

# 11 Anforderungsspezifikation mit Szenarien oder Anwendungsfällen

## 11.1 Grundideen 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**
- + Modelliert die **Funktionalität** aus **Benutzersicht**: leicht verstehbar und überprüfbar
- + Hilft bei der **Abgrenzung** zwischen **System** und **Kontext**
- + **Dekomposition** möglich
- **Zusammenhänge** und **Abhängigkeiten** zwischen den Szenarien werden häufig nicht modelliert
- **Statische Struktur**, insbesondere notwendige Daten, werden nicht modelliert

## Zur Terminologie

**Szenario (scenario).** Eine **geordnete Menge** von **Interaktionen** zwischen Partnern, in der Regel zwischen einem System und einer Menge systemexterner Akteure. Kann sowohl eine konkrete Interaktionsfolge (**Beispielszenario**) als auch 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).** **Rolle**, welche ein **externes System** oder ein **Benutzer** gegenüber dem System einnehmen kann.

- 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.

## 11.2 Darstellung von Szenarien

- Modellierung **einzelner Szenarien**
  - Freier Text
  - Strukturierter Text
  - Zustandsautomaten bzw. Statecharts
  - Aktivitätsdiagramme (→ UML)
- **Übersichten**
  - Anwendungsfall-Diagramm (→ UML)
  - Einbettung in Objektmodell oder Subsystem-Modell
- **Zusammenhänge**
  - ?
  - Möglich mit: Statecharts, Jackson-Diagrammen, speziellen Abhängigkeitsdiagrammen

## 11.2.1 Darstellung eines einzelnen Szenarios

### Freier Text

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

# Strukturierter Text

- + Flexibel und ausdrucksmächtig
- + Von Anwendungsexperten les- und schreibbar
- + Präziser als freier Text, weniger Auslassungen und Fehler
- Für kritische Szenarien immer noch zu unpräzise und fehlerträchtig
- Zusammenhänge mit anderen Szenarien werden nicht erfasst
- ☆ Häufig zur Beschreibung von Typszenarien / Anwendungsfällen verwendet

Typszenario: Bücher ausleihen

Akteur: Bibliotheksbenutzer

Normalfall

1 Benutzer liest Bibliothekskarte ein

System validiert die Karte und gibt sie zurück; zeigt Benutzerdaten an; zeigt Auswahl-Dialog an

2 Benutzer wählt Funktion Ausleihen

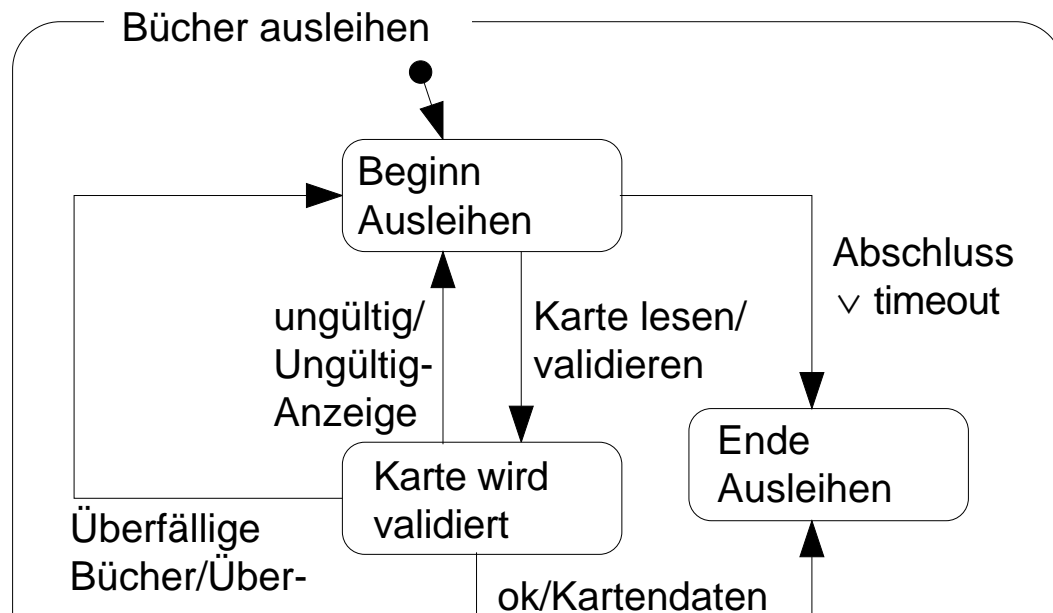
System zeigt Ausleihen-Dialog an

3 Benutzer erfasst Strichcodeetikett auf Buch

...

# Zustandsautomaten oder Statecharts

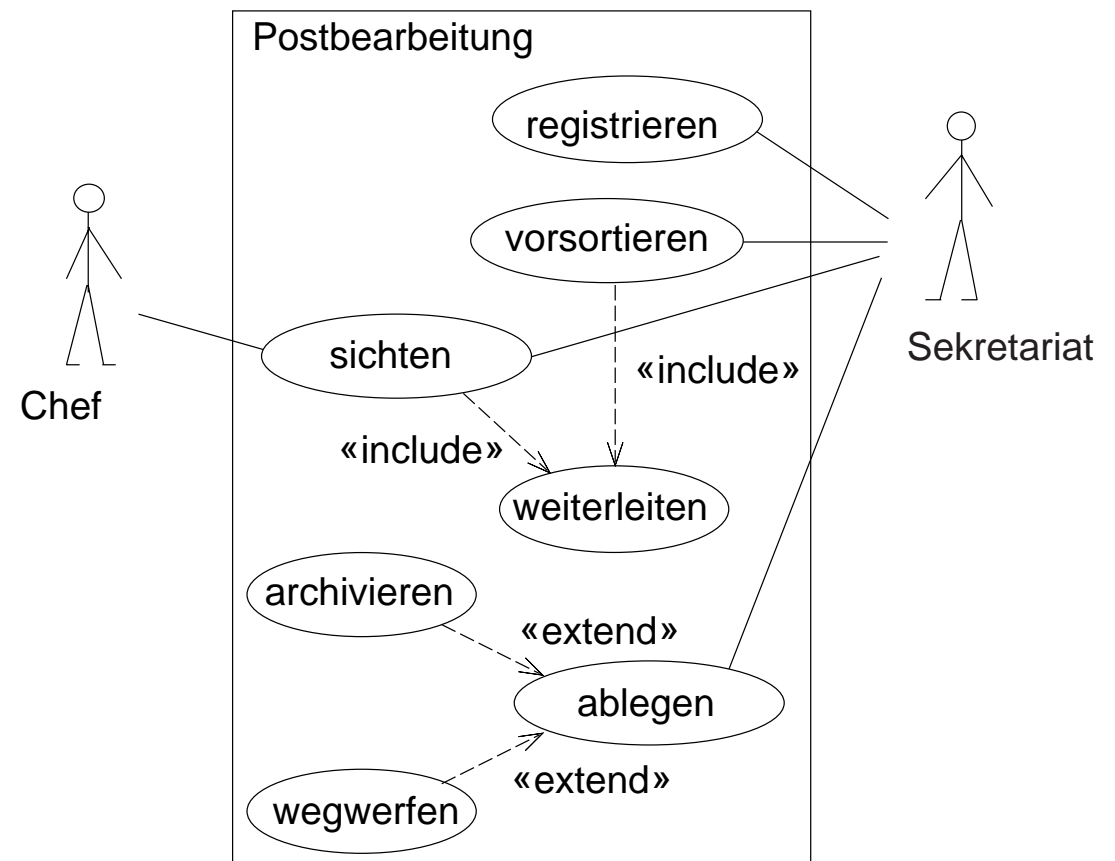
- + Aussagekräftig
- + Wählbarer Grad an Präzision (in der Beschreibung der Ereignisse und Aktionen)
- + Zusammenhänge mit anderen Szenarien sind modellierbar
- o Von Anwendungsexperten **verstehbar** (erfordert Unterstützung oder Ausbildung)
- Braucht Modellierungsexperten zur Erstellung



## 11.2.2 Darstellung von Szenarienübersichten

### Anwendungsfall-Diagramm (→ UML)

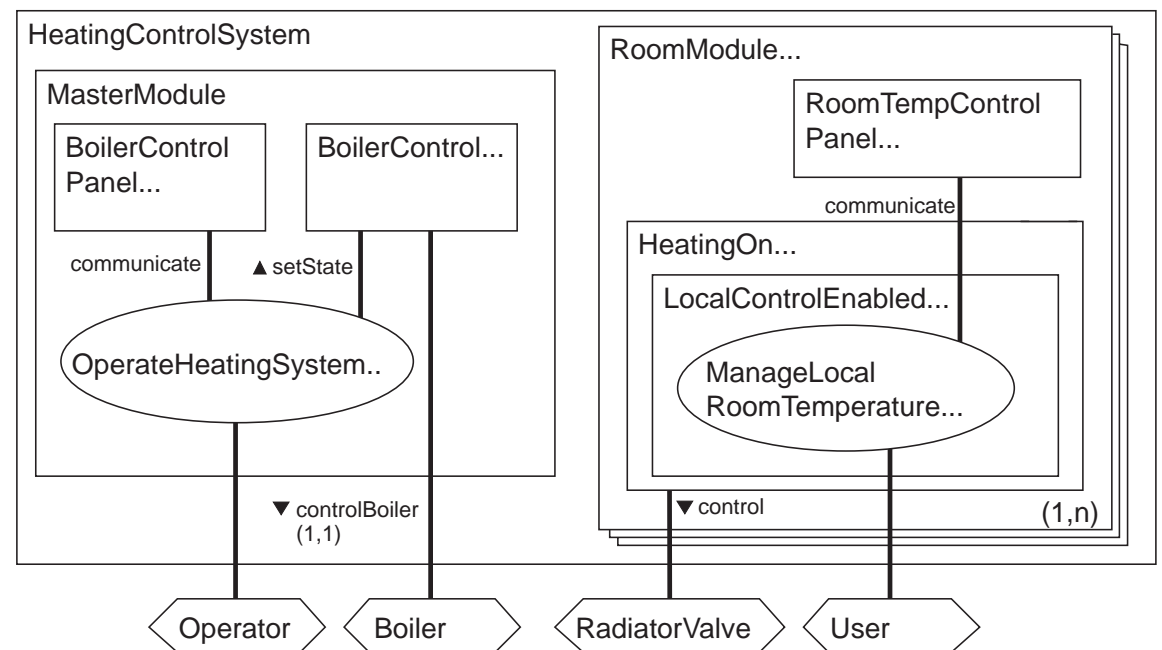
- Gibt einen **Überblick** über alle **Anwendungsfälle** (Typszenarien) eines Systems
- Ist eine Art **Kontextdiagramm**
- + Zeigt, welche Akteure in welchen Anwendungsfällen mit dem System interagieren
- + Bei kleinen Modellen übersichtlich
- Modelliert Zusammenhänge zwischen Anwendungsfällen nur rudimentär
- Modelliert die Inhalte der Anwendungsfälle nicht
- Kennt **keine Zerlegung**



# Einbettung von Szenarien in ein Objektmodell

- Das System wird **hierarchisch** in Teilprobleme zerlegt
- Jedes Teilproblem wird durch ein **abstraktes Objekt** modelliert
- Szenarien werden den Teilproblemen zugeordnet, zu denen sie gehören

Beispiel: Eine Heizungssteuerung, modelliert in der Spezifikationsprache ADORA



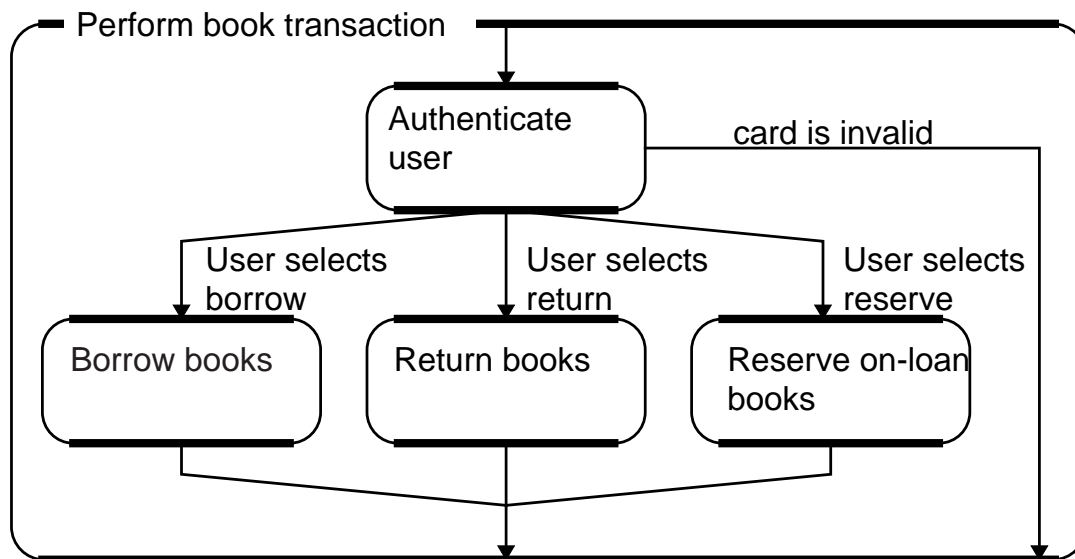
- In UML ist eine Einbettung von Anwendungsfällen in Subsysteme (und damit indirekt eine gewisse Dekomposition) möglich



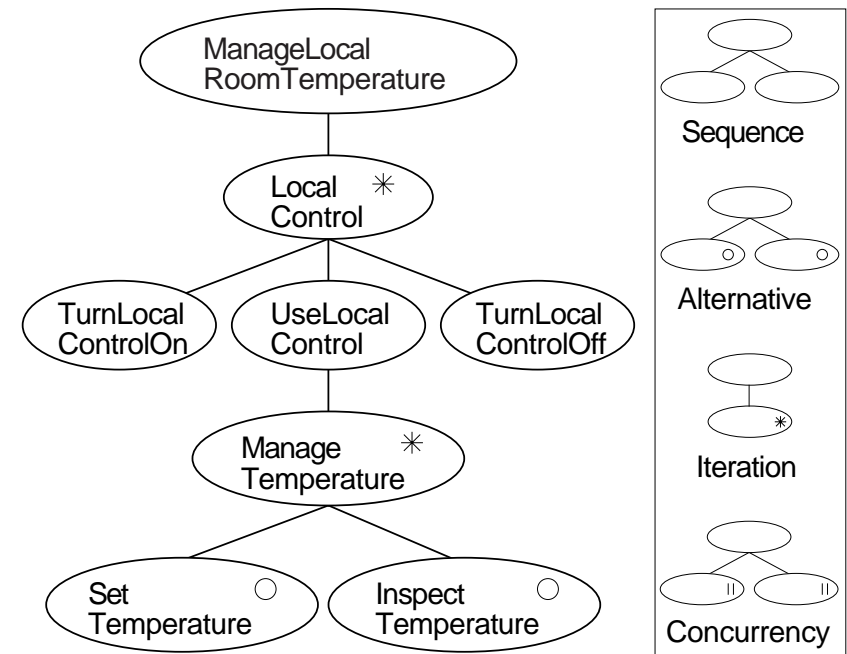
## 11.2.3 Darstellung von Zusammenhängen

- Problem wird bisher weitgehend **ignoriert**
- **Möglich** beispielsweise mit

### Statecharts



### Jackson-Diagrammen



## 11.3 Methodik der Spezifikation mit Szenarien

### 11.3.1 Szenarienanalyse

- **Systemgrenzen** und **Akteure** im **Systemkontext** bestimmen
- Die **Hauptfunktionen** des Systems und ihre Akteure auflisten
- **Hauptfunktionen** ggf. in Teilfunktionen **zerlegen**\*
- Pro Funktion **ein Szenario modellieren**
  - Ziel sind Typszenarien (Anwendungsfälle)
  - Als Weg dorthin kann es sinnvoll sein, ausgewählte Beispielinteraktionen in Beispielszenarien darzustellen und diese dann zu Typszenarien zu abstrahieren
- **Szenarienübersichten** modellieren, zum Beispiel Anwendungsfall-Diagramm
- **Szenarienzusammenhänge** analysieren und darstellen

\*Achtung: nicht über zu viel Stufen, sonst resultiert eine funktionale Dekomposition

# Beispiel: Bibliothekssystem

- Bestimmung der **Hauptfunktionen**

<i>Hauptfunktion</i>	<i>Auslösender Akteur</i>
Bibliothek benutzen	Benutzer
Benutzer pflegen	Bibliothekarin
Bestand pflegen	Bibliothekarin
Alarmieren	Schleuse

...

- **Zerlegung** der Funktion 'Bibliothek benutzen'

Buch ausleihen

Buch zurückgeben

Buch vormerken

...

# Beispielszenario „Ein einzelnes Buch erfolgreich ausleihen“

Szenario „Ein einzelnes Buch erfolgreich ausleihen“

Akteur: BenutzerIn

Ablauf:

- 1 BenutzerIn liest ihre Karte ein  
System prüft und validiert die Karte
- 2 BenutzerIn wählt „Ausleihen“  
System aktiviert die Ausleihfunktion
- 3 Benutzerin liest Buchcode ein  
System identifiziert das Buch, registriert die Ausleihe,  
deaktiviert das Diebstahletikett
- 4 Benutzerin beendet Ausleihe  
System druckt Leihschein

# Beispielszenario „Ein vorgemerktes Buch versuchen auszuleihen“

Szenario „Ein vorgemerktes Buch versuchen auszuleihen“

Akteur: BenutzerIn

Ablauf:

- 1 BenutzerIn liest ihre Karte ein  
System prüft und validiert die Karte
- 2 BenutzerIn wählt „Ausleihen“  
System aktiviert die Ausleihfunktion
- 3 Benutzerin liest Buchcode ein  
System identifiziert das Buch, gibt Vormerkwarnung aus
- 4 Benutzerin beendet Ausleihe  
System quittiert den Abschluss

# Resultierendes Typszenario

Typszenario: Bücher ausleihen

Akteur: BenutzerIn

## Normalfall

- 1 Benutzer liest Bibliothekskarte ein  
System validiert die Karte und gibt sie zurück; zeigt Benutzerdaten an;  
zeigt Auswahl-Dialog an
- 2 Benutzer wählt Funktion Ausleihen  
System zeigt Ausleihen-Dialog an
- 3 Benutzer erfasst Strichcodeetikett auf Buch  
System identifiziert Buch; registriert Buch als ausgeliehen; deaktiviert Sicherheitsetikett;  
zeigt die Buchdaten an

...

## Ausnahmefälle

...

- 3' Wenn Buch für eine andere Person vorgemerkt ist: System zeigt Vorgemerkt-Dialog an

## 11.3.2 Ereignis-Reaktions-Analyse (vgl. Kapitel 10)

- Alle **Ereignisse**, die eine Reaktion des Systems erfordern, **auflisten**
- Für jedes Ereignis die erforderlichen **Reaktionen** bestimmen
  
- **Interaktionen bestimmen** durch
  - Gruppierung von Ereignissen, die vom gleichen externen Akteur stammen, zu logischen Sequenzen von Ereignissen und Systemreaktionen
  - Beschreiben solcher Sequenzen in Szenarien

# Beispiel: Ereignis-Reaktions-Analyse für ausleihbare Bücher (Auszug)

(vgl. Kapitel 10.4)

## **Ereignis**

- 1 neues Buch trifft ein
- 2 BenutzerIn will Buch ausleihen
  - 2.1 BenutzerIn liest ihre Karte ein
  - 2.2 BenutzerIn wählt „Ausleihen“
  - 2.3 Benutzerin liest Buchcode ein
  - 2.4 Benutzerin beendet Ausleihe

...

## **Reaktion**

- Buch klassifizieren
- Buch nicht ausgeliehen und nicht vorgemerkt:  
ausleihen  
sonst: vormerken
- Karte prüfen und validieren
- Ausleihe aktivieren
- Buch identifizieren, Ausleihe registrieren,  
Diebstahletikett deaktivieren
- Leihschein drucken



# Gruppierung einer zusammengehörenden Ereignis-Reaktionssequenz zu einem (Beispiel-)Szenario

Szenario „Ein einzelnes Buch erfolgreich ausleihen“

Akteur: BenutzerIn

Ablauf:

- 1 BenutzerIn liest ihre Karte ein  
System prüft und validiert die Karte
- 2 BenutzerIn wählt „Ausleihen“  
System aktiviert die Ausleihfunktion
- 3 Benutzerin liest Buchcode ein  
System identifiziert das Buch, registriert die Ausleihe,  
deaktiviert das Diebstahletikett
- 4 Benutzerin beendet Ausleihe  
System druckt Leihschein

## 11.4 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 → Forschungsgegenstand
- **Querverweise**, vor allem aus den Szenarien ins Objekt- bzw. Klassenmodell

Beispiel:

3 Benutzerin liest Buchcode ein

System identifiziert das Buch (Buch.Identifizieren), registriert die Ausleihe (Buch.Ausleihen), deaktiviert das Diebstahletikett