

2. Einleitung und Überblick über UML

2.1 Einleitung

2.2 Überblick

2.1 Einleitung

- Visualisierung, Spezifikation, Konstruktion und Dokumentation eines Softwaresystems erfordert mehrere *Sichten* (Views)

Einleitung (Forts.)

- Bearbeitung und Nutzung des Systems durch verschiedene Interessensgruppen
 - Endbenutzer
 - Analytiker und Entwickler
 - System-Administrator, -Integrator
 - Tester
 - Personen, die Dokumentation schreiben
 - Projektleiter

Einleitung (Forts.)

- UML bietet Vielzahl von Konzepten und Konstrukten zur Realisierung verschiedener Sichten an
- Grobe Unterteilung der Sichten in
 - Strukturelle Klassifikation
 - Dynamisches Verhalten
 - Modell Management

Einleitung (Forts.)

- *Strukturelle Klassifikation* beschreibt die Objekte eines Systems und deren Beziehung zu anderen Objekten
- Bereich *Dynamisches Verhalten* umfasst Sichten, die Änderungen des Systems in Abhängigkeit der Zeit beschreiben
- *Modell Management* beschäftigt sich mit der Organisation des Modell selbst

Einleitung (Forts.)

- UML besitzt *Erweiterungsmechanismen*, die auf alle Elemente angewendet werden können

UML Sichten und Diagramme

| <i>Bereich</i> | <i>Sicht</i> | <i>Diagramme</i> | <i>Hauptkonzepte</i> |
|-----------------|-------------------------|---------------------------|---|
| Struktur | Statische Sicht | Klassendiagramm | Klasse, Assoziation, Generalisierung, Abhängigkeit, Realisierung, Schnittstelle |
| | Anwendungsfall-Sicht | Anwendungsfalldiagramm | Anwendungsfall, Akteur, Assoziation, extend, include, Anwendungsfall, Generalisierung |
| | Implementations-Sicht | Komponentendiagramm | Komponente, Schnittstelle, Abhängigkeit, Realisierung |
| | Verteilungs-Sicht | Verteilungsdiagramm | Knoten, Komponente, Abhängigkeit, Ort |
| Dynamisch | Zustandsmaschinen-Sicht | Zustandsübergangsdiagramm | Zustand, Ereignis, Übergang, Aktion |
| | Aktivitäts-Sicht | Aktivitätsdiagramm | Zustand, Aktivität, Beendigungsübergang, Verzweigung, Synchronisation |
| | Interaktions-Sicht | Sequenzdiagramm | Interaktion, Objekt, Nachricht, Aktivierung |
| | | Kollaborationsdiagramm | Kollaboration, Interaktion, Kollaborationsrolle, Nachricht |
| Modell Mgmt. | Modell Mgmt. Sicht | Klassendiagramm | Paket, Subsystem, Modell |
| Erweiterbarkeit | Alle | Alle | Einschränkung, Stereotyp, Eigenschaftswert |

2.2 Überblick

- Überblick über die verschiedenen Sichten anhand eines Beispiels
- **Beispiel:** EDV-System für eine Theaterkasse (stark vereinfacht)

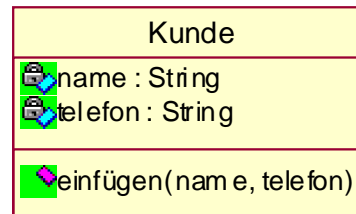
Statische Sicht (Static View)

- Modellierung der Konzepte des Anwendungsbereichs sowie interner Konzepte
- kein zeitabhängiges Verhalten
- Hauptbestandteile: *Klassen* (Classes) und ihre *Beziehungen* (Relationships) in Form von *Klassendiagrammen* (Class diagrams)

Statische Sicht (Forts.)

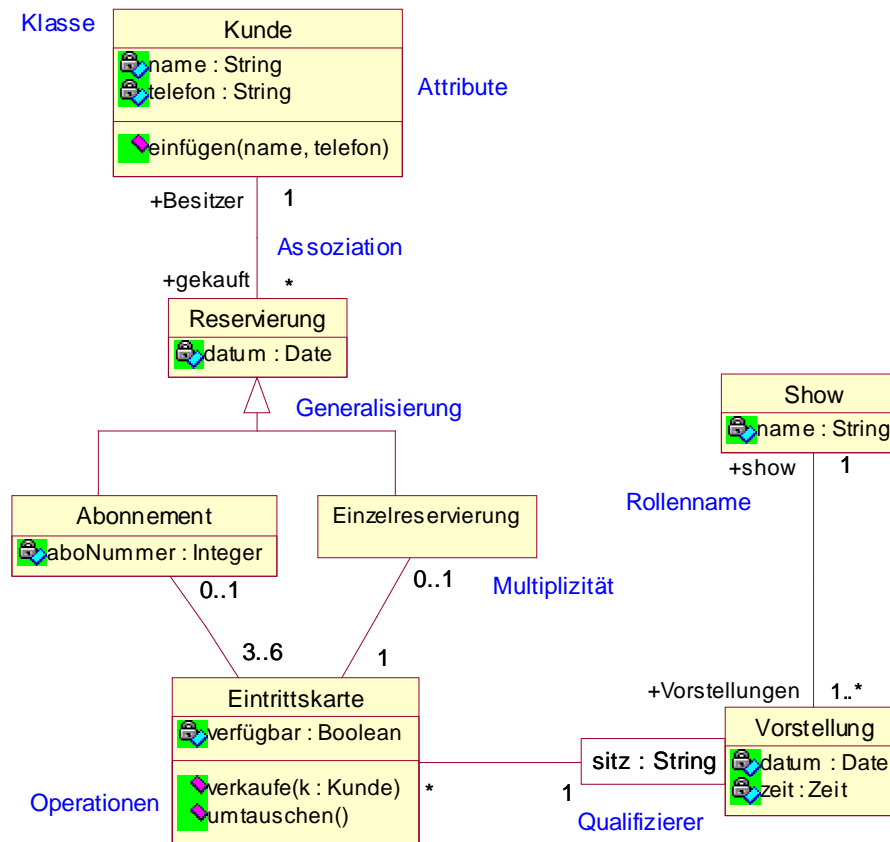
Graphische Darstellung:

- *Klassen* = Rechtecke mit getrennten Bereichen für *Attribute* (Attributes) und *Operationen* (Operations)



- *Beziehungen* = Linien zwischen betroffenen Klassen; Art der Beziehung unterscheidbar durch verschiedene Linienarten, -enden und -beschriftungen

Klassendiagramm *Theaterkasse*



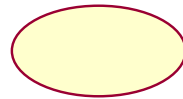
Anwendungsfall-Sicht (Use Case View)

- Modellierung der Funktionalität des Systems, wie sie von aussenstehenden Benutzern, sog. *Akteuren* (Actors), wahrgenommen wird
- Anwendungsfall beschreibt typische Interaktion zwischen Benutzer und System (*WAS* macht das System, aber *nicht WIE*)
- Als Akteure können auch andere Systeme auftreten

Anwendungsfall-Sicht (Forts.)

Graphische Darstellung:

- *Anwendungsfälle* = Ellipsen



kaufe Eintrittskarten

- *Akteure* = Strichmännchen

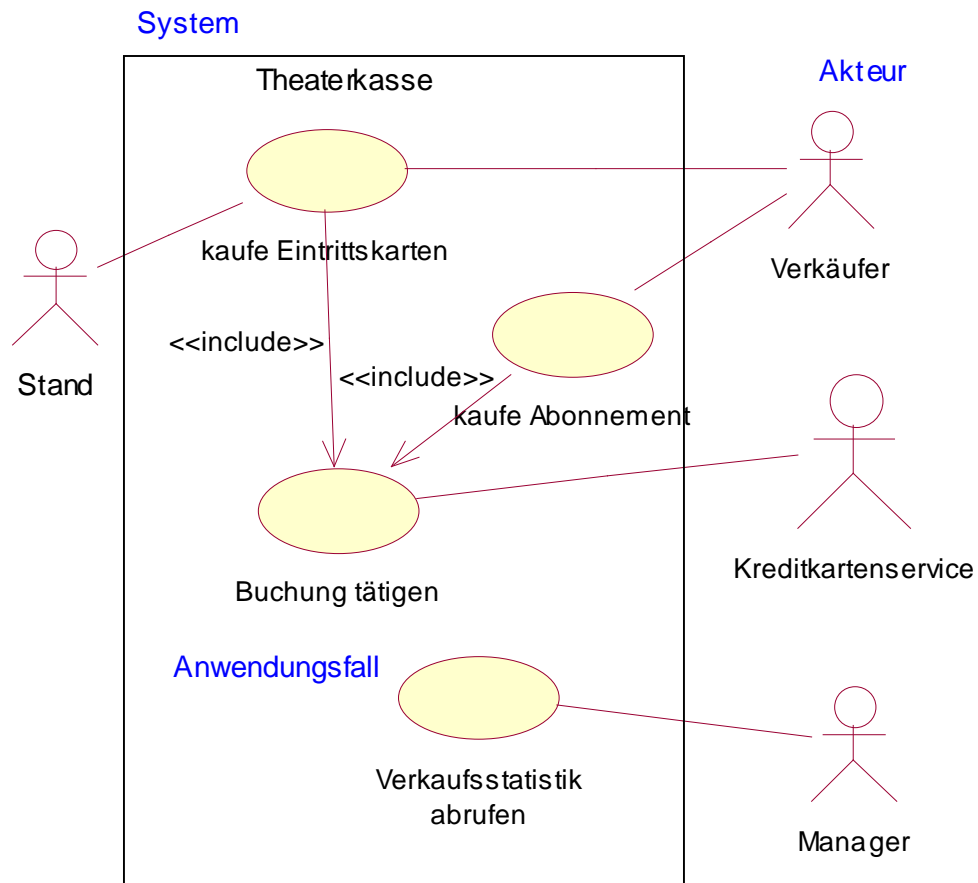


Verkäufer

- *Systemgrenzen* werden durch Rechtecke visualisiert

Anwendungsfalldiagramm

Theaterkasse



Interaktions-Sicht (Interaction View)

- Modellierung von Verhalten eines Systems durch Darstellung des Nachrichtenaustauschs zwischen verschiedenen Objekten
- Zwei verschiedene Diagramme für die Interaktions-Sicht:
 - *Sequenzdiagramm* (Sequence diagram)
 - *Kollaborationsdiagramm* (Collaboration diagram)

Sequenzdiagramm

- Zeigt den Nachrichtenaustausch zwischen verschiedenen Objekten unter Betonung der zeitlichen Abfolge

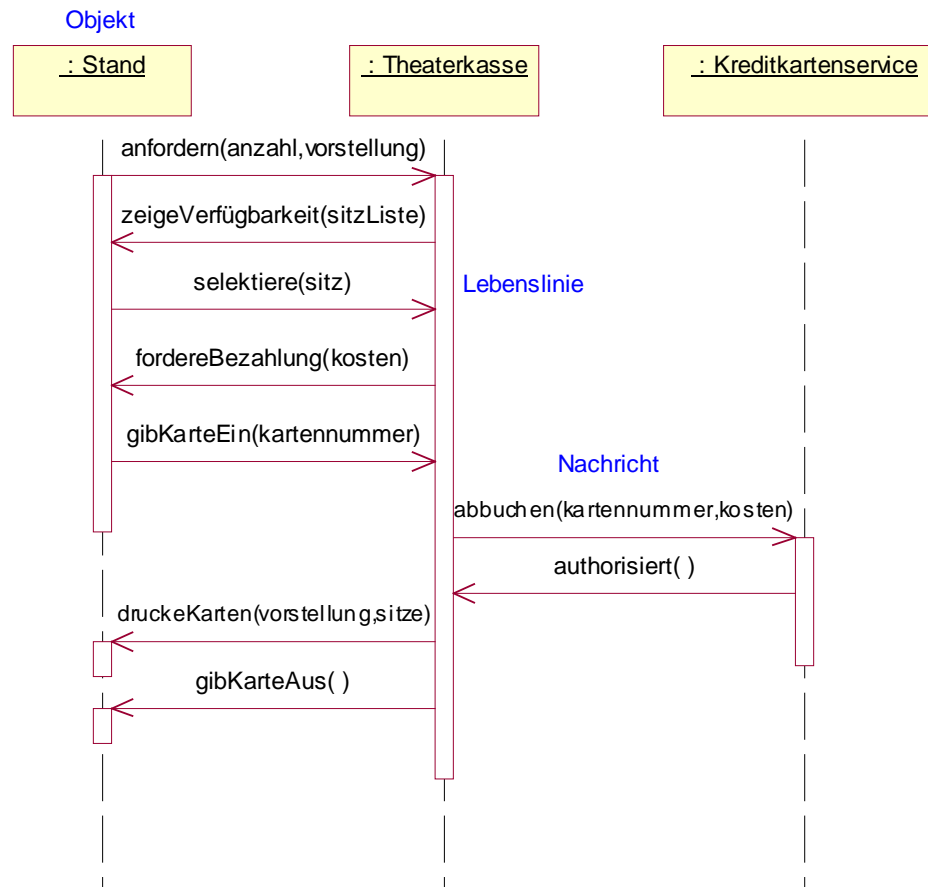
Sequenzdiagramm (Forts.)

Graphische Darstellung:

- beteiligte Objekte werden horizontal angeordnet
- jedes Objekt hat eine *Lebenslinie* (Lifeline), die vertikal aufgetragen wird
- *Nachrichten* (Messages) = Pfeile zwischen den Lebenslinien der beteiligten Objekte

Sequenzdiagramm

kaufe Eintrittskarten



Kollaborationsdiagramm

- Zeigt den Nachrichtenaustausch zwischen verschiedenen Objekten unter Betonung der Beziehung zwischen den Objekten

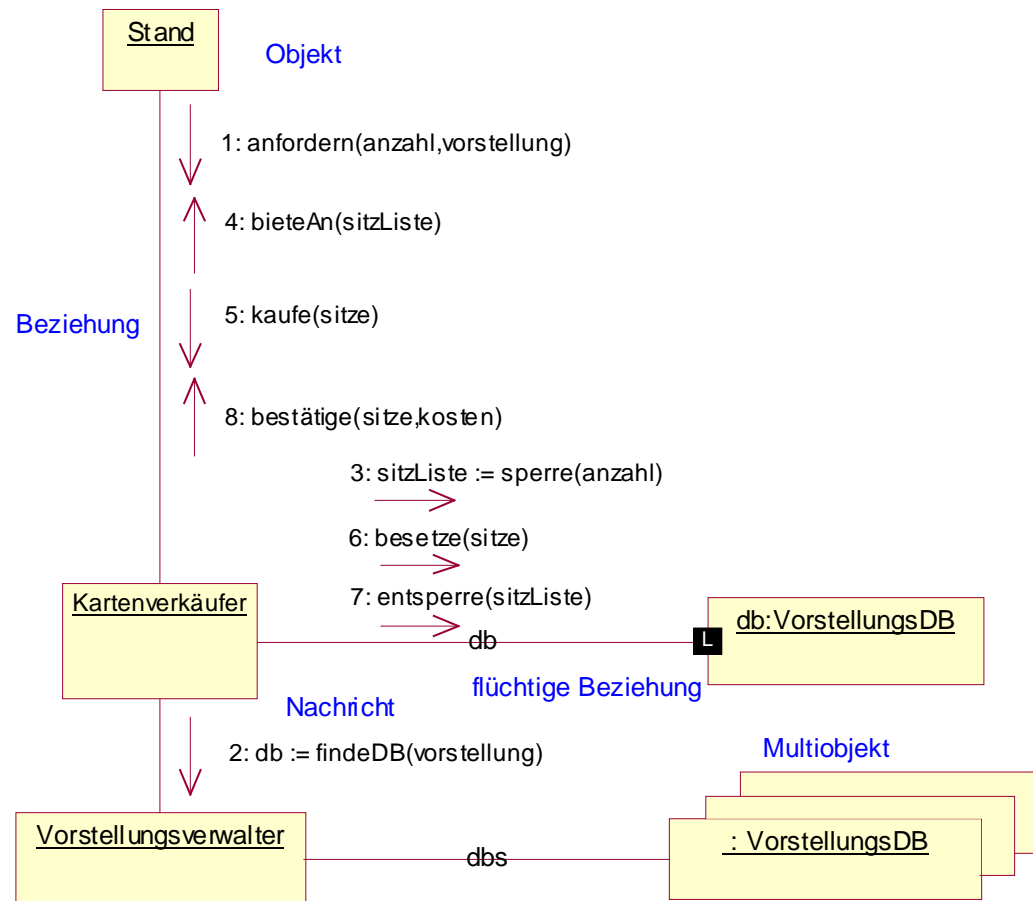
Kollaborationsdiagramm (Forts.)

Graphische Darstellung:

- Anordnung der beteiligten Objekte gemäß ihrer Beziehung
- Darstellung der Beziehungen durch Verbindungslinien
- *Nachrichten* = Pfeile entlang der Beziehungslinien
- zeitliche Abfolge der Nachrichten wird durch Numerierung festgelegt

Kollaborationsdiagramm

reserviere Eintrittskarten



Interaktions-Sicht (Forts.)

- Sowohl Sequenz- als auch Kollaborationsdiagramme zeigen Interaktion zwischen Objekten jedoch mit Betonung unterschiedlicher Aspekte
- Sequenzdiagramm
 - graph. Veranschaulichung der zeitlichen Abfolge
 - Beziehung der Objekte implizit
- Kollaborationsdiagramm
 - graph. Veranschaulichung der Beziehung
 - untergeordnete Darstellung der zeitlichen Abfolge

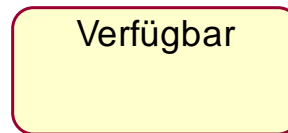
Zustandsmaschinen-Sicht (State Machine View)

- Modellierung der möglichen *Zustände* (States), die ein Objekt im Laufe seines Lebens einnehmen kann, sowie der Stimuli, die zu einem *Zustandsübergang* (State Transition) führen

Zustandsmaschinen-Sicht (Forts.)

Graphische Darstellung:

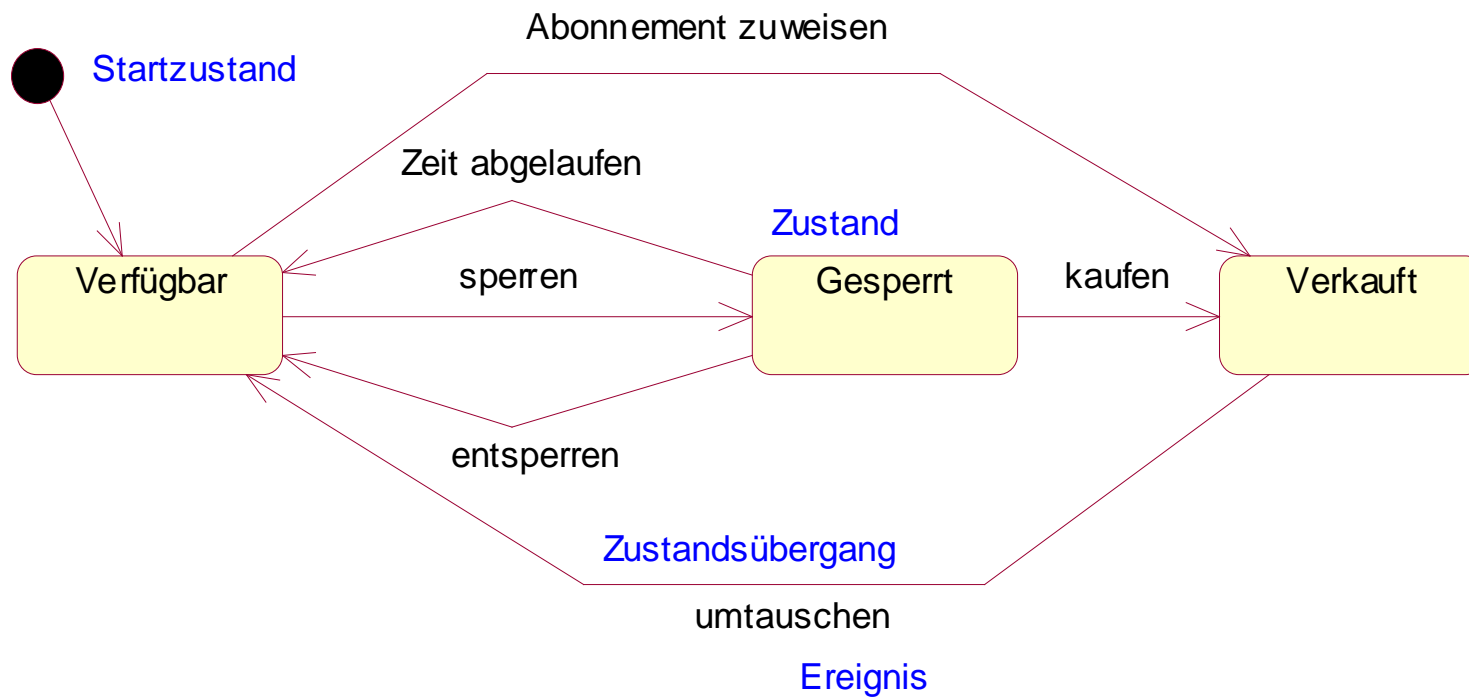
- *Zustände* = Rechtecke mit abgerundeten Ecken



- *Zustandsübergänge* = Pfeile zwischen betroffenen Zuständen
- *Startzustand* = schwarzer ausgefüllter Kreis ●
- *Endzustand* = in Kreis eingeschlossener schwarzer ausgefüllter Kreis ⊙

Zustandsübergangsdiagramm

Eintrittskarte



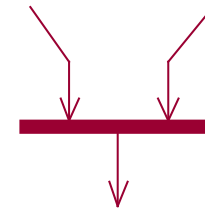
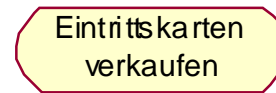
Aktivitäts-Sicht (Activity View)

- Modellierung der Ablaufmöglichkeiten eines Systems durch Angabe der einzelnen Aktivitäten
- Aktivitätsdiagramm ist spezielle Form eines Zustandsübergangsdiagrammes
- *Aktivität* (Activity) ist Zustand mit interner Aktion und einem oder mehreren ausgehenden Zustandsübergängen, die automatisch der Beendigung der internen Aktion folgen

Aktivitäts-Sicht (Forts.)

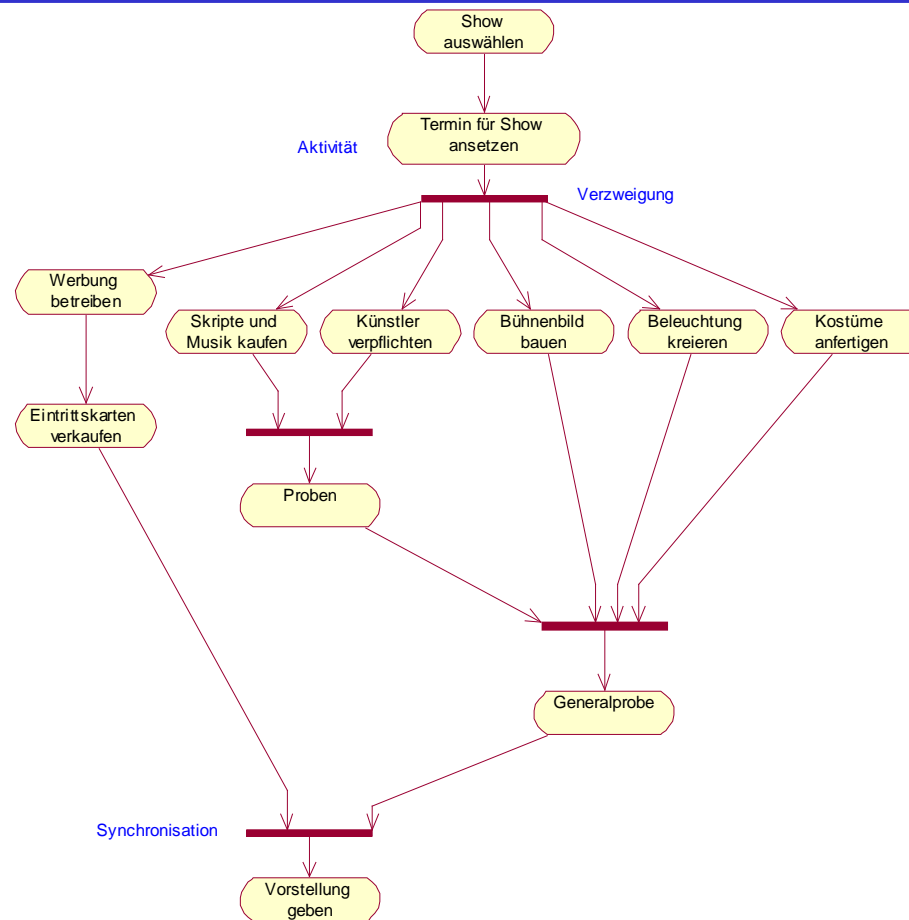
Graphische Darstellung:

- *Aktivitäten* = Rechtecke, deren linke und rechte Seite Kreisbögen sind
- *Übergänge* = Pfeile zwischen den betroffenen Aktivitäten
- *Verzweigung* (Fork), *Synchronisation* (Join) = dickere Balken, bei denen sich Pfeile verzweigen bzw. von denen Pfeile abgehen



Aktivitätsdiagramm

Planung/Ausführung einer Show



Physische Sichten (Physical Views)

- Bisherige Sichten modellierten *logische Struktur* der Applikation
- Physische Sichten modellieren
 - *Implementationsstruktur* der Applikation
(**Implementations-Sicht**)
 - *Verteilung der Komponenten* auf Rechnerknoten
(**Verteilungs-Sicht**)

Implementations-Sicht (Implementation View)

- Zeigt die Komponenten (sowie deren Schnittstellen) eines Systems und deren Abhängigkeiten untereinander
- *Komponente* (Component) = Software Einheit, aus der die Applikation zusammengebaut wird (z.B. Quellcode, Executable, Library, ...)
- *Schnittstelle* (Interface) = extern sichtbares Verhalten einer Komponente

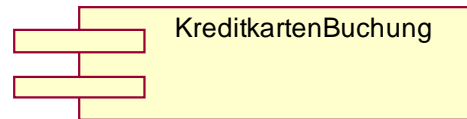
Implementations-Sicht (Forts.)

- Implementations-Sicht wird durch *Komponentendiagramme* (component diagrams) dargestellt

Implementations-Sicht (Forts.)

Graphische Darstellung:

- *Komponenten* = Rechtecke, die am linken Rand zwei kleine Rechtecke tragen



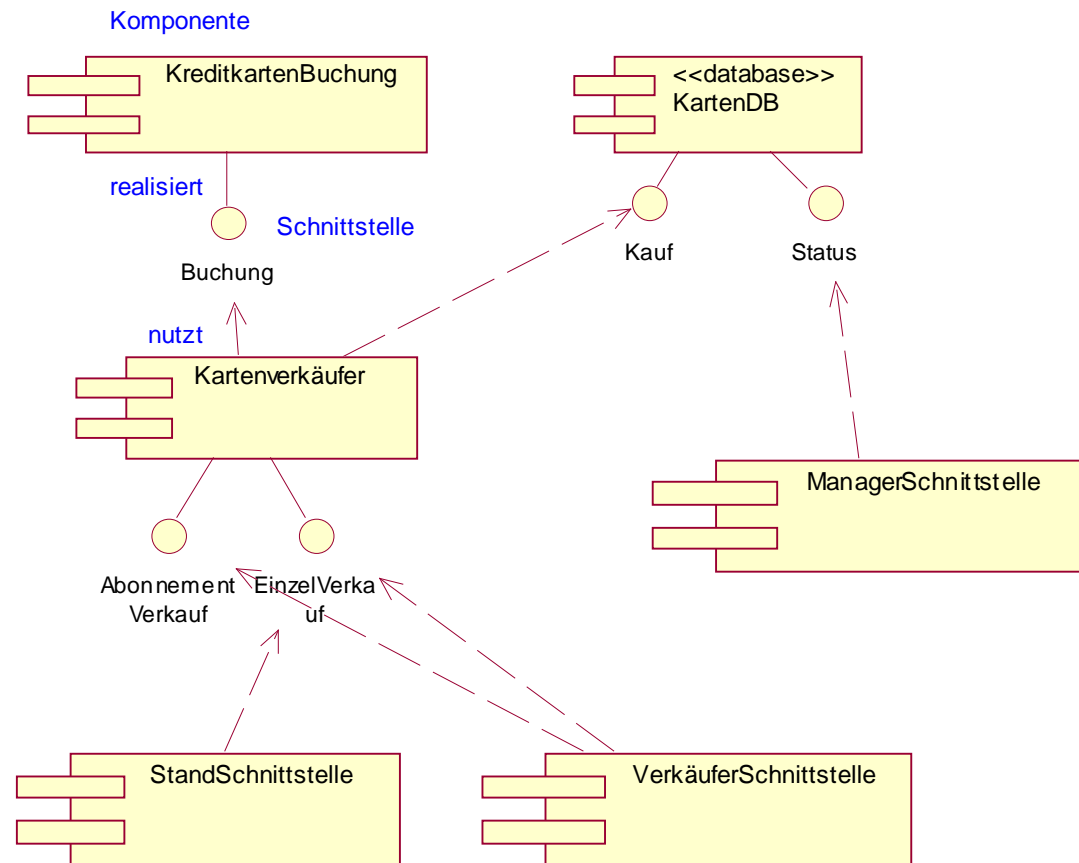
- *Schnittstelle* = Kreis
- Komponente *bietet* Schnittstelle *an* = Komponente und Schnittstelle durch *Assoziation* verbunden
- Komponente *nutzt* Schnittstelle = Komponente und Schnittstelle durch *Abhängigkeit* verbunden



Buchung

Komponentendiagramm

Theaterkasse



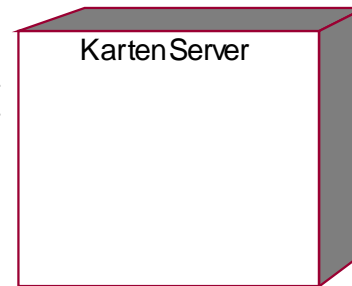
Verteilungs-Sicht (Deployment View)

- Zeigt Anordnung der Komponenten auf den zur Verfügung stehenden Knoten zur Laufzeit
- *Knoten* (Node) = zur Laufzeit physisch vorhandenes Objekt, das über Rechenleistung bzw. Speicher verfügt
(z.B. Computer, Eingabeterminal, RAID, ...)
- Verteilungs-Sicht wird durch *Verteilungs-diagramme* (Deployment diagrams) dargestellt

Verteilungs-Sicht (Forts.)

Graphische Darstellung:

– *Knoten* = Quader

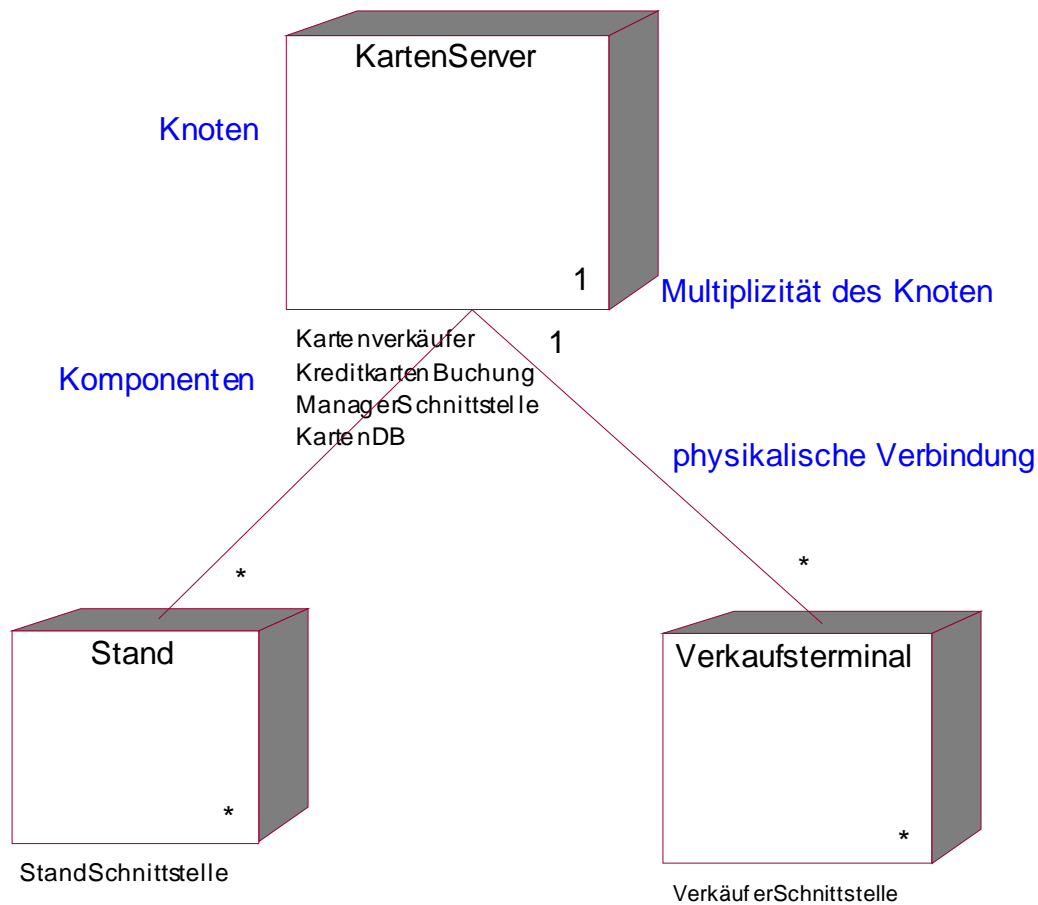


Kartenverkäufer
KreditkartenBuchung
ManagerSchnittstelle
KartenDB

- Physikalisch Verbindungen zwischen Knoten = Assoziationsverbindung
- Platzierung von Komponenten und deren Abhängigkeiten innerhalb von Knoten möglich

Verteilungsdiagramm

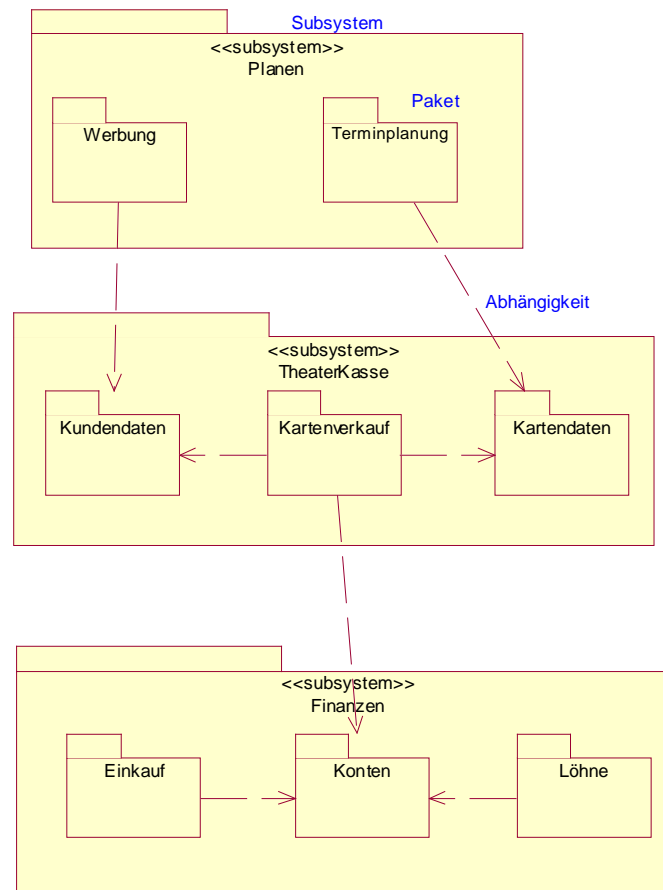
Theaterkasse



Modell Management Sicht

- Organisation und Strukturierung des Modells selbst in Pakete und Subsysteme
- *Modell* (Model) = komplette Beschreibung des Systems
- *Paket* (Package) enthält Modellierungselemente und u. U. weitere Pakete
- *Subsystem* = Teil des Systems, der isoliert betrachtet ein eigenständiges System darstellt

Organisation des Modells *Theater*

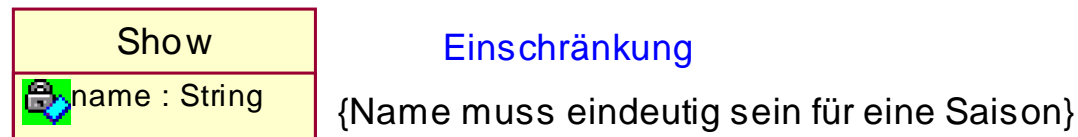


Erweiterbarkeitskonstrukte (Extensibility Constructs)

- UML enthält drei Klassen von Erweiterbarkeitskonstrukten
 - *Einschränkungen* (Constraints)
 - *Stereotypen* (Stereotype)
 - *Eigenschaftswerte* (Tagged values)

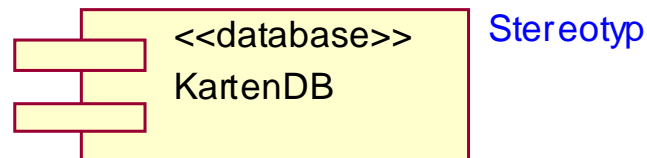
Einschränkung (Constraint)

- Ausdruck, der die möglichen Inhalte, Zustände oder die Semantik eines Modellelements einschränkt und stets erfüllt sein muss



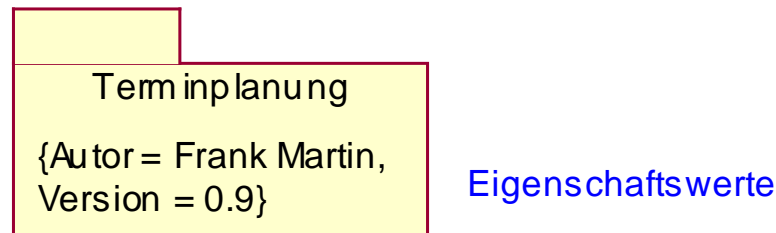
Stereotyp

- Ein *neuartiges Modellierungselement*, welches
- basierend auf einem vorhandenen - vom
Modellierer entworfen werden kann



Eigenschaftswert (Tagged value)

- Eine mit Attributen versehene Information, die an ein beliebiges Modellierungselement angehängt werden kann



Notizen (Notes)

- Konstrukte, die zur Darstellung eines Kommentars oder anderer textueller Information dienen, z.B.
 - Source-Code Fragmente
 - Einschränkungen

Notizen (Forts.)

Graphische Darstellung:

- *Notiz* = Rechteck mit Eselsohr, welches Text oder Verweis auf Dokument enthält
- Verbindung zum zugehörigen Element = gestrichelte Linie

