

informatik

Harald Störrle

# UML 2 für Studenten

Mit UML-Syntax-Poster

# UML 2 für Studenten

# UML 2 für Studenten

## Inhaltsverzeichnis

### UML 2 für Studenten

#### Inhaltsverzeichnis

Vorwort

#### Teil I Einführung

Kapitel 1 UML (nicht nur) für Studenten

Kapitel 2 UML im Überblick

Kapitel 3 UML im Kontext

Kapitel 4 UML in der Praxis

#### Teil II Struktur

Kapitel 5 Klassen und Beziehungen

Kapitel 6 Architektur und Komponenten

Kapitel 7 Nichtfunktionale Anforderungen

Kapitel 8 Object Constraint Language

#### Teil III Verhalten

Kapitel 9 Nutzfälle

Kapitel 10 Zustandsautomaten

Kapitel 11 Aktivitäten

Kapitel 12 Interaktionen

#### Teil IV Anhänge

Anhang A UML-Syntax

Anhang B UML Diagramme und Modelle

Anhang C UML Erweiterbarkeit

Anhang D UML Metamodell

Anhang E Glossar

#### Literaturverzeichnis

# Inhaltsverzeichnis

Sachregister

Vorwort

Teil I Einführung

UML (nicht nur) für Studenten

UML im Überblick

UML im Kontext

UML in der Praxis

Teil II Struktur

Klassen und Beziehungen

Architektur und Komponenten

Nichtfunktionale Anforderungen

Object Constraint Language

Teil III Verhalten

Nutzfälle

Zustandsautomaten

Aktivitäten

Interaktionen

Teil IV Anhänge

Anhang A UML-Syntax

A.1 Grafische Notationselemente

A.1.1 Allgemein

A.1.2 Klassendiagramme 1 Klassen und Objekte

A.1.3 Klassendiagramme 2 ungerichtete Beziehungen

A.1.4 Klassendiagramme 3 gerichtete Beziehungen

A.1.5 Montagediagramme

A.1.6 Paketdiagramme

A.1.7 Verteilungsdiagramme

A.1.8 Nutzfalldiagramme

A.1.9 Zustandsdiagramme

# Inhaltsverzeichnis

A.1.10 Aktivitätsdiagramme

A.1.11 Interaktionsdiagramme

## A.2 Anschriften

Klassen und Assoziationen

Pakete

OCL

Zustandsautomaten

Interaktionen

## A.3 Eindeutige Bezeichner

### A.4 Namen

Systeme

Domänen

Prozesse, Ereignisse

Funktionen, Dienste, Aktivitäten

Operationen

Zustände

Zustandsautomaten

Klassen, Attribute, Pakete, Komponenten

Zulässige Zeichen

### A.5 Layout

Form follows function

Beschränkung auf das Wesentliche

Leserichtung beachten

Gleiches gleich darstellen

## Anhang B UML Diagramme und Modelle

Klassendiagramme

Architekturdiagramme

Kollaborationen

Installationsdiagramme

Nutzfalldiagramme

Zustandsautomaten

Aktivitätsdiagramme

Interaktionsdiagramme

# Inhaltsverzeichnis

Anhang C UML Erweiterbarkeit

C.1 Stereotype

C.2 Tagged Values

C.3 Profile

Anhang D UML Metamodell

Anhang E Glossar

Literaturverzeichnis

Sachregister

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

# Inhaltsverzeichnis

U

V

W

X

Z

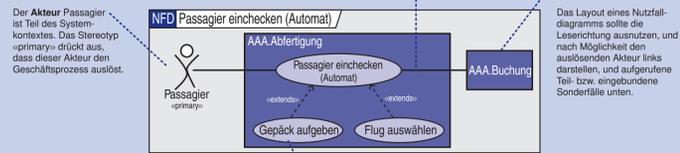
Ins Internet: Weitere Infos zum Buch, Downloads, etc.

© Copyright

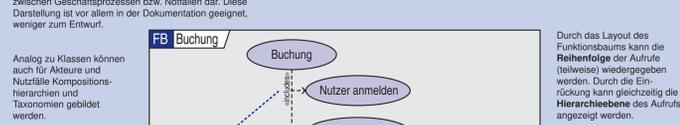
UML 2.0 Syntax Poster

## Nutzfalldiagramme

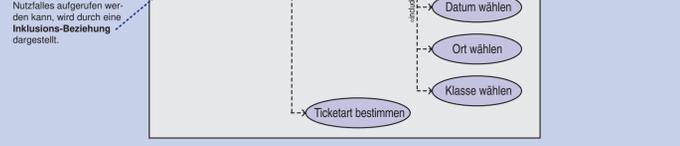
**Nutzfalldiagramme** sind vor allem als Übersichtsdarstellungen nützlich. Sie stellen elementare Elemente eines Systemkontextes dar (also Akteure und Nachbarsysteme), andererseits Geschäftsprozesse bzw. Nutzfälle des Systems. Daneben können auch Abhängigkeiten zwischen Geschäftsprozessen bzw. Nutzfällen mit Inklusions- und Erweiterungsbeziehungen angezeigt werden.



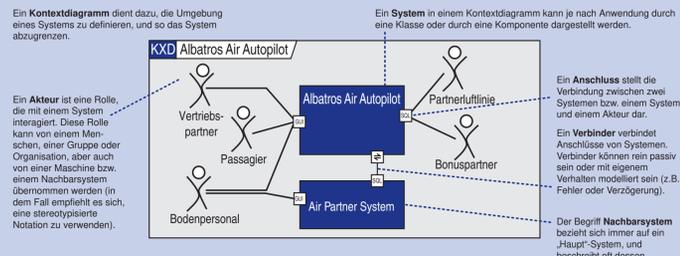
Die **Assoziation** zeigt an, dass der Akteur bzw. das Nachbarsystem an diesem Geschäftsprozess teilnimmt. Das Layout eines Nutzfalldiagramms sollte die Lesartigkeit ausnutzen, und nach Möglichkeit den auslösenden Akteur links darstellen, und aufgerufene Teil- bzw. eingebundene Sonderfälle unten.



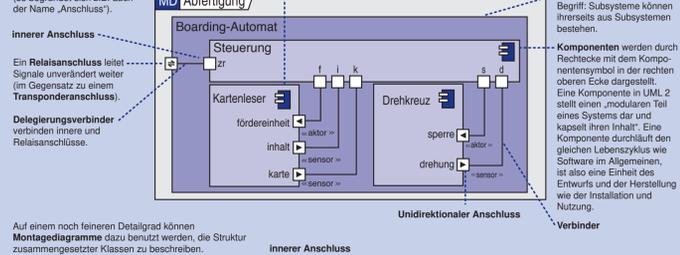
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



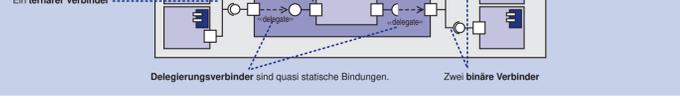
Ein **Kontextdiagramm** dient dazu, die Umgebung eines Systems zu definieren, und so das System abzugrenzen. Ein **Akteur** ist eine Rolle, die mit einem System interagiert. Diese Rolle kann von einem Menschen, einer Gruppe oder Organisation, aber auch von einer Maschine bzw. einem Nachbarsystem übernommen werden.



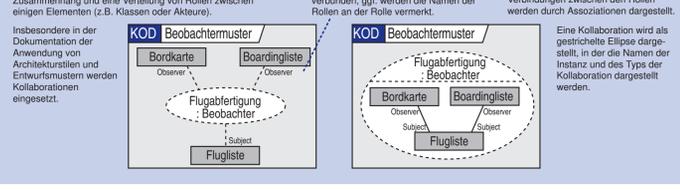
Ein **Anschluss** stellt die Verbindung zwischen zwei Systemen bzw. einem System und einem Akteur dar. Ein **Verbinders** verbindet Anschlüsse von Systemen. Ein **Aufbauhinweis** definiert einen Bereich einer Aktivität, in dem Massendaten nach verschiedenen Modi in groß verarbeitet werden.



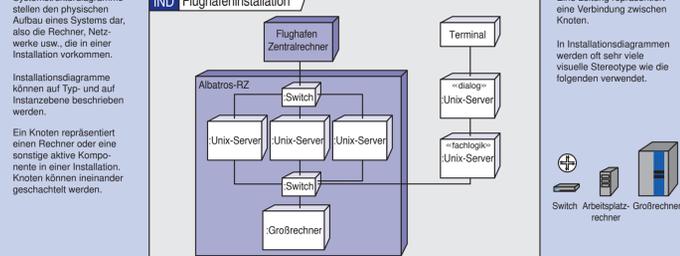
Ein **Relaisanschluss** leitet Signale unverändert weiter (im Gegensatz zu einem Transponderanschluss). Ein **Relaisanschluss** verbindet innere und Relaisanschlüsse. Ein **Relaisanschluss** verbindet innere und Relaisanschlüsse.



Ein **Pin** ist ein notwendiger Parameter für den Aufruf einer Aktion, bzw. ein garantiertes Resultat. Ein **Pin** ist ein notwendiger Parameter für den Aufruf einer Aktion, bzw. ein garantiertes Resultat.



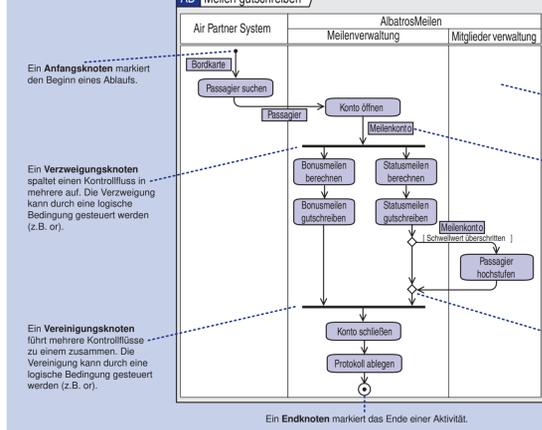
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



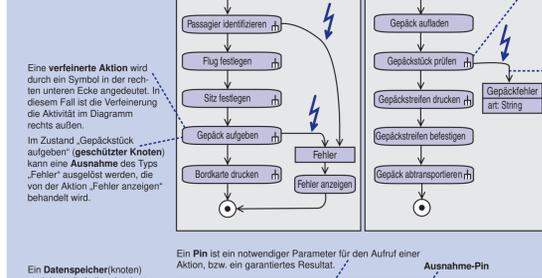
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

## Aktivitätsdiagramme

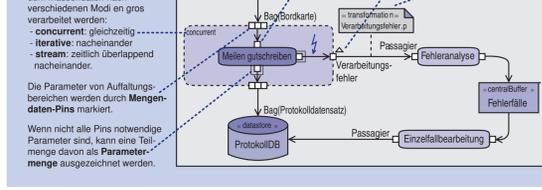
Ein **Aktivitätsdiagramm** kann zur Modellierung von Abläufen jeder Art benutzt werden, z.B. für Geschäftsprozesse, Nutzfälle oder algorithmische Abläufe. Ein **Aktivitätsdiagramm** kann zur Modellierung von Abläufen jeder Art benutzt werden, z.B. für Geschäftsprozesse, Nutzfälle oder algorithmische Abläufe.



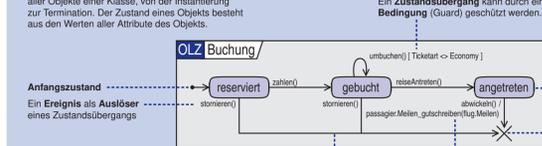
Ein **Startknoten** markiert den Beginn eines Ablaufs. Ein **Startknoten** markiert den Beginn eines Ablaufs. Ein **Startknoten** markiert den Beginn eines Ablaufs.



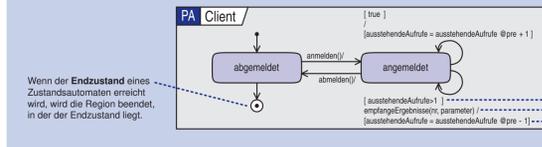
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



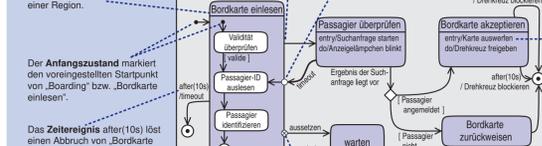
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

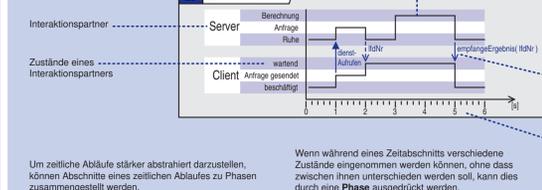


Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

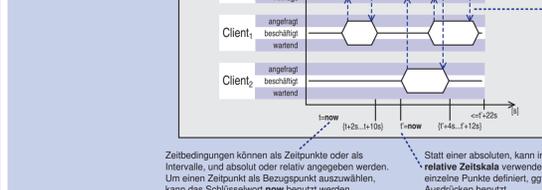


## Zeitdiagramme

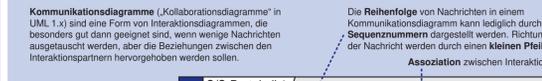
**Zeitdiagramme** sind eine Form von Interaktionsdiagrammen, die besonders zur Modellierung und Visualisierung komplexer zeitlicher Abläufe geeignet sind. **Zeitdiagramme** sind eine Form von Interaktionsdiagrammen, die besonders zur Modellierung und Visualisierung komplexer zeitlicher Abläufe geeignet sind.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



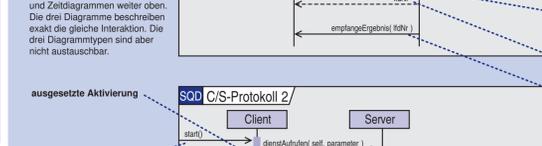
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



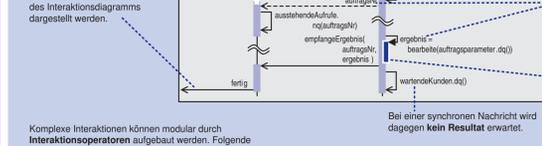
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



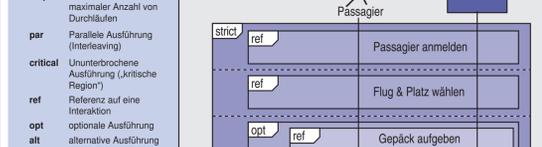
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



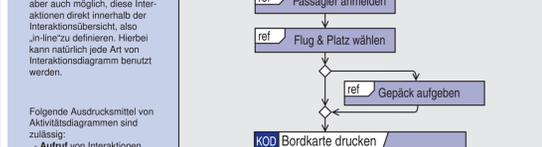
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

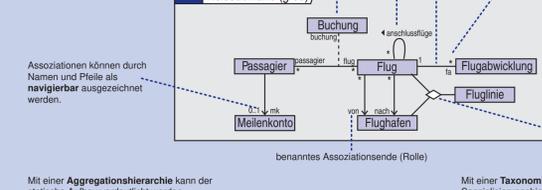


Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

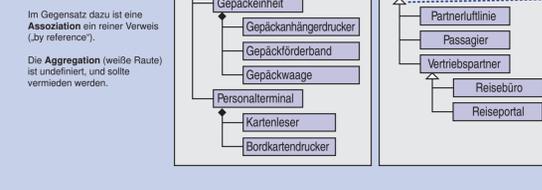


## Klassendiagramme

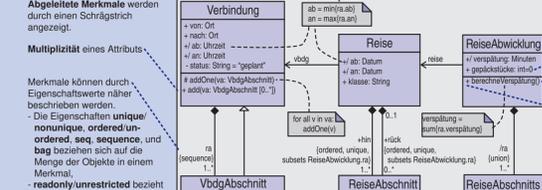
**Analyse-Klassendiagramme** stellen Geflechte von Konzepten dar, wobei Konzepte durch Klassen und ihre Beziehungen durch Assoziationen dargestellt werden. **Analyse-Klassendiagramme** stellen Geflechte von Konzepten dar, wobei Konzepte durch Klassen und ihre Beziehungen durch Assoziationen dargestellt werden.



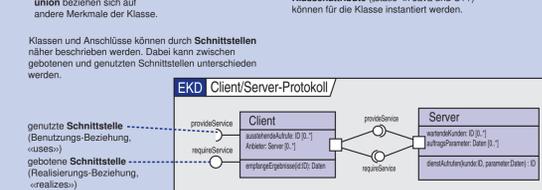
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



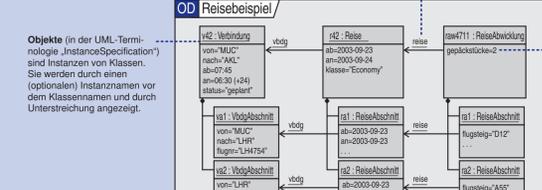
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



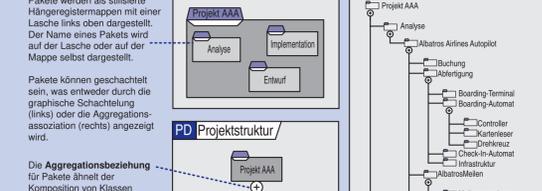
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



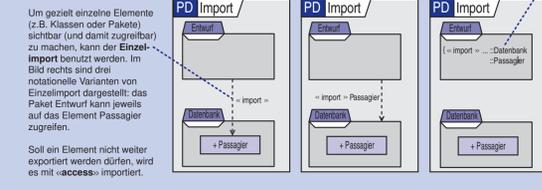
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.

## Paketdiagramme

**Paketdiagramme** stellen die Gruppierung von Modellelementen zu Namensräumen dar. **Paketdiagramme** stellen die Gruppierung von Modellelementen zu Namensräumen dar.



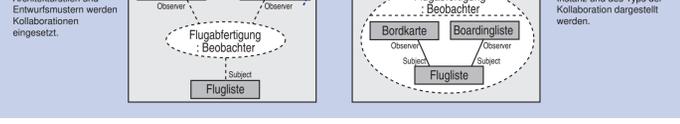
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



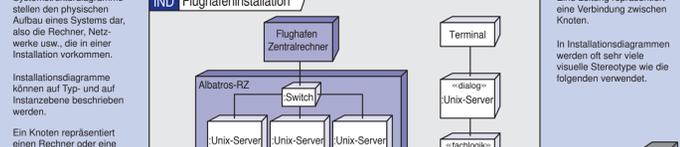
Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.

## Kollaborationen

**Kollaborationen** definieren einen strukturellen Zusammenhang und eine Verteilung von Rollen zwischen einzelnen Elementen (z.B. Klassen oder Akteure). **Kollaborationen** definieren einen strukturellen Zusammenhang und eine Verteilung von Rollen zwischen einzelnen Elementen (z.B. Klassen oder Akteure).



Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.



Ein **Funktionsbaum** stellt die Aufrufabhängigkeiten zwischen Geschäftsprozessen bzw. Notfällen dar. Diese Darstellung ist vor allem in der Dokumentation geeignet, weniger zum Entwurf. Analog zu Klassen können auch für Akteure und Nutzfälle Kompositionshierarchien und Taxonomieen gebildet werden.

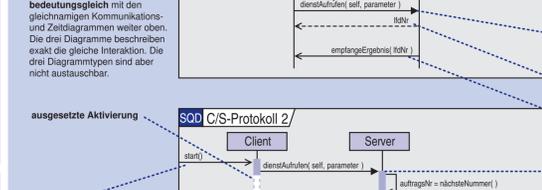


## Kommunikationsdiagramme

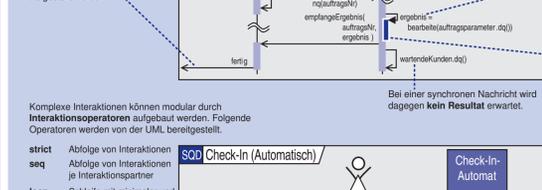
**Kommunikationsdiagramme** („Kollaborationsdiagramme“ in UML 1.x) sind eine Form von Interaktionsdiagrammen, die besonders gut dann geeignet sind, wenn wenige Nachrichten ausgetauscht werden, aber die Beziehungen zwischen den Interaktionspartnern hervorgehoben werden sollen. **Kommunikationsdiagramme** („Kollaborationsdiagramme“ in UML 1.x) sind eine Form von Interaktionsdiagrammen, die besonders gut dann geeignet sind, wenn wenige Nachrichten ausgetauscht werden, aber die Beziehungen zwischen den Interaktionspartnern hervorgehoben werden sollen.



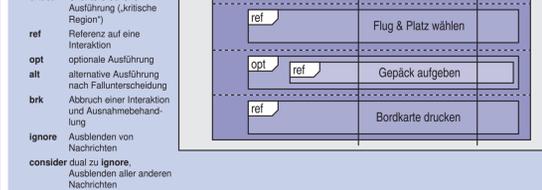
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



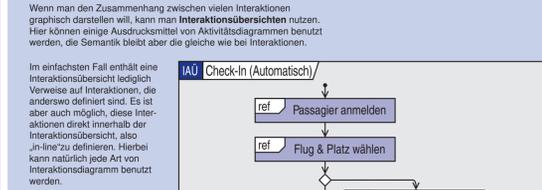
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



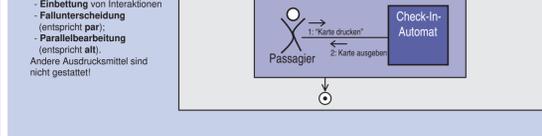
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

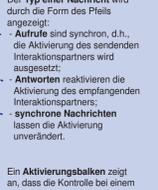


## Sequenzdiagramme

**Sequenzdiagramme** sind eine Form von Interaktionsdiagrammen, die besonders gut dann geeignet sind, wenn wenige Nachrichten ausgetauscht werden, aber die Beziehungen zwischen den Interaktionspartnern hervorgehoben werden sollen. **Sequenzdiagramme** sind eine Form von Interaktionsdiagrammen, die besonders gut dann geeignet sind, wenn wenige Nachrichten ausgetauscht werden, aber die Beziehungen zwischen den Interaktionspartnern hervorgehoben werden sollen.



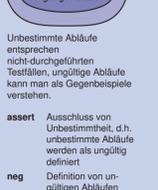
Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.



Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher. Ein **Objektzustandsautomat** beschreibt ein Verhalten ausschließlich über Zustände und Übergänge, aber ohne Seiteneffekte und Speicher.

