Spezifikation von Anwendungskernen (mit UML)

Vorlesung: Software-Engineering für große Informationssysteme

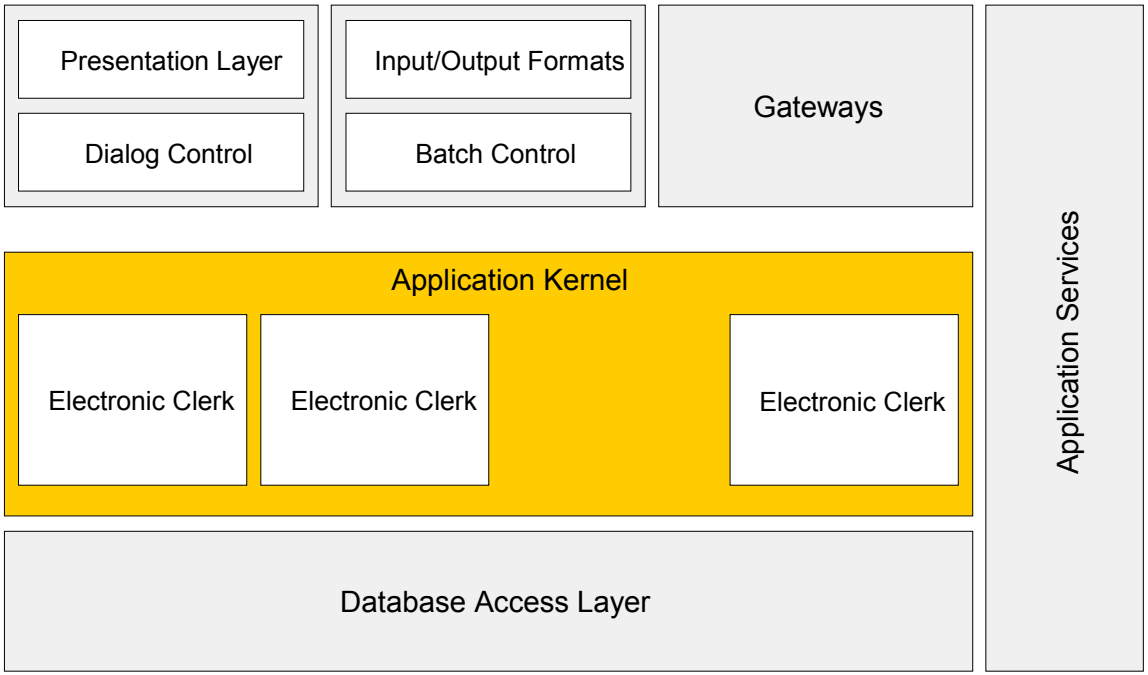
TU-Wien, Sommersemester 2003

Bernhard Anzelettl

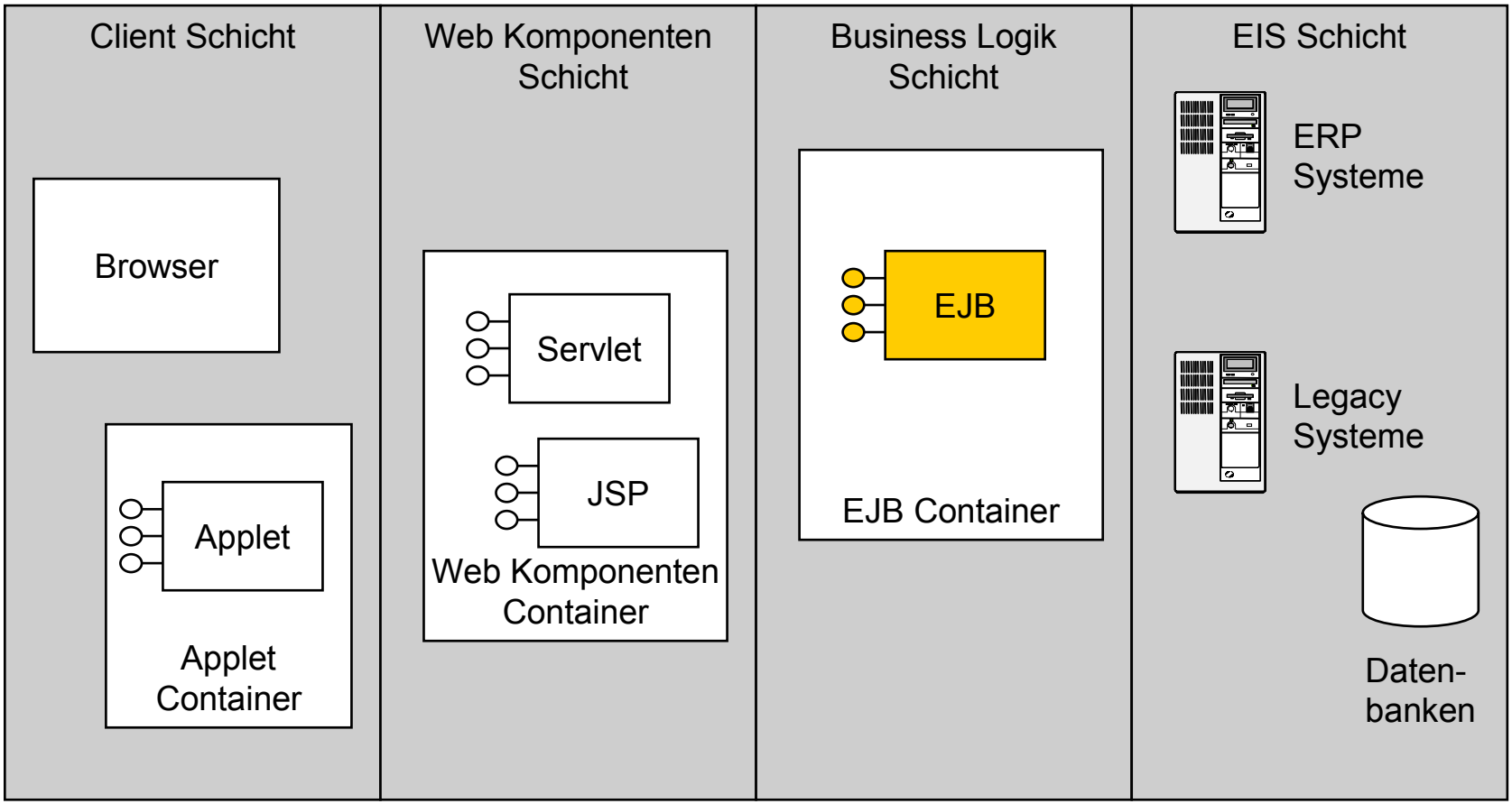
Überblick

- Anwendungskern – wo finde ich den?
- Spezifikation versus „Runterprogrammieren“ versus UML- Bilder
- Dennoch UML!

Anwendungskern
Wo finde ich den?



Anwendungskern
Wo finde ich den?



Anwendungskern Verantwortlichkeiten

- Ein Anwendungskern implementiert fachliche Funktionalität
- Seine Abstraktionen sind „fachliche Objekte “ und er hat dementsprechend „fachliche Aufträge“.
- Der Anwendungskern muß auch ohne GUI oder Oberfläche funktionieren

Grundlage für Implementierung von Software

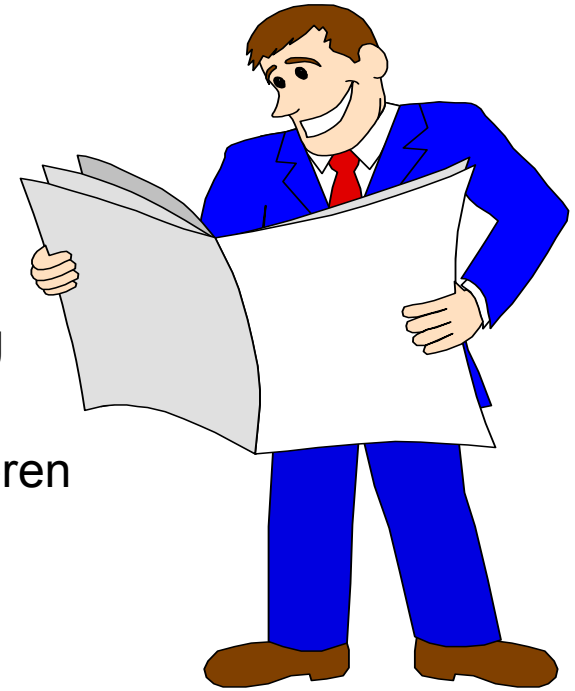
- Das fachliche **WAS** muß präzise spezifiziert sein
- **WIE** etwas implementiert wird, muß nicht spezifiziert werden.
- Präzision muß so gut sein, daß **zwei** Software-anbieter aus **einer** Spezifikation Software bauen können, die sich identisch verhält und austauschbar ist.

Nicht ausreichend sind also....

- Nennung von Funktionen
  - Es muß genau beschrieben werden, was die Funktionen tun
  - Die Ein-/ und Ausgabeparameter müssen genau beschrieben werden
- Nennung von Entitäten und Attributen
  - Beziehungen müssen beschrieben werden
  - Attribute müssen typgenau beschrieben sein
  - fachliche Beschreibungen müssen vorhanden sein
- Aufzählung von Anforderungen
  - sind wichtig für den späteren Abgleich mit der spezifizierten Funktionalität

Ihre Leser sind ...

- Zuallerst **Sie** selbst
  Sie strukturieren damit Ihre Arbeit
- **Entwickler** aus anderen Teilteams
  - die werden Sie reviewen und sich informieren
- **und auch** Mitarbeiter aus der Fachabteilung
  - wir erinnern uns – dass Fachmitarbeiter Datenmodelle lesen und Datentypen definieren ist Wunschdenken



Vor der Spezifikation des Anwendungskerns sollten Sie ...

- Anforderungen gelistet haben
- Geschäftsprozesse und deren Teilprozesse genannt und beschrieben haben
- Geschäftsvorfälle genannt haben
- Wesentlichen Systemfunktionen katalogisiert haben

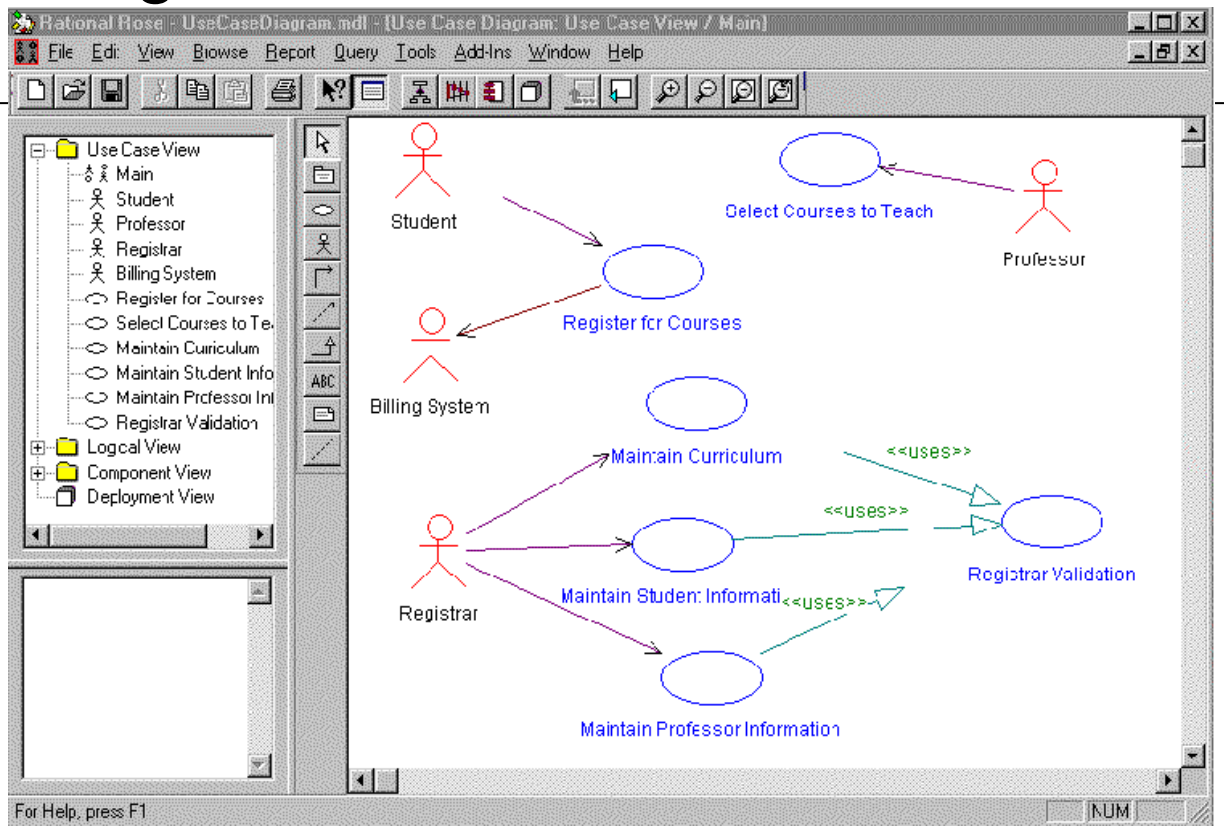
Und wie bitte beschreibe ich einen Anweungskern?

wait and see ☺

Spezifikation versus „Runterprogrammieren“ versus UML-Bilder

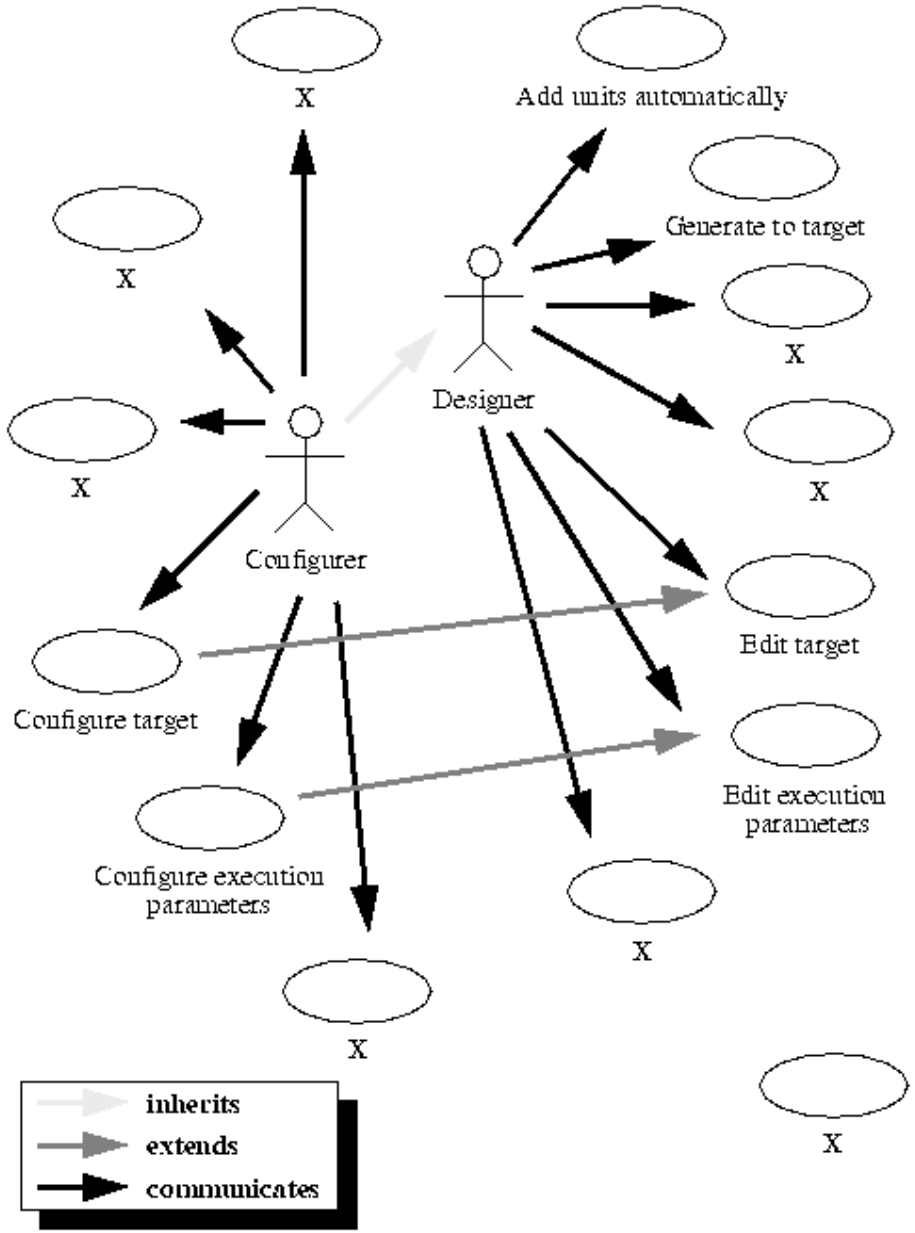
Bilder ...

- ein Bild sagt mehr als 1000 Worte
- aber was sagen 1000 Bilder dieser Art ...



Bilder ...

- noch ein schönes Beispiel ...

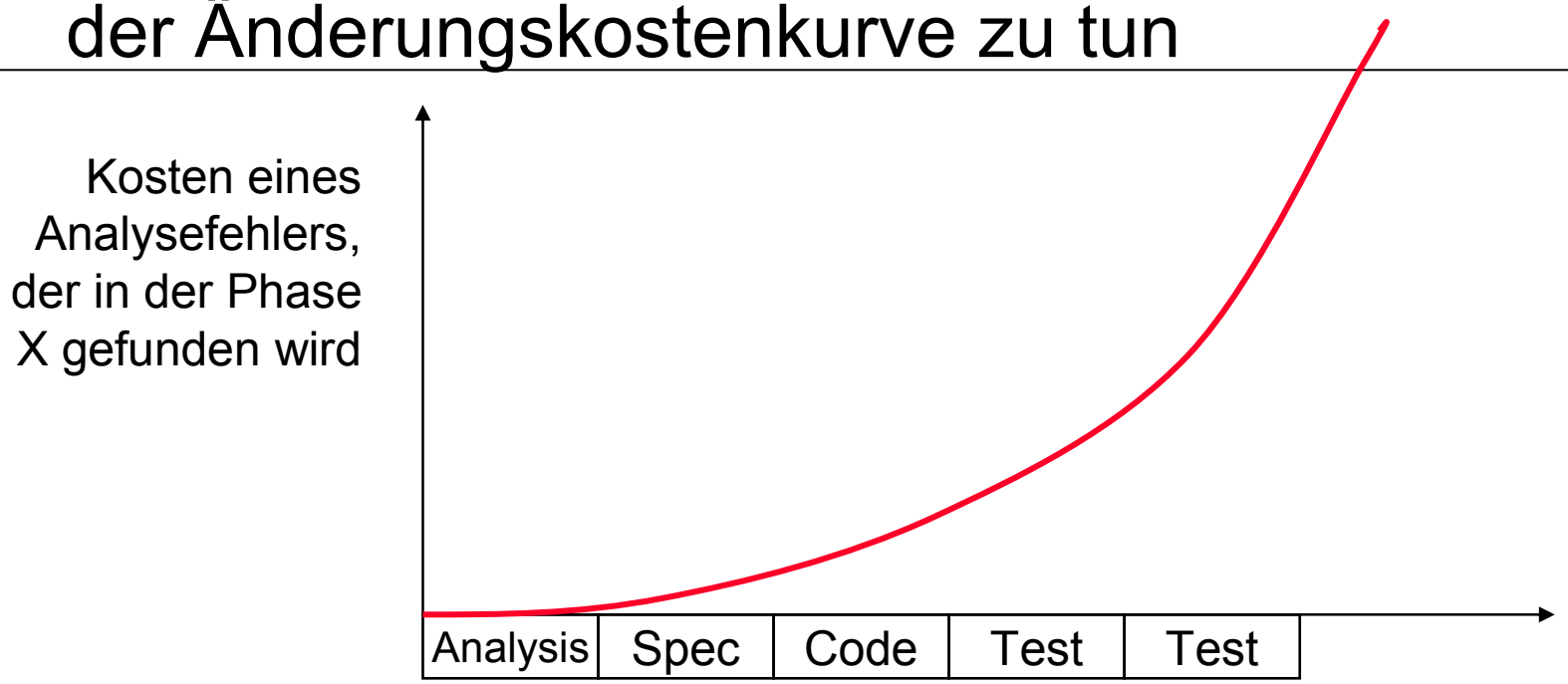


einfach „Runterprogrammieren“?

- das geht solange man sich im Umfeld von XP bewegt
- Wir erinnern uns an die Voraussetzungen:
  - entsprechende Entwicklungsumgebung
  - Debuggen, Änderungen und Refactoring werden einfach unterstützt
  - Kurze Compile oder Change Zyklen
  - in großen Team nicht erprobt

einfach „Runterprogrammieren“?

- wenn Sie die Voraussetzungen von XP oder agilen Prozessen nicht erfüllen, haben Sie es mit der Änderungskostenkurve zu tun



Folgerung

- einfach Runterprogrammieren: Für große Systeme wegen der Arbeitsteiligkeit nicht geeignet
  - und für „ältere Umgebungen“ mit langsamen Zyklen schon garnicht
- 1000 Bilder
  - es kann passieren, dass man dabei den Überblick verliert
- Also: Auf eine gesunde Mischung kommt es an aus
  - Bilder, Texten, Generierung ....

Dennoch - UML!

Was ist UML – kurze Historie

- Bei der Entstehung von OO waren Analyse- und Designmethoden gesucht

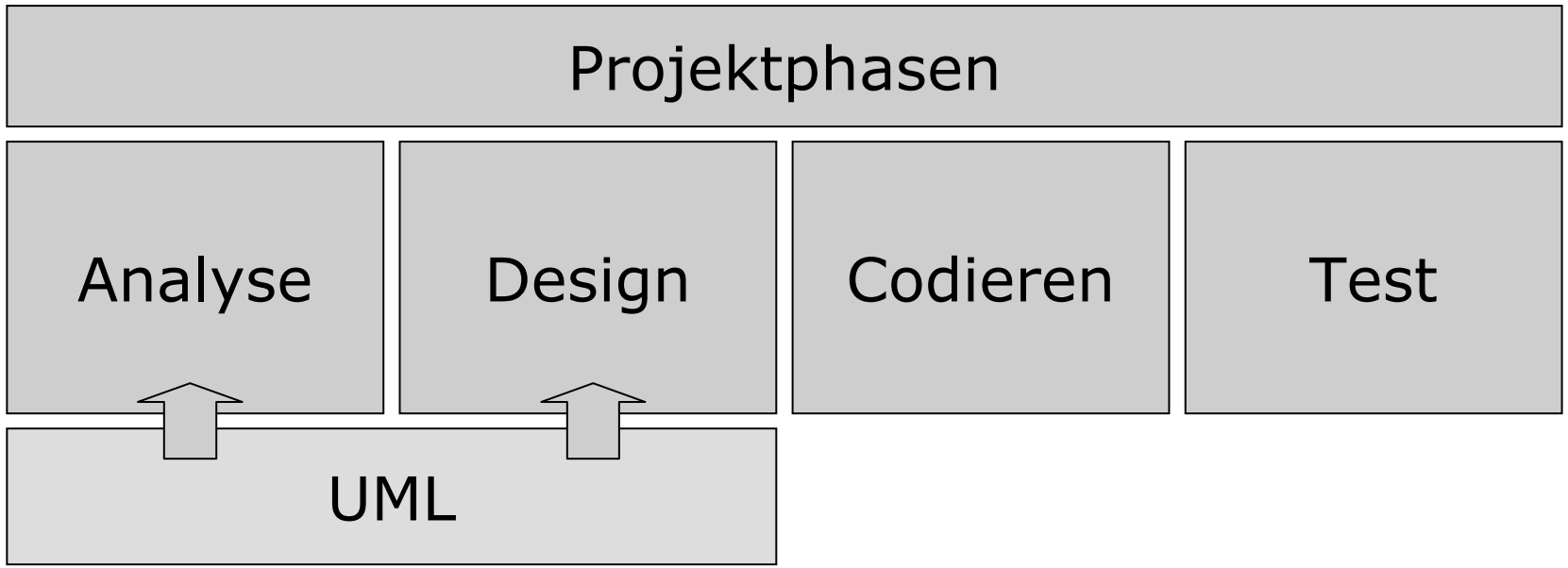
- Methoden tauchten auf....
  - Jacobson – Use Cases
  - Rumbaugh – OMT
  - Booch – OOA,OOD
- Leider: Jeder hat alles anders aufgeschrieben...

- UML = Vereinheitlichte Notation
  - Objektorientierte Analyse und Design
  - Gründung „Los tres amigos" bei Rational
  - Schaffung des Rational Unified Process
  - Hauptsächlich Grafiken
  - „Prä-Programmierung"

- XMI = UML auf XML (Grafiken in XML-Code)

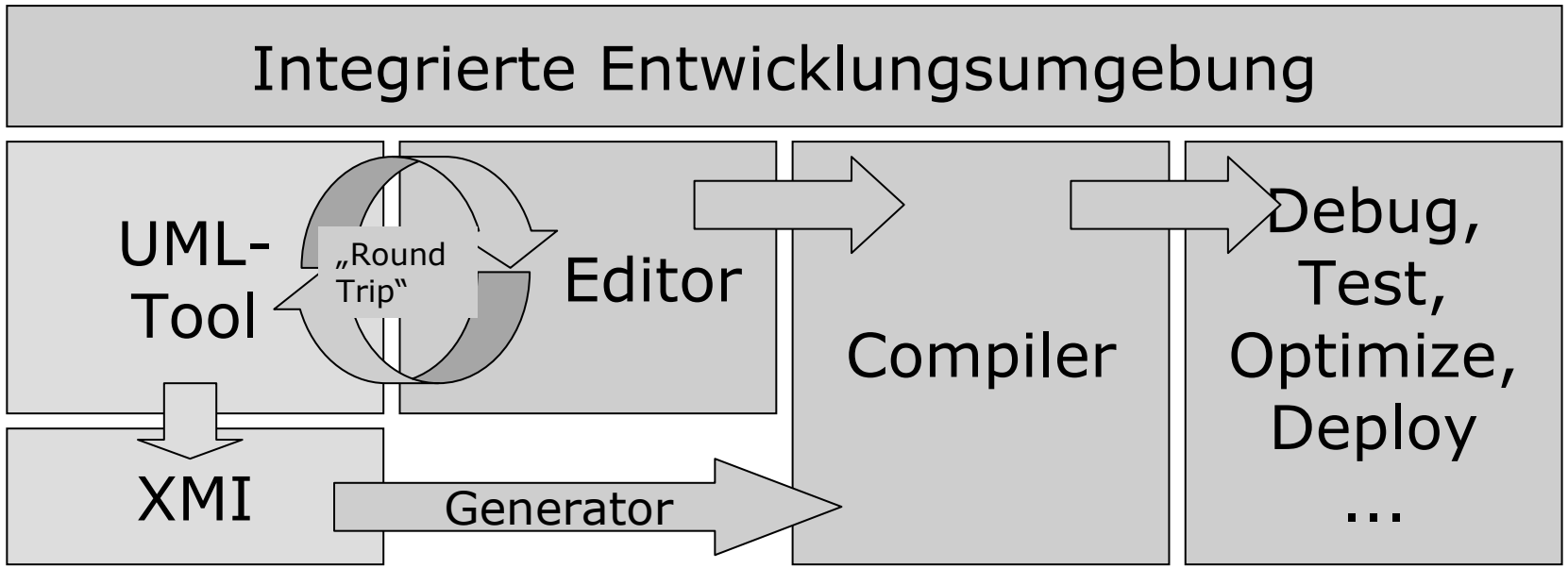
DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Grundidee UML



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

UML und Programmieren



- UML Tools
  - Rational Rose
  - Poseidon (OpenSource)
  - MagicDraw UML ☺

Beispiel-Generator-Tipp: UML2EJB
(http://uml2ejb.sourceforge.net/)

DI Bernhard Anzeletti
Lange Gasse 3/2
A-1080 Wien
Aon.912020365@aon.at

UML – Sprache der Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

DI Bernhard Anzeletti
Lange Gasse 3/2
A-1080 Wien
Aon.912020365@aon.at

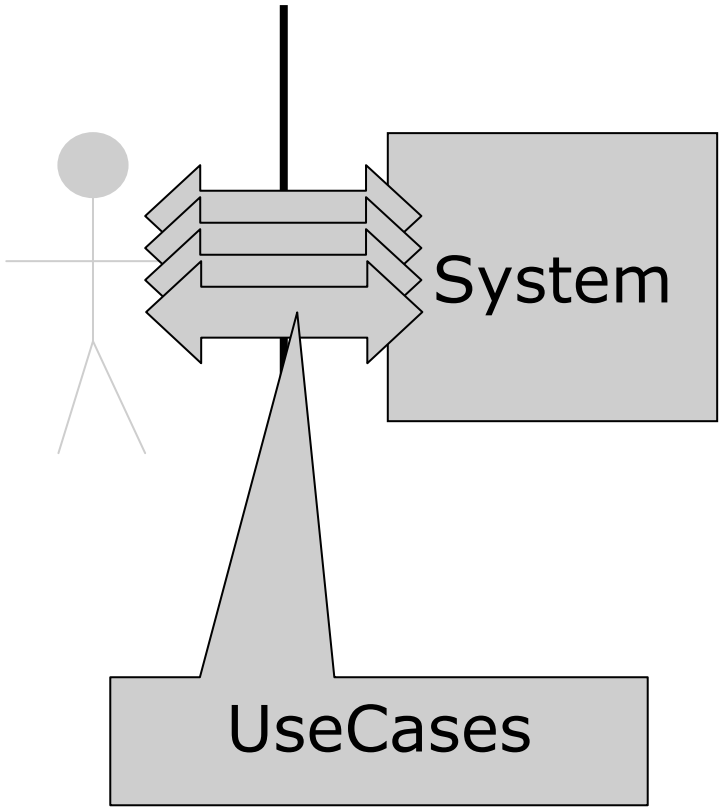
Was ist ein UseCase?

• Ein UseCase beschreibt die Schnittstelle zwischen Benutzer (Akteur) und System für einen logischen Geschäftsvorgang.

•Er hat einen Standardablauf.

•Ausnahmefälle werden in „Szenarios" dargestellt.

•UseCases dienen zur Abstimmung mit den Auftraggebern.



DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Gute Benutzer

- Endbenutzer (als Rollen!)
- Andere Systeme (aktive und passive)
- Eventuell auch: Die Zeit

Die Systemgrenze muß klar definiert sein.

DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Namensänderung – Standard-UseCase

- Der Makler drückt auf „Person suchen“.
- Das System liefert die Suchmaske.
- Der Makler füllt die Suchkriterien aus und drückt auf „Suche“.
- Das System liefert eine Liste von gefundenen Personen.
- Der Makler doppelklickt auf die zu ändernde Person.
- Das System liefert die Personendetailmaske
- Der Makler ändert den Namen
- Der Makler drückt auf „Änderung durchführen“
- Das System antwortet mit der „Name geändert“ Nachricht.

DI Bernhard Anzeletti
Lange Gasse 3/6
A-1080 Wien
Aon.912020365@aon.at

Namensänderung – Szenario Person nicht gefunden

{Vorbedingung: gesuchte Person ist nicht im System}

- Der Makler drückt auf „Person suchen“.
- Das System liefert die Suchmaske.
- Der Makler füllt die Suchkriterien aus und drückt auf „Suche“.
- Das System liefert die Suchmaske mit der Rückmeldung „Für Ihre Eingabe wurde keine Person gefunden“.

DI Bernhard Anzeletti
Lange Gasse 3/2
A-1080 Wien
Aon.912020365@aon.at

Namensänderung – Szenario Person nicht gefunden

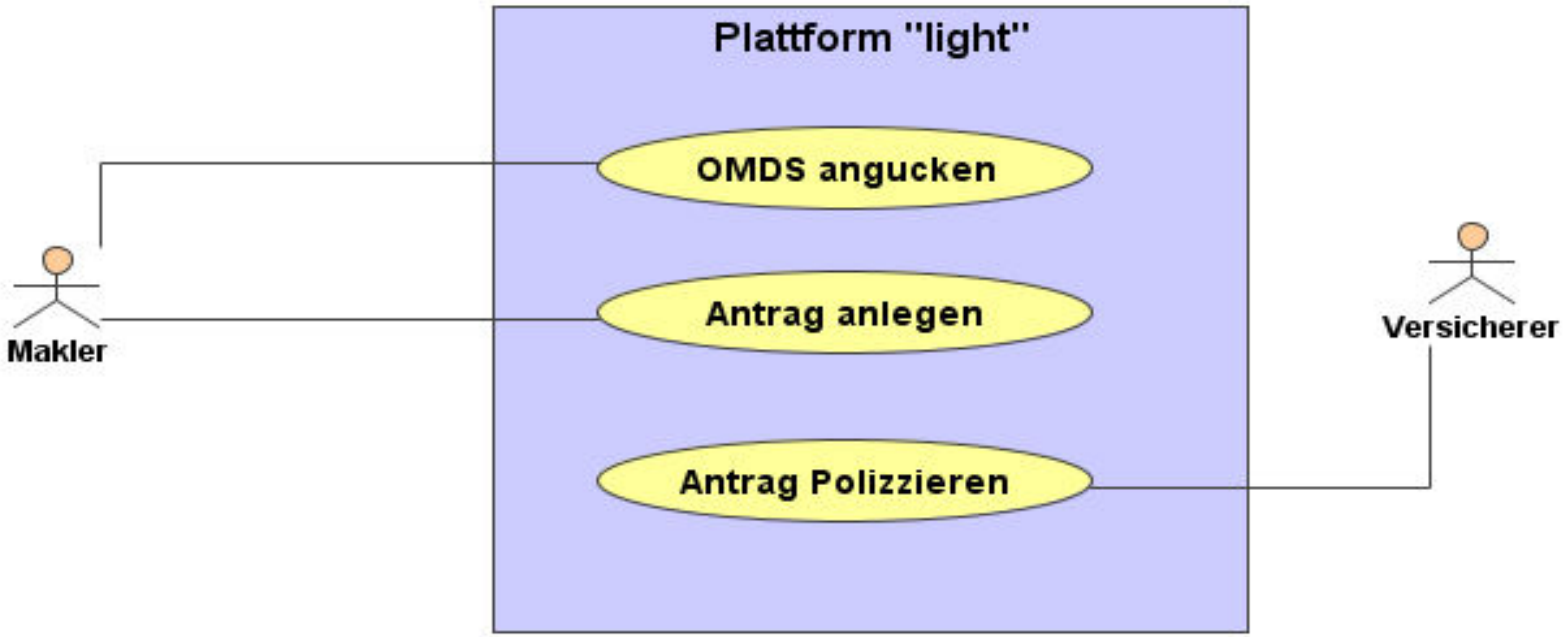
{Vorbedingung: gesuchte Person ist nicht im System}

- Der Makler drückt auf „Person suchen“.
- Das System liefert die Suchmaske.
- Der Makler füllt die Suchkriterien aus und drückt auf „Suche“.
- Das System liefert die Suchmaske mit der Rückmeldung „Für Ihre Eingabe wurde keine Person gefunden“.

Diesen Text muß man hier zweimal schreiben

DI Bernhard Anzeletti
Lange Gasse 3/2
A-1080 Wien
Aon.912020365@aon.at

UseCase Diagram -Beispiel (ohne Worte)



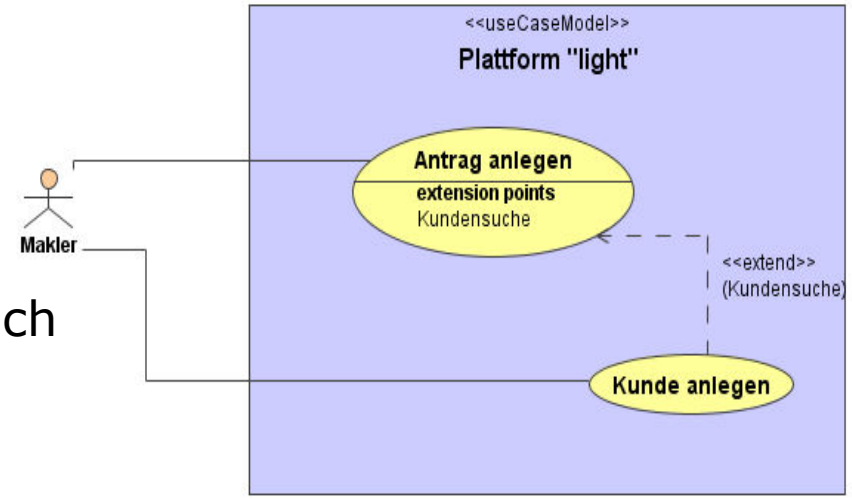
DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

„extends“ - Sonderfunktion

• Makler können Anträge anlegen (UseCase 1)
•Makler können auch Kunden anlegen (UseCase2)

•Beim Antragsanlegen können auch Kunden angelegt werden.

•UseCase „Kunden anlegen“ <<extends>> UseCase „Antrag anlegen“.



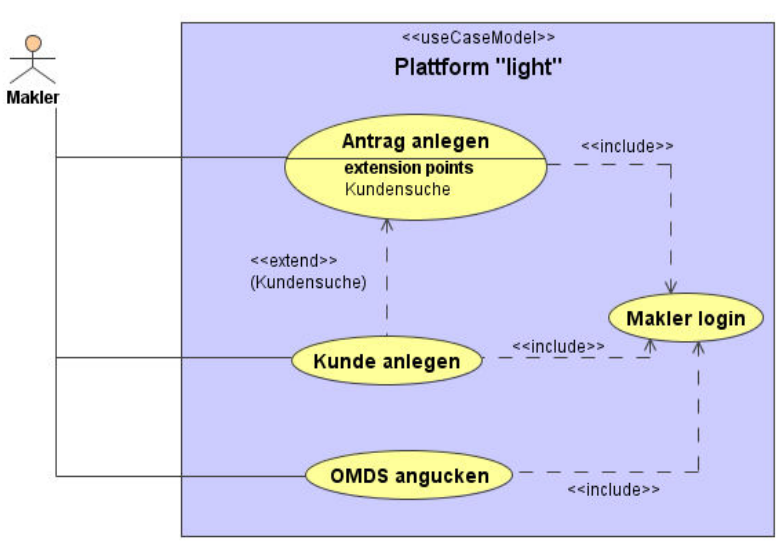
DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

„includes" - Sonderfunktion

• Login an sich existiert nicht (nehmen wir mal an...)

•Bei egal welchem UseCase kann „Login" gemacht werden.

•„Login" ist in den anderen UseCases <<included>>.



DI Bernhard Anzeletti
Lange Gasse 3/2
A-1080 Wien
Aon.912020365@aon.at

UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Class Diagrams

- Bilden statische Aspekte ab (früher: Datenmodell)
- Werden im Lauf der Zeit „entdeckt“
- Klassen, Methoden, Attribute, Beziehungen, Vererbung etc

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Darstellung einer Klasse

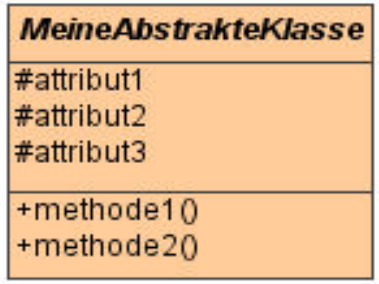
- Eine Klasse wird als Rechteck dargestellt.

- Der Name ist in der Überschrift. Er beginnt mit einem Großbuchstaben. Bei abstrakten Klassen ist er *kursiv*.

- Extra Segmente im Rechteck für Attribute und Methoden

- Sichtbarkeits-"Marker" können verwendet werden

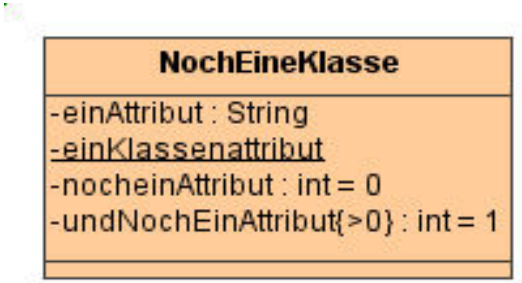
  -... Private
  +... Public
  #... Protected



DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Attribute

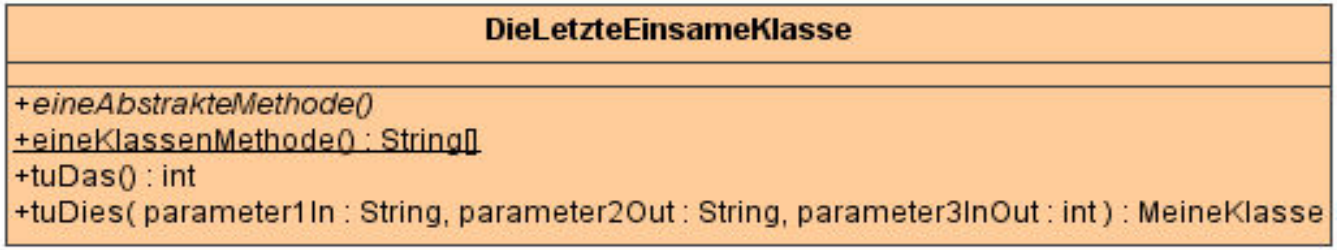
- Attribute können Typen haben
- Sie können einen Default Wert haben
- Man kann sie mit „Constraints“ (Zusicherungen) versehen
- Natürlich existieren auch Klassenattribute.
- Bei konstanten Attributen kommt „@“.



DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Methoden

- Beliebige Return- und Parametertypen (auch andere Klassen)
- Diverse Modifier ([], &, *) – je nach Programmiersprache
- abstrakt, Klassenmethode etc. natürlich auch darstellbar



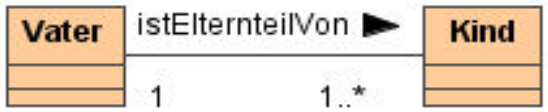
DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Mehrere Klassen ...

- Allgemeine Verbindungen – „Associations“
- Das Ganze und seine Teile – „Aggregations“
- Vererbung

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Associations – Grundvariante



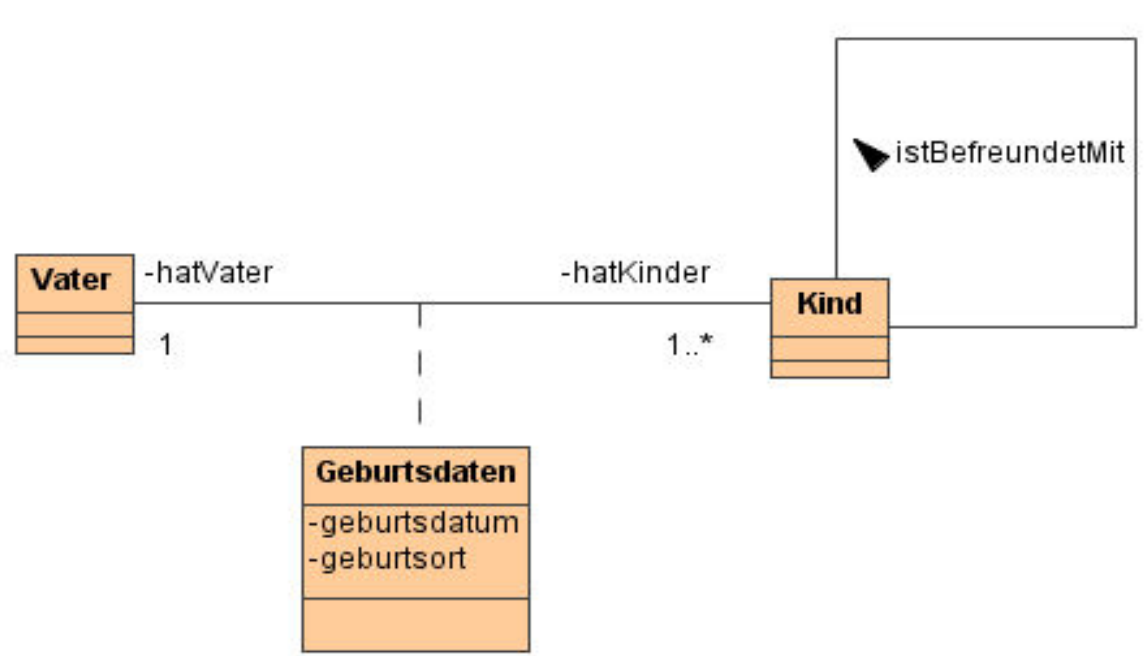
- Verbinden zwei Klassen
- Haben einen Namen
- Man kann Mengen der verbundenen Objekte einschränken – steht immter bei „Ziel“

*	...	Beliebig viele (Null bis n)
1	...	Genau 1
0..1	...	Optional, „Keins oder eins“
2..*	...	Mindestens 2
1..7,9,13..*		Kombinierter Ausdruck

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Associations – Ausbaustufen

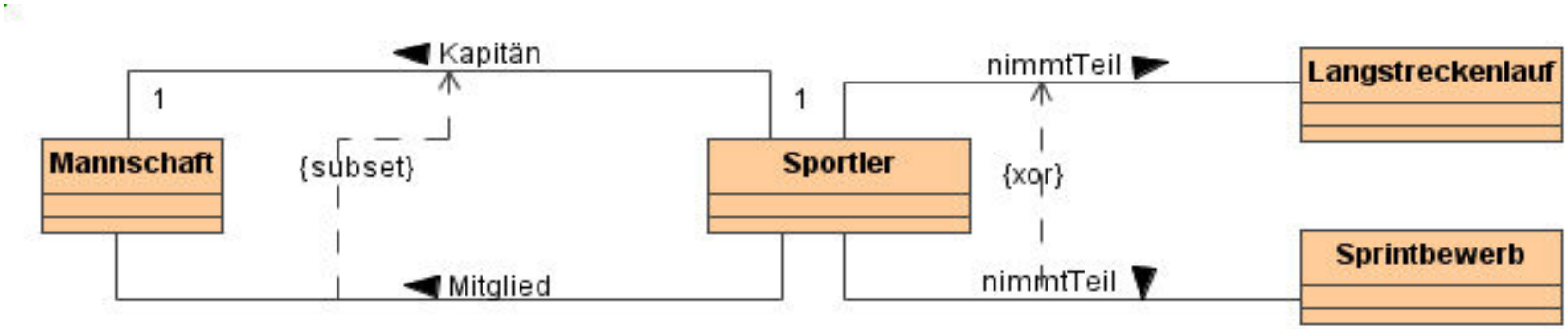
- Eigener Name für jede Richtung
- zusätzliche Attribute (Association Class)
- Reflexive Assozationen



DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Associations – Zusicherungen

- Zusicherungen machen die Semantik stärker
- Subset, or, etc



DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

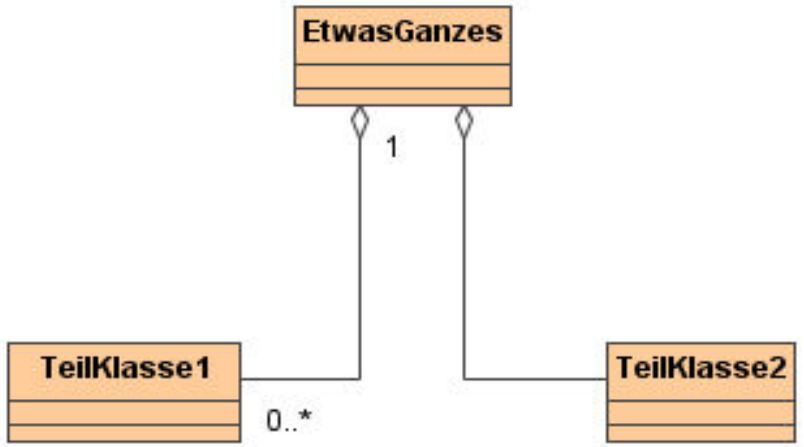
Mehrere Klassen ...

- Allgemeine Verbindungen – „Associations“
- Das Ganze und seine Teile – „Aggregations“
- Vererbung

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Aggregation

- Ein Ganzes
- besteht aus mehreren Teilen
- Die Teile können „für sich“ (also ohne das Ganze) existieren



DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

Composition

- Ein Ganzes
- besteht aus mehreren Teilen
- Die Teile können „für sich“ (also ohne das Ganze)

NICHT

existieren

DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

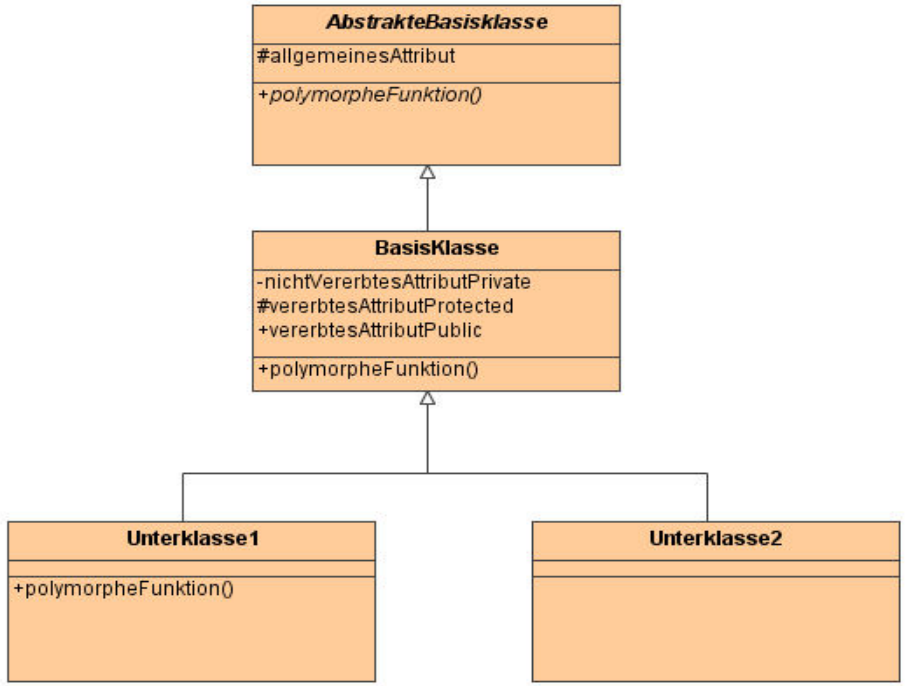
Mehrere Klassen ...

- Allgemeine Verbindungen – „Associations“
- Das Ganze und seine Teile – „Aggregations“
- Vererbung

DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

Vererbung

- Klar, auch darstellbar:
- Einfach- oder Mehrfachvererbung
- Abstrakte Klassen können nicht instanziiert werden, müssen also mindestens eine Unterklasse haben



DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

Stereotypes

- Abgekürzte Notation für Standard-Klassenarten
  - Boundary, Entity, Control
  - Metaclass
  - Process
  - ...

DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Gute Klassendiagramme – Meine persönliche Meinung

- Ein System braucht oft viele Klassendiagramme zur Beschreibung.
- Ein Diagramm „reift" über mehrere Versionen hinweg.
- Nicht „die Welt" modellieren, sondern das „Wesen des Geschäfts".

- Ein Diagramm sollte ca. 1 DIN A4 Seite sein
  - Zum Überblick ohne Methoden und Attribute
- Es sollte einen Aspekt darstellen.
  - Wird in UML durch „Pakete" gestützt

- Vorsicht mit Verbindungen
  - Keine „Kreise" (redundante Assoziationen)
  - Keine „Platinenlayouts"
  - E/R und Entity-Dependency in Überlegung einbeziehen
  - Typische Muster (Patterns) verwenden
  - Stereotypen verwenden

- Siehe Sonderkapitel „Patterns"

DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

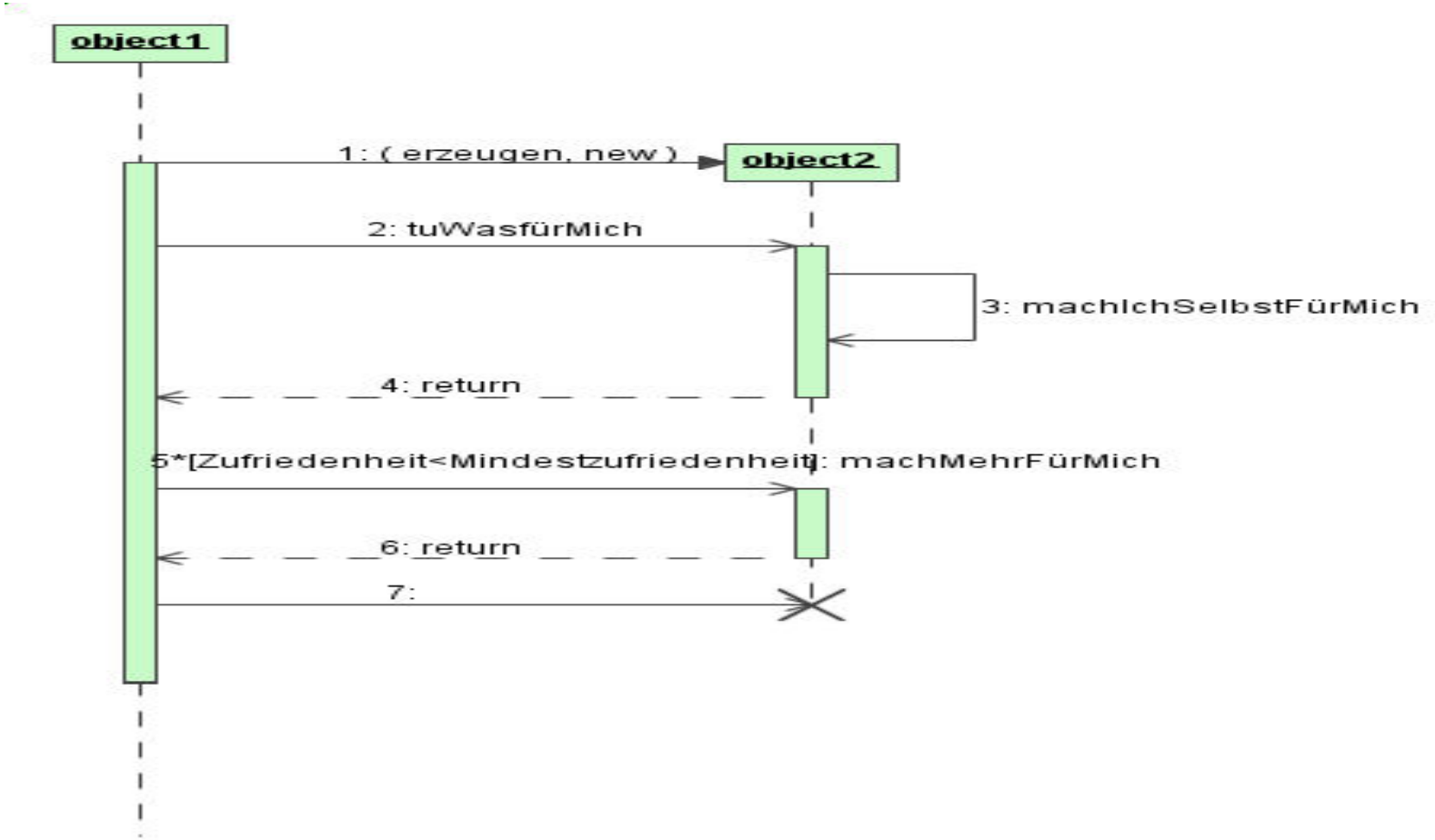
DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

Sequence Diagrams

- Bilden dynamische Aspekte ab (Aufruf-Baum)
- Können aus UseCases gebildet werden
- Müssen mit dem Klassendiagramm übereinstimmen

DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

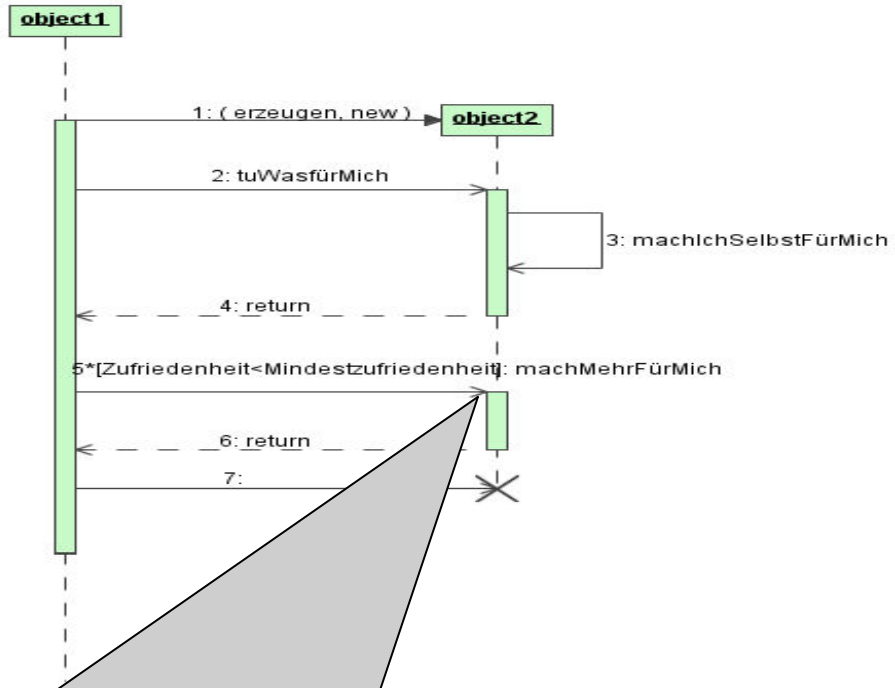
Sequence Diagrams – Beispiel



DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Sequence Diagrams – Ausdruckmöglichkeiten 1

- Standard: Synchroner Aufruf
- new, delete möglich
- Return möglich
- Bedingungen, Iteration

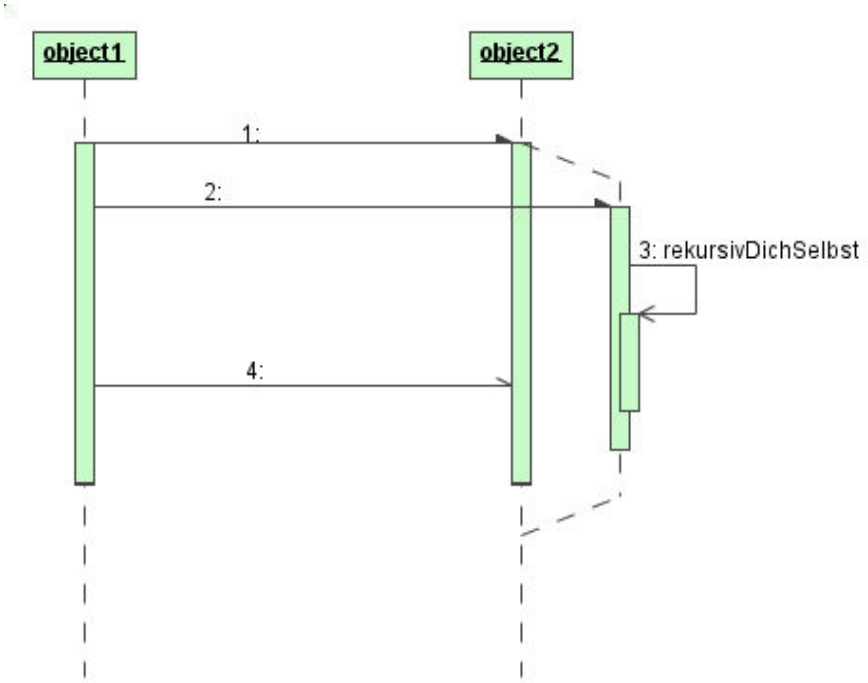


Man erkennt die Bedeutung an der Gestalt der Pfeilspitze....

DI Bernhard Anzeletti
Lange Gasse 3/4
A-1080 Wien
Aon.912020365@aon.at

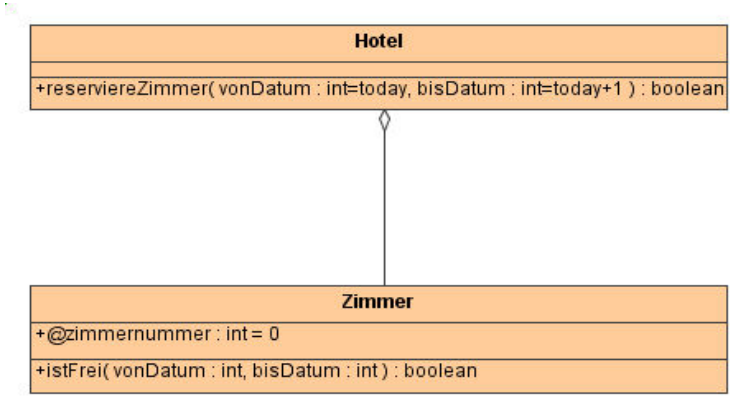
Sequence Diagrams – Ausdruckmöglichkeiten asynchron

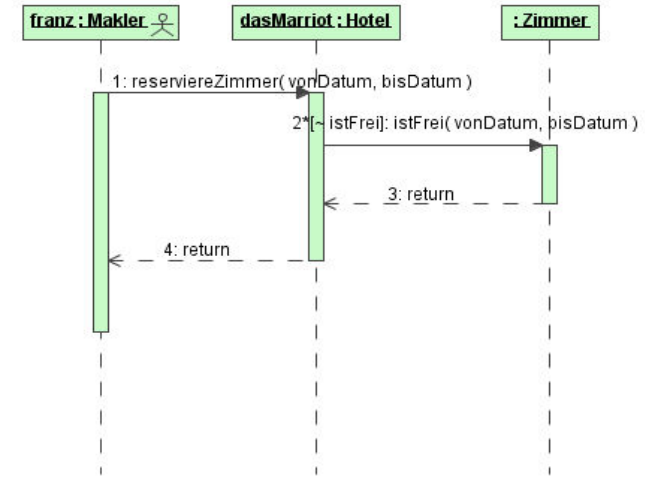
- Asynchroner Start
- Asynchroner Stopp
- Parallele Threads in gleichem Objekt



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Sequence- und Klassendiagramm





- Methoden im Klassendiagramm passen mit Aufrufen im Sequencediagramm zusammen.

- Um aufrufen zu können, gibt es einen Weg (Association etc) vom Aufrufer zum Aufgerufenen.

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

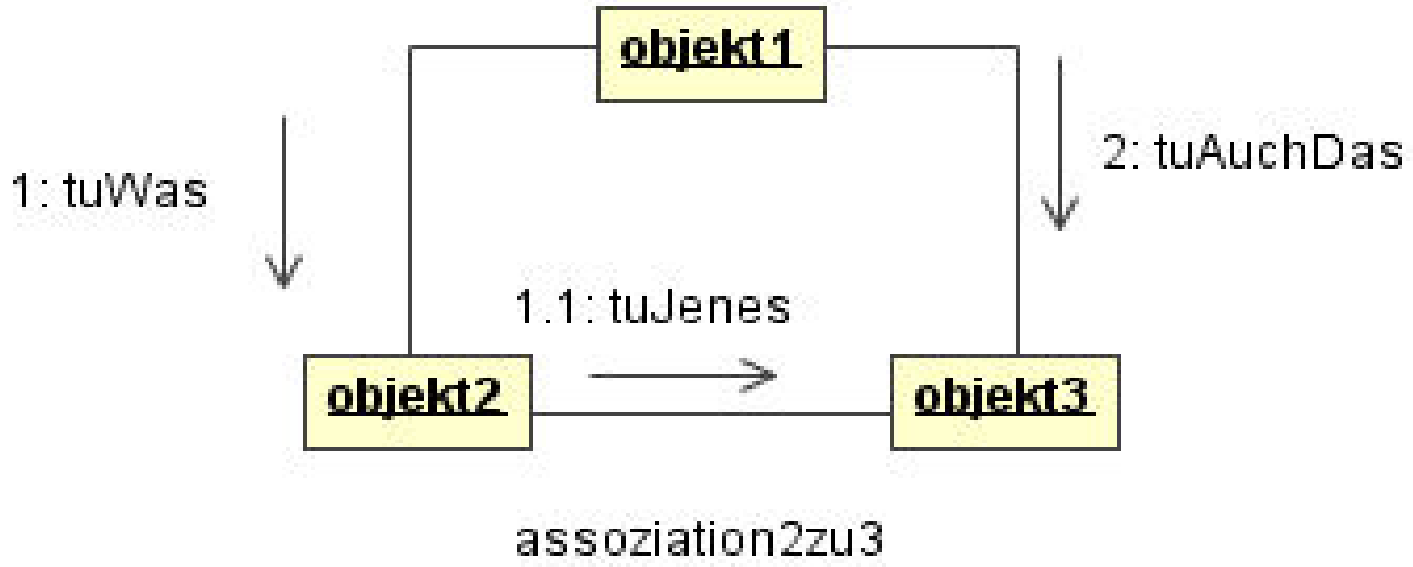
DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Collaboration Diagrams

- Bilden dynamische Aspekte ab (Aufruf-Baum)
- Können aus UseCases gebildet werden
- Müssen mit dem Klassendiagramm übereinstimmen
- Sind „Sequence-Diagrams“ im Überblick, d.h. für frühe Phasen besser geeignet
- Vorteil: Navigation über Assoziationen wird dargestellt

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Collaboration Diagrams – Beispiel (ohne Worte)



- „Lyrik" wie Sequence Diagrams...

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

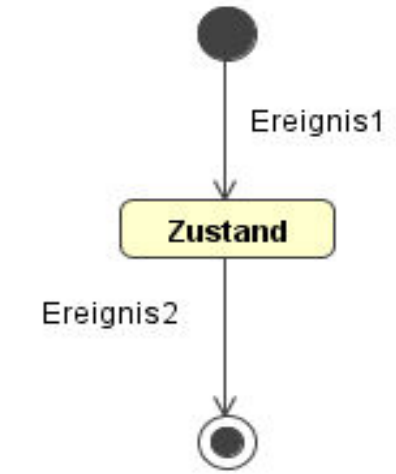
Status Diagrams

- Bilden dynamische Aspekte ab (Aufruf-Baum)
- Lebenszyklus eines Objekts
- Ereignisse (=Methodenaufrufe) bewirken Statusübergänge
- Für bestimmte Klassen sinnvoll
- Gut mit Benutzer abstimmbar!

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

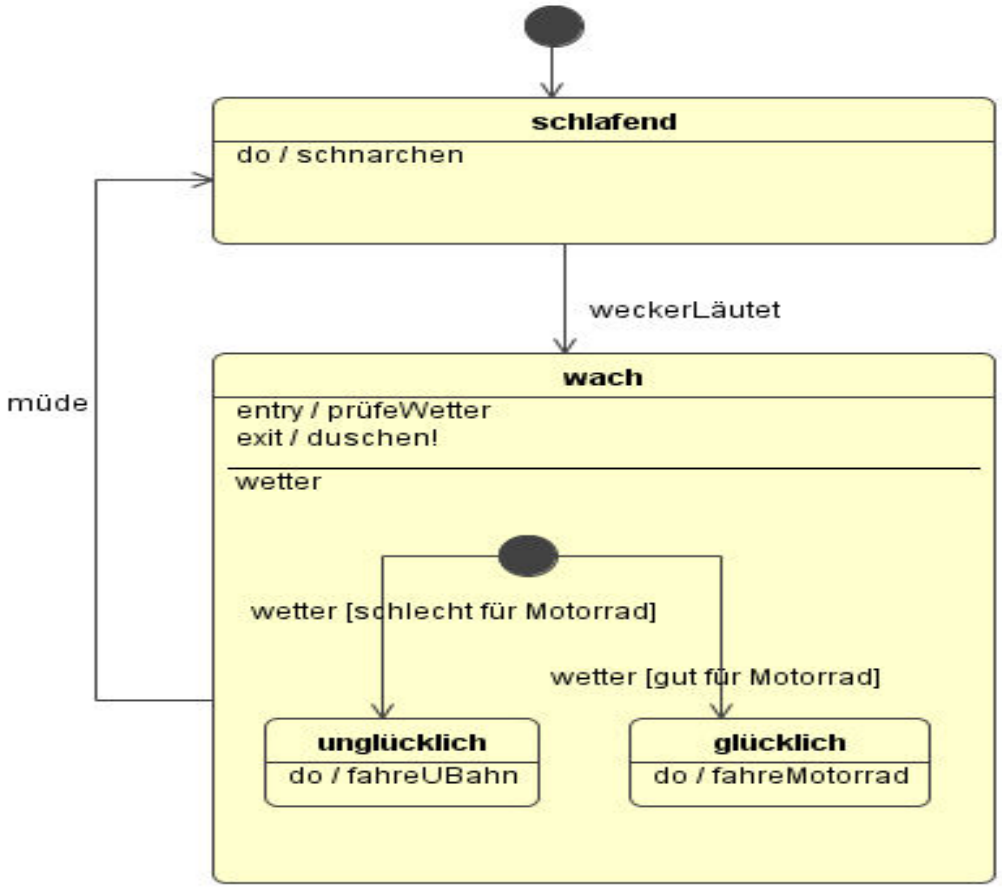
Status Diagrams – Beispiel

- Ein Status Diagramm bezieht sich auf eine Klasse
- Ein Zustand hat einen Namen
- Der Startzustand ist schwarz, ohne „Kringel"
- Ein Endzustand ist schwarz „mit Kringerl"
- Ein Übergang tritt aufgrund eines Ereignisses ein



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Status Diagrams – Erweitertes Beispiel (ohne Worte)



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Status Diagrams – Erweitertes Beispiel

- Zustände haben auch Aktionen

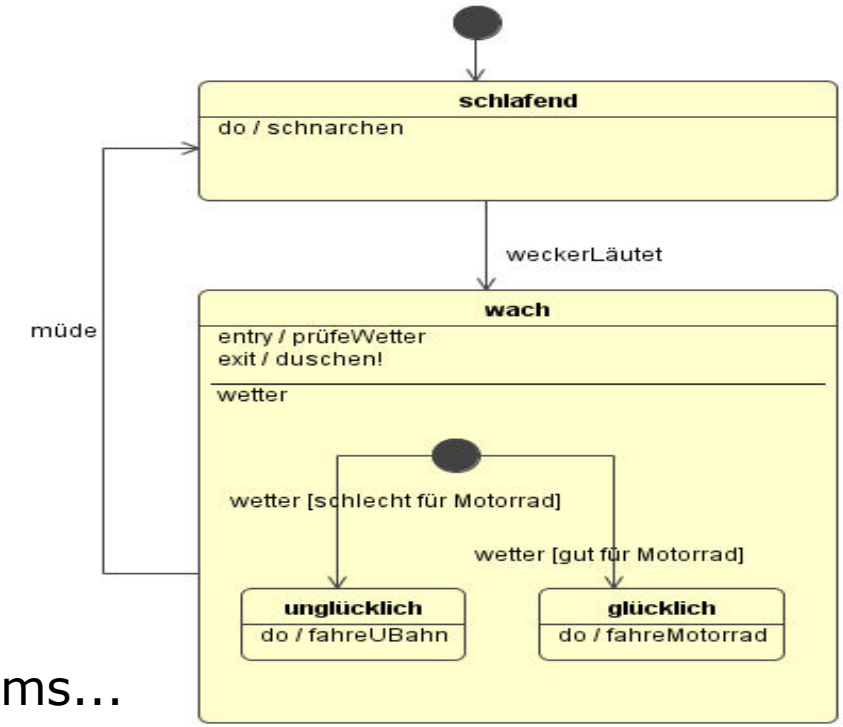
  Do ... Solange in diesem Zustand
  Entry ... Beim Eintritt in Zustand
  Exit ... Beim Austritt aus Zustand

- Zustände haben auch Attribute

  zB Wetter

- Aktionen können an Bedingungen geknüpft sein

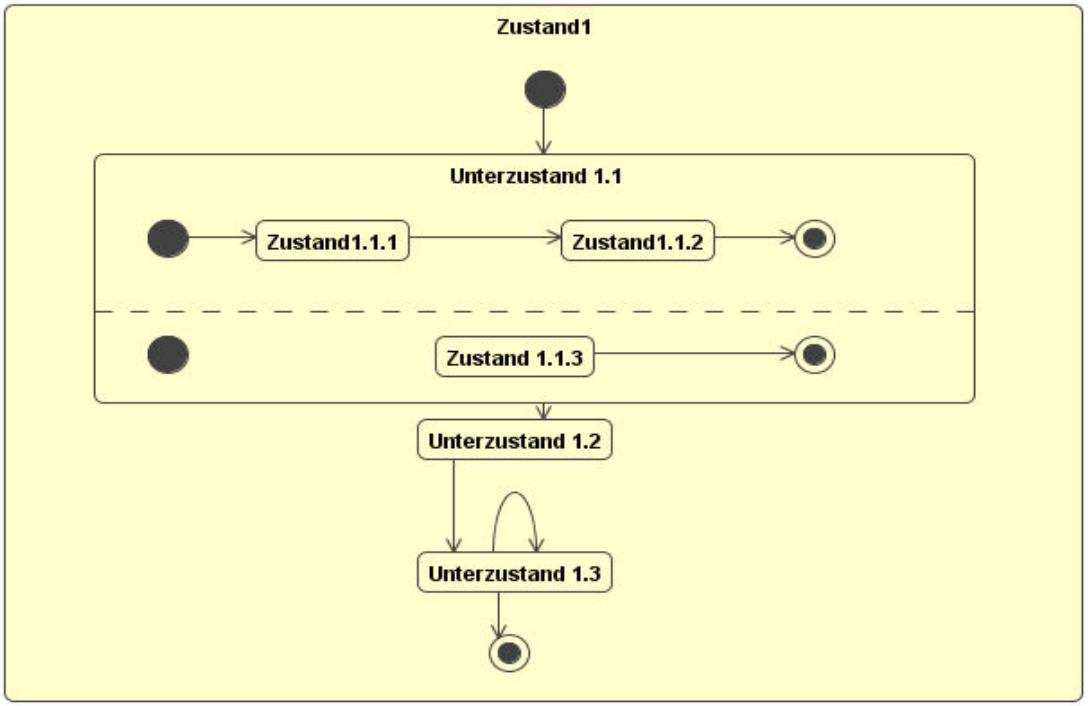
- Aktionen sonst wie Sequence Diagrams...



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

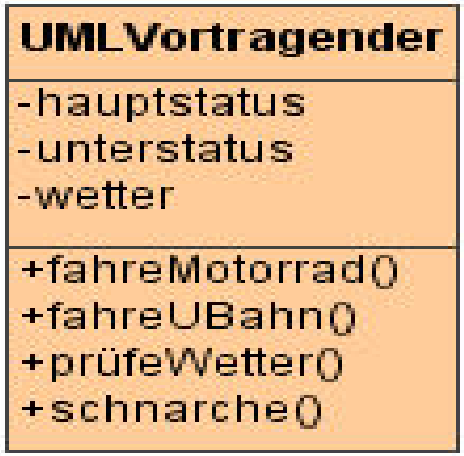
Status Diagrams – Erweitertes Beispiel mit Parallelität

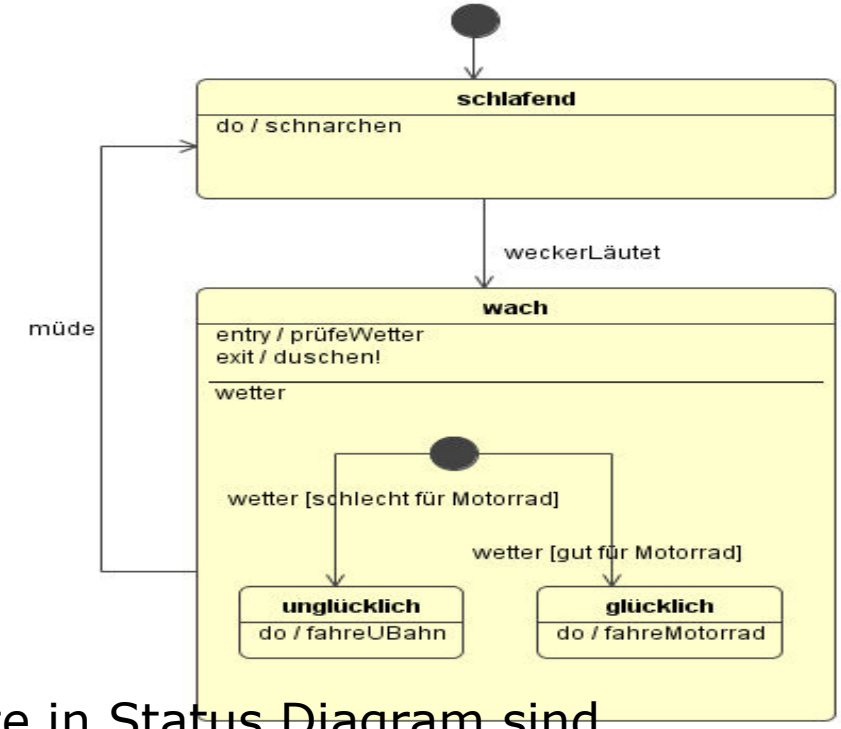
- Unterzustände können parallel verfeinert werden



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Status Diagrams – Class Diagrams





- Zustände, Zustandsattribute in Status Diagram sind Attribute der Klasse
- Methoden im Status Diagram sind Methoden der Klasse

DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams

- Bilden Prozesse ab
- (Wer kennt ADONIS oder ARIS?)
- Schaut aus wie Status-Diagrams
- Besonderheit: Zeit fliesst „von oben nach unten“

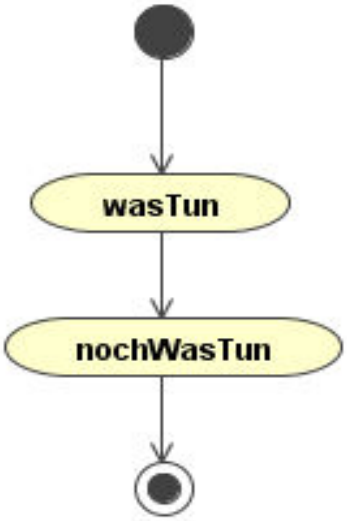
DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams 1 - Grundlagen

- Es gibt wieder einen Start (ohne Kringerl)
- Es gibt wieder ein Ende (mit Kringerl)
- Die Kästchen sind diesmal Aktivitäten
- Die Übergänge sind – Aktivitätsfolgen

- Achtung: Was genau ist eine Aktivität?

  Ende bei
  1) Übergabe eines Dings
  2) Übergabe der Verantwortung
  3) ... ?

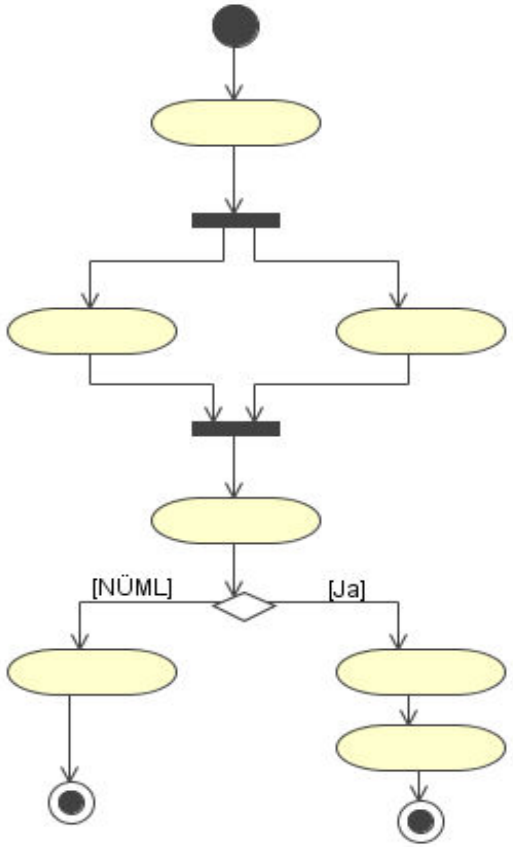


DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams 2 – Entscheidungen und Parallelitäten

- Es gibt auch Entscheidungen
- ... Und Parallelität

1) Split
2) Join



DI Bernhard Anzeletti
Lange Gasse 3/
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams 3 – Swim Lanes

- Man kann auch sagen, wer es macht.

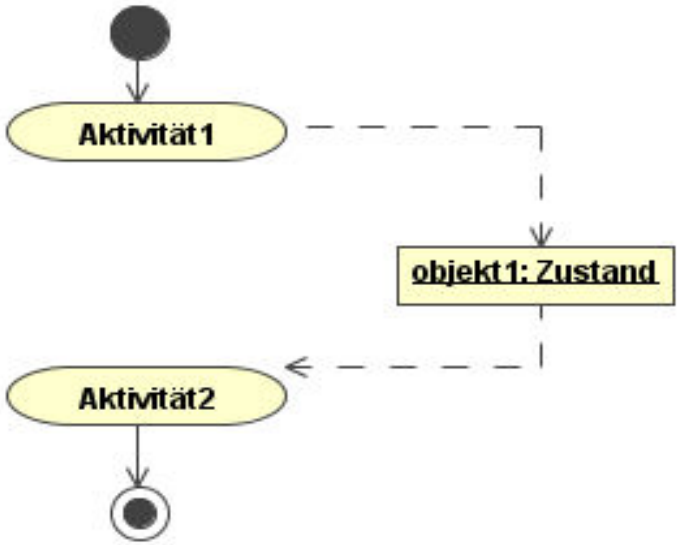
DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams 4 – Objektzustände

- ... Und mit Objektzuständen verbinden.

Pfeil zu Objekt: Aktivität ändert Zustand im Objekt

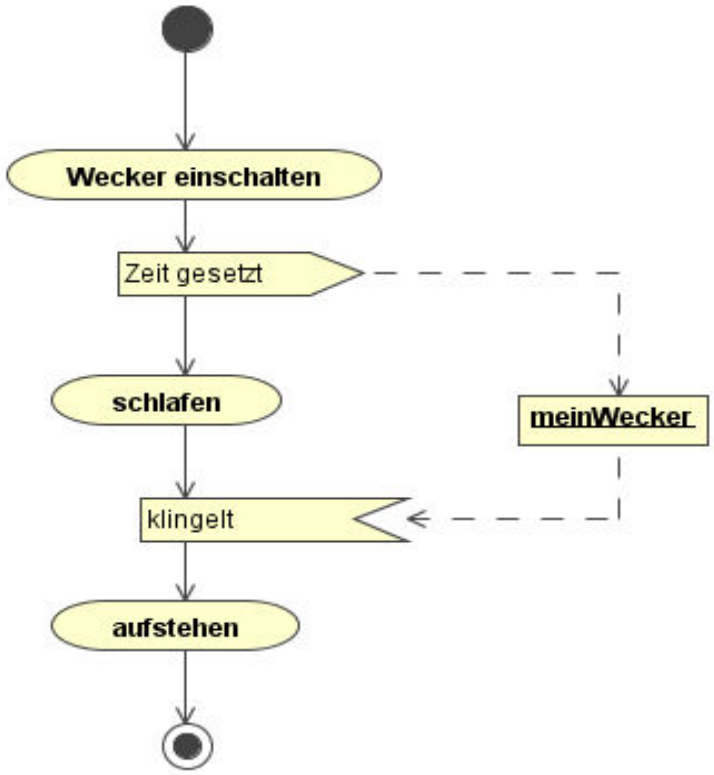
Pfeil zu Aktivität: Aktivität kann nur dann fortgesetzt werden, wenn Objektzustand erreicht



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

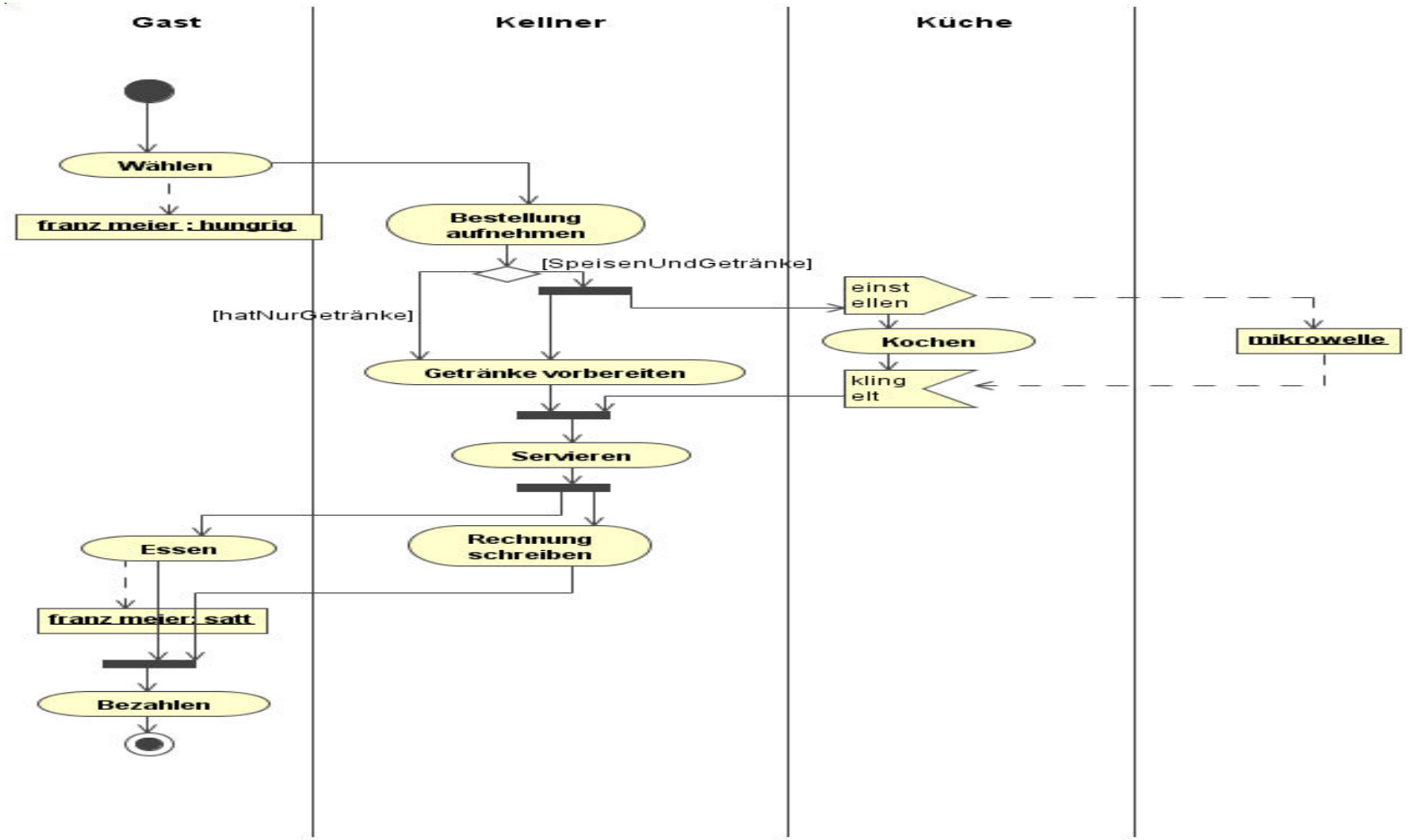
Activity Diagrams 5 - Signale

- ... Und mit Objekten reden
- ... Und Objekten zuhören



DI Bernhard Anzeletti
Lange Gasse 3/5
A-1080 Wien
Aon.912020365@aon.at

Activity Diagrams – Beispiel (ohne Worte) - Schocktherapie



UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

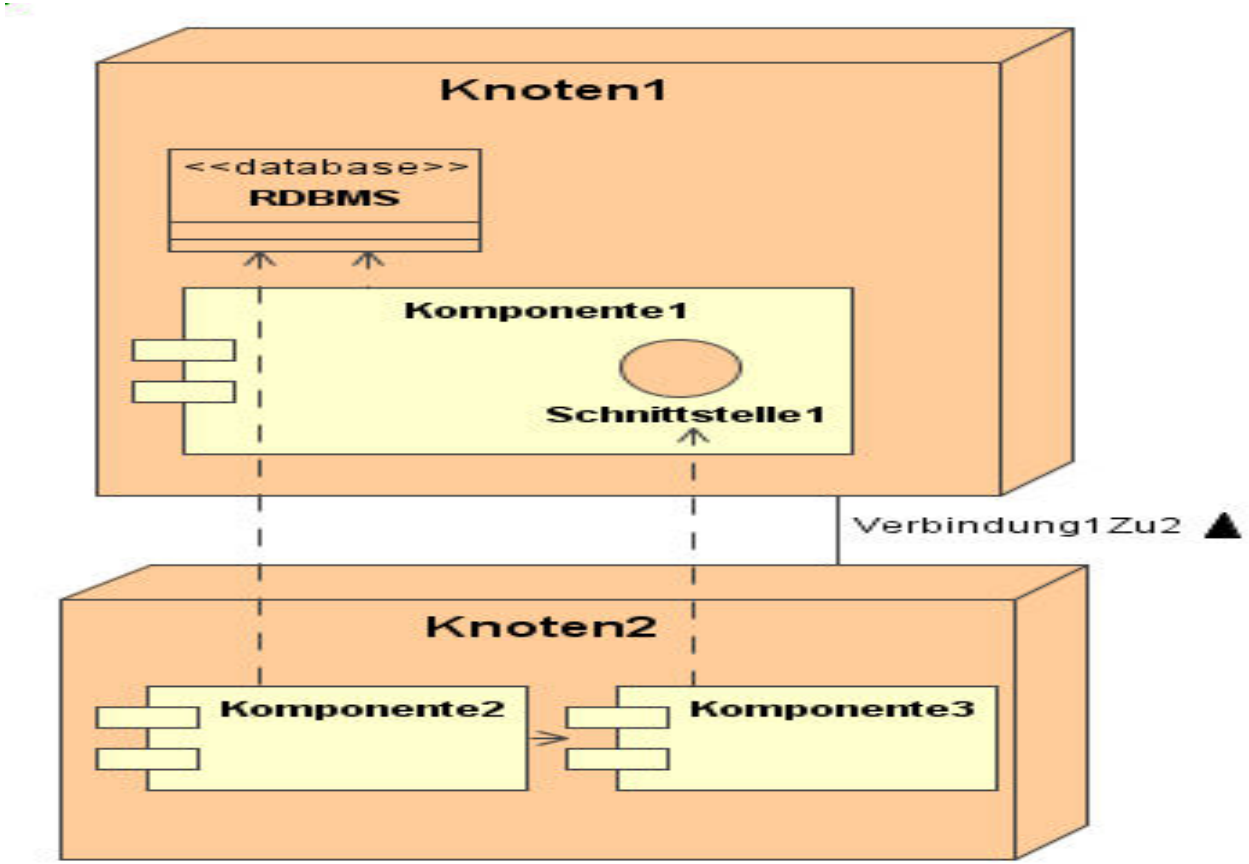
Implementation Diagrams

- Bilden statische Aspekte ab (Architektur)
- Knoten, Verbindungen
- Komponenten
- In Architekturüberlegungen sinnvoll
- Gut mit Eigentümer abstimmbar

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

Implementation Diagrams – Beispiel

- Knoten und Verbindungen
- Komponenten
- Schnittstellen
- Pfeil bedeutet „abhängig von“



UML-Diagramme

- Use Case Diagramms
- Class Diagramms
- Sequence-Diagramms
- Collaboration-Diagrams
- Status-Diagrams
- Activity-Diagrams
- Implementation Diagrams

- Zusammenfassung

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at

UML Links, Literatur etc.

- Theorie...
  - http://www.uml.org
  - http://www.omg.org/uml
  - http://www.rational.com/products/rup/index.jsp

- Und Praxis
  - http://www.rational.com/products/rose/index.jsp
  - http://www.magicdraw.com
  - http://www.gentleware.com (Poseidon)

DI Bernhard Anzeletti
Lange Gasse 3/9
A-1080 Wien
Aon.912020365@aon.at