

Software Engineering in der Praxis

Praktische Übungen

Objektorientiertes Design

Matthias Meitner Marc Spisländer

Lehrstuhl für Software Engineering
Friedrich-Alexander-Universität Erlangen-Nürnberg

1 Inhalt

2 Nachlese

- Lernziele der letzten Woche(n)
- Objektorientierte Analyse

3 Objektorientiertes Design

- Strukturmodellierung
- Verhaltensmodellierung

Model Checking

- CTL-Formeln
- NuSMV: Synchroner und asynchroner Automaten
- Lebendigkeit als Begriff mit Abstufungen
- NuSMV: Nichtdeterminismus
- interaktive Simulation und automatischer Beweis

Analysewerkzeuge der UML

■ Anwendungsfalldiagramme

- Bestimmung, welche Leistungen des Systems erbracht werden
- Vorwiegend aktionsorientiert
- Sicht auf die Systemgrenzen

■ Aktivitätsdiagramme

- Kontroll- und Datenfluss kann dargestellt werden
- Swimlane ordnet Aktionen und Kontrollflusselemente zu Systemstrukturen

■ Zustandsautomaten / Statecharts

- Zu jedem Zeitpunkt eindeutiger Systemzustand
- Ereignisse können Übergänge auslösen: Trigger [Guard] / Effekt

Analysewerkzeuge der UML

- Anwendungsfalldiagramme
 - Bestimmung, welche Leistungen des Systems erbracht werden
 - Vorwiegend aktionsorientiert
 - Sicht auf die Systemgrenzen
- Aktivitätsdiagramme
 - Kontroll- und Datenfluss kann dargestellt werden
 - Swimlane ordnet Aktionen und Kontrollflusselemente zu Systemstrukturen
- Zustandsautomaten / Statecharts
 - Zu jedem Zeitpunkt eindeutiger Systemzustand
 - Ereignisse können Übergänge auslösen: Trigger [Guard] / Effekt

Analysewerkzeuge der UML

- Anwendungsfalldiagramme
 - Bestimmung, welche Leistungen des Systems erbracht werden
 - Vorwiegend aktionsorientiert
 - Sicht auf die Systemgrenzen
- Aktivitätsdiagramme
 - Kontroll- und Datenfluss kann dargestellt werden
 - Swimlane ordnet Aktionen und Kontrollflusselemente zu Systemstrukturen
- Zustandsautomaten / Statecharts
 - Zu jedem Zeitpunkt eindeutiger Systemzustand
 - Ereignisse können Übergänge auslösen: Trigger [Guard] / Effekt

Abgrenzung

- Objektorientierte Analyse als **Lernprozess**
- Heute: **gestalterischer** Design-Prozess
- Vorgehensmodell: meist keine streng getrennten Phasen
- zum Teil gleiche Werkzeuge (s. Aktivitätsdiagramme)

Ziel des Objektorientierten Designs

- Modellieren, *wie* das System die Aufgaben löst
- Nahe an der Implementierung

Abgrenzung

- Objektorientierte Analyse als **Lernprozess**
- Heute: **gestalterischer** Design-Prozess
- Vorgehensmodell: meist keine streng getrennten Phasen
- zum Teil gleiche Werkzeuge (s. Aktivitätsdiagramme)

Ziel des Objektorientierten Designs

- Modellieren, *wie* das System die Aufgaben löst
- Nahe an der Implementierung

Abgrenzung

- Objektorientierte Analyse als **Lernprozess**
- Heute: **gestalterischer** Design-Prozess
- Vorgehensmodell: meist keine streng getrennten Phasen
 - zum Teil gleiche Werkzeuge (s. Aktivitätsdiagramme)

Ziel des Objektorientierten Designs

- Modellieren, *wie* das System die Aufgaben löst
- Nahe an der Implementierung

Abgrenzung

- Objektorientierte Analyse als **Lernprozess**
- Heute: **gestalterischer** Design-Prozess
- Vorgehensmodell: meist keine streng getrennten Phasen
- zum Teil gleiche Werkzeuge (s. Aktivitätsdiagramme)

Ziel des Objektorientierten Designs

- Modellieren, *wie* das System die Aufgaben löst
- Nahe an der Implementierung

Strukturmodellierung in der UML

- **Klassendiagramme**
- Komponentendiagramme
- Objektdiagramme
- Paketdiagramme, Kompositionsstrukturdiagramme, Verteilungsdiagramme

Strukturmodellierung in der UML

- Klassendiagramme
- Komponentendiagramme
- Objektdiagramme
- Paketdiagramme, Kompositionsstrukturdiagramme, Verteilungsdiagramme

Strukturmodellierung in der UML

- Klassendiagramme
- Komponentendiagramme
- Objektdiagramme
- Paketdiagramme, Kompositionsstrukturdiagramme, Verteilungsdiagramme

Strukturmodellierung in der UML

- Klassendiagramme
- Komponentendiagramme
- Objektdiagramme
- Paketdiagramme, Kompositionsstrukturdiagramme, Verteilungsdiagramme

Komponentendiagramme

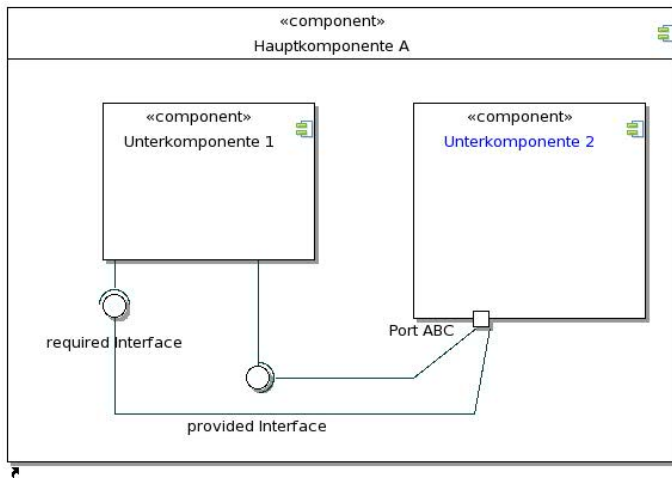
Komponente

»A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment«¹

- Komponentendiagramme stellen die Struktur eines Systems zur Laufzeit dar
- Schnittstellen, Ports (“Kommunikationspunkte”), ...

¹http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML

Beispiel



Klassendiagramme

- Klassen: Kapselung von Attributen und Operationen
- Schnittstellen: meist nur Operationen (UML2: auch Attribute)
- Beziehungen:
 - Generalisierung, Vererbung
 - Realisierung, Implementierung
 - Komposition
 - Aggregation
 - Assoziation

Klassendiagramme

- Klassen: Kapselung von Attributen und Operationen
- Schnittstellen: meist nur Operationen (UML2: auch Attribute)
- Beziehungen:
 - Generalisierung, Vererbung
 - Realisierung, Implementierung
 - Komposition
 - Aggregation
 - Assoziation

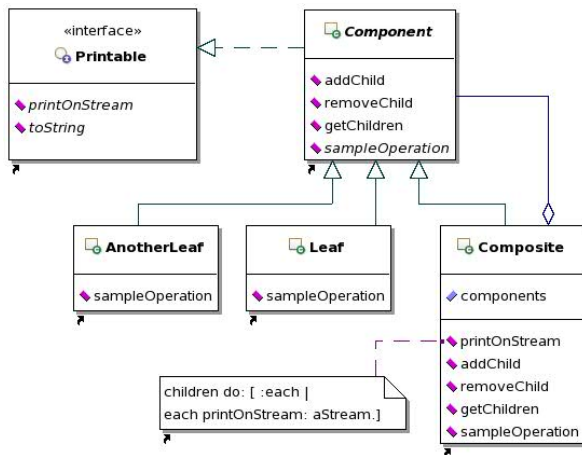
Klassendiagramme

- Klassen: Kapselung von Attributen und Operationen
- Schnittstellen: meist nur Operationen (UML2: auch Attribute)
- Beziehungen:
 - Generalisierung, Vererbung
 - Realisierung, Implementierung
 - Komposition
 - Aggregation
 - Assoziation

Klassendiagramme

- Klassen: Kapselung von Attributen und Operationen
- Schnittstellen: meist nur Operationen (UML2: auch Attribute)
- Beziehungen:
 - Generalisierung, Vererbung
 - Realisierung, Implementierung
 - Komposition
 - Aggregation
 - Assoziation

Klassendiagramme



Verhaltensmodellierung in der UML

- **Aktivitätsmodell**
 - »Was geschieht in welcher Reihenfolge?«
 - z.B. Aktivitätsdiagramm
- Interaktionsmodell
 - »Wann ruft wer wen wie auf?«
 - z.B. Sequenzdiagramm
- Zustandsmodell
 - »Wie reagiert ein Objekt auf Ereignisse?«
 - Zustandsautomaten
- ... uvm

Verhaltensmodellierung in der UML

- **Aktivitätsmodell**
 - »Was geschieht in welcher Reihenfolge?«
 - z.B. Aktivitätsdiagramm
- **Interaktionsmodell**
 - »Wann ruft wer wen wie auf?«
 - z.B. Sequenzdiagramm
- **Zustandsmodell**
 - »Wie reagiert ein Objekt auf Ereignisse?«
 - Zustandsautomaten
- ... uvm

Verhaltensmodellierung in der UML

- **Aktivitätsmodell**
 - »Was geschieht in welcher Reihenfolge?«
 - z.B. Aktivitätsdiagramm
- **Interaktionsmodell**
 - »Wann ruft wer wen wie auf?«
 - z.B. Sequenzdiagramm
- **Zustandsmodell**
 - »Wie reagiert ein Objekt auf Ereignisse?«
 - Zustandsautomaten
- ... uvm

Verhaltensmodellierung in der UML

- **Aktivitätsmodell**
 - »Was geschieht in welcher Reihenfolge?«
 - z.B. Aktivitätsdiagramm
- **Interaktionsmodell**
 - »Wann ruft wer wen wie auf?«
 - z.B. Sequenzdiagramm
- **Zustandsmodell**
 - »Wie reagiert ein Objekt auf Ereignisse?«
 - Zustandsautomaten
- ... uvm

UML-Zustandsautomaten

- **Zustände: Name, Verhalten (3×)**
- Übergänge: Trigger [Guards] / Verhalten
- Entscheidungen
- Pseudozustände, Historien, Hierarchien, Regionen ...

UML-Zustandsautomaten

- Zustände: Name, Verhalten (3×)
- Übergänge: Trigger [Guards] / Verhalten
- Entscheidungen
- Pseudozustände, Historien, Hierarchien, Regionen ...

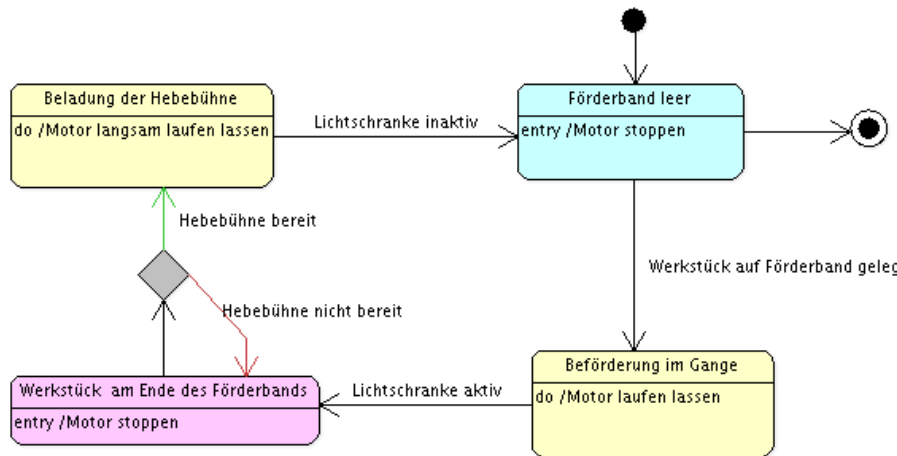
UML-Zustandsautomaten

- Zustände: Name, Verhalten (3×)
- Übergänge: Trigger [Guards] / Verhalten
- Entscheidungen
- Pseudozustände, Historien, Hierarchien, Regionen ...

UML-Zustandsautomaten

- Zustände: Name, Verhalten (3×)
- Übergänge: Trigger [Guards] / Verhalten
- Entscheidungen
- Pseudozustände, Historien, Hierarchien, Regionen ...

Zustandsautomat



Sequenzdiagramm

- Stellt ein oder mehrere Szenarien als *eine Interaktion* dar
- Objekte, Lebenslinien, Nachrichten (Methoden)
- darüber hinaus: Kontrollfluss und Schachtelung
 - optionale Ausführung
 - alternative Ausführungen
 - Referenzen auf andere Interaktionsdiagramme
 - Sprungmarken

Sequenzdiagramm

- Stellt ein oder mehrere Szenarien als *eine Interaktion* dar
- Objekte, Lebenslinien, Nachrichten (Methoden)
- darüber hinaus: Kontrollfluss und Schachtelung
 - optionale Ausführung
 - alternative Ausführungen
 - Referenzen auf andere Interaktionsdiagramme
 - Sprungmarken

Sequenzdiagramm

- Stellt ein oder mehrere Szenarien als *eine Interaktion* dar
- Objekte, Lebenslinien, Nachrichten (Methoden)
- darüber hinaus: Kontrollfluss und Schachtelung
 - optionale Ausführung
 - alternative Ausführungen
 - Referenzen auf andere Interaktionsdiagramme
 - Sprungmarken

Sequenzdiagramme

