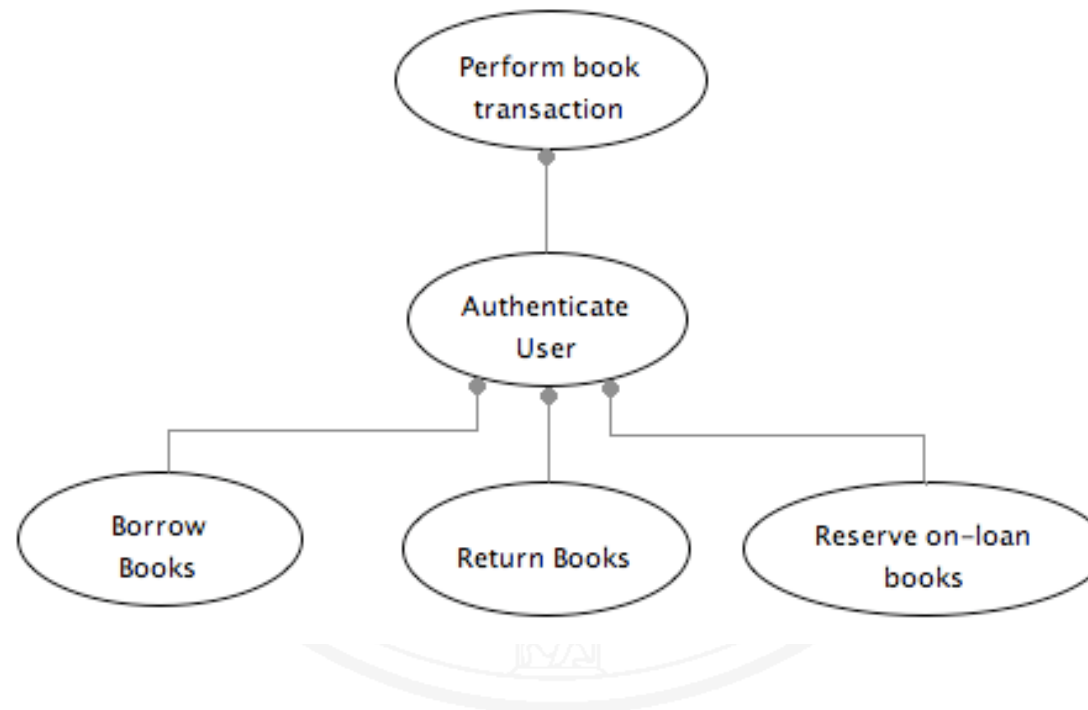
- Jedes Szenario wird als **Elementarzustand** oder als **Statechart** mit genau einem Anfangs- und Endzustand* modelliert (Glinz 2000a)
- Die klassischen Zusammenhänge (**Sequenz**, **Alternative**, **Iteration** und **Parallelität**) sind modellierbar
- **Forschungsergebnis**; in der Praxis (noch)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



*symbolisiert durch Balken oben und unten

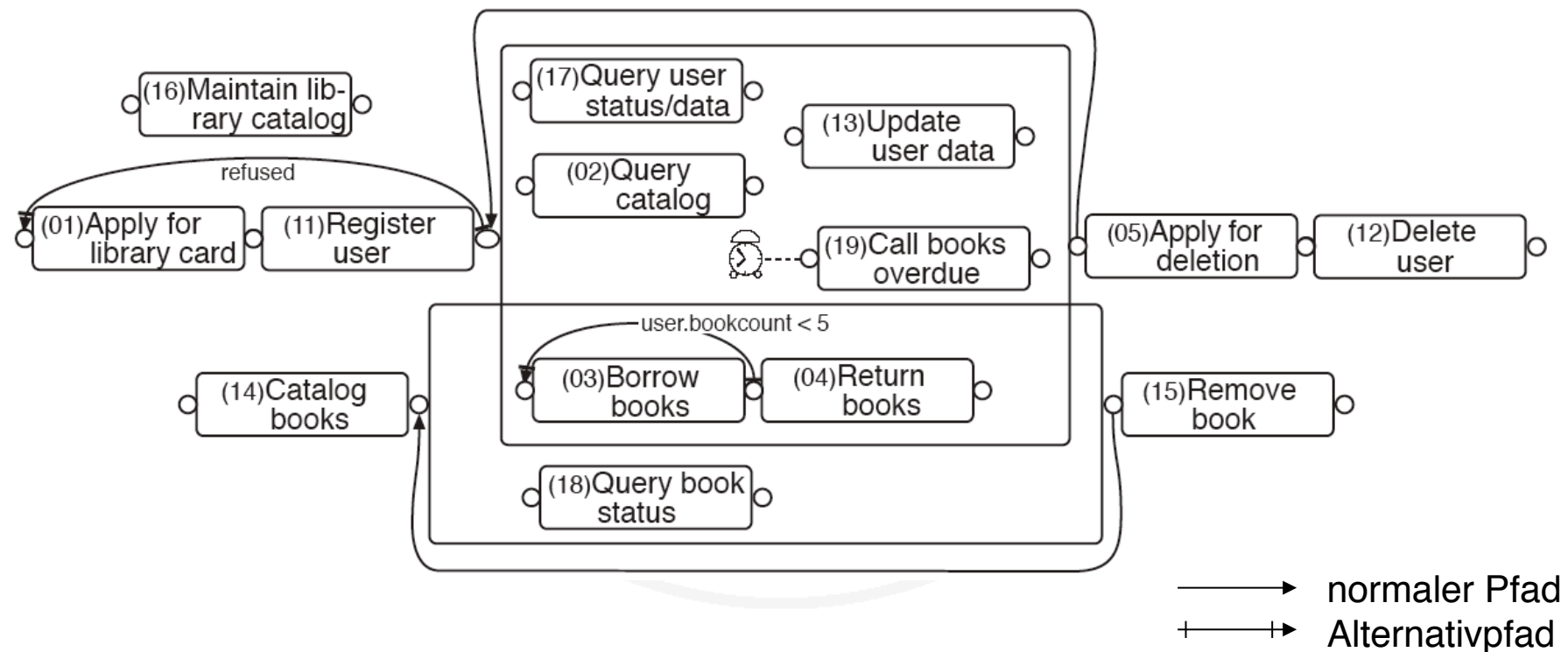
Erweiterte Jackson-Diagramme

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



Abhängigkeitsdiagramme

- **Spezielle Notation** zur Modellierung der Abhängigkeiten (Ryser und Glinz 2001)
- **Forschungsergebnis**; in der Praxis (noch) nicht verwendet



8.3 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 Kapitel 5 und Vorlesung Informatik IIa: Modellierung, Kapitel 10

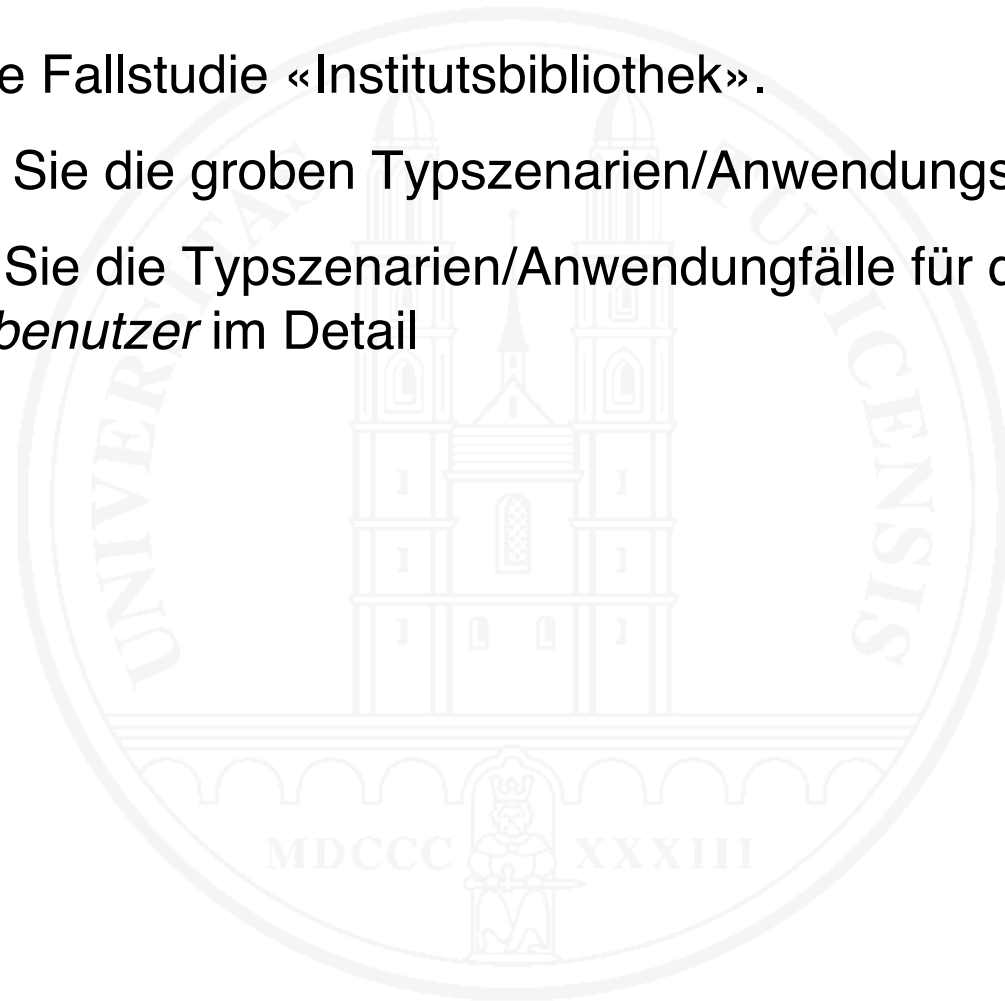
Methodik der Modellierung mit Szenarien – 2

- 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 8.2: 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



8.4 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

Literatur

Carroll, J. (1995). The Scenario Perspective on System Development. In Carroll, J., Ed.: *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York: John Wiley & Sons. 1-18.

Glinz, M. (1995). An Integrated Formal Model of Scenarios Based on Statecharts. In Schäfer, W. and Botella, P. (eds.): *Software Engineering – ESEC '95*. Proceedings of the 5th European Software Engineering Conference. Berlin, etc.: Springer. 254-271.

Glinz, M. (2000a). Improving the Quality of Requirements with Scenarios. *Proceedings of the Second World Congress on Software Quality*. Yokohama. 55-60.

Glinz, M. (2000b). Problems and Deficiencies of UML as a Requirements Specification Language. *Proceedings of the Tenth International Workshop on Software Specification and Design*. San Diego. 11-22.

Holbrook, H. (1990). A Scenario-Based Methodology for Conducting Requirements Elicitation. *ACM Software Engineering Notes* **15**, 1. 95-104.

Jacobson, I., M. Christerson, P. Jonsson, G. Övergaard (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Amsterdam, Reading, Mass. [u.a.]: Addison-Wesley.

Oestereich, B. (1998). *Objektorientierte Softwareentwicklung*. München: Oldenbourg.

Literatur – 2

OMG (2005). *UML Superstructure Specification, v2.0*. OMG document formal/05-07-04.
<http://www.omg.org/cgi-bin/doc?formal/05-07-04>

Ryser, J., M. Glinz (2001). Dependency Charts as a Means to Model Inter-Scenario Dependencies. In G. Engels, A. Oberweis and A. Zündorf (eds.): *Modellierung 2001*. GI-Workshop, Bad Lippspringe, Germany. GI-Edition - Lecture Notes in Informatics, Vol. P-1.

Sutcliffe, A. (1998). Scenario-Based Requirements Analysis. *Requirements Engineering Journal* **3**, 1. 48-65.

Weidenhaupt, K., K. Pohl, M. Jarke, P. Haumer (1998). Scenarios in System Development: Current Practice. *IEEE Software* **15**, 2. 34-45.