Unified Modeling Language 2

Marvin Frommhold

17.11.2008

Gliederung

Was ist die UML?

- standardisierte Sprache für die Modellierung von Software (ISO/IEC 19501)

Was ist die UML?

- standardisierte Sprache für die Modellierung von Software (ISO/IEC 19501)

definiert:

- Bezeichner für Begriffe aus der Modellierung und legt Beziehungen dieser fest

Was ist die UML?

- standardisierte Sprache für die Modellierung von Software (ISO/IEC 19501)

definiert:

- Bezeichner für Begriffe aus der Modellierung und legt Beziehungen dieser fest
- graphische Notationen für diese Begriffe und für Modelle von statischen Strukturen und dynamischen Abläufen formuliert durch diese Begriffe

Was ist die UML?

- standardisierte Sprache für die Modellierung von Software (ISO/IEC 19501)

definiert:

- Bezeichner für Begriffe aus der Modellierung und legt Beziehungen dieser fest
- graphische Notationen für diese Begriffe und für Modelle von statischen Strukturen und dynamischen Abläufen formuliert durch diese Begriffe

dominierende Sprache für Modellierung von betrieblichen Anwendungs- und Softwaresystemen

Geschichte

- Anfang der 90er aufkommende objekt-orientierte Softwareentwicklung
- Bedarf an geeigneten Modellierungssprachen und -methoden
- 1997 als Standard von der OMG akzeptiert und übernommen

Geschichte

- Anfang der 90er aufkommende objekt-orientierte Softwareentwicklung
- Bedarf an geeigneten Modellierungssprachen und -methoden
- 1997 als Standard von der OMG akzeptiert und übernommen

$\rightarrow$ UML 1.x

Geschichte

- Anfang der 90er aufkommende objekt-orientierte Softwareentwicklung
- Bedarf an geeigneten Modellierungssprachen und -methoden
- 1997 als Standard von der OMG akzeptiert und übernommen

$\rightarrow$ UML 1.x

- OMG für Standardisierung, Pflege und Weiterentwicklung verantwortlich
- 1999 beginn der Entwicklung von UML2
- September 2004 endgültig abgenommene Dokumente

Geschichte

- Anfang der 90er aufkommende objekt-orientierte Softwareentwicklung
- Bedarf an geeigneten Modellierungssprachen und -methoden
- 1997 als Standard von der OMG akzeptiert und übernommen

→ UML 1.x

- OMG für Standardisierung, Pflege und Weiterentwicklung verantwortlich
- 1999 beginn der Entwicklung von UML2
- September 2004 endgültig abgenommene Dokumente

→ UML 2.0

Geschichte

- Anfang der 90er aufkommende objekt-orientierte Softwareentwicklung
- Bedarf an geeigneten Modellierungssprachen und -methoden
- 1997 als Standard von der OMG akzeptiert und übernommen

→ UML 1.x

- OMG für Standardisierung, Pflege und Weiterentwicklung verantwortlich
- 1999 beginn der Entwicklung von UML2
- September 2004 endgültig abgenommene Dokumente

→ UML 2.0

aktuelle Version vom 21.10.2008: 2.2 Beta 1

Strukturierung der Spezifikation

- insgesamt vier Teile
- *UML 2 Infrastructure Specification* beschreibt die häufigst verwendeten Elemente (z.B.: Klassen, Assoziation, . . .)

Strukturierung der Spezifikation

- insgesamt vier Teile
- *UML 2 Infrastructure Specification* beschreibt die häufigst verwendeten Elemente (z.B.: Klassen, Assoziation, . . .)
- *UML 2 Superstructure Specification* legt die verschiedenen Spracheinheiten fest (z.B. Aktivität, Zustandsautomat)

Strukturierung der Spezifikation

- insgesamt vier Teile
- *UML 2 Infrastructure Specification* beschreibt die häufigst verwendeten Elemente (z.B.: Klassen, Assoziation, . . .)
- *UML 2 Superstructure Specification* legt die verschiedenen Spracheinheiten fest (z.B. Aktivität, Zustandsautomat)
- *UML 2 Object Constraint Language* spezifiziert die Object Constraint Language (legt Bedingungen fest: Constraints)

Strukturierung der Spezifikation

- insgesamt vier Teile
- *UML 2 Infrastructure Specification* beschreibt die häufigst verwendeten Elemente (z.B.: Klassen, Assoziation, . . .)
- *UML 2 Superstructure Specification* legt die verschiedenen Spracheinheiten fest (z.B. Aktivität, Zustandsautomat)
- *UML 2 Object Constraint Language* spezifiziert die Object Constraint Language (legt Bedingungen fest: Constraints)
- *UML 2 Diagram Interchange* spezifiziert Layout der Diagramme sowie Austauschformat (XMI)

Diagramme

- visuelle Darstellung der Modellierung
- 13 verschiedene Diagrammtypen
- Einteilung in Verhaltens- und Strukturdiagramme

Strukturdiagramme

- Klassendiagramm
- Kompositionsstrukturdiagramm (Montagediagramm)
- Komponentendiagramm
- Verteilungsdiagramm
- Objektdiagramm
- Paketdiagramm

Strukturdiagramme

- Klassendiagramm
- Kompositionsstrukturdiagramm (Montagediagramm)
- Komponentendiagramm
- Verteilungsdiagramm
- Objektdiagramm
- Paketdiagramm

Komponentendiagramm

- Darstellung umfasst typischerweise Komponenten und deren Schnittstellen bzw. Ports
  - Port: Menge von angebotenen/benötigten Schnittstellen
- Abhängigkeitsbeziehungen und Konnektoren zwischen Komponenten
- Komponenteninneres wird durch Notationselemente aus anderen Diagrammtypen beschrieben
- Einsatz: komponentenbasierte Systeme

Beispiel Komponentendiagramm

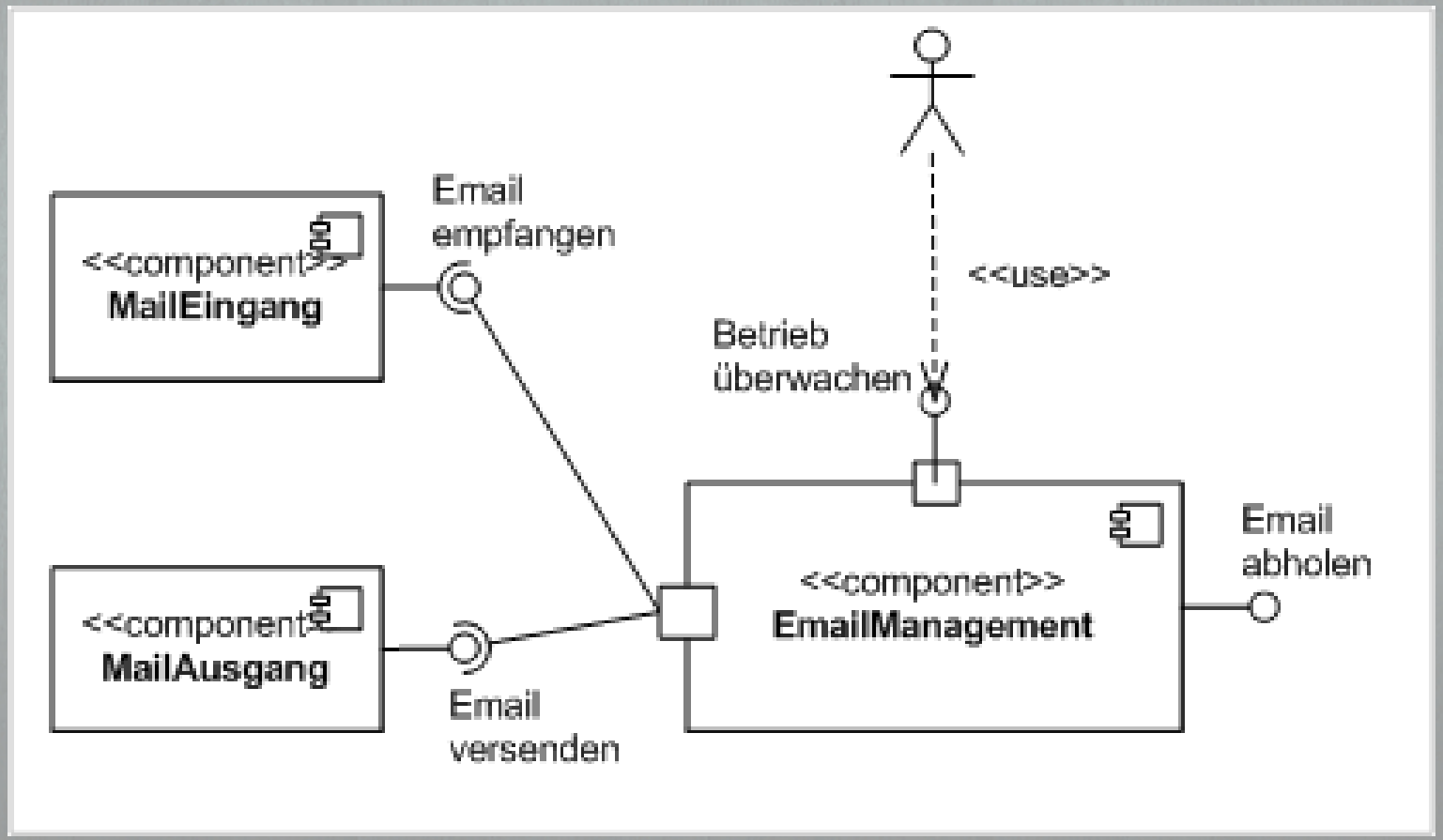


Abbildung: http://upload.wikimedia.org/wikipedia/de/7/74/Component-4.png

Paketdiagramm

- Darstellung der Schichtung und/oder Unterteilung des Software-Systems
- bspw. Paketstruktur bei Java-Programmen

Beispiel Paketdiagramm

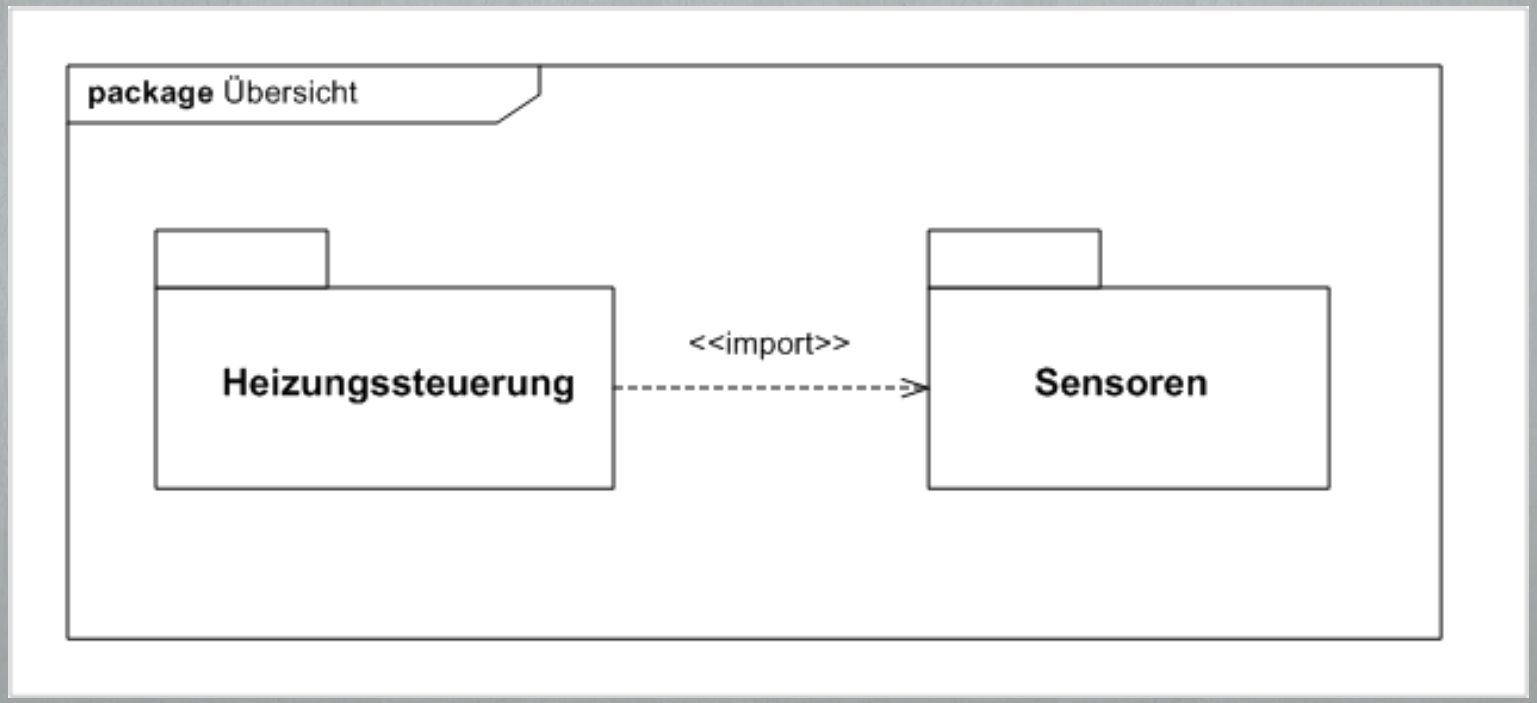
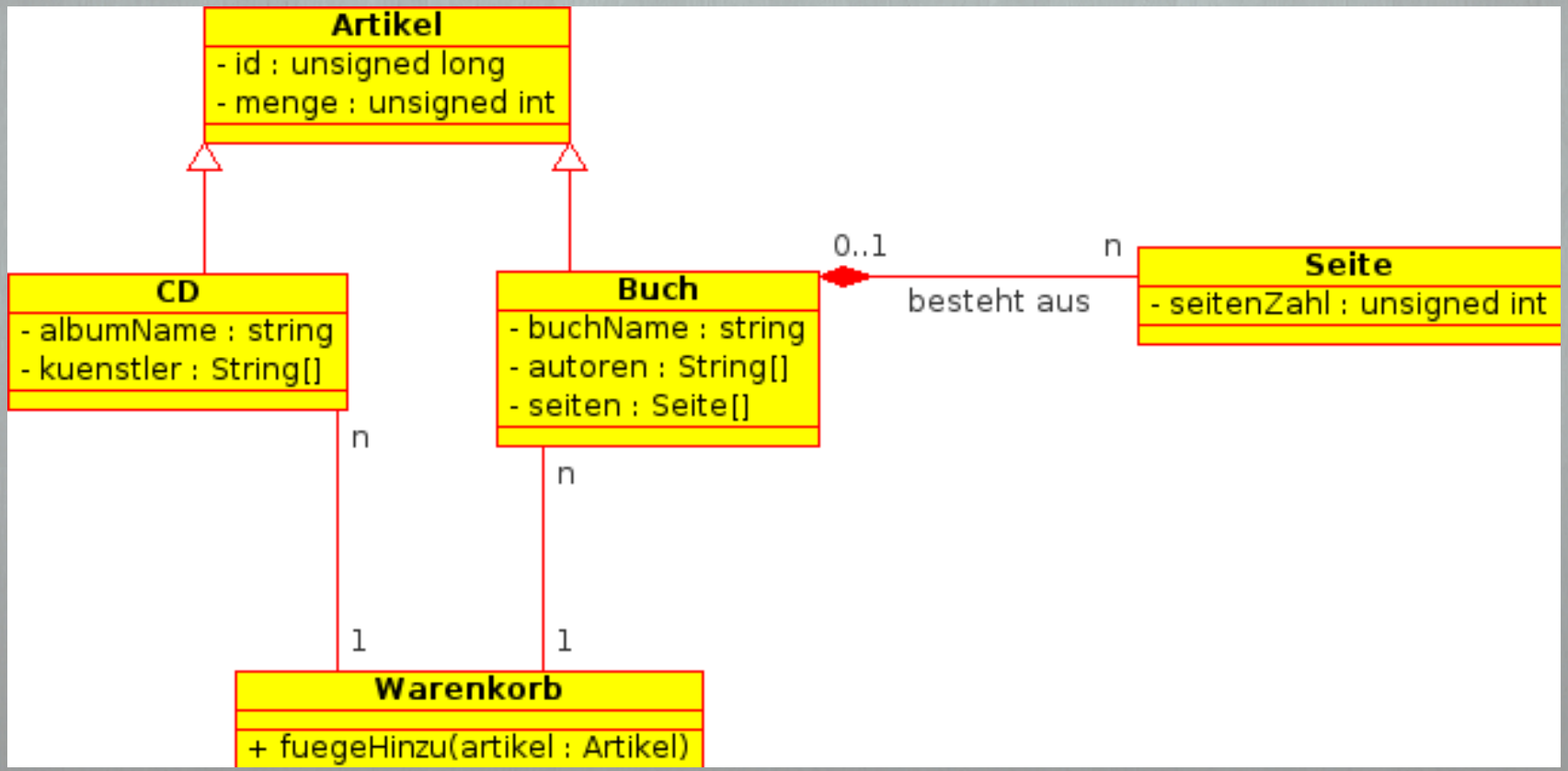


Abbildung: http://upload.wikimedia.org/wikipedia/de/7/77/Packageimport-2.png

Klassendiagramm

- Übersicht über Klassen und deren Attribute und Methoden sowie der Beziehungen von Klassen untereinander
- wichtigste Beziehungen (Assoziationen):
  - **Assoziation:** einfache Beziehung untereinander
  - **Generalisierung (Vererbung):** Ableitung einer Klasse aus einer/mehreren anderen
  - **Komposition:** eine Klasse ist Teil einer anderen (Komposition kann allein nicht existieren)
  - **Aggregation:** schwache Komposition (kann allein existieren)
- meist auch Angabe von **Kardinalitäten** (wie viele Instanzen einer Klasse stehen in Beziehung zu wie vielen Instanzen einer anderen Klasse)

Beispiel Klassendiagramm



Verhaltensdiagramme

- Aktivitätsdiagramm
- Anwendungsfalldiagramm (Use-Case/Nutzfalldiagramm)
- Interaktionsübersichtsdiagramm
- Kommunikationsdiagramm
- Sequenzdiagramm
- Zeitverlaufsdiagramm
- Zustandsdiagramm

Verhaltensdiagramme

- Aktivitätsdiagramm
- Anwendungsfalldiagramm (Use-Case/Nutzfalldiagramm)
- Interaktionsübersichtsdiagramm
- Kommunikationsdiagramm
- Sequenzdiagramm
- Zeitverlaufsdiagramm
- Zustandsdiagramm

Aktivitätsdiagramm

- Beschreibung des Ablaufs eines Anwendungsfalls
- **Rechtecke:** Aktivitätsknoten (Übergabe/Empfang von Werten)
- **abgerundete Rechtecke:** Aktivität
- **Kästchen:** Pins (zeigen Objektfluss an)
- **Schwarzer Punkt:** Startpunkt der Aktivität

Beispiel: Aktivitätsdiagramm

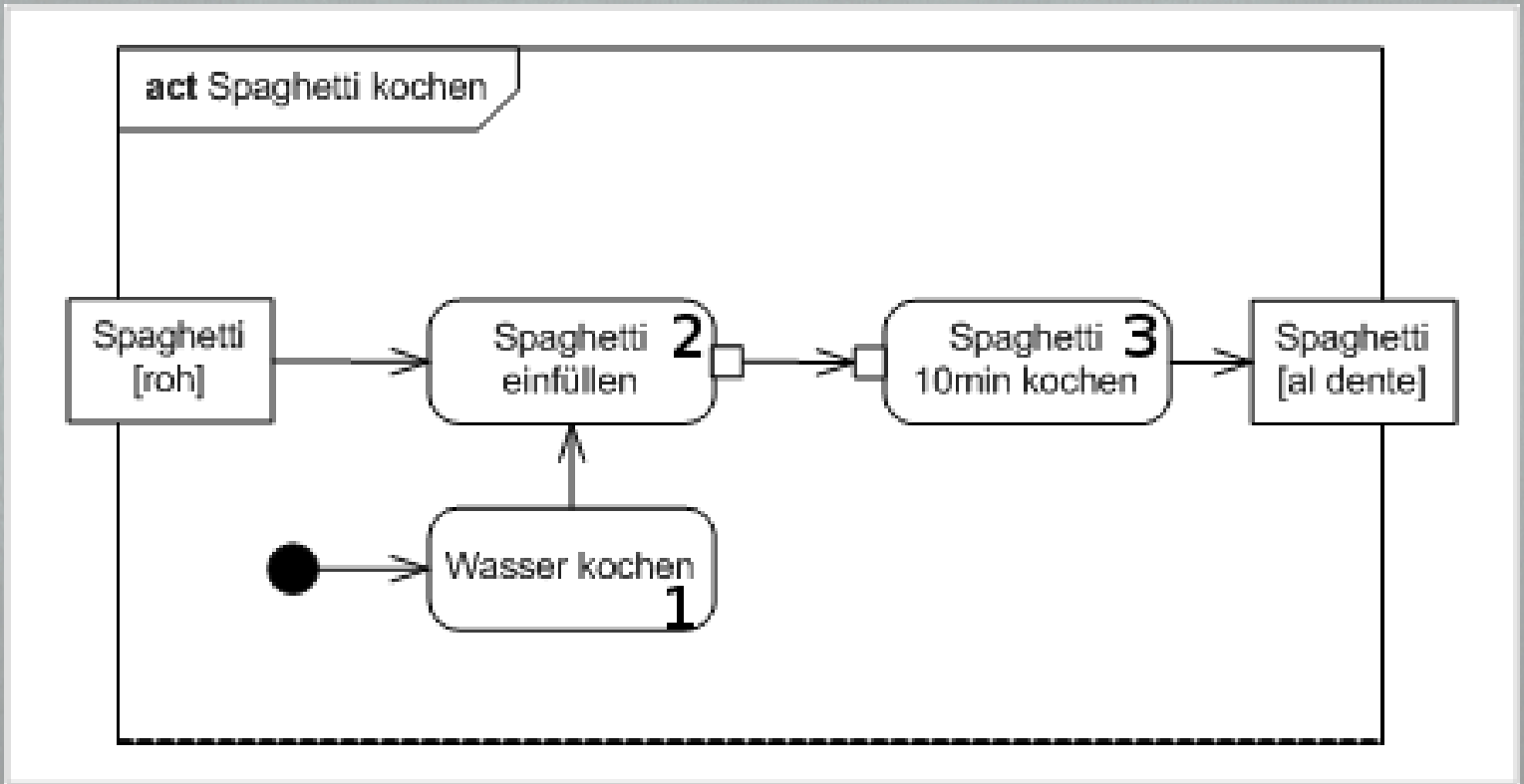


Abbildung:
http://upload.wikimedia.org/wikipedia/commons/1/12/Activity_diagram_-1-.png

Use-Case-Diagramm

- Anwendungsfälle und Akteure mit ihren Abhängigkeiten und Beziehungen zueinander
- **Ovale:** Anwendungsfall
- **Männchen:** Akteure
- Anwendungsfälle können andere einschließen (<<*include*>>) bzw. erweitern (<<*extends*>>)

Beispiel Use-Case-Diagramm

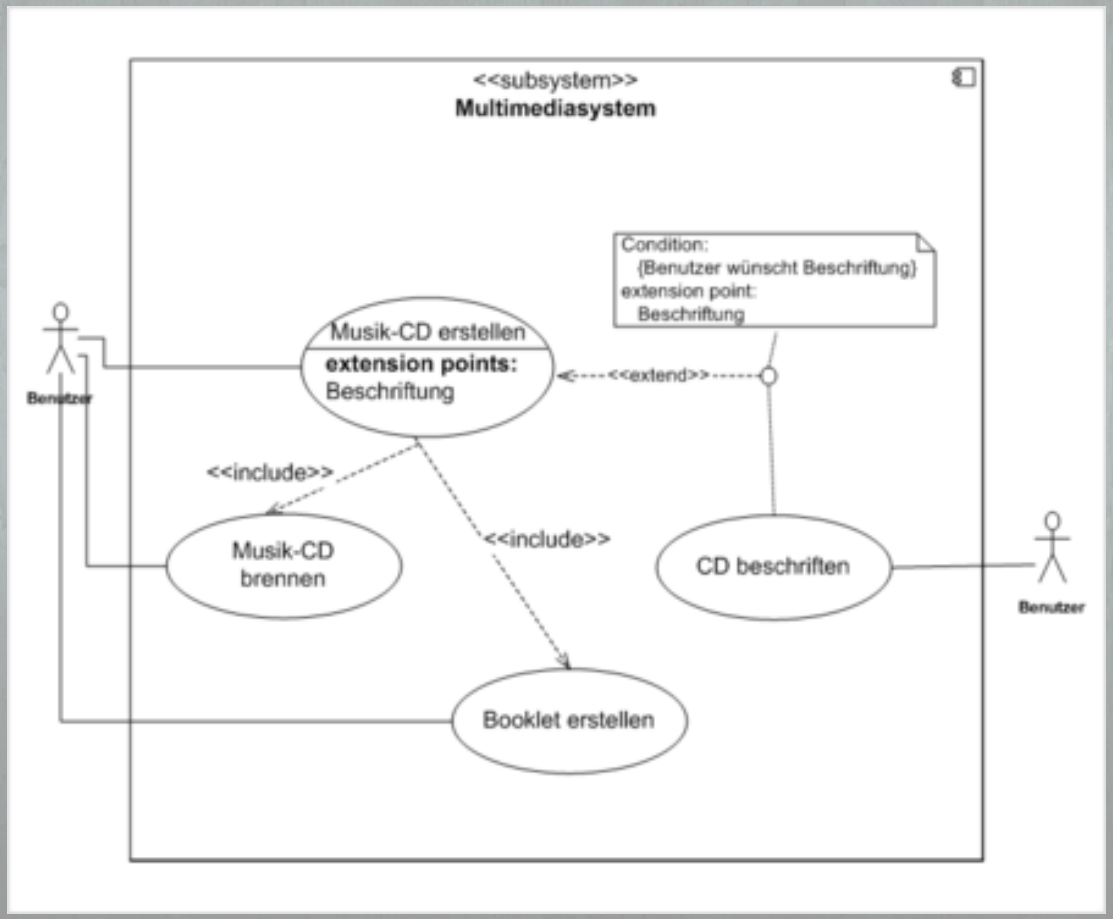
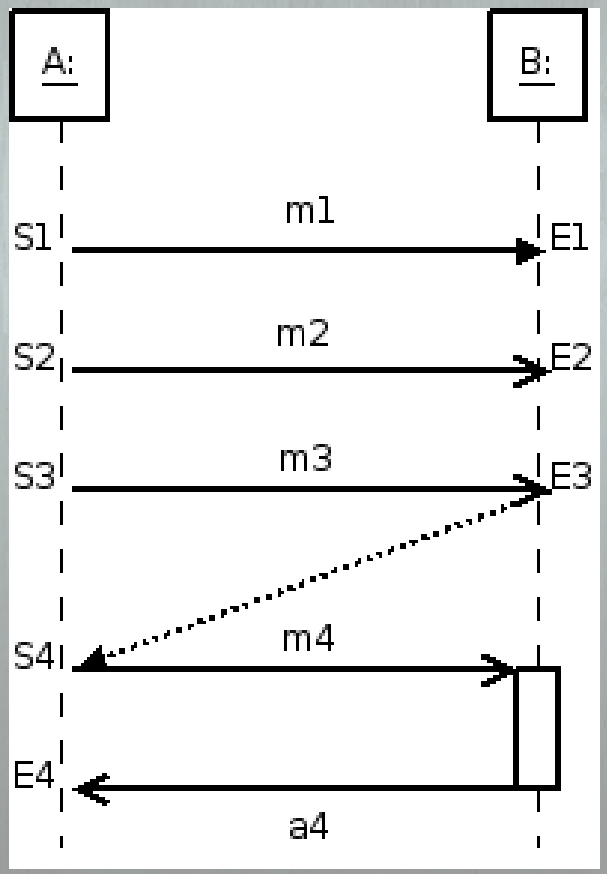


Abbildung: http://upload.wikimedia.org/wikipedia/commons/9/99/Use-case-8.png

Sequenzdiagramm

- Grafische Darstellung einer Interaktion
- zeigt Operationsaufrufe zwischen den beteiligten Komponenten in zeitlicher Reihenfolge
- **synchrone Aufrufe:** zeitliche Reihenfolge entscheidend
- **asynchrone Aufrufe:** zweiter Aufruf kann vor Empfangsnachricht des ersten Aufrufs geschehen
- **Balken auf Lebenslinien:** Objekt hat Kontrollfluss

Beispiel Sequenzdiagramm



Zustandsdiagramm

- Übersicht über Zustände, die Objekt/Teilsystem unter bestimmten Bedingungen annehmen kann
- auf Ereignisse folgen Zustandsänderungen/-übergange
- **Verhaltenszustandsdiagramm:** modelliert Verhalten eines Objekts
- **Protokollzustandsautomat:** spezifiziert zulässige Nutzung der Elemente eines Objekts
- **abgerundetes Rechteck:** Zustand
- **Raute:** Kreuzung oder Entscheidung
- **Pfeile:** Übergänge (möglichst beschriften)

Beispiel Zustandsdiagramm

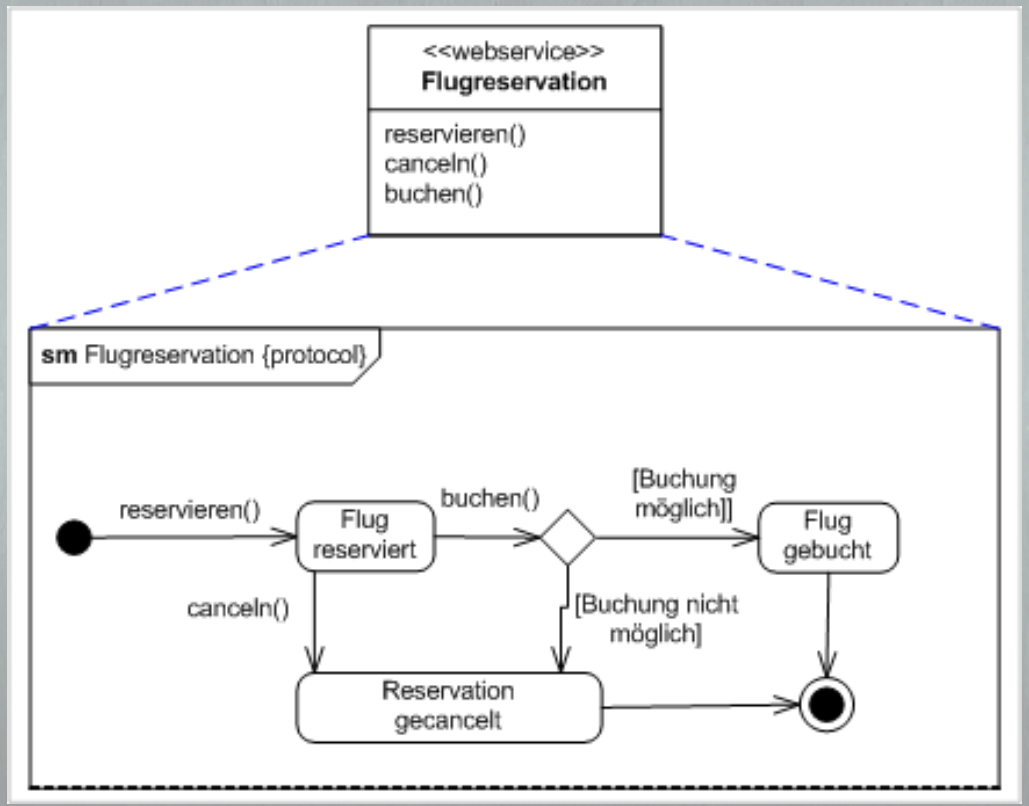


Abbildung:
http://upload.wikimedia.org/wikipedia/commons/1/13/Statemachine-6.png

Quicksheets der Notationselemente

http://www.oose.de/notationuml14.htm

CASE-Werkzeug: Umbrello

- Editor zum Erstellen von UML-Diagrammen
- basiert auf KDE4/QT4; funktioniert auf allen Plattformen, für die KDE4 verfügbar ist

CASE-Werkzeug: Umbrello

- Editor zum Erstellen von UML-Diagrammen
- basiert auf KDE4/QT4; funktioniert auf allen Plattformen, für die KDE4 verfügbar ist
  $\rightarrow$ Linux/Unix, Windows und Mac OS X

CASE-Werkzeug: Umbrello

- Editor zum Erstellen von UML-Diagrammen
- basiert auf KDE4/QT4; funktioniert auf allen Plattformen, für die KDE4 verfügbar ist
  $\rightarrow$ Linux/Unix, Windows und Mac OS X
- kostenlos, Open-Source-Lizenz
- nutzt XMI-Dateiformat zur Speicherung

Features:

- fertige Elemente “zusammenfügen” /modellieren

Features:

- fertige Elemente “zusammenfügen” /modellieren
- gute Modellierungsmöglichkeiten für UML-Klassendiagramm

Features:

- fertige Elemente “zusammenfügen” /modellieren
- gute Modellierungsmöglichkeiten für UML-Klassendiagramm
- Quellcode-Generierung für verschiedene Sprachen (C++, Java, PHP, HTML, . . .)

Features:

- fertige Elemente “zusammenfügen” /modellieren
- gute Modellierungsmöglichkeiten für UML-Klassendiagramm
- Quellcode-Generierung für verschiedene Sprachen (C++, Java, PHP, HTML, . . .)
- Diagramme exportierbar in verschiedene Bild-Formate (PNG, JPG, SVG, . . .)

Features:

- fertige Elemente “zusammenfügen” /modellieren
- gute Modellierungsmöglichkeiten für UML-Klassendiagramm
- Quellcode-Generierung für verschiedene Sprachen (C++, Java, PHP, HTML, . . .)
- Diagramme exportierbar in verschiedene Bild-Formate (PNG, JPG, SVG, . . .)

Probleme:

- nicht alle Diagrammtypen verfügbar
- begrenzte Modellierungsmöglichkeiten bei den Diagrammen

Features:

- fertige Elemente “zusammenfügen” /modellieren
- gute Modellierungsmöglichkeiten für UML-Klassendiagramm
- Quellcode-Generierung für verschiedene Sprachen (C++, Java, PHP, HTML, . . .)
- Diagramme exportierbar in verschiedene Bild-Formate (PNG, JPG, SVG, . . .)

Probleme:

- nicht alle Diagrammtypen verfügbar
- begrenzte Modellierungsmöglichkeiten bei den Diagrammen
- scheinbar nur UML 1.x

CASE-Werkzeug: Dia

- Zeichenprogramm für Diagramme aller Art
- basiert auf GTK+
- verfügbar für Linux/Unix und Windows
- kostenlos, Open-Source-Lizenz

CASE-Werkzeug: Dia

- Zeichenprogramm für Diagramme aller Art
- basiert auf GTK+
- verfügbar für Linux/Unix und Windows
- kostenlos, Open-Source-Lizenz
- nutzt eigenes Format zur Speicherung der Diagramme
- reines Zeichenprogramm, keine Code-Generierung usw.

Quellen

- Wikipedia, Die freie Enzyklopädie: *Unified Modeling Language*, Wikimedia Foundation Inc.
- http://www.oose.de/uml.htm: *Unified Modeling Language*, oose. Innovative Informatik
- http://uml.sourceforge.net/: *Umbrello UML Modeller*
- http://projects.gnome.org/dia/: *Dia, a drawing program*

Vielen Dank für Eure Aufmerksamkeit!