



wirtschaftsinformatik
managementinformationssysteme

systems landscape engineering - übung -

Wintersemester 2010 /2011
Arbeitsgruppe Wirtschaftsinformatik
- Managementinformationssysteme -
Dipl. Wirt.-Inform. Sven Gerber



ablauf

- Modellierung
- Einführung Grundsätze ordnungsgemäßer Modellierung
- UML
- Fachliche Spezifikation



modell (I)

- Muster, Form, Entwurf oder Person als Gegenstand bildender Kunst



- Repräsentation eines Originals im Sinne
 1. eines Abbild von etwas oder Vorbild für etwas
 2. eines Individuums als Beispiel für eine Menge von originalen Individuen für die künstlerische Gestaltung

Stachowiak, 1973

Quelle: Herden, S.: Skript System Landscape Engineering (2009)

modell (II)

- Definition:
 - Ein Modell ist ein abstraktes, immaterielles Abbild realer Strukturen bzw. des realen Verhaltens für Zwecke des Subjekts.
 - Ein Modell ist damit ein
 - adäquates
 - vereinfachendes
 - idealisierendes Abbild der Realität.



grundsätze ordnungsgemäßer modellierung

Notwendige Grundsätze

Grundsatz der Richtigkeit

Grundsatz der Relevanz

Grundsatz der
Wirtschaftlichkeit

Ergänzende Grundsätze

Grundsatz der Klarheit

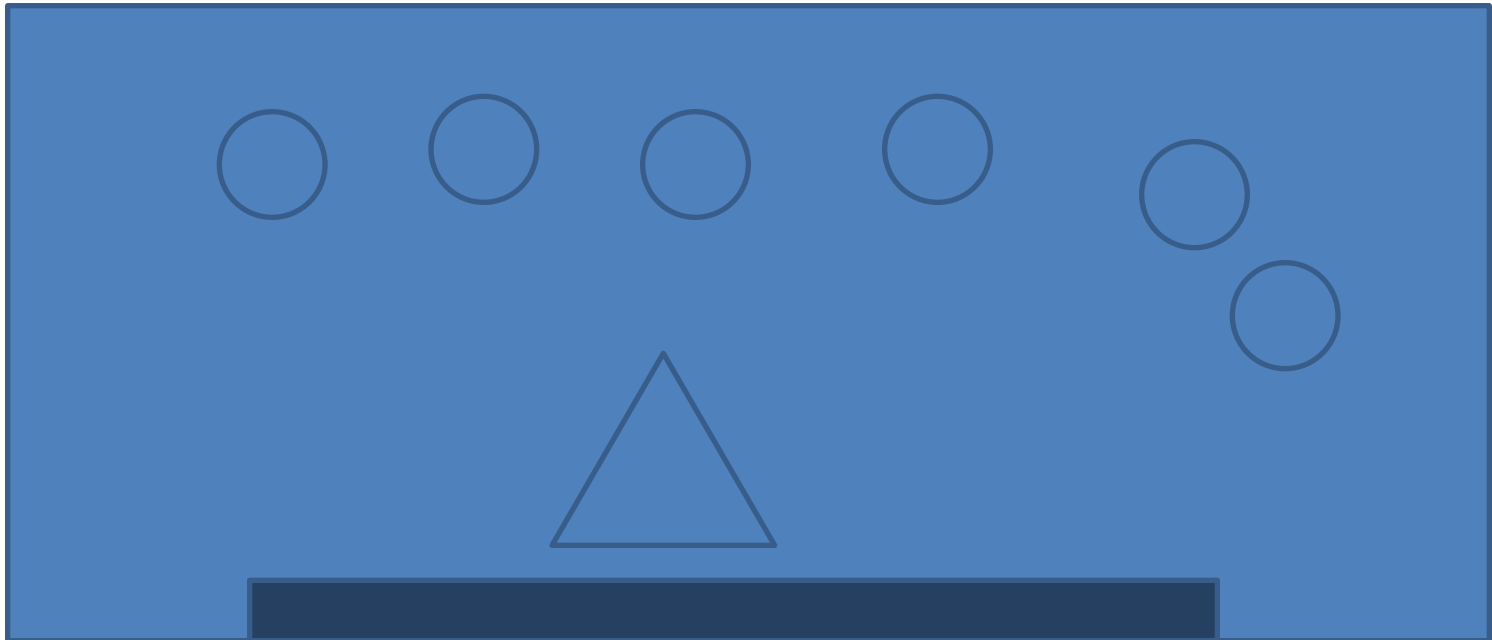
Grundsatz der
Vergleichbarkeit

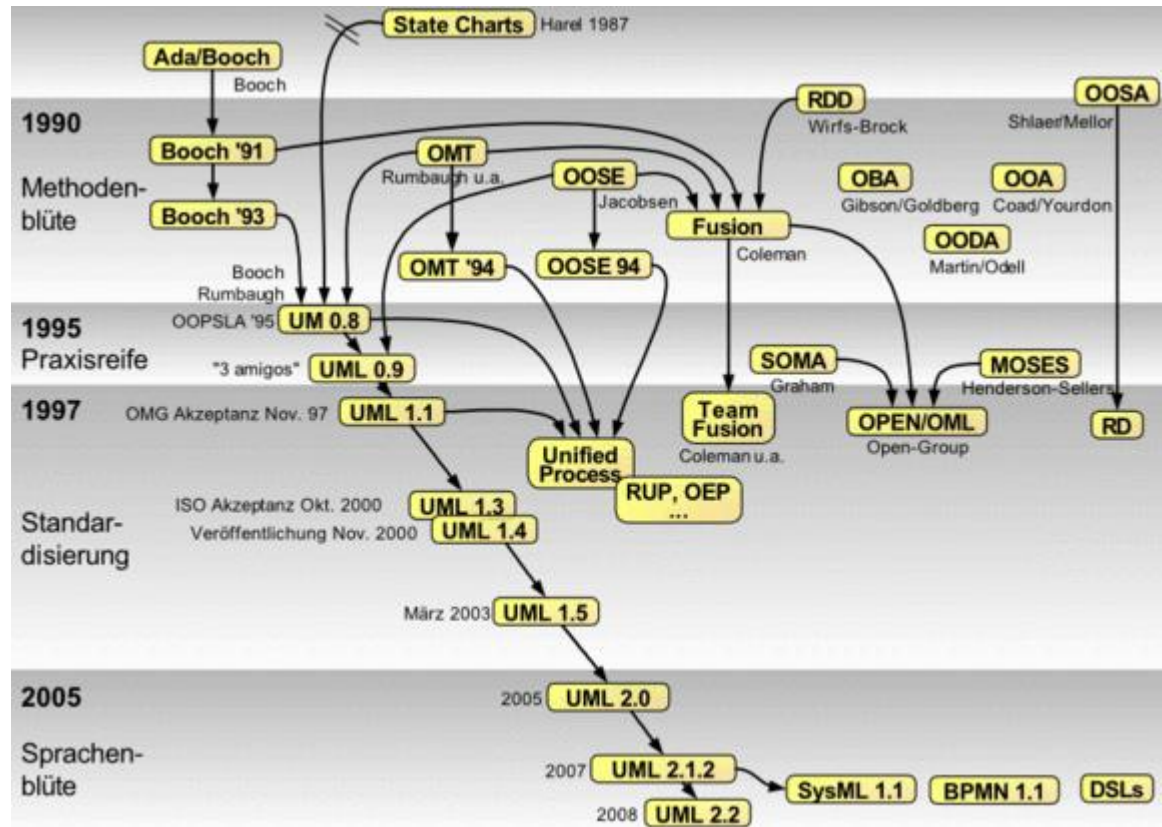
Grundsatz des
systematischen Aufbaus

Quelle: in Anlehnung: Becker, Mathias, Winkelmann: Geschäftsprozessmanagement



beispiel





Quelle: OOSE - <http://www.oose.de/swe/uml/historie.html>

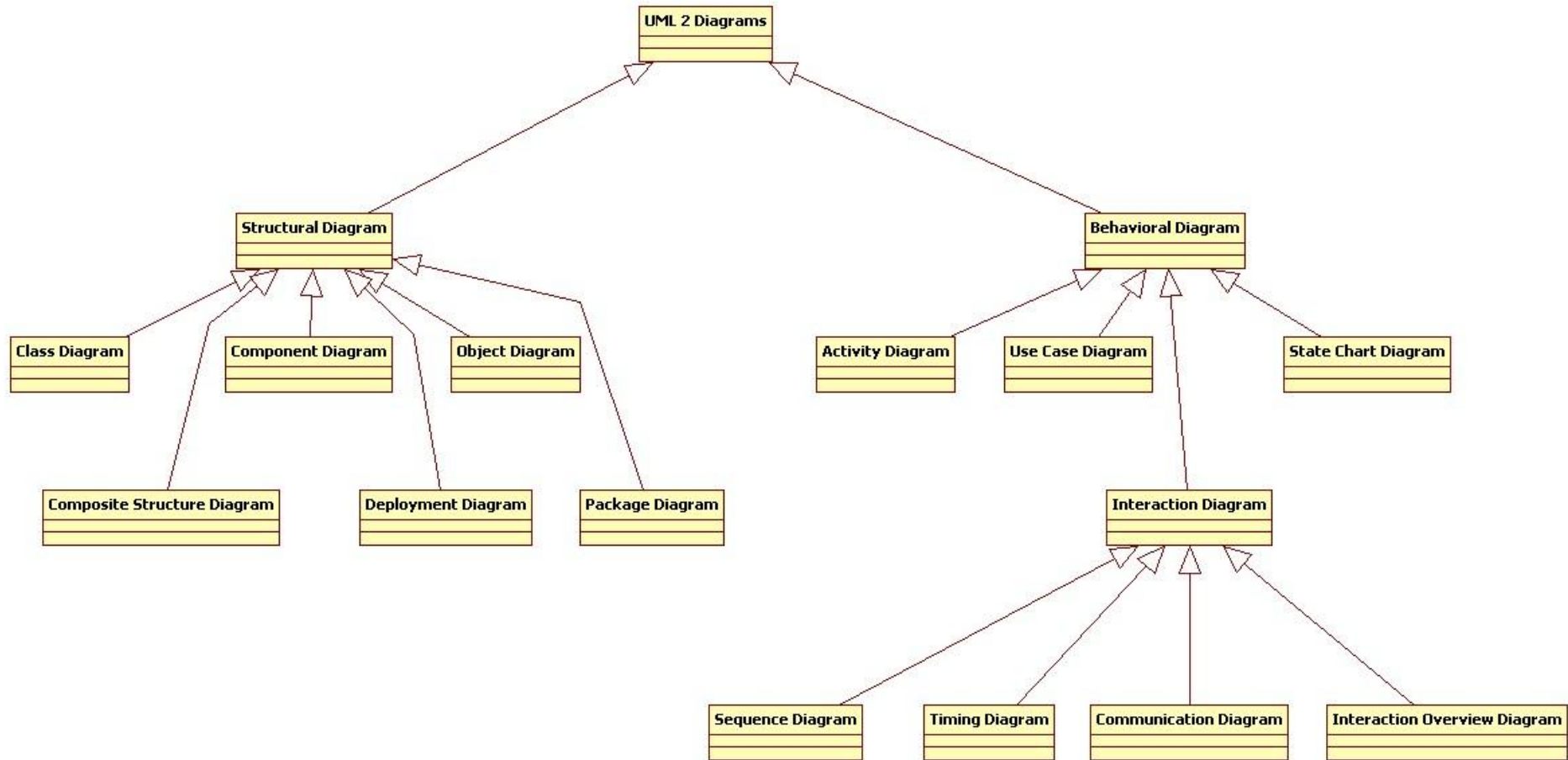


uml - begriffserklärung

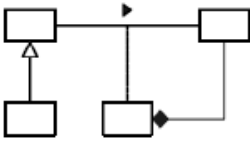
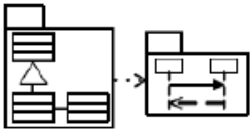
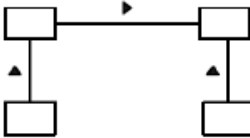
- Allgemeine, standardisierte Notationssprache zur Spezifikation, Dokumentation von Modellen für komplexe Softwaresysteme
- Konsistente Darstellung einer Vielzahl von Elementen von Softwaresystemen mittels einer einheitlichen Notation in einem Modell unter Nutzung vieler unterschiedlicher Diagrammtypen
- Unterscheidung in Struktur- und Verhaltensdiagramme



uml - übersicht



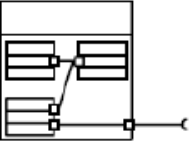
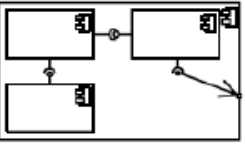
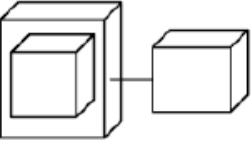
uml – diagrammtypen (I)

Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
<p>Klassendiagramm</p> 	<p>Aus welchen Klassen besteht mein System und wie stehen diese untereinander in Beziehung?</p>	<p>Beschreibt die statische Struktur des Systems. Enthält alle relevanten Strukturzusammenhänge/Datentypen. Brücke zu dynamischen Diagrammen. Normalerweise unverzichtbar.</p>
<p>Paketdiagramm</p> 	<p>Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?</p>	<p>Logische Zusammenfassung von Modellelementen. Modellierung von Abhängigkeiten/Inklusion möglich.</p>
<p>Objektdiagramm</p> 	<p>Welche innere Struktur besitzt mein System zu einem bestimmten Zeitpunkt zur Laufzeit (Klassendiagrammschnappschuss)?</p>	<p>Zeigt Objekte u. Attributbelegungen zu einem bestimmten Zeitpunkt. Verwendung beispielhaft zur Veranschaulichung Detailniveau wie im Klassendiagramm. Sehr gute Darstellung von Mengenverhältnissen.</p>

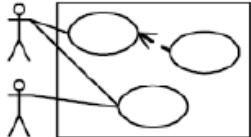
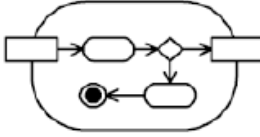
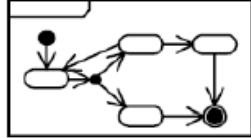
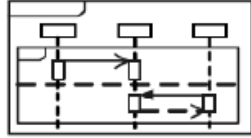
Quelle: Rupp, C. (2003): UML 2 -- Ballast oder Befreiung?
- <http://www.jeckle.de/uml-glasklar/AgilityDays2003.pdf>



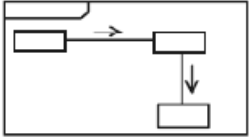
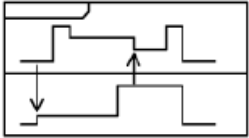
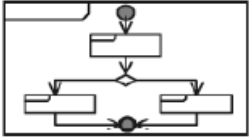
uml – diagrammtypen (II)

Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
<p>Kompositionsstrukturdiagramm</p> 	<p>Wie sieht das Innenleben einer Klasse, einer Komponente, eines Systemteils aus?</p>	<p>Ideal für die Top-Down-Modellierung des Systems (Ganz-Teil-Hierarchien). Zeigt Teile eines „Gesamtelements“ und deren Mengenverhältnisse. Präzise Modellierung der Teile-Beziehungen über spezielle Schnittstellen (Ports) möglich.</p>
<p>Komponentendiagramm</p> 	<p>Wie werden meine Klassen zu wieder verwendbaren, verwaltbaren Komponenten zusammengefasst und wie stehen diese in Beziehung?</p>	<p>Zeigt Organisation und Abhängigkeiten einzelner technischer Systemkomponenten. Modellierung angebotener und benötigter Schnittstellen möglich.</p>
<p>Verteilungsdiagramm</p> 	<p>Wie sieht das Einsatzumfeld (Hardware, Server, Datenbanken, ...) des Systems aus? Wie werden die Komponenten zur Laufzeit wohin verteilt?</p>	<p>Zeigt das Laufzeitumfeld des Systems mit den „greifbaren“ Systemteilen. Darstellung von „Softwareservern“ möglich. Hohes Abstraktionsniveau, kaum Notationselemente.</p>

uml – diagrammtypen (III)

Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Use-Case-Diagramm 	Was leistet mein System für seine Umwelt (Nachbarsysteme, Stakeholder)?	Außensicht auf das System. Geeignet zur Kontextabgrenzung. Hohes Abstraktionsniveau, einfache Notationsmittel.
Aktivitätsdiagramm 	Wie läuft ein bestimmter fluss-orientierter Prozess oder ein Algorithmus ab?	Sehr detaillierte Visualisierung von Abläufen mit Bedingungen, Schleifen, Verzweigungen. Parallelisierung und Synchronisation. Darstellung von Datenflüssen.
Zustandsautomat 	Welche Zustände kann ein Objekt, eine Schnittstelle, ein Use Case, ... bei welchen Ereignissen annehmen?	Präzise Abbildung eines Zustandsmodells mit Zuständen, Ereignissen, Nebenläufigkeiten, Bedingungen, Ein- und Austrittsaktionen. Schachtelung möglich.
	Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus?	Darstellung d. Informationsaustauschs zwischen Kommunikationspartnern Sehr präzise Darstellung der zeitlichen Abfolge auch mit Nebenläufigkeiten.

uml – diagrammtypen (IV)

Diagrammtyp	Diese zentrale Frage beantwortet das Diagramm	Stärken
Kommunikationsdiagramm 	Wer kommuniziert mit wem? Wer „arbeitet“ im System zusammen?	Stellt den Informationsaustausch zwischen Kommunikationspartnern dar. Überblick steht im Vordergrund (Details und zeitliche Abfolge weniger wichtig).
Timingdiagramm 	Wann befinden sich verschiedene Interaktionspartner in welchem Zustand?	Visualisiert das exakte zeitliche Verhalten von Klassen, Schnittstellen,.. Geeignet für die Detailbetrachtungen, bei denen es wichtig ist, dass ein Ereignis zum richtigen Zeitpunkt eintritt.
Interaktionsübersichtsdiagramm 	Wann läuft welche Interaktion ab?	Verbindet Interaktionsdiagramme (Sequenz-, Kommunikation- und Timingdiagramme) auf Top-Level-Ebene. Hohes Abstraktionsniveau.

verhaltensdiagramme

- **Use-Case-Diagramm**
 - Funktionale Sicht auf das System
- **Aktivitätsdiagramm**
 - Visualisierung von möglichen Abläufen mit unterschiedlichem Detailierungsgrad
- **State-Chart-Diagram (Zustandsdiagramm)**
 - Verhalten durch Zustände und Transitionen
- **Interaktionsdiagramme**
 - Nachrichten- und Datenaustausch zwischen kommunizierenden Partnern
 - Sequence Diagram
 - Communication Diagram
 - Timing Diagram
 - Interaction Overview Diagram

Quelle: SLE-Übung 2008



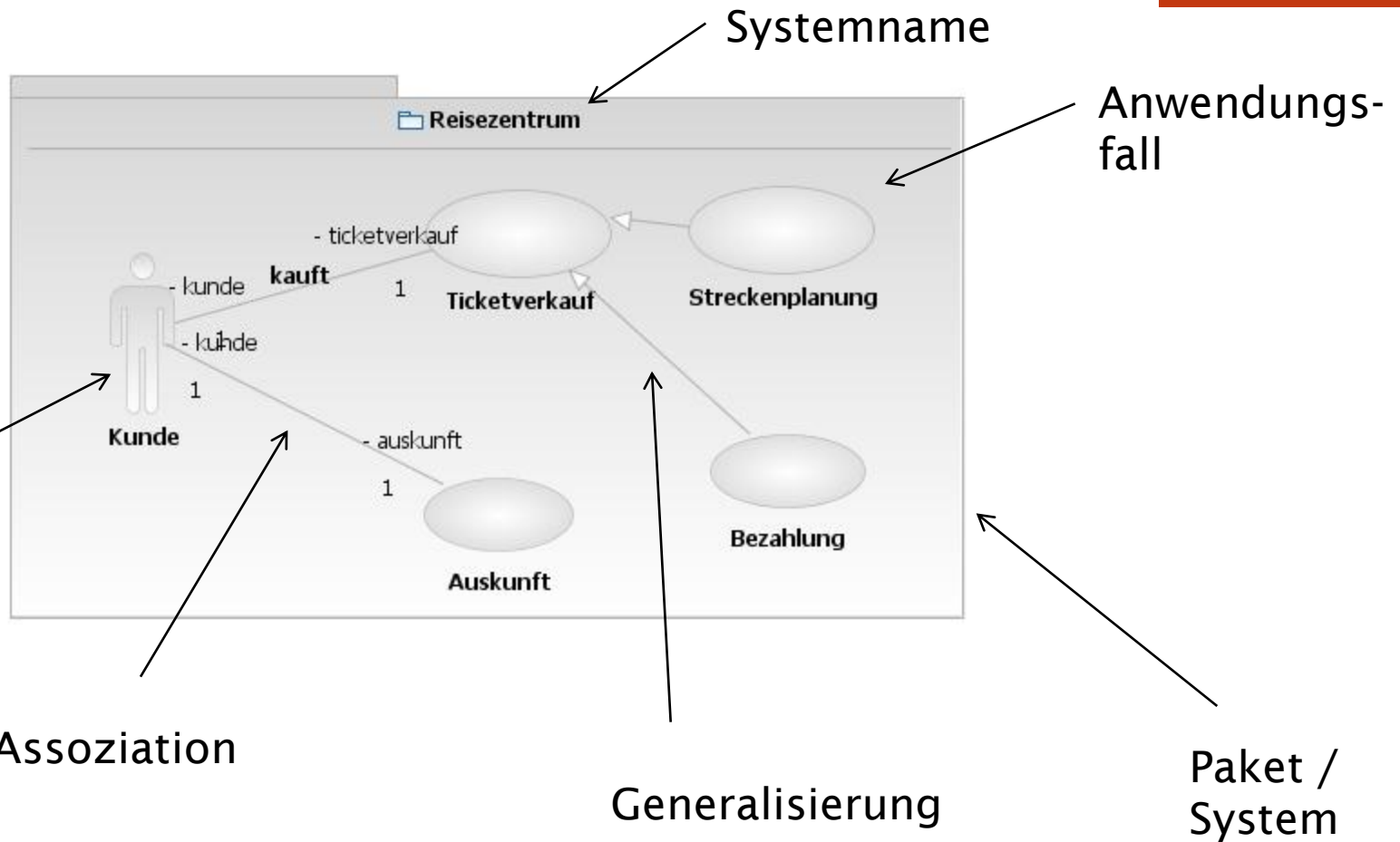
use-case-diagramm (I)

- Zweck
 - Funktionale Sicht auf das System
 - Modellierung der Benutzerinteraktionen mit dem System
 - Szenarioentwicklung
- Ziele innerhalb eines Projektes
 - Erste konzeptionelle Beschreibung
 - Festlegen von Systemgrenzen
 - Grundlage für die Kommunikation mit den Projektbeteiligten (Stakeholders)
 - Visualisierung von Anforderungen
 - Zerlegung von Arbeitspaketen
 - Work breakdown structure (WBS)

Quelle: SLE-Übung 2008



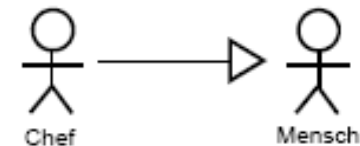
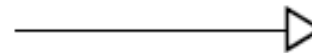
use-case-diagramm (II)



use-case-diagramm – include & extend

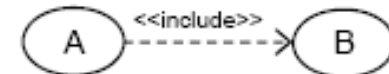
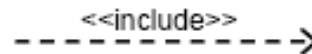
Generalisierung:

- Akteure und Anwendungsfälle
- Kommunikationsbeziehungen zwischen Akteuren und Anwendungsfällen werden weiter vererbt



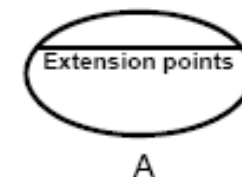
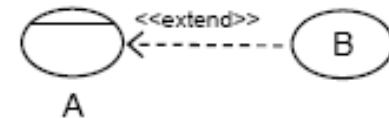
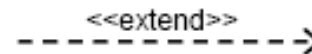
Include:

- Anwendungsfall A importiert das Verhalten des Anwendungsfalls B
- Verbot zyklischer Abhängigkeiten



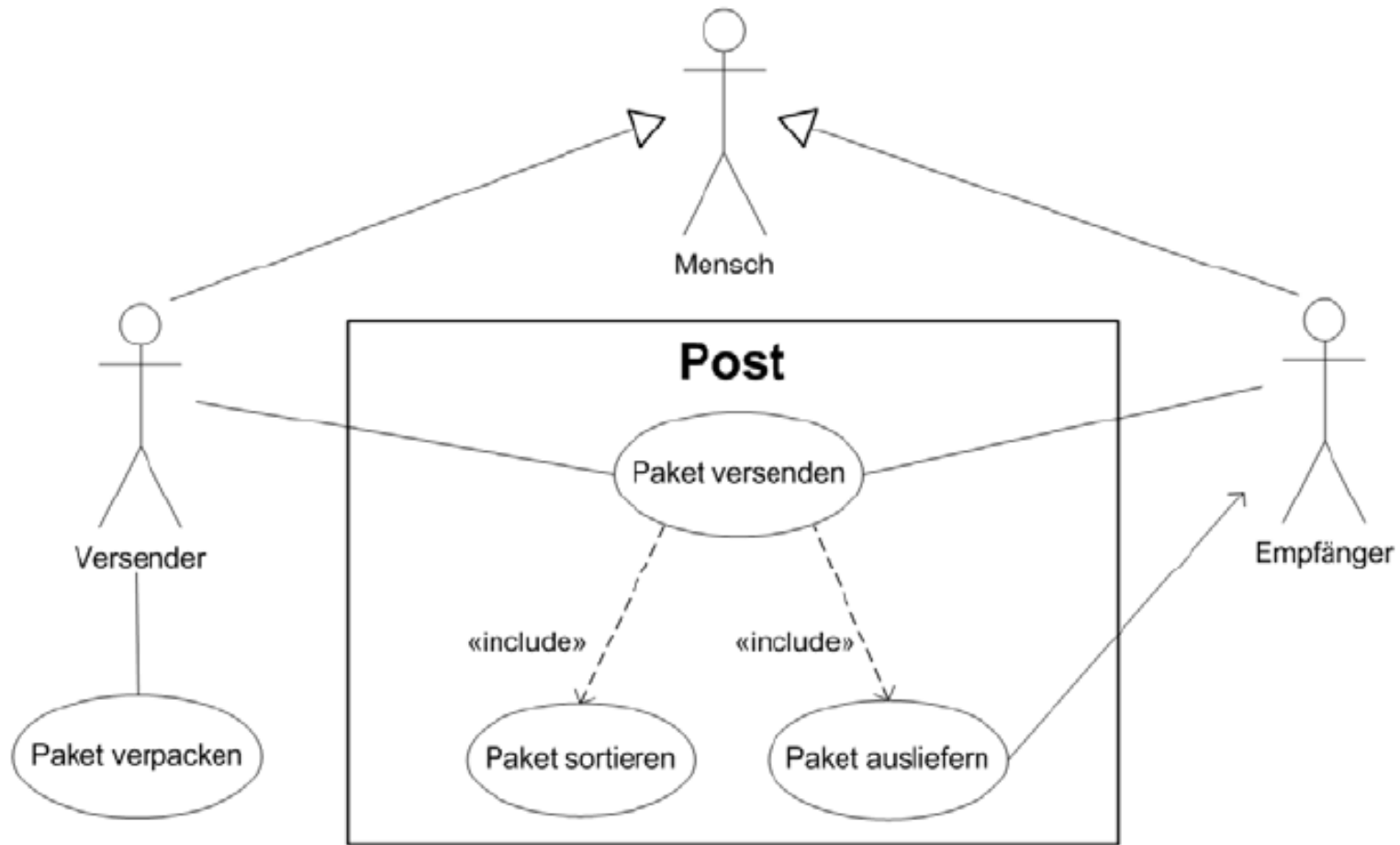
Extend:

- Das Verhalten des Anwendungsfalls A kann (muss nicht) durch einen Anwendungsfall B erweitert werden



Quelle: Modellierung Übung 2008

beispiel



Quelle: Modellierung Übung 2008