



**Universität  
Zürich** <sup>UZH</sup>

**Institut für Informatik**

# Informatik II: Modellierung

## Prof. Dr. Martin Glinz

### Kapitel 10

# Interaktionsmodelle

# Inhalt

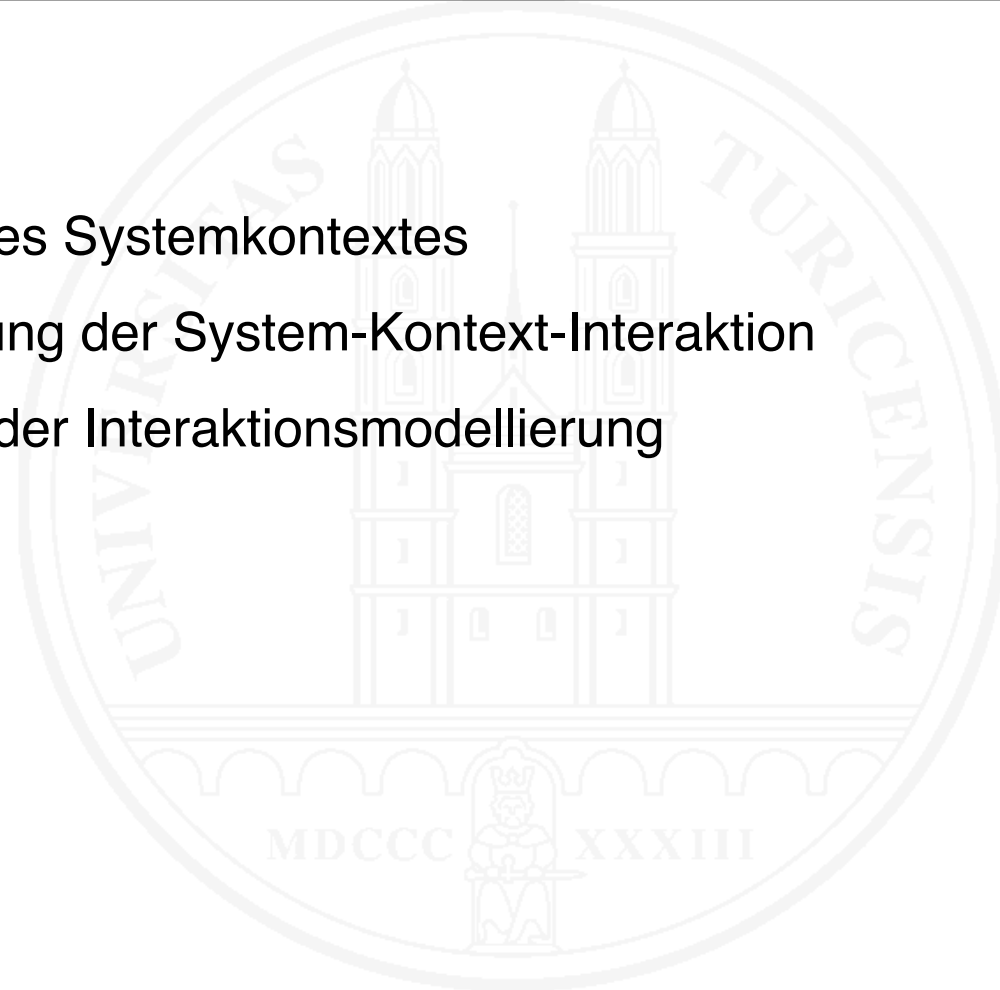
---

10.1 Motivation

10.2 Modelle des Systemkontextes

10.3 Modellierung der System-Kontext-Interaktion

10.4 Methodik der Interaktionsmodellierung



# 10.1 Motivation

Modellierung von ...



Ein System

Systemkontext (statisch)



Interaktion zwischen System und Kontext (dynamisch)

## Zeit stoppen

1. Benutzer drückt Start/Stop-Knopf kurz. Stoppuhr schaltet sich ein und zeigt fortlaufend die Zeit ab dem Drücken an.
2. Benutzer drückt Start/Stop-Knopf kurz. Stoppuhr zeigt die Zeit zum Zeitpunkt des Drückens an.
3. Benutzer drückt Start/Stop-Knopf lang. Stoppuhr schaltet aus, Anzeige verschwindet.

# Motivation – 2

---

Bei der Betrachtung von Systemen interessiert man sich neben dem System selbst auch für den System**kontext** und die **Interaktion** zwischen dem System und seinem Kontext

## Systemkontext (statisch):

- Identifikation der **Systemgrenzen**
- Modellierung der **externen**, mit dem System kommunizierenden **Akteure**
- Modellierung der **Information**, welche die **Systemgrenzen überquert**
  - **welche** Information
  - **von** wem – **für** wen

# Motivation – 3

---

## Interaktion (dynamisch):

- Modellierung der **Abläufe** in der **Kooperation** zwischen **Systembenutzern** und **System**
  - **Was** geschieht **wann**
  - **Wer** handelt oder ist **beteiligt**
- Mögliche **Akteure**:
  - Menschen
  - Sensoren, Maschinen
  - Andere (Informatik-)Systeme

## 10.2 Modelle des Systemkontextes

---

- Der **Kontext** eines Systems besteht aus den (systemexternen) **Akteuren**, welche mit dem System kommunizieren sowie der **Information**, welche diese **Akteure** mit dem **System austauschen**.
- Ein Kontextmodell ist **statisch**, d.h. es beschreibt die **strukturelle Einbettung** eines Systems in seinen Kontext und die dabei möglichen **Informationsflüsse**.
- Die **Dynamik** des Informationsaustauschs zwischen System und externen Akteuren wird **nicht** modelliert.

# Behandlung des Systemkontextes

---

Der Systemkontext wird in verschiedenen Modellierungstechniken sehr unterschiedlich behandelt:

- In der **Datenmodellierung** wird der Kontext überhaupt **nicht** betrachtet
- **Funktionsmodelle** mit Strukturierter Analyse beinhalten ein **explizites Kontextmodell** (Kontextdiagramm, siehe unten)
- **Verhaltensmodelle** enthalten **Kontextelemente**, z.B. in Form externer Ereignisse, modellieren aber typischerweise den Kontext nicht explizit
- **Klassen- und Objektmodelle** betrachten den Kontext **nicht**. Im praktischen Einsatz wird dieses Manko oft durch eine Kombination mit einem Anwendungsfallmodell (siehe unten) kompensiert

# Das Kontextdiagramm der Strukturierten Analyse

---

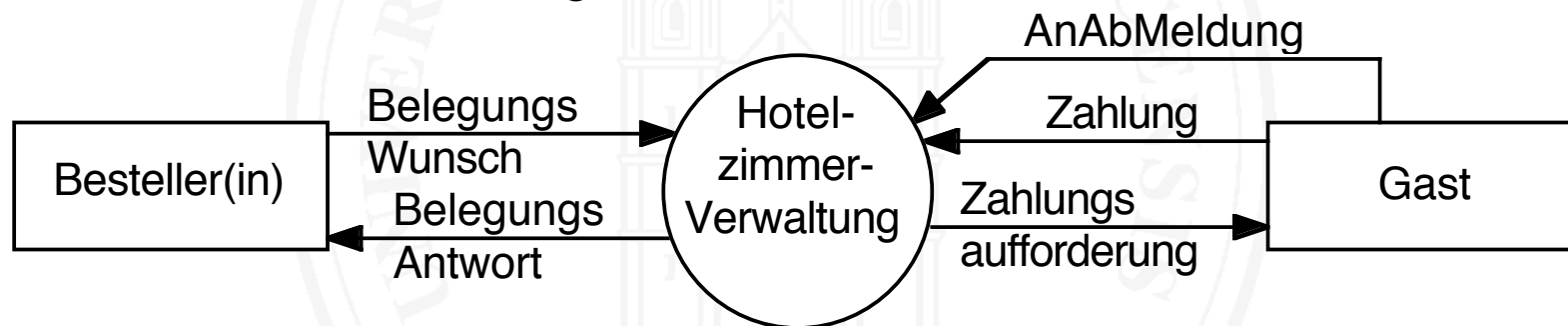
- Das **Kontextdiagramm (context diagram)** in der Strukturierten Analyse (vgl. Kapitel 6.3) ist ein typischer Vertreter eines Kontextmodells.
- Sonderform des **Datenflussdiagramms**
- Bildet die **Spitze der Zerlegungshierarchie** der Datenflussdiagramme (vgl. Kapitel 6.3)
- **Notation:**
  - Betrachtetes **System** als Ganzes im **Zentrum** des Diagramms
  - Systemexternen **Akteure darum herum** angeordnet
  - **Pfeile** repräsentieren **Informationsaustausch** (Datenfluss).



# Beispiel eines Kontextdiagramms

---

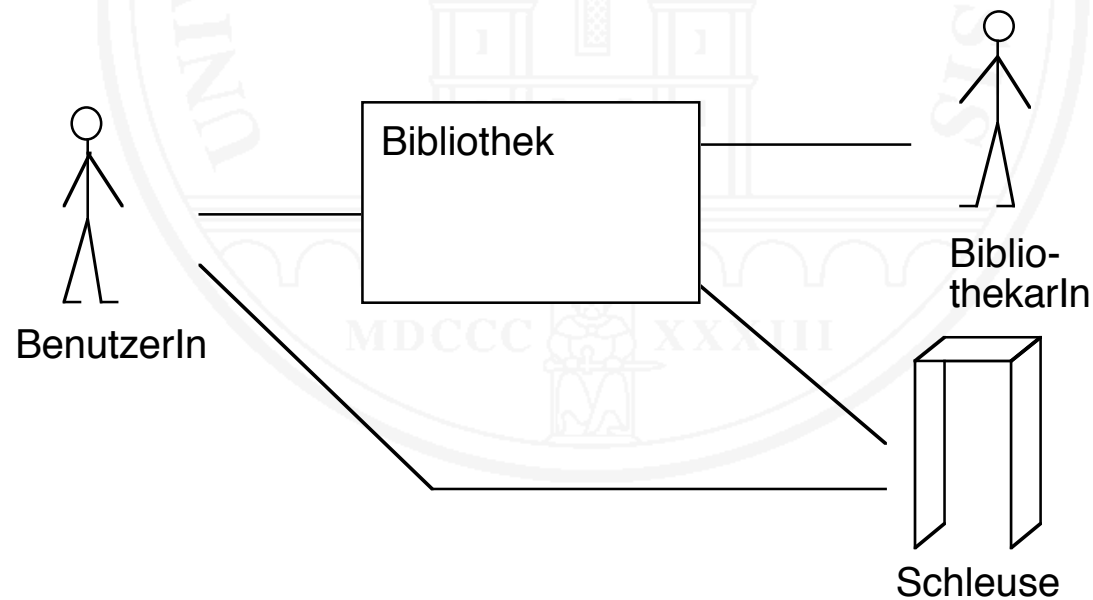
Kontext-Diagramm  
Hotelzimmer-Verwaltung



# Allgemeine Kontextdiagramme

---

- Das betrachtete **System** als Ganzes im **Zentrum**
- Systemexterne **Akteure** darum herum angeordnet
- Linien repräsentieren **Interaktionen** (bei Bedarf beschriftet)
- Beziehungen zwischen **Akteuren untereinander** modellierbar



# Aufgabe 10.1

---

Ein Unternehmen (200 Mitarbeiter, alle am gleichen Standort) hat Sie beauftragt, eine Studie über die Beschaffung und Einführung eines Arbeitszeiterfassungssystems zu erstellen.

Um den Auftrag präziser fassen zu können, wollen Sie als erstes die Systemgrenzen abklären. Als Grundlage für die Diskussion mit Ihrem Auftraggeber modellieren Sie einen Vorschlag.

Modellieren Sie Ihren Vorschlag in Form eines allgemeinen Kontextdiagramms.

## 10.3 Modellierung der System-Kontext-Interaktion

---

**Szenarien** und **Anwendungsfälle** modellieren die **Dynamik der Interaktion** zwischen einem System und seiner Umgebung.

- **Szenario (scenario)** – Eine **geordnete Menge von Interaktionen** zwischen Partnern
- In der Regel zwischen einem **System** und einer Menge **systemexterner Akteure**
- Zwei mögliche **Interpretationen**
  - Konkrete Interaktionsfolge (**Beispielszenario**)
  - Menge möglicher Interaktionen (**Typszenario**).
- Manche Autoren beschränken die Bezeichnung „Szenario“ auf Beispielszenarien

# Szenarien und Anwendungsfälle

---

- Für Typszenarien hat sich der Name **Anwendungsfall** (use case, Jacobson et al. 1992) eingebürgert.

**Anwendungsfall (use case)** – Eine durch einen Akteur angestoßene Folge von Systemereignissen, welche für den Akteur ein Ergebnis produziert und an welcher weitere Akteure teilnehmen können.

- Jeder Anwendungsfall
  - modelliert eine Systemfunktion
  - aus der Sicht des benutzenden Akteurs
  - durch Angabe der möglichen Abläufe von Interaktionsschritten
- Zusammenhänge zwischen den Systemfunktionen und die Einzelheiten ihrer Berechnung werden typisch nicht modelliert
- Daher oft mit Klassen- und Verhaltensmodellen kombiniert

# Notationen für Szenarien und Anwendungsfälle

---

## A. Modellierung einzelner Szenarien bzw. Anwendungsfälle

- Freier Text
- Strukturierter Text
- Zustandsautomaten bzw. Statecharts
- Aktivitätsdiagramme
- Interaktionsdiagramme

## B. Übersicht über die Anwendungsfälle eines Systems

- Anwendungsfall-Diagramm

# Gemeinsames Beispiel: Ein Bibliothekssystem

---

- Freihandbibliothek mit Ausleihe
- System verwaltet Benutzer, Katalog und Ausleihen
- Mitnahme nicht ausgeliehener Bücher wird durch Sicherungsetiketten in den Büchern und eine Sicherungsschleuse verhindert

# 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

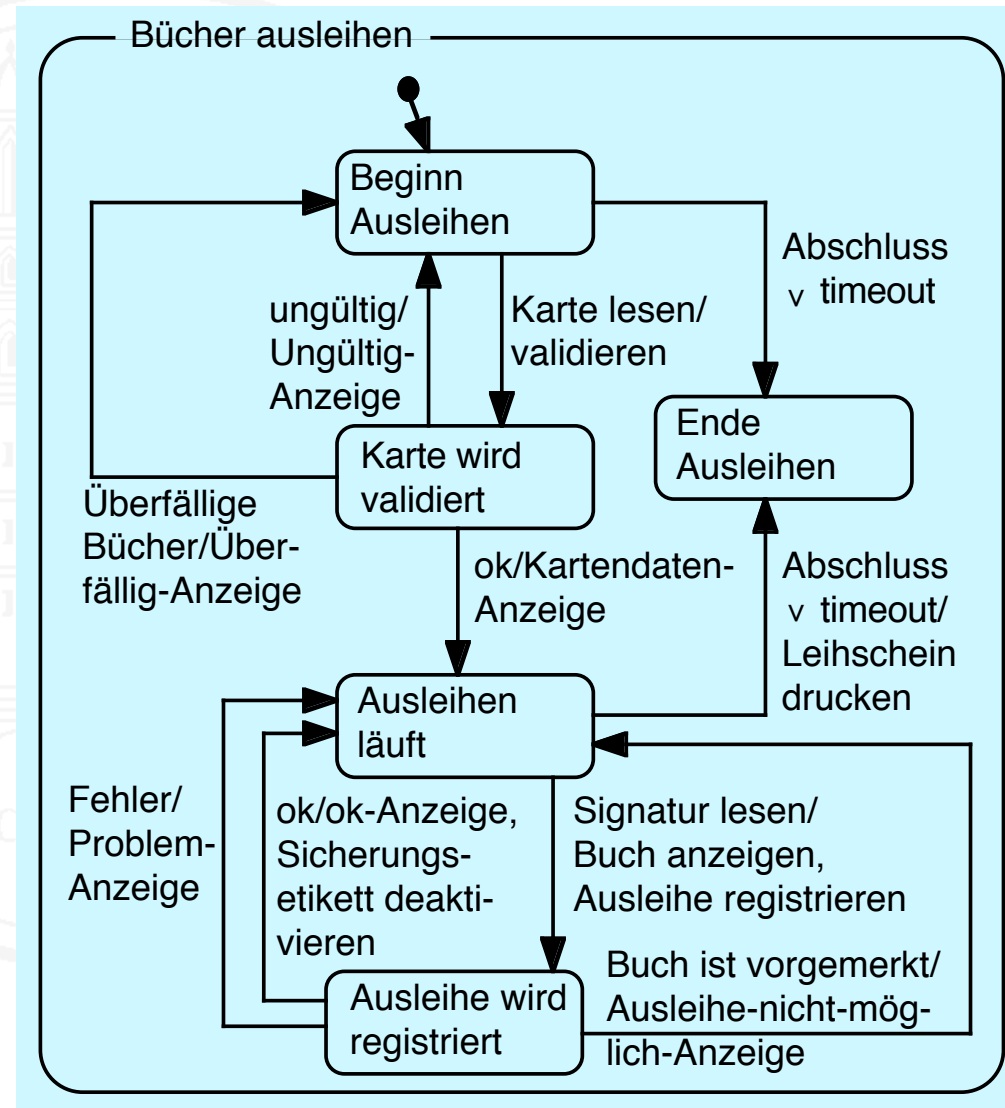
# Szenariobeschreibung mit strukturiertem Text – 4

---

- + 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**

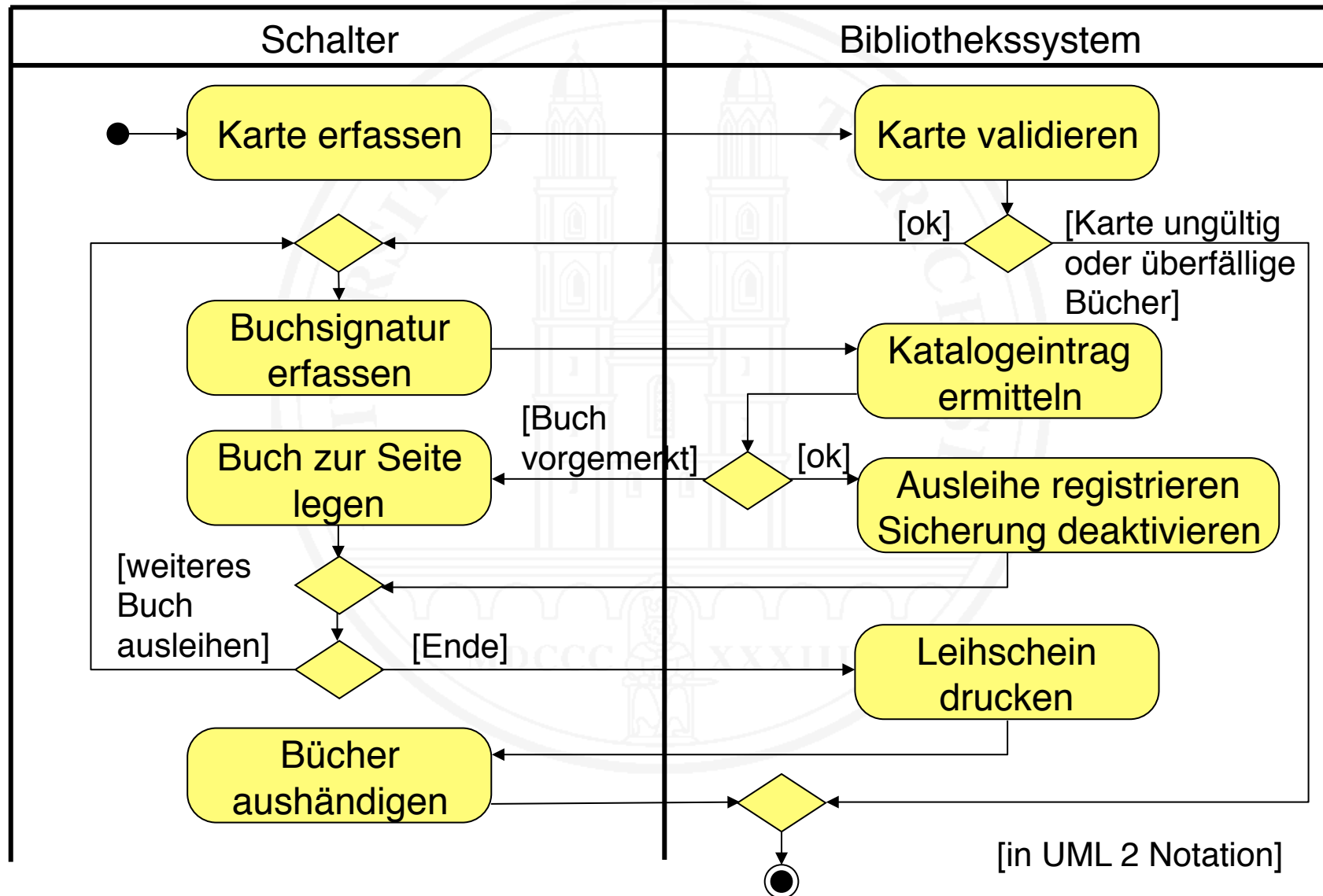


# Szenariobeschreibung mit Statecharts – 2

---

- + 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**

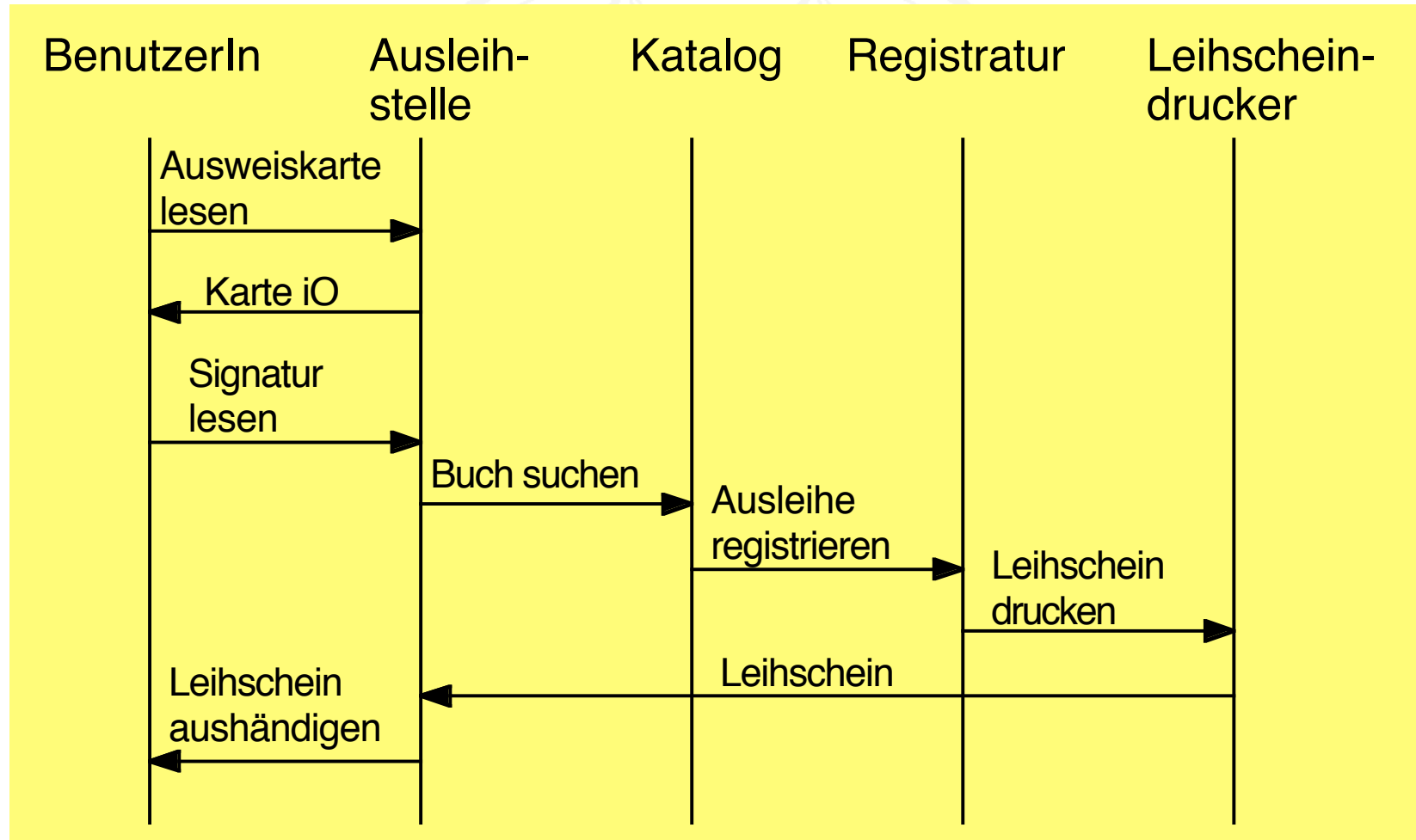


# Szenariobeschreibung mit Interaktionsdiagrammen

---

- 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 unter verschiedenen Namen: Sequenzdiagramme (sequence diagrams), Message sequence diagrams, MSCs
- + Eingängige, intuitiv verständliche Notation**
- Nur für Beispielszenarien**

# Szenariobeschreibung mit Interaktionsdiagrammen – 2



# Erweiterte Interaktionsdiagramme

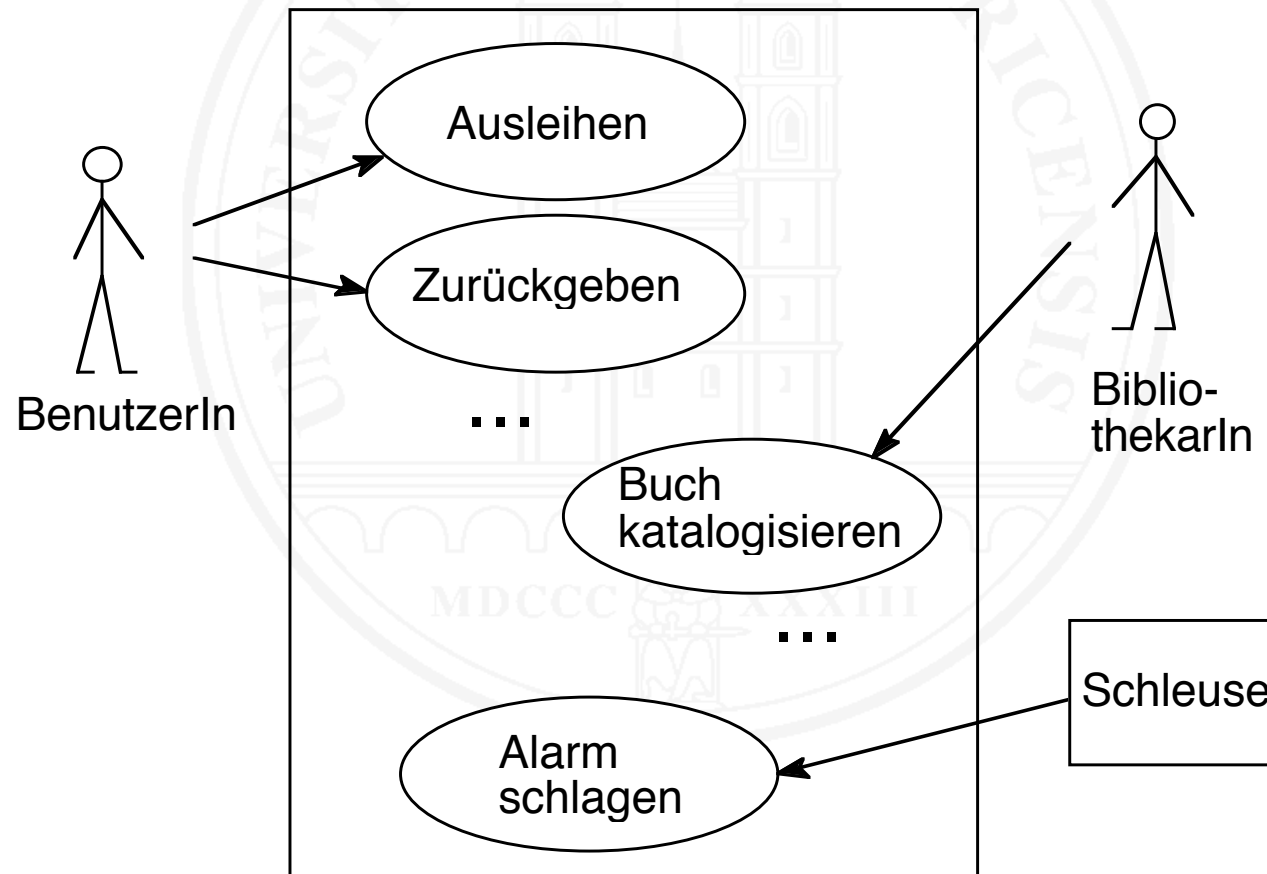
---

- Es gibt verschiedene Ansätze, Interaktionsdiagramme um Konstrukte für die Modellierung von **Alternativen** und **Iterationen** zu **erweitern**
- **UML-Sequenzdiagramme** (vgl. Kapitel 4) beispielsweise enthalten eine Erweiterung, welche dies gestattet
- + Sequenzdiagramme gestatten die Modellierung eines vollständigen Anwendungsfalls
- Die Notation ist bei komplexen Anwendungsfällen nur noch mit Mühe lesbar

# Anwendungsfall-Diagramm

---

Übersicht über alle Akteure und Anwendungsfälle eines Systems



# Anwendungsfalldiagramm – 2

---

- Gibt einen **Überblick** über alle Anwendungsfälle eines Systems
- Zeigt, **welche Akteure** in welchen Anwendungsfällen mit dem System interagieren
- Modelliert **Zusammenhänge** zwischen Anwendungsfällen **nur rudimentär**
- Ist eine Art **Kontextdiagramm**

## 10.4 Methodik der Interaktionsmodellierung

---

- Derzeit **noch keine allgemein anerkannte Methodik**
- Das nachfolgend beschriebene Vorgehen funktioniert in der Regel

### **Primäre Informationsquellen:**

Aufgabenbeschreibungen, Gespräche/Gesprächsnotizen, Beobachtung/Diskussion bestehender Verfahren/Arbeitsweisen/Prozesse

### **Hilfsmittel bei der Informationsgewinnung:**

- **Beispielszenarien** erstellen/diskutieren
- Liste wichtiger **Systemfunktionen** zusammenstellen
- Liste von **Ereignissen** zusammenstellen, auf die das System **reagieren** muss

# Methodik der Interaktionsmodellierung – 2

---

## Systemgrenzen und externe Akteure bestimmen:

- Die Zuweisung/Diskussion, wer für was verantwortlich ist, hilft bei der Abgrenzung
- **Kontextdiagramm** modellieren

## Anwendungsfälle (d.h. Typszenarien) ermitteln:

- **Akteur-System-Interaktionsfolgen** bestimmen, 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

# Methodik der Interaktionsmodellierung – 3

---

## Anwendungsfälle (d.h. Typszenarien) ermitteln (Fortsetzung):

- Logisch zusammengehörige Interaktionsfolgen/Beispielszenarien **zu Anwendungsfällen abstrahieren**
- Jeden **Anwendungsfall beschreiben** (mit strukturiertem Text oder präziser mit Statecharts bzw. Aktivitätsdiagrammen)
- Dabei **Normal- und Ausnahmeabläufe** beschreiben
- **Anwendungsfall-Diagramm** als **Übersicht** modellieren
- Bei vielen, feingranular beschriebenen Anwendungsfällen: Übersicht schaffen durch **Zusammenfassen** zusammengehöriger Anwendungsfälle zu einem übergeordneten, **vergrößerten** Anwendungsfall



# Methodik der Interaktionsmodellierung – 4

---

## Teilmodelle und fertiges Modell prüfen:

- Diskutieren der Modelle
- Durchspielen von Beispielszenarien
- Immer unter Beteiligung der Fachleute, die den modellierten Problem-  
bereich kennen

## Hinweis:

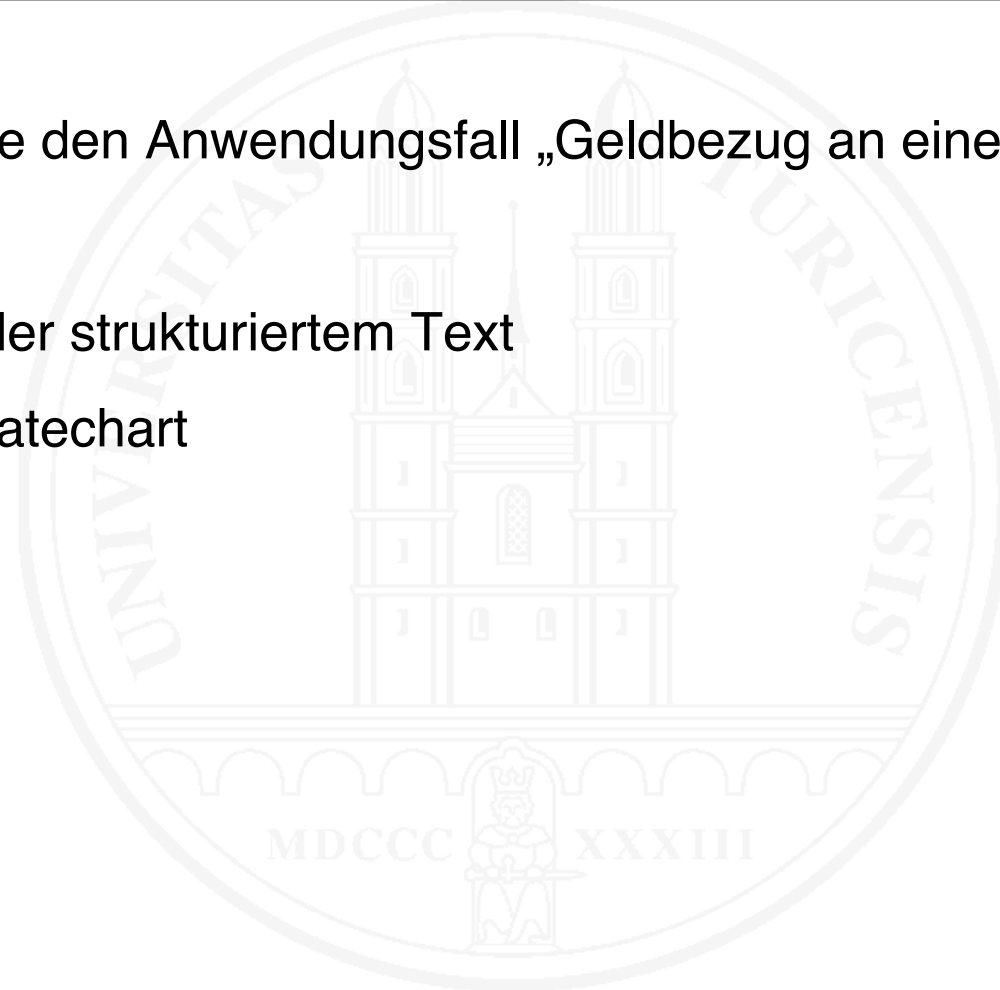
- Bisher gibt es nur rudimentäre Verfahren zur systematischen Modellierung von Zusammenhängen zwischen Anwendungsfällen
- Ebenso gibt es keine Methodik für das Modellieren der Zusammenhänge zwischen einem Interaktionsmodell und einem Klassen- oder Objektmodell des gleichen Problems.

## Aufgabe 10.2

---

Beschreiben Sie den Anwendungsfall „Geldbezug an einem Bancomaten“

- a) mit freiem oder strukturiertem Text
- b) mit einem Statechart



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

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.

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

Yourdon, E. (1989). *Modern Structured Analysis*. Englewood Cliffs, N.J.: Prentice Hall.

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