Spezifikation von Anwendungskernen für große Informationssysteme

Vorlesung: Software-Engineering für große, betriebliche Informationssysteme

für Universität Leipzig, Sommersemester 2004
Institut für Software- und Systementwicklung
Professur Wirtschaftsinformatik, insbes. Informationsmanagement

Hans Hartmann (Generali VIS Informatik Ges.m.b.H., Wien)
Wolfgang Keller (AMB Generali Informatik Services GmbH, Aachen)

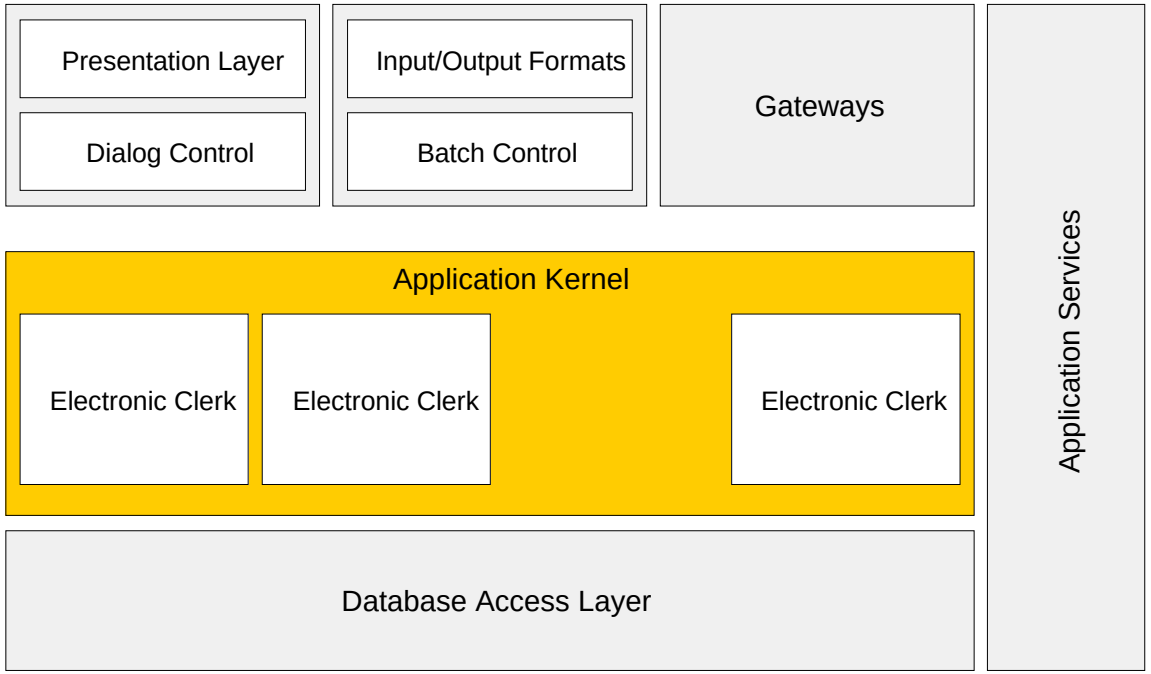
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Überblick

- Anwendungskern – wo finde ich den?
- Spezifikation versus „Runterprogrammieren“ versus UML- Bilder
- Ein Beispiel für textuelle Spezifikation: Die Sachbearbeitermethode
- Die Sachbearbeitermethode und OO
  - Deltas zu UML und hartem OO
  - Die Sachbearbeitermethode und J2EE Patterns

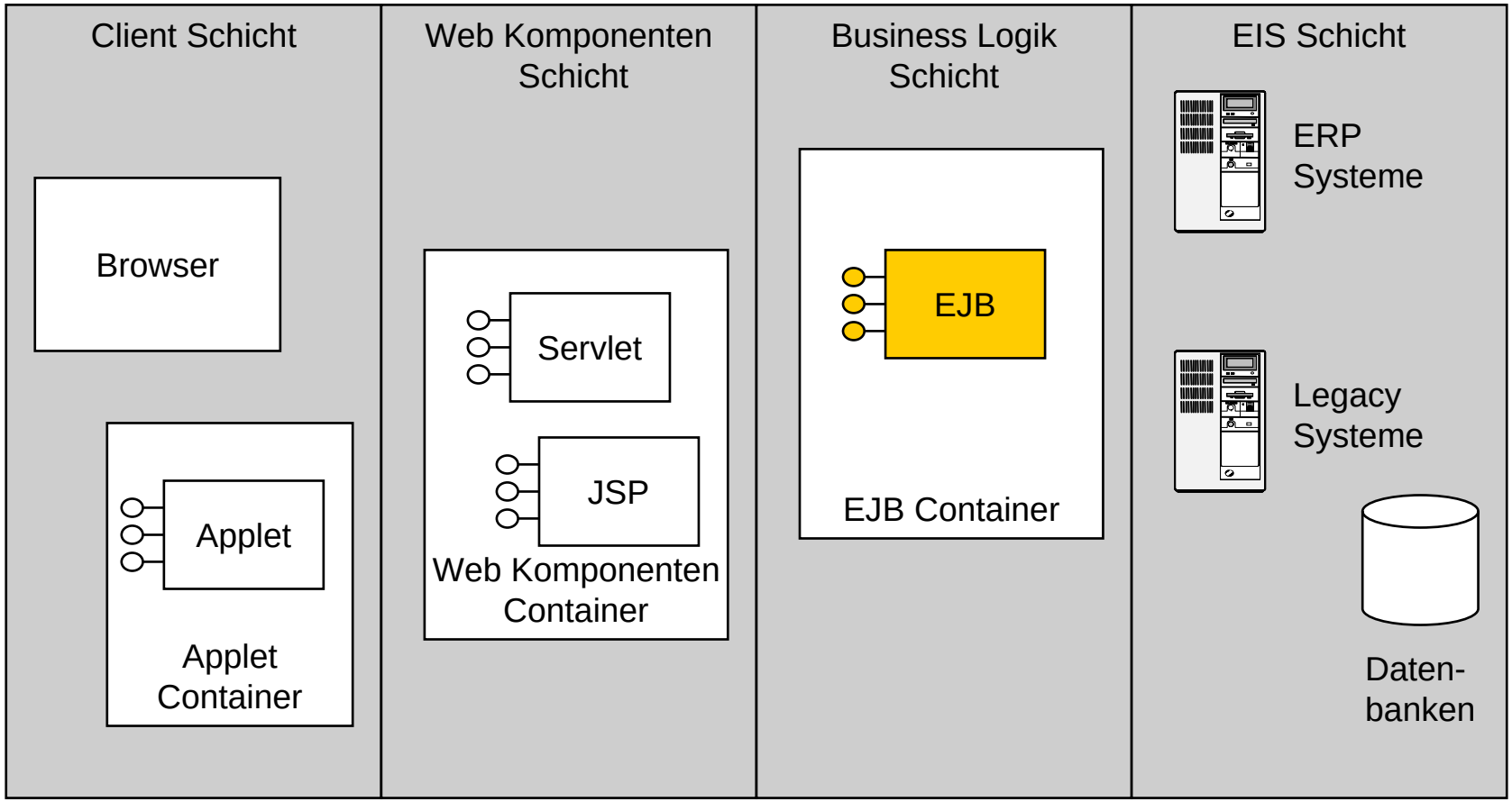
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Anwendungskern
Wo finde ich den?



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Anwendungskern
Wo finde ich den?



Vorlesung „Software Engineering für große, betriebliche Informationssysteme“,
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

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

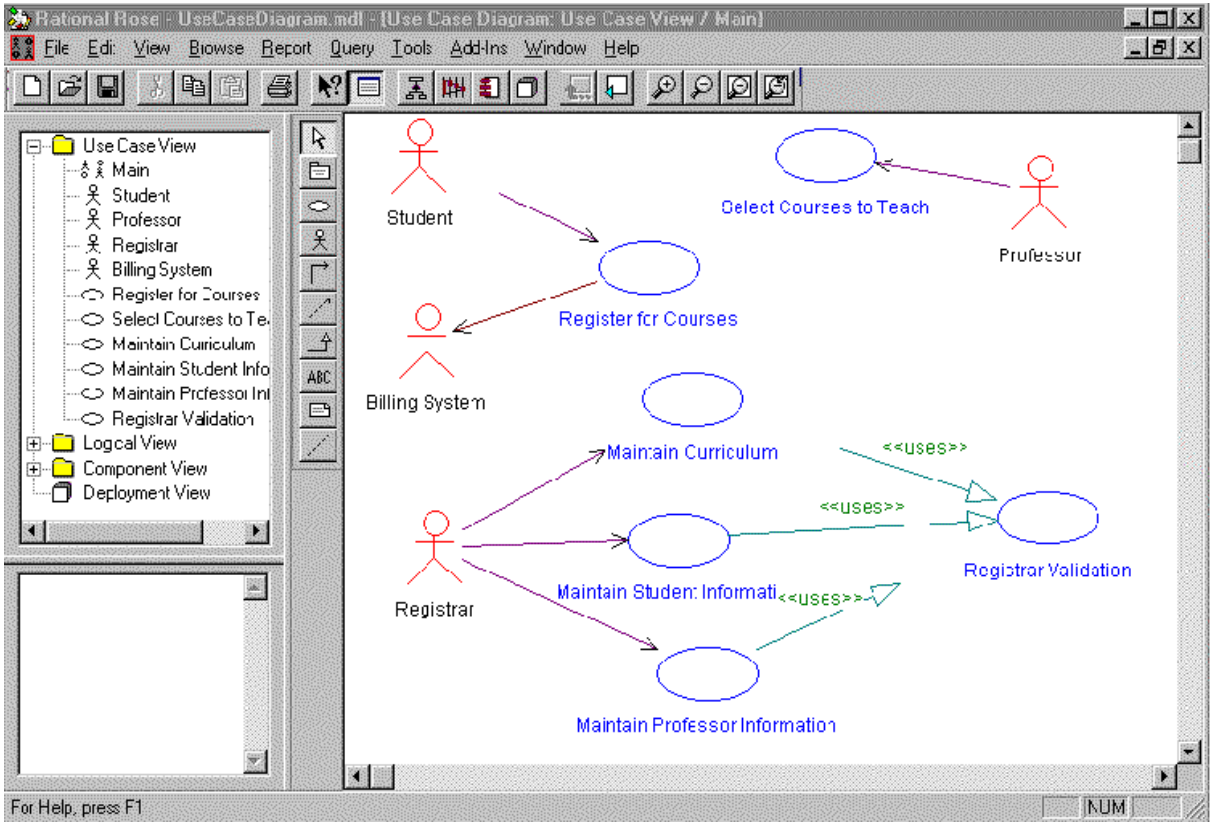
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Spezifikation versus „Runterprogrammieren“ versus UML- Bilder

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Bilder ...

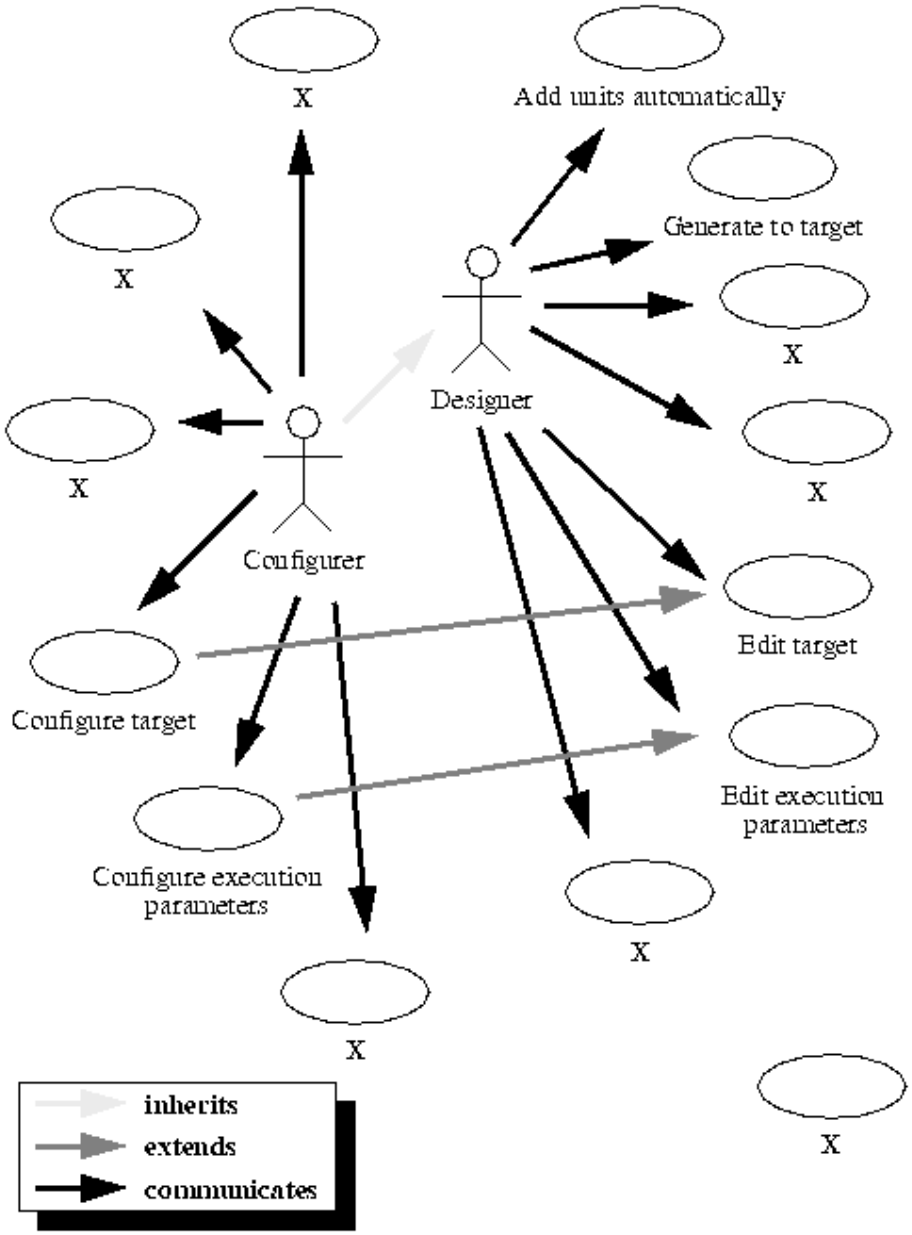
- ein Bild sagt mehr als 1000 Worte
- aber was sagen 1000 Bilder dieser Art ...



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Bilder ...

- noch ein schönes Beispiel ...
- Kapieren Sie was???



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

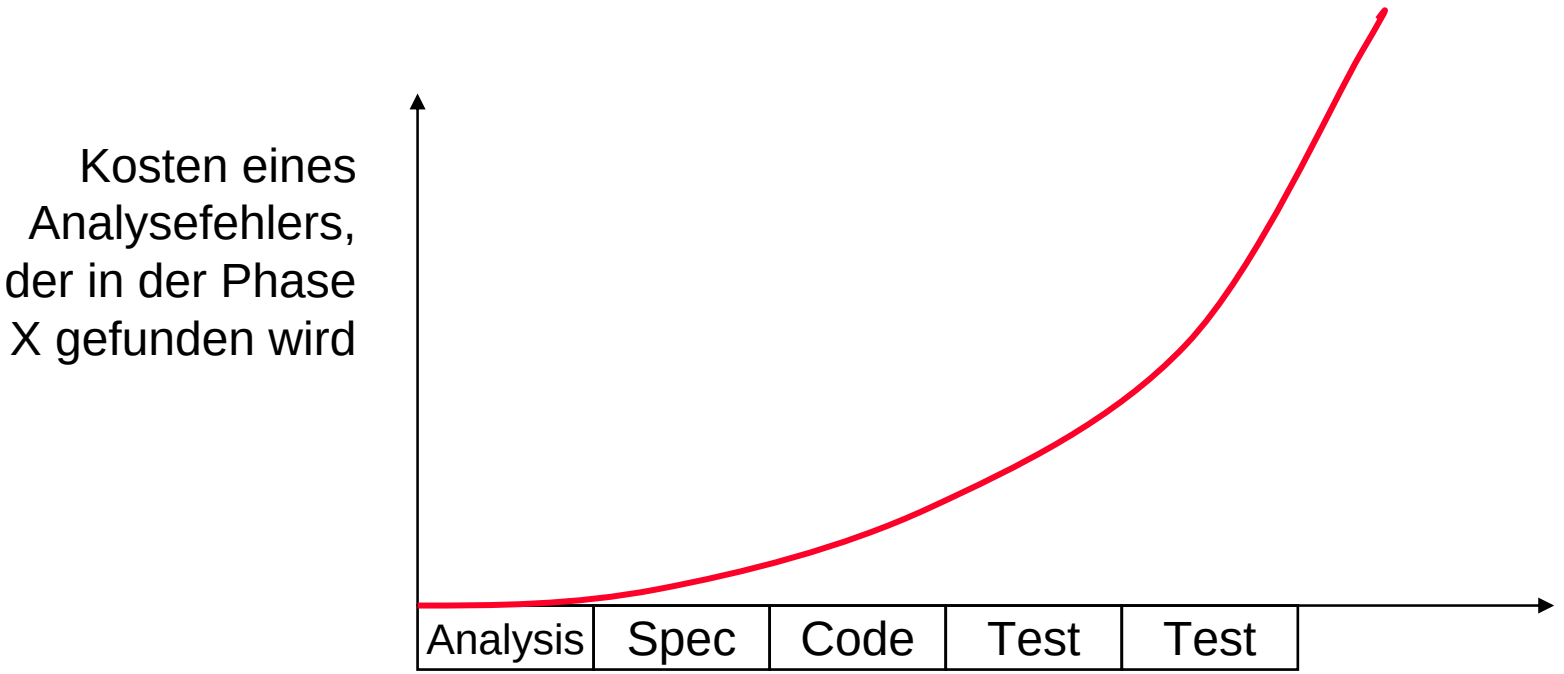
einfach „Runterprogrammieren“?

- 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

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

einfach „Runterprogrammieren“?

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



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Folgerung

- einfach Runterprogrammieren: Für große Systeme wegen der Arbeitsteiligkeit nicht geeignet
  - und für „ältere Umgebungen“ mit langsamen Zyklen schon gar nicht
- 1000 Bilder
  - es kann passieren, dass man dabei den Überblick verliert

- Also: Auf eine gesunde Mischung kommt es an aus
  - Bilder, Texten, Generierung ....
- Damit Sie die finden, heute mal ein Beispiel, wie man es rein textuell macht – einfach als Contrapunkt und weil das sonst selten gezeigt wird.

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Ein Beispiel für textuelle Spezifikation: Die Sachbearbeitermethode

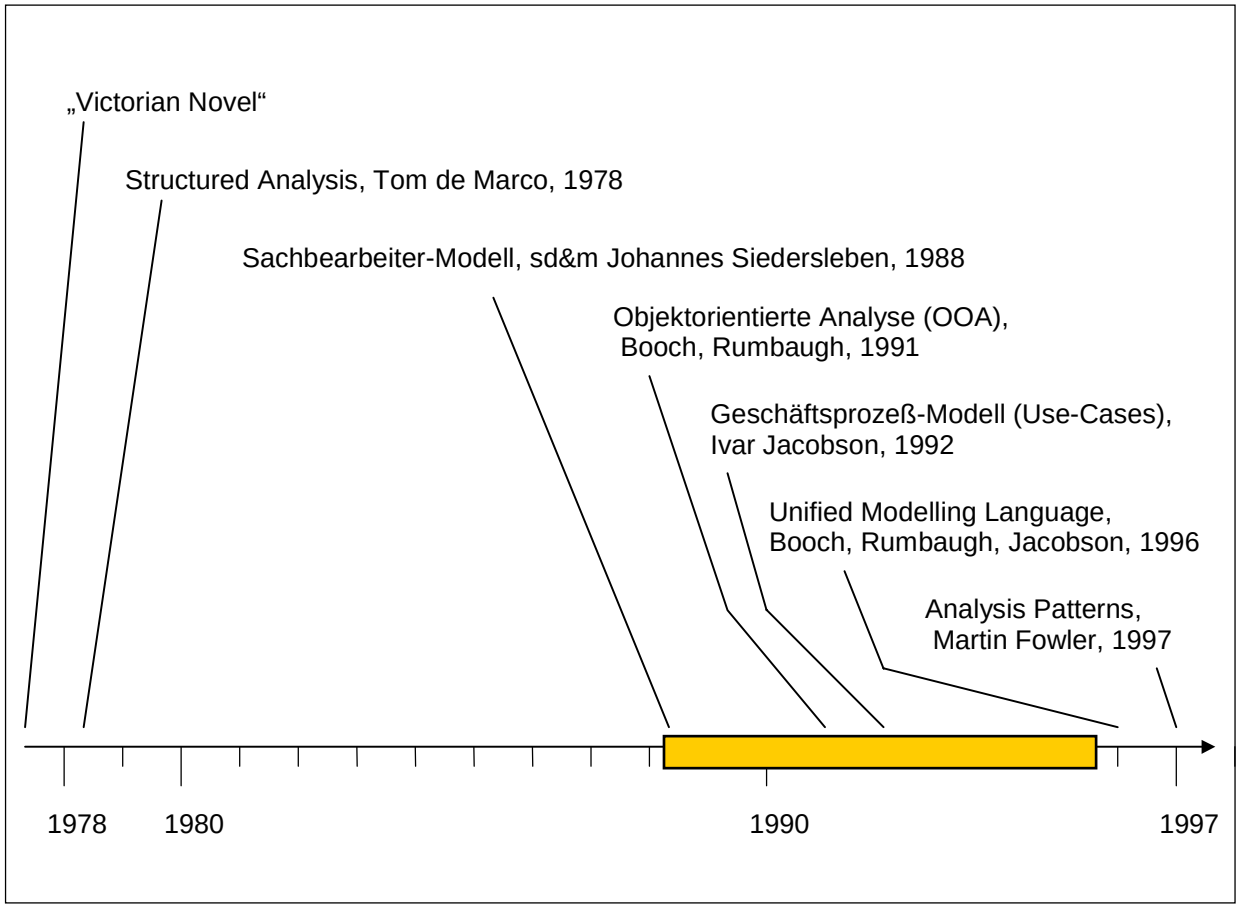
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Spezifikation mit der Sachbearbeitermethode

- Wann war das „state of the art“?
- Was zeichnet eine gute Spezifikation aus

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Wo befinden wir uns dabei in der Geschichte der Software ..



2004 aktuell
- UML 2.0
- MDA

Quelle sd&m

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

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 Softwareanbieter aus einer Spezifikation Software bauen können, die sich identisch verhält und austauschbar ist.

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

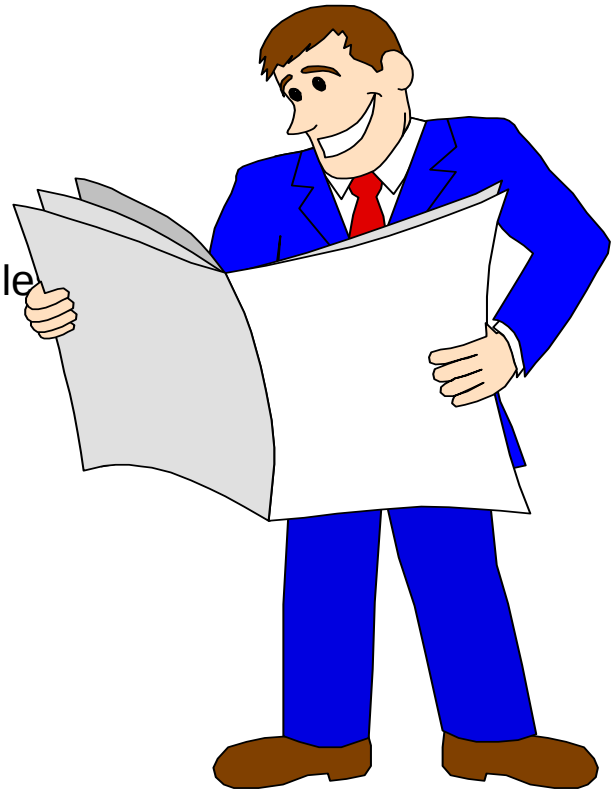
Nicht ausreichend sind also....

- 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

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Ihre Leser sind ...

- Zuallererst **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 le
    und Datentypen definieren
    ist Wunschdenken



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

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

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Und wie bitte beschreibe ich einen Anwendungskern?

wait and see ☺

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Spezifikation mit der Sachbearbeiter-Methode

- Sachbearbeiter erbringen Dienste
- Sachbearbeiter haben Gedächtnis
  - Karteikästen
- Sachbearbeiter beauftragen sich gegenseitig
  - Sie als Spezifikateure sind die Organisatoren
- Die Dienste müssen genau spezifiziert werden
  - Aufträge der Sachbearbeiter

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

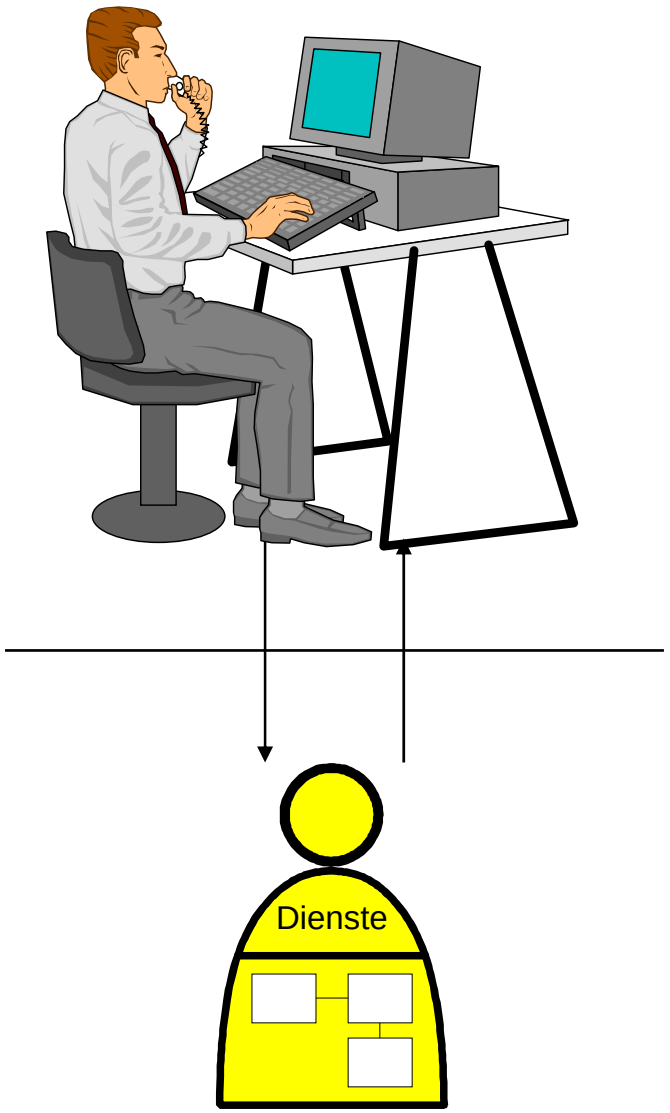
Spezifikation mit der Sachbearbeiter-Methode

- Die Dinge, die ein Sachbearbeiter verwaltet haben ein Zustandsmodell
- Zusammenhang Zustandsmodell und Dienste
- Der Karteikasten ist eine Datensicht
- Diskussion
  - Verwendbarkeit der Metapher
  - Methodische Einordnung

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter erbringen Dienste

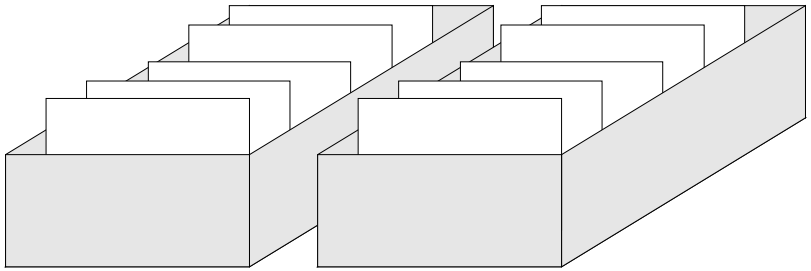
- EDV-Sachbearbeiter erledigen Aufgaben für menschliche Sachbearbeiter
- Sie erbringen Dienste
- Sie werden aus den Geschäftsprozessen, aus der Benutzungsschnittstelle, aus Batches beauftragt



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter haben Gedächtnis

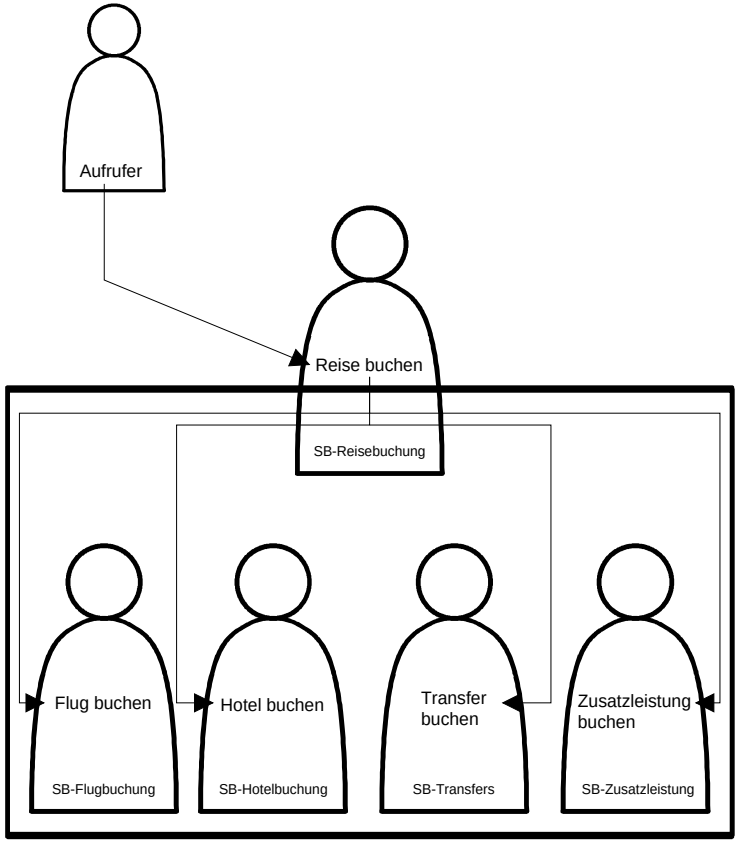
- Karteikästen bilden das Gedächtnis der Sachbearbeiter
- Jeder Sachbearbeiter darf nur in seinen Karteikästen schreiben
- Lesen darf er überall
  (mit Einschränkungen)



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter beauftragen sich gegenseitig

- Sachbearbeiter können sich Aufträge erteilen
- Aufträge, die direkt aus Geschäftsprozessen kommen, heißen initiale Aufträge oder Geschäftsvorfälle



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Die Dienste müssen genau spezifiziert werden

- Name des Dienstes
- Parameter
- Vor- und Nachbedingungen
- Effektbeschreibung

Auftrag ablehnen

Braucht	Einstellungsdatum	:	DATE
Braucht	Forderungsnummer	:	Forderungsnummer
Braucht	Begründung	:	Kommentar
Braucht	AutomatischerBrief	:	BOOL

Vorbedingung

Eine entsprechende Forderung ist existent.

Vorbedingung

Es handelt sich auch wirklich um einen Anspruch

Effekt

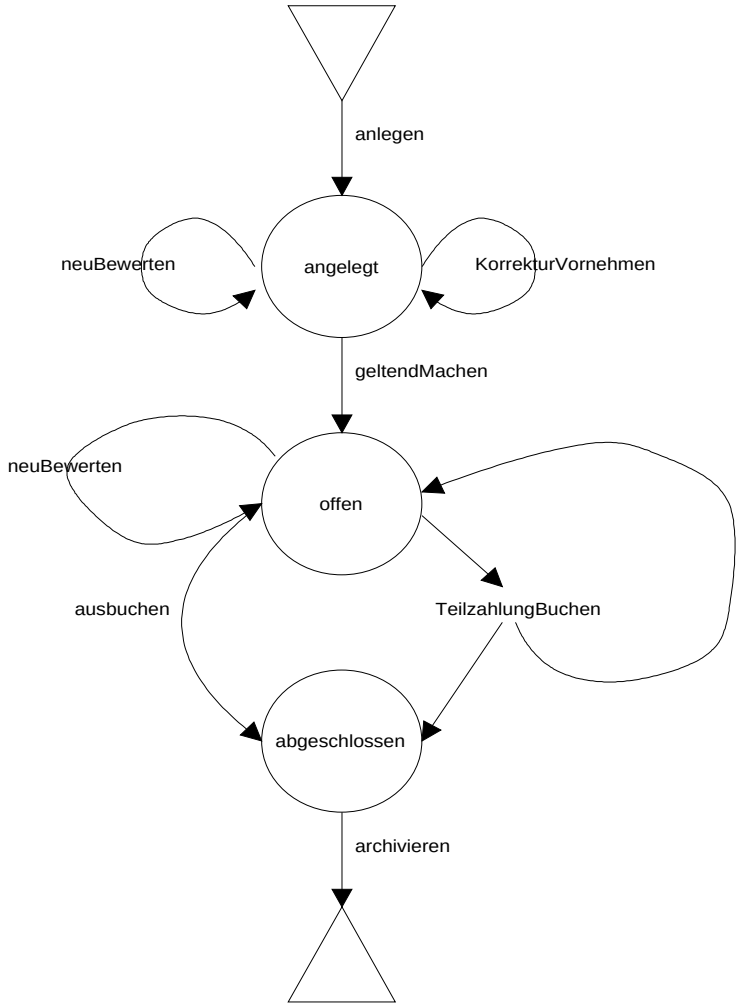
Es wird der Zustand abgelehnt eingenommen. Es wird durch das a Korrespondenzsystem (Sachbearbeiter Korrespondenz.BriefErstellen....) eine der Ablehnung erstellt, falls dies vom Sachbearbeiter gewünscht ist.

Die Begründung wird zusammen mit anderen organisatorischen Da *Forderung.Historie* festgehalten

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Die Dinge, die ein Sachbearbeiter verwaltet, haben ein Zustandsmodell

- Sachbearbeiter verwalten meist Dinge (zum Beispiel Regresse zu Schäden)
- Solche Dinge haben ein Zustandsmodell



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

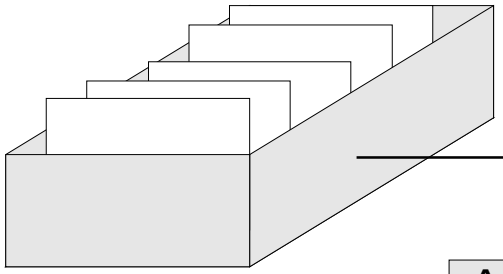
Zusammenhang Zustandsmodell und Dienste

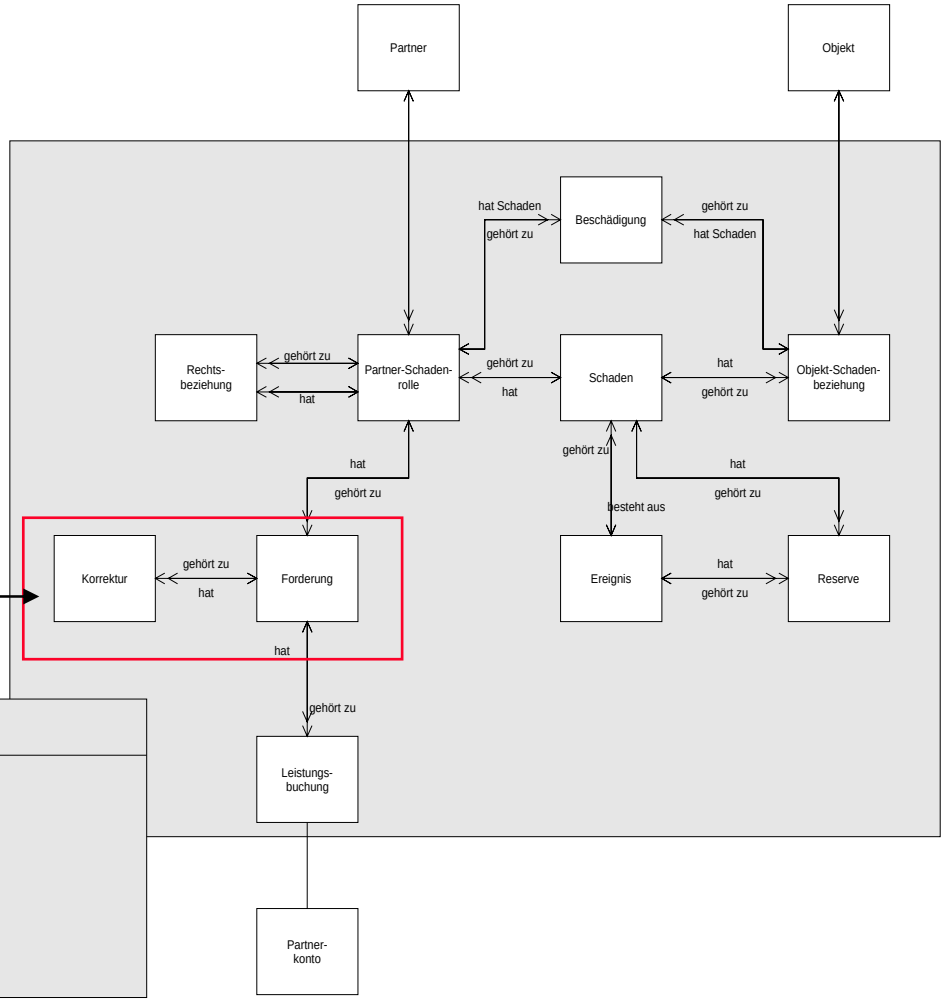
- Zustandsübergänge werden durch Aufträge bewirkt
- Man findet die Dienste oft über das Nachdenken über ein Zustandsmodell
- Es gibt auch mehr als einen Folgezustand
  Beispiel: Teilzahlung buchen

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Der Karteikasten ist eine Datensicht

Beispiel: Anspruch



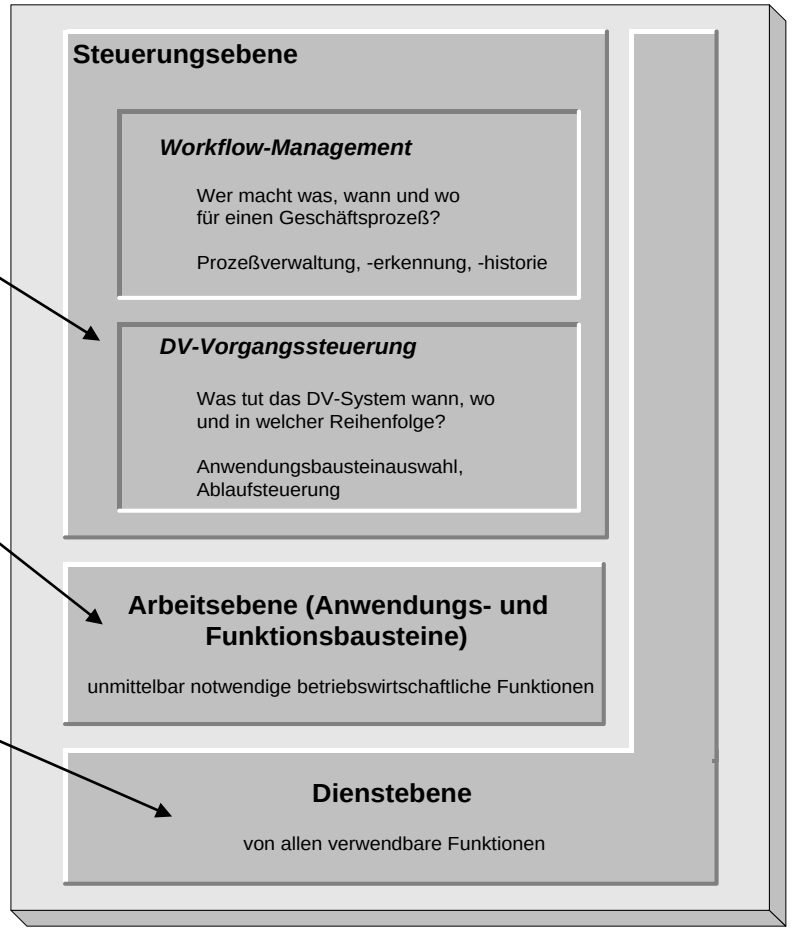


Anspruch
Korrekturen +200 -100

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter und zum Beispiel VAA-Architektur

- Die Aufträge kommen aus der DV-Vorgangssteuerung
- Sachbearbeiter bilden die Arbeitsebene der VAA-Architektur
- Das Datenmodell wird im Datenmanager verwaltet. (Dienstebene)

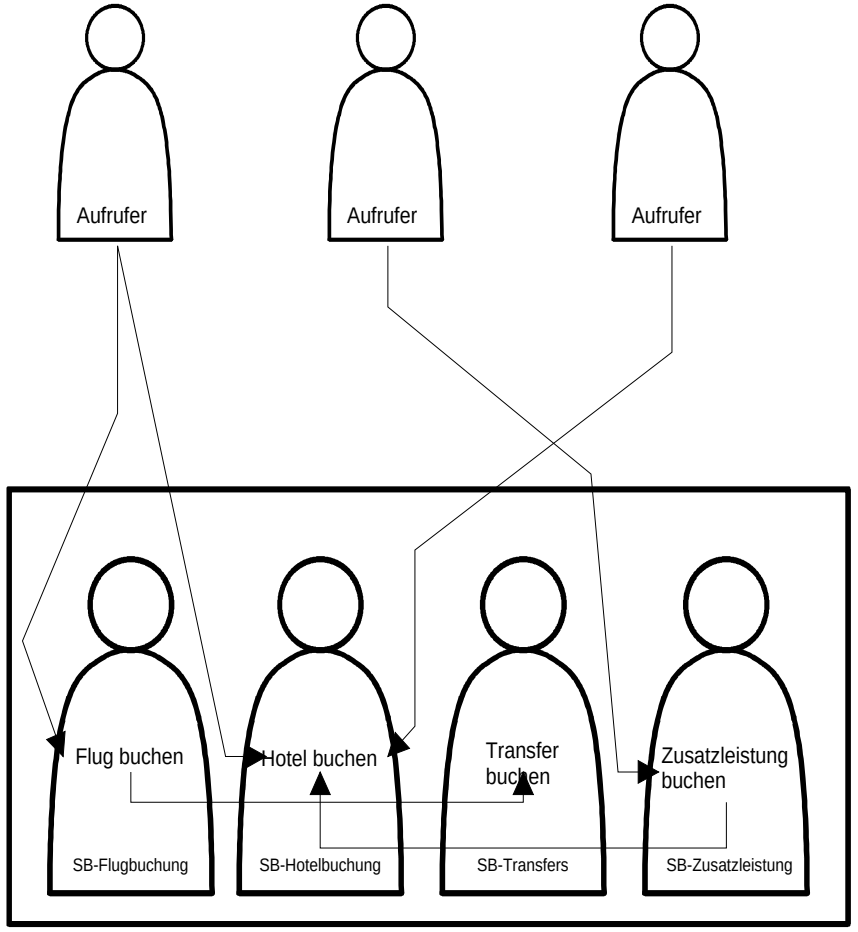


© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter und Bausteine: Warum ist Clusterung notwendig?

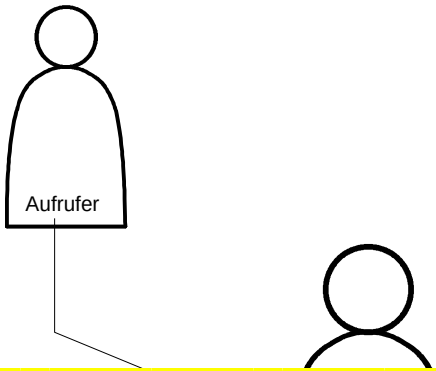
ohne Clusterung

Änderungen im Team brechen Organisation der Aufrufer



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter und VAA-Bausteine: Warum ist Clusterung notwendig?



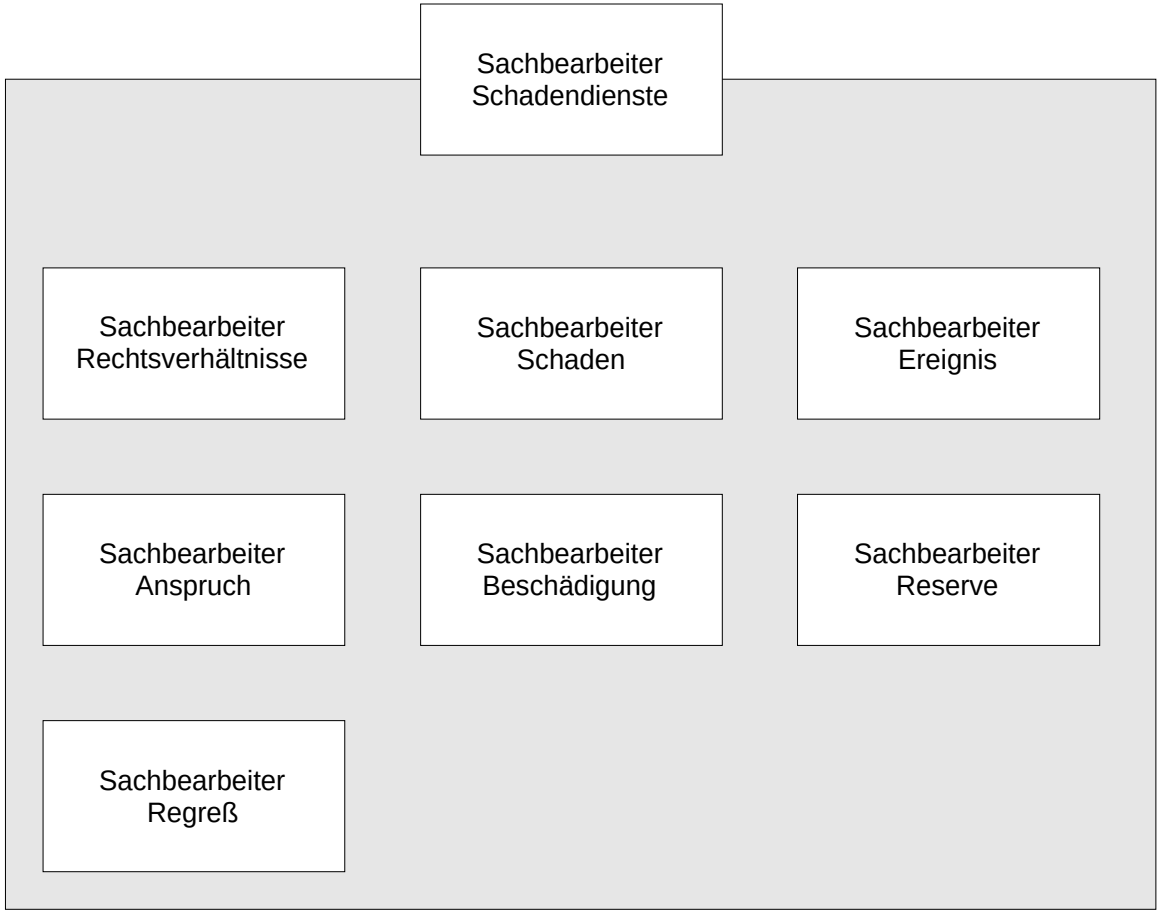
mit Clusterung

Haben Sie das Pattern erkannt?
Muster „Fassade“

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

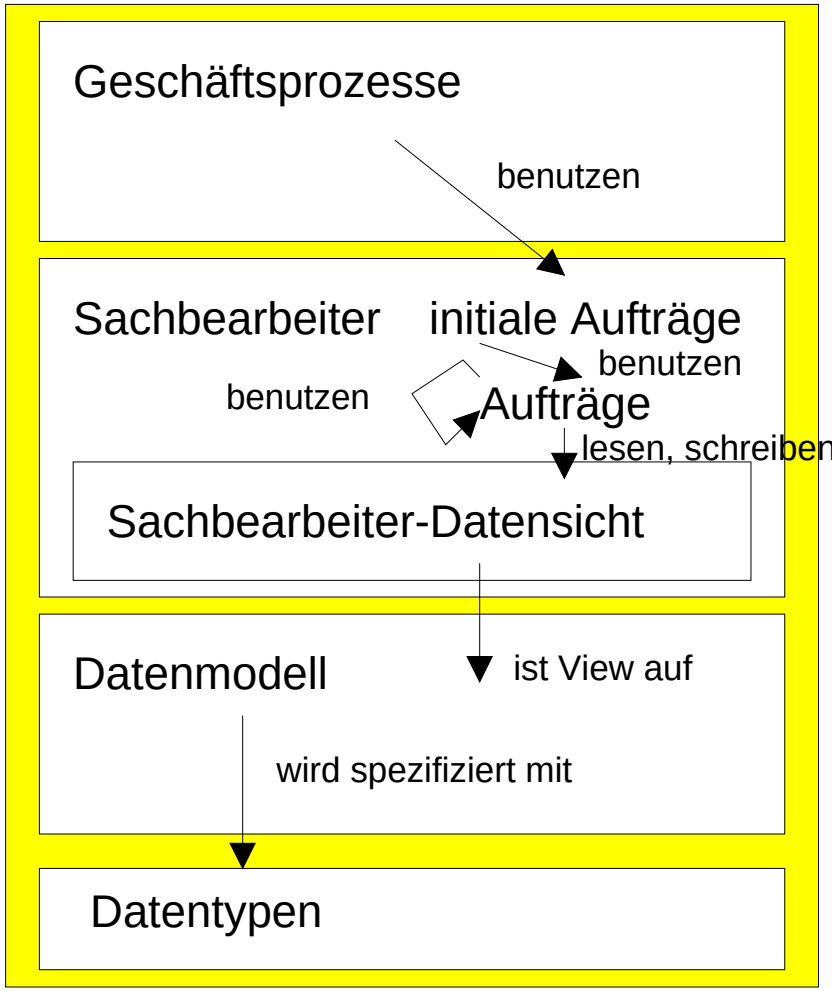
Sachbearbeiter und Anwendungsbausteine

Beispiel Schaden/Leistung



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter: Zusammenfassung



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Diskussion
Verwendbarkeit der Metapher

- Sachbearbeiter sind für die Spezifikation technischer Subsysteme nicht geeignet, Grund: Bruch der Metapher
- Sachbearbeiter eignen sich gut für fachliche Bausteine: Sachbearbeiter-Metapher fügt sich gut in "Sachbearbeitung" mit Workflows ein.
- Einordnung in eine Schichten oder J2EE-Architektur einfach
- Verträglich mit Datenzugriffsschichten
- konventionelle Datenmodellierung kann beibehalten werden
- Kapselung durch Sachbearbeiter erhöht Übersicht der Modelle

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Diskussion
Methodische Einordnung

Methode	SA/SD	Sachbearbeiter	Objektorientiert
Charakteristik	-Daten -Funktionen	+ Kapselung	+ Vererbung + Polymorphie
Einordnung		Soft-OO	Hard-OO
Verwendung		fachliche Spezifikationen (Versicherungs-logik)	Architektur-Spezifikationen (Technik)

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Beschreibung von Datenmodellen

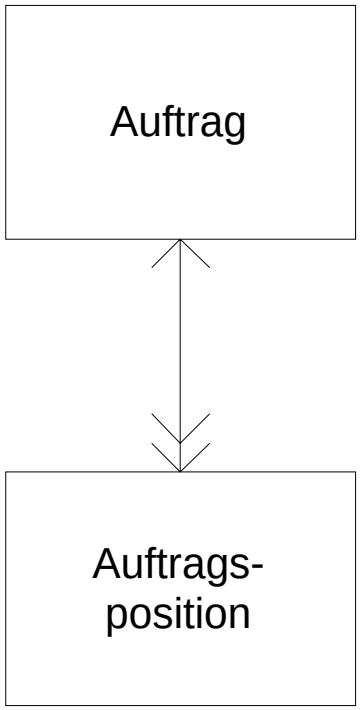
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Datenmodellierung und die Sachbearbeiter-Methode

- Datenmodellierung soll hier nicht grundlegend eingeführt werden
  - siehe Literatur
  - siehe diverse Datenmodellierungshandbücher
- Übersicht über Konstrukte
- Was man beachten sollte
- Datentypen

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

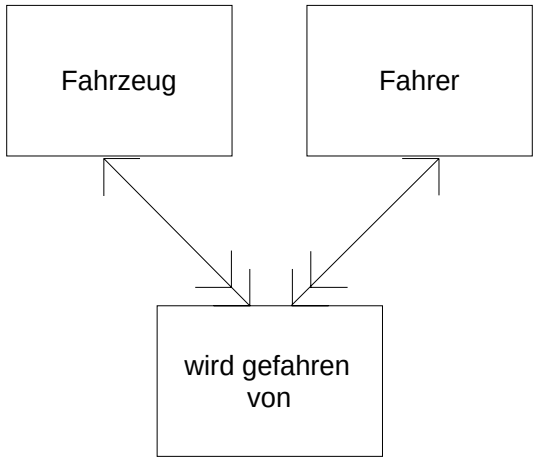
Übersicht über Konstrukte Datenmodell
Graphische Notation



1:n Beziehungen
einschließlich
1:0,n; 1:1 etc.

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

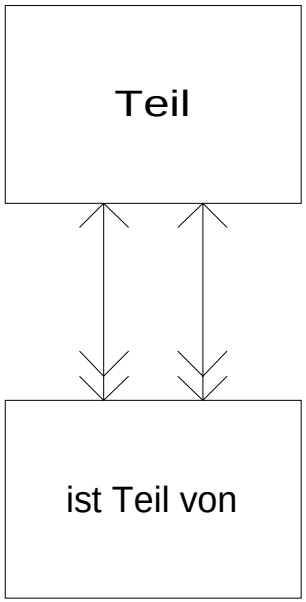
Übersicht über Konstrukte Datenmodell
Graphische Notation



n:m Beziehungen
einschließlich aller Varianten

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Übersicht über Konstrukte Datenmodell
Graphische Notation

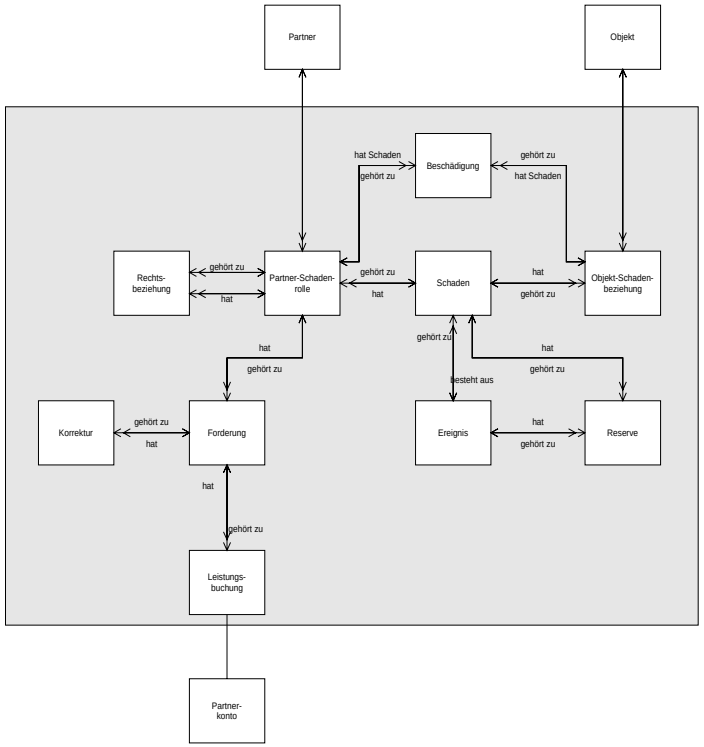


Selbstbeziehung
(is part of)

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Datenmodellierung
Was man beachten sollte

- Übersichten müssen textuell erläutert werden
  - jede Entität
  - jede Beziehung
  - Beispiele sind nicht schädlich, sondern hilfreich
- siehe "Beispielspezifikation zum Methodenhandbuch"



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Datenmodellierung
Elemente der textuellen Beschreibung

(Regel) *Entitätsbeschreibung* ::=

Entität *Identifikator*

Beschreibung

Freitext

Primärschlüssel

*Attributname**

{**Schlüsselkandidat**

*Attributname** } *

Attribute

(*Attributdeklaration* | *Beziehungsdeklaration*) +

(Regel) *Attributdeklaration* ::=

hat *AttributName* *Arität* *DatentypName*

Beschreibungstext

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Datenmodellierung
Was man beachten sollte

- Entitäten sollen ausführlich fachlich beschrieben werden.

Entität Schaden

Beschreibung

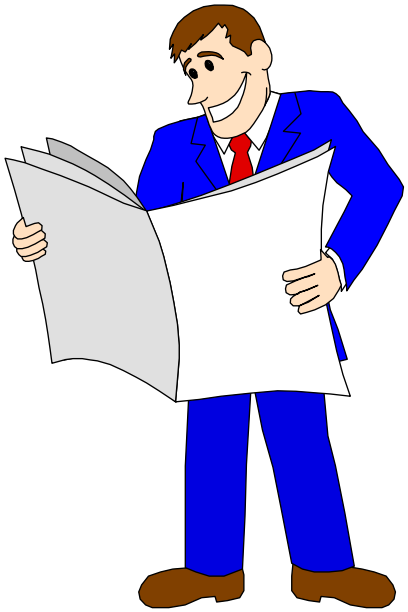
Zentrales Objekt des Bausteins Schaden/Leistung ist der ***Schaden***. Ein Schaden hat dabei selbst nicht weiter viele Eigenschaften oder Attribute. Man kann sich den Schaden eher als einen Anker vorstellen oder einen elektronischen Aktendeckel, unter dem man alle relevanten Informationen zu einem Schadensfall finden kann.

Insbesondere wird der Schaden hier nicht als elektronische Akte modelliert. Was der Sachbearbeiter also in seiner Schadensakte generell an Unterlagen hat, wird hier nicht weiter dargestellt.

Ein Schaden ist immer genau einer Sparte zugeordnet, die sich aus dem Vertrag oder dem Risiko ergibt. Ein Attribut Sparte wäre allerdings redundant, da es aus dem Risiko oder dem Vertrag entnommen werden kann. Wenn ein Schadensereignis mehr als eine Sparte betrifft, so sind entsprechend viele Schadenobjekte anzulegen. Beispiel hierfür ist ein Auffahrunfall eine Vollkasko versicherten Fahrzeuges. Der geschädigte Unfallgegner, dem der Versicherungsnehmer aufgefahren ist, wird als K-Haftpflichtschaden geführt. Der Schaden am Fahrzeug des Versicherungsnehmers wird als K-Kaskoschaden geführt. Es werden zwei Schäden angelegt, die mittels eines Ereignisses gruppiert werden können, damit Gutachter zum Beispiel nur einmal bestellt werden. .....

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Attribute werden durch einen Datentyp und eine Definition beschrieben



nicht so...

hat Schadendatum

eher so...

hat Schadendatum : DATE

Datum an dem der Schaden eingetreten ist. Dieses Datum ist relevant für die Prüfung der Deckung. Wenn am Schadendatum keine Deckung durch ein Risiko bestand, wenn der Vertrag also aus irgendwelchen Gründen am Schadendatum nicht deckend war, so kann eine Deckung abgelehnt werden.

Das Schadendatum ist im allgemeinen mit dem Datum der Schadensmeldung nicht identisch.

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Datentypkonzept

- Wozu ein Datentypkonzept?
- Elemente
  - Basistypen
  - davon abgeleitete Typen

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Warum ein Datentypkonzept?

üblich in der Datenmodellierung

Attribute

hat Kontonummer : PIC 9(5)
Ordnungsmerkmal für das Konto - künstlicher Schlüssel.

besser

Attribute

hat Kontonummer : KtoNummer
Ordnungsmerkmal für das Konto - künstlicher Schlüssel.

Und zusätzlich den Datentyp KtoNummer

Datentypen

DIE KtoNummer
IST EIN NAT(5)

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Basistypen

Datentyp	Bedeutung
BOOL	Wahrheitswerte, boolesche Werte mit zweiwertiger Logik. Konstanten **TRUE** und **FALSE**
INT(<n>)	Ganzzahl der Länge <n> mit Vorzeichen, zum Beispiel **INT(5)**
NAT(<n>)	Natürliche Zahl der Länge <n> ohne Vorzeichen, zum Beispiel **NAT(3)**
FIX(<n>,<m>)	BCD-Arithmetik mit <n> Vorkommastellen und <m> Nachkommastellen.
FLOAT(<n>)	Gleitkommazahl mit n-stelliger Genauigkeit.
STRING(<n>)	Zeichenkette mit <n> Zeichen Länge. Zum Beispiel **STRING(10)** für eine Zeichenkette der Länge 10. Konstante **EMPTY** für die leere Zeichenkette.
DATE	Datum der Form TT/MM/JJJJ
TIME	Zeit in der Form HH:MM:SS

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Abgeleitete Typen

- Aufzählungstypen
- Feldtypen
- Einschränkungstypen
- Verbundtypen

Beispiele siehe Handbuch und Musterspezifikation

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Modellierung von Geschäftsprozessen

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Beschreibung von Geschäftsprozessen

- Sinn und Unsinn der Geschäftsprozessmodellierung
  - Wann macht GP-Modellierung Sinn?
  - Was ist ein Arbeitsschritt? Was ist ein Prozess/Teilprozess?
  - Beispiel
- Die graphischen Beschreibungselemente
- Beschreibung als Text
- Relevanz für Schaden/Leistung

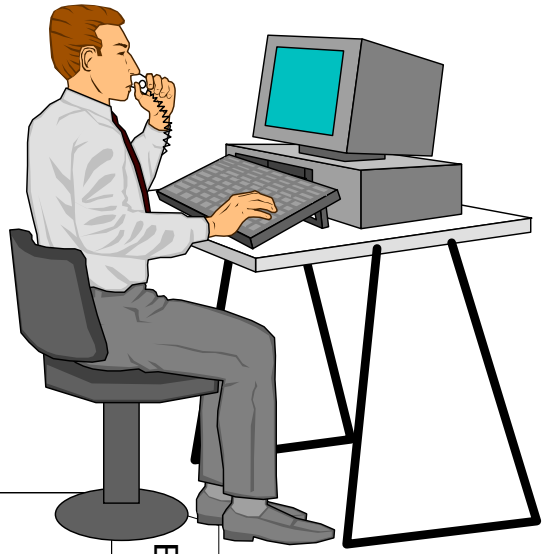
© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

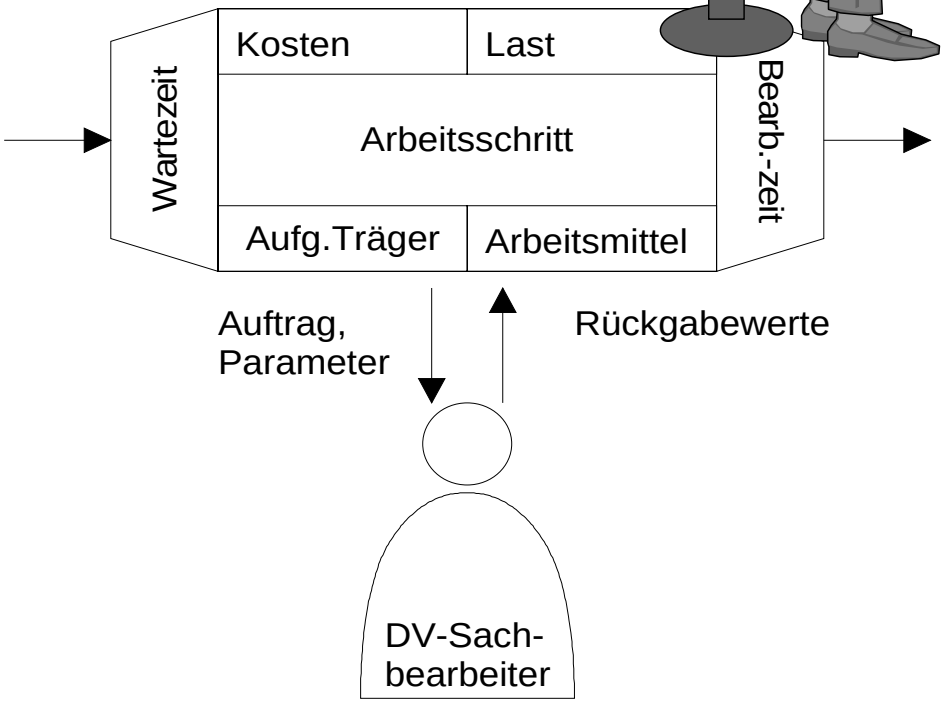
Ziele GP-Modellierung

- Allgemein: Denken in Vorgängen
  - über Abteilungsgrenzen hinweg.
  - über Grenzen von Anwendungssystemen hinweg.
  - Unterstützung Reengineering.
  - Kostenrechnung, Organisation, weiteres.....

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Zusammenhang: Geschäftsprozesse EDV-Sachbearbeiter





© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sinn und Unsinn
Geschäftsprozessmodell NEIN für ...

- Prozess "Sicherung des Geschäftserfolges"
- Prozess "Strategische Informationssystemplanung"

- Kriterien:
  - keine Unterstützung durch EDV
  - nicht strukturierbar
  - kaum maschinell unterstützbar

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sinn und Unsinn
Geschäftsprozessmodell JA für ...

- Mehrstufige Bearbeitung von Immobilienkrediten

- Kriterien:
  - Man kann sich eine Akte vorstellen, die durch die Organisation wandert
  - Die Akte besucht verschiedene menschliche Sachbearbeiter
  - Die Bearbeitung ist strukturierbar
  - Die Bearbeitung wird teilweise durch EDV unterstützt

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Abgrenzung

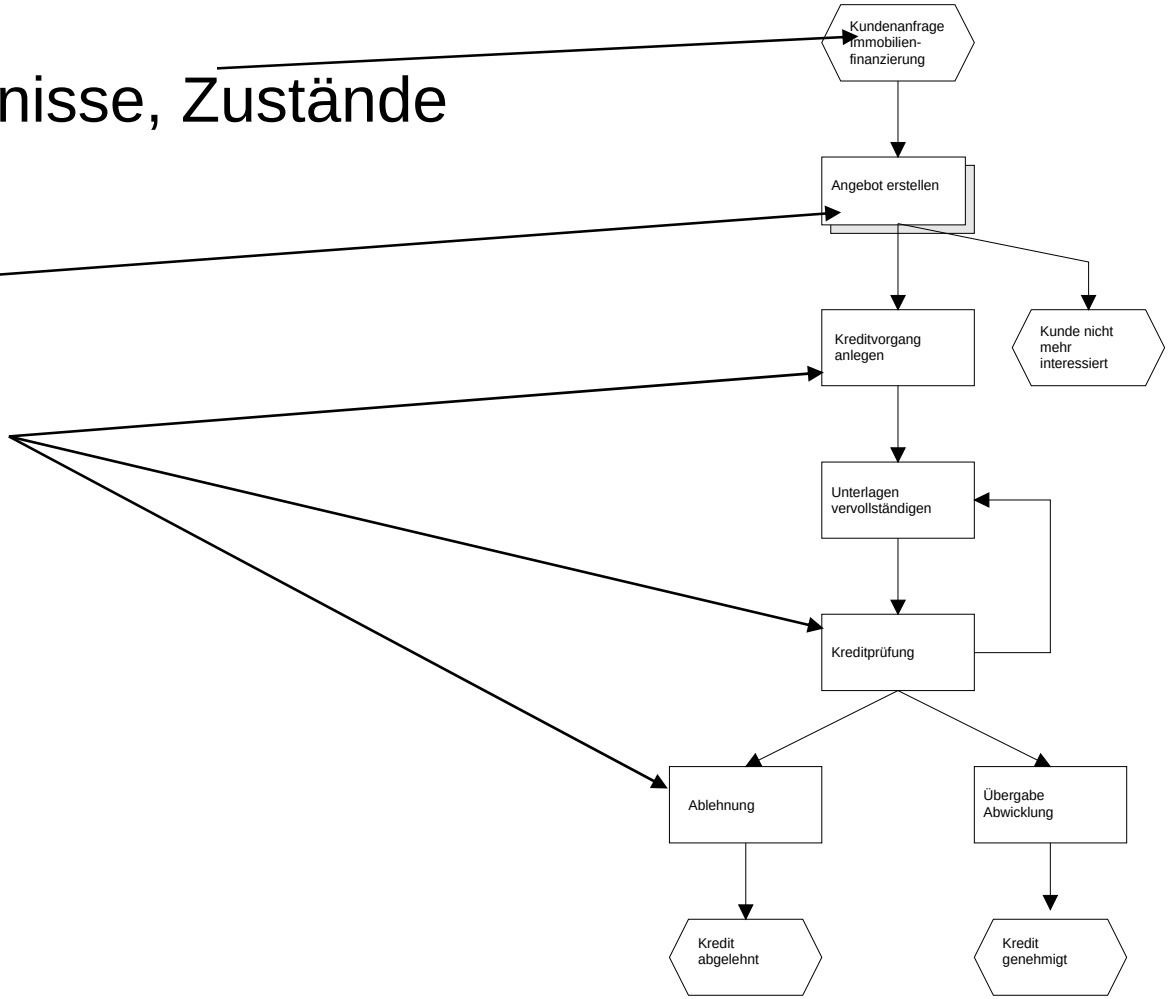
- Wann ist es ein
  - Geschäftsprozess?
  - Teilprozess?
  - Arbeitsschritt?

- Kriterien:
  - Geschäftsprozesse werden in keinem anderen Prozess als "Teilprozess" verwendet
  - Arbeitsschritte
    - Wechsel des Bearbeiters
    - Wechsel des Bearbeitungsmittels
    - Wechsel der Phase: Planung, Durchführung, Kontrolle
  - siehe betriebswirtschaftliche Organisationslehre

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Graphische Beschreibungselemente

- Ereignisse, Ergebnisse, Zustände
- Teilprozesse
- Arbeitsschritte



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Arbeitsschritt Kreditvorgang anlegen

Auslöser

Bemerkung: Auslöser ist hier der Abschluß des vorhergehenden Arbeitsschrittes - ein solch „trivialer Auslöser“ sollte besser gar nicht erst beschrieben werden.

Bearbeiter Kreditsachbearbeiter

Arbeitsmittel EDV-System Kreditbearbeitung

Beschreibung

Ein Angebot ist erstellt - der Kunde möchte das Angebot annehmen. Die Eckdaten des Angebotes werden im Kreditbearbeitungssystem erfaßt (**Sachbearbeiterauftrag** Kreditvorgang.erfassen).

Arbeitsschritt Unterlagen vervollständigen

Bearbeiter Kreditsachbearbeiter

Arbeitsmittel Scanner, EDV-System Kreditbearbeitung

Beschreibung

Abhängig von den Eckdaten ermittelt das Kreditbearbeitungssystem die benötigten Unterlagen, die von Wertgrenzen abhängen. (**Sachbearbeiterauftrag** Kreditvorgang.UnterlagenErmitteln).

Unterlagen werden vom Kunden eingereicht und mit dem **Sachbearbeiterauftrag** Kreditvorgang.UnterlageZuordnen dem Vorgang zugeschrieben. Sind alle Unterlagen vorhanden, so erfolgt eine Freigabe mit dem **Sachbearbeiterauftrag** Kreditvorgang.UnterlageZurPrüfungWeiterreichen.

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

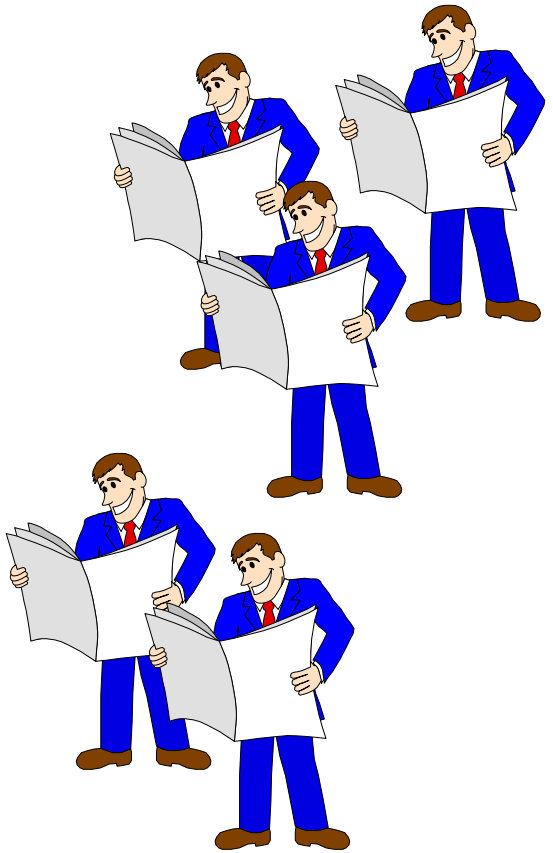
Eine Mustergliederung für Spezifikationen fachlicher Anwendungsbausteine

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Mustergliederung - Grundidee

- Fachliche Beschreibung
  - Anforderungen
  - Lösungsalternativen
  - Beschreibung des Fachgebietes
  - Szenarien für die Benutzung

- "formale" Beschreibung
  - Geschäftsprozesse
  - Sachbearbeitermodell
  - Datenmodell
  - Datentypen



unterschiedliche Leser

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Mustergliederung

Mustergliederung

- Überblick
  - fachlicher Teil
  - "formaler Teil"

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Verbindlichkeit der Mustergliederung

- Konformität zu einer Mustergliederung macht noch keine gute Spezifikation
  - Qualitätssicherung ist wichtiger
  - Formalismus ersetzt kein Nachdenken über die Inhalte
  - Es gibt "formal richtige" Spezifikationen, die eine miserable Qualität haben
- Die Mustergliederung ist ein Leitfaden
  und eine (unvollständige) Checkliste
- Die Mustergliederung muss nicht immer 1:1 eingehalten werden
- Gute Beispiele sind wichtiger

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Mustergliederung
Überblick "Fachlicher Teil"

- 1 Überblick und Abgrenzung des Anwendungsbausteins
- 1.1 Fachlicher Überblick
- 1.2 Abgrenzung

- 2 Fachliche Beschreibung des Anwendungsbausteins
- 2.1 Wesentliche Anforderungen
- 2.2 Lösungsalternativen
- 2.3 Beschreibung des Bausteins

- 3 Szenarien

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Mustergliederung
Überblick "Formaler Teil"

- 4 Formale Beschreibung

- 4.1 Geschäftsprozesse
- 4.1.1 Geschäftsprozess <Beispiel 1>
- 4.1.n Geschäftsprozess <Beispiel n>

- 4.2 Sachbearbeitermodell
- 4.2.1 Sachbearbeiter <Beispiel 1>
- 4.2.n Sachbearbeiter <Beispiel n>

- 4.3 Batches
- 4.3.1 Batch <Batch 1>
- 4.3.n Batch <Batch n>

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Mustergliederung
Überblick "Formaler Teil" (Forts.)

- 4.4 Datenmodell
- 4.4.1 Überblick
- 4.4.2 Entität <Entität 1>
- 4.4.n Entität <Entität n>

- 4.5 Datentypen

- 4.6 Offene Schnittstellen

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Die Sachbearbeitermethode und OO

In der Sachbearbeiter-Methode stecken viele Patterns – ohne dass das 1990 bewusst war

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Die Sachbearbeitermethode und OO

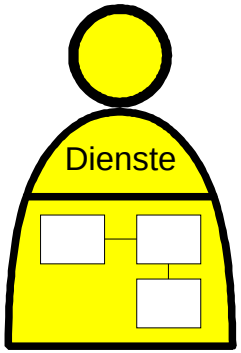
- Hartes OO enthält zusätzlich noch Vererbung und Polymorphie ..

- Implementierungsvererbung in „fachlichen Systemen“ wurde inzwischen als „schädlich“ beschrieben – siehe auch J2EE Patterns
- Polymorphie, sofern sie „fachlich“ ist, fällt ebenfalls darunter

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Objektorientierte Methoden: Beispiel UML

- UML: Grafikorientiert
  - Anwendungsfälle (Use Cases)
  - Sequenzdiagramm:
    Modellierung objektübergreifender Abläufe
  - plus Beschreibung der Objekttypen
    - und pro Objekt Zustandsdiagramm:
      Modellierung objektinterner Abläufe
  - Typenmodell



© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004

Sachbearbeiter sind nicht von gestern ☺
Man findet darin folgende J2EE Patterns

- Man kann den Sachbearbeiter auch als Business Delegate bezeichnen
  - http://developer.java.sun.com/developer/restricted/patterns/BusinessDelegate.html
- Die Fassade, die wir gesehen haben, heißt dort Session Facade
  - http://developer.java.sun.com/developer/restricted/patterns/SessionFacade.html
- Aggregate Entity entspricht der Datensicht eines Sachbearbeiters
  - http://developer.java.sun.com/developer/restricted/patterns/AggregateEntity.html
- Die Aufträge kann man mit Value Objects implementieren
  - http://developer.java.sun.com/developer/restricted/patterns/ValueObject.html

© 2004 Hans Hartmann, Wolfgang Keller, all rights reserved, Universität Leipzig, SS2004