OLDENBURGER FORSCHUNGS- UND ENTWICKLUNGSINSTITUT
FÜR INFORMATIK-WERKZEUGE UND -SYSTEME

# Eine Infrastruktur für den Austausch von Fachkomponenten

5. Workshop
„Komponentenorientierte Betriebliche Anwendungssysteme“
Augsburg, 25./26. Februar 2003


Holger Jaekel
holger.jaekel@offis.de

Thorsten Teschke
thorsten.teschke@offis.de


## Gliederung

- Einleitung: OFFIS-Projekt KOSOBAR
- Vorgehensmodell
- Systemarchitektur
- Broker: Geschäftsprozessorientierte Komponentensuche
- Zusammenfassung

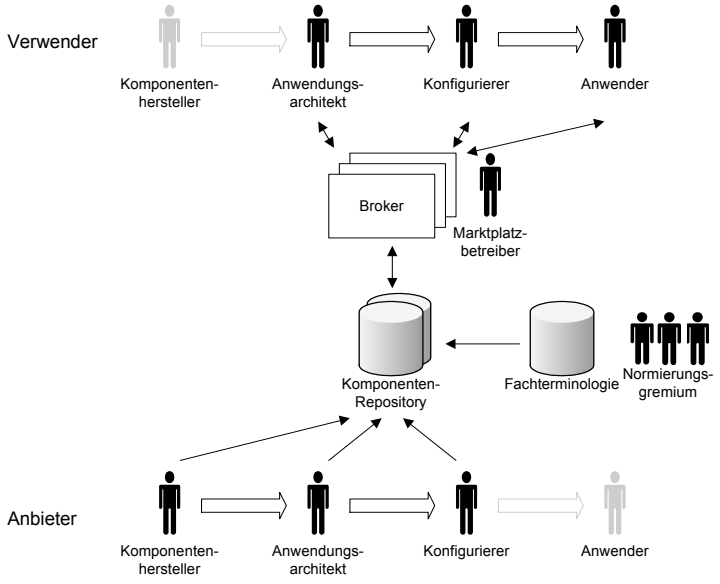
## KOSOBAR: Übersicht

- OFFIS-Projekt KOSOBAR:
  Komponentenbasierte Softwareentwicklung auf Basis von Referenzmodellen
- Zielsetzung: Entwicklung von Sprachen, Methoden und Werkzeugen zur Unterstützung komponentenbasierter Softwareentwicklungsprozesse über internetbasierte Komponentenmärkte
- Zentraler Gedanke:
  - „Komponentenorientierte Softwarereferenzmodelle“
  - Kommunikation zwischen Marktteilnehmern mittels standardisierter, fachlich orientierter Komponentenbeschreibungen

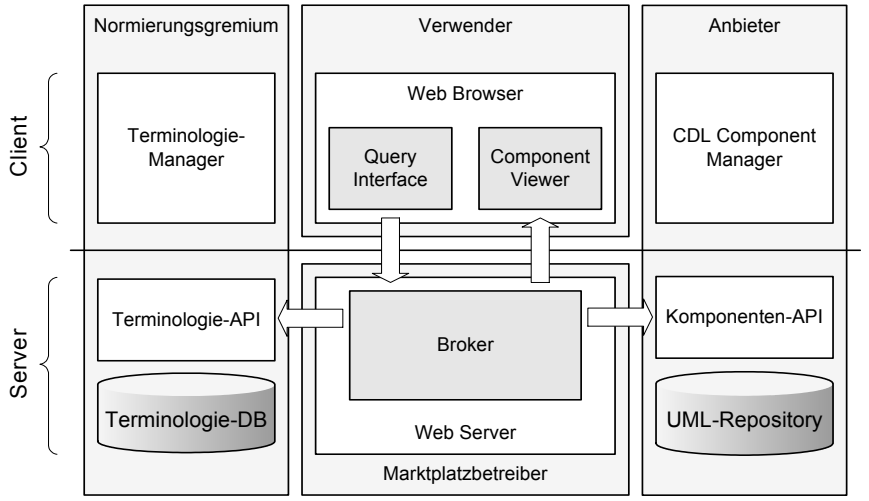
- Parallele Arbeit zu Memorandum zur Vereinheitlichung der Spezifikation von Fachkomponenten



## KOSOBAR: Vorgehensmodell



## KOSOBAR: Systemarchitektur






## Komponentensuche: Beobachtung

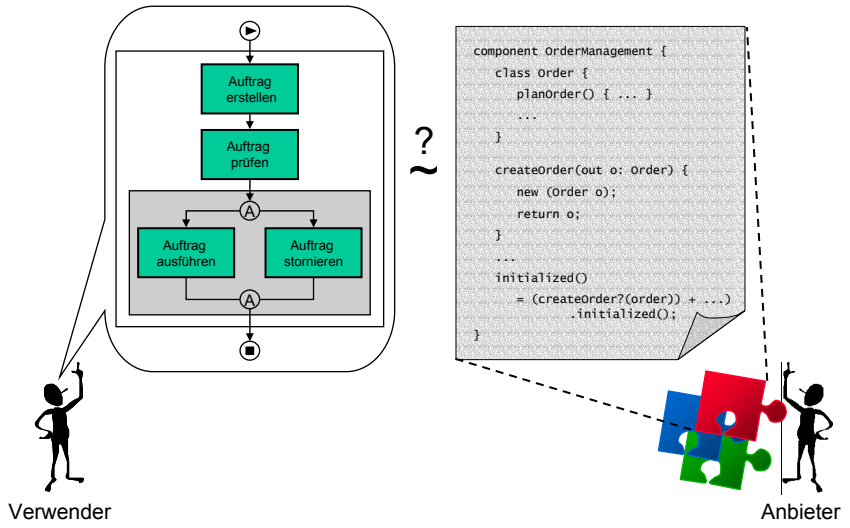
- Einsatz verschiedener Modellierungstechniken in der Analysephase komponentenbasierter Entwicklungsprojekte
- Geschäftsprozessmodelle beinhalten umfangreiches Wissen über die fachlichen Anforderungen eines Unternehmens:
  - Aktivitäten
  - Organisationsstruktur
  - Datenstrukturen

  } Prozesse

➢ Wissen wird bislang nicht in ausreichendem Maße für die Suche nach Komponenten eingesetzt

➢ Vielfach Ausrichtung der Geschäftsprozesse eines Unternehmens an Fähigkeiten der Software erforderlich

## Geschäftsprozessorientierte Komponentensuche






## Kernkonzept: Substituierbarkeit

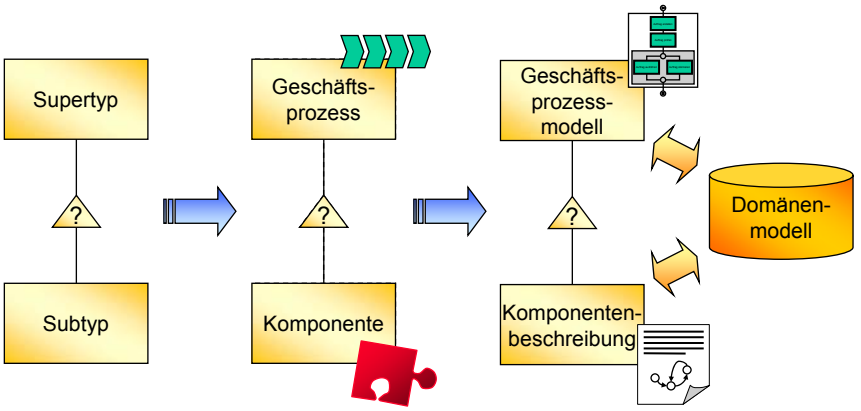
- Komponentensuche: Ist die (Wieder-)Verwendung einer betrachteten Komponente in gegebenem Kontext möglich?
- Wiederverwendbarkeit in objektorientiertem Paradigma ist eng mit Frage der Substituierbarkeit verknüpft
- Prinzip der Substituierbarkeit stützt sich auf *Subtyping*:

> *„An instance of a subtype can always be used in any context in which an instance of a supertype was expected.“*
> *[Wegner/Zdonik, 1988]*

- „Konventionelles“ Subtyping betrachtet nur Signaturen
- Aber: Typen können spezifisches Verhalten definieren, das ihre (Wieder-)Verwendbarkeit einschränkt
- *Behavioural Subtyping* berücksichtigt neben Signaturen auch das Verhalten der Typen

## Komponentensuche als Subtyping-Problem

Übertragung des Behavioural Subtyping auf die geschäftsprozessorientierte Komponentensuche:





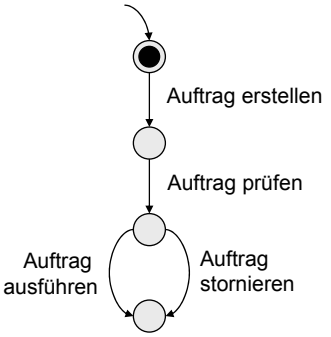

## Ansatz: Zwei Teilprobleme

| *Interoperabilitätsebene* | *Subtyping-Ansatz* | *Form der Berücksichtigung* |
|---|---|---|
| Signaturebene | „Konventionelles“ Subtyping (Signature Matching) | — |
| Protokollebene | Behavioural Subtyping (verhaltensorientiert) | • beschriftete Transitionssysteme zur Repräsentation des Verhaltens von Geschäftsprozessen und Komponenten<br>• Subtyping-Relation auf Transitionssystemen |
| Semantikebene | Behavioural Subtyping (zustandsbasiert) | • Normsprachlicher Ansatz zur Definition der Semantik von Geschäftsprozessmodellen und Komponenten auf Grundlage domänenspezifischer Terminologien<br>• Spezialisierungsrelation auf normsprachlichen Aussagen |

*„An instance of a subtype can always be used in any context in which an instance of a supertype was expected.“*
*[Wegner/Zdonik, 1988]*

## Protokollebene: Geschäftsprozessmodelle

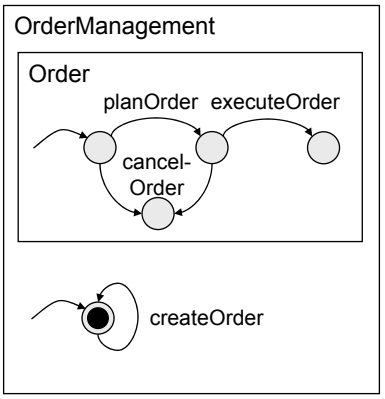
- Spezifizierung von Geschäftsprozessmodellen: *Lineare Prozessmodelle*
- Formale Grundlage: Prozessalgebra
- Konzentration auf Ablauforganisation: Formale Semantik linearer Prozessmodelle als beschriftete Transitionssysteme



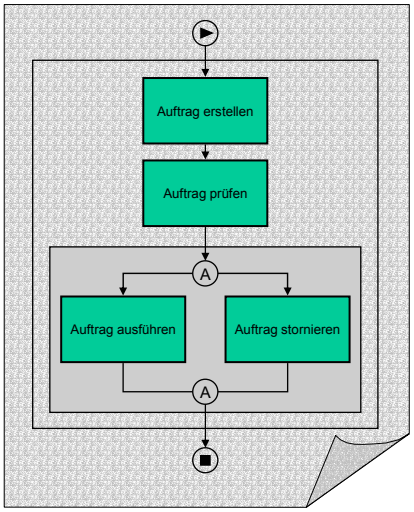



## Protokollebene: Komponentenbeschreibungen

- Spezifizierung von Komponenten: *Component Description Language (CDL)*
  - Abstraktion der Komponentenmodelle EJB, COM und CCM
  - integrierte Spezifikation von Struktur und Verhalten
  - Verhalten von Komponenten und Klassen: Interaktionsprotokolle
  - Referenzierung neu erzeugter bzw. bestehender Objekte
- Formale Semantik der Interaktionsprotokolle als beschriftete Transitionssysteme



## Beispiel:
## Simulation eines Geschäftsprozessmodells

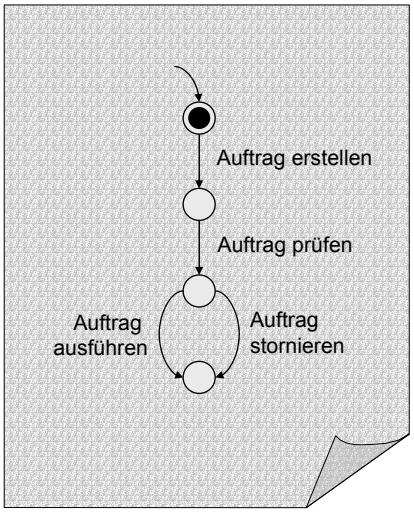



```
component OrderManagement {
   class Order {
      planOrder() { ... }
      ...
   }

   createOrder(out o: Order) {
      new (Order o);
      return o;
   }
   ...
   initialized()
      = (createOrder?(order)) + ...)
        .initialized();
}
```



## Beispiel:
## Simulation eines Geschäftsprozessmodells

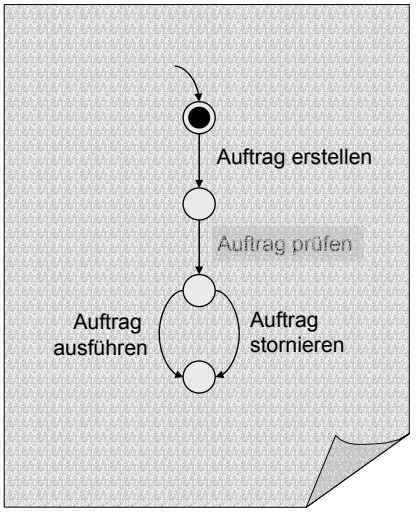


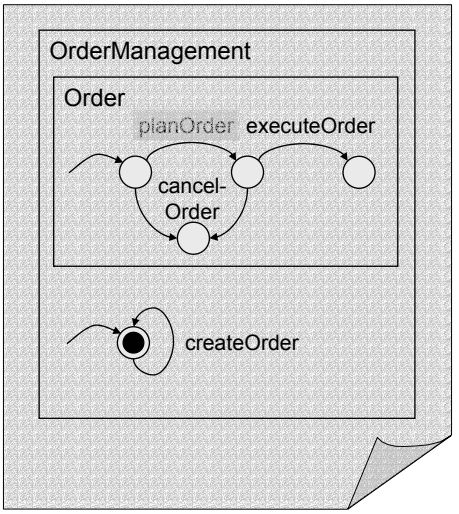

```
component OrderManagement {
   class Order {
      planOrder() { ... }
      ...
   }

   createOrder(out o: Order) {
      new (Order o);
      return o;
   }
   ...
   initialized()
      = (createOrder?(order)) + ...)
        .initialized();
}
```

Jaekel / Teschke Eine Infrastruktur für den Austausch von Fachkomponenten 13 / 21

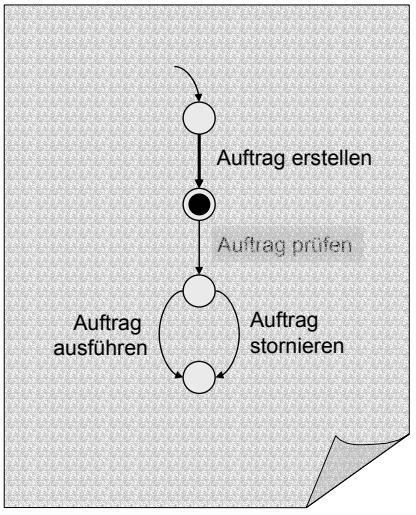
## Beispiel:
## Simulation eines Geschäftsprozessmodells

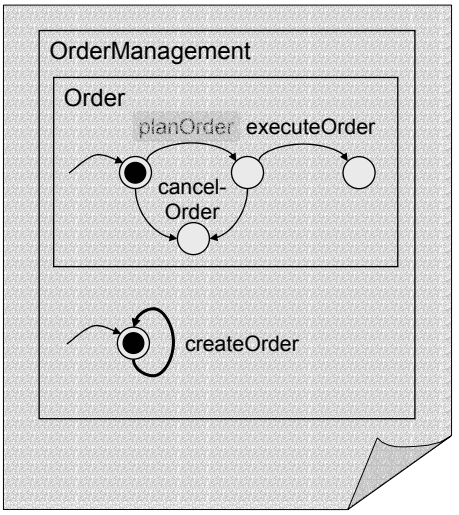


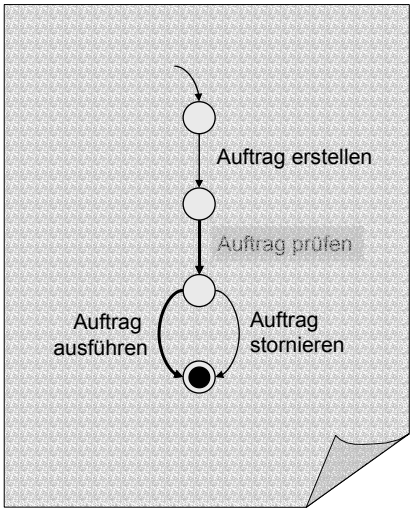


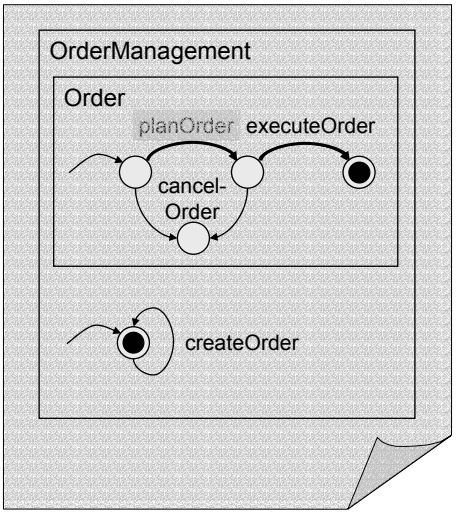

## Beispiel:
## Simulation eines Geschäftsprozessmodells



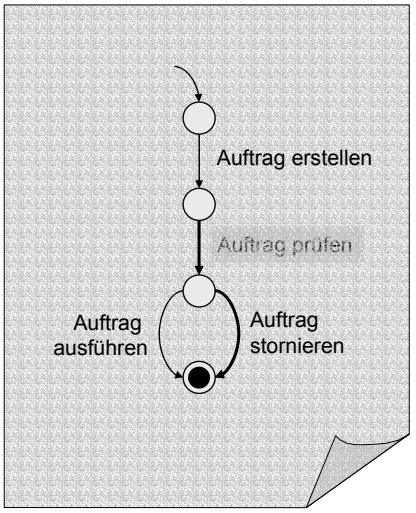
## Beispiel:
## Simulation eines Geschäftsprozessmodells

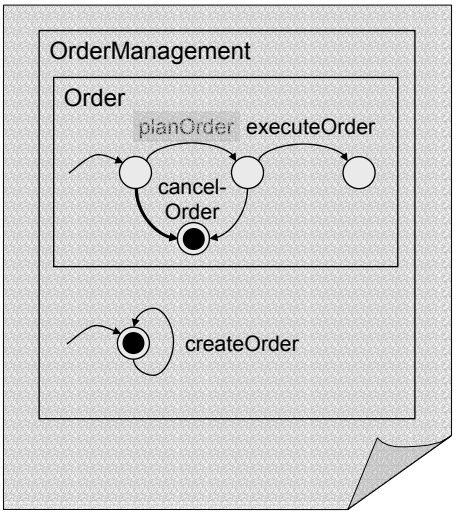




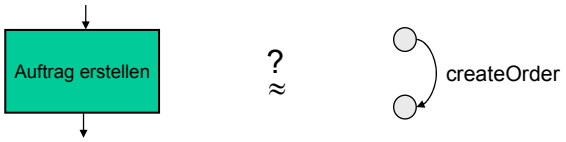

## Beispiel:
## Simulation eines Geschäftsprozessmodells



## Semantikebene: Repräsentation von Semantik

- Syntaktischer Vergleich von Geschäftsprozessmodellen und Komponenten (*Signature Matching*) nicht sinnvoll:

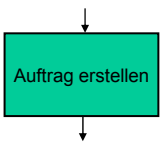



  ⇒ Vergleich der Semantik erforderlich
- Zustandsbasiertes Behavioural Subtyping: Formale Spezifikation der Semantik von Methoden über Vor- und Nachbedingungen
- Aber: Formale Spezifikationen nicht dem Fachexperten zugänglich, daher nicht geeignet für Erstellung von Geschäftsprozessmodellen



## Semantikebene: Aktivitäten und Methoden als Sprachhandlungen

Hypothese: Ausführungen von Aktivitäten und Methoden lassen sich als Handlungen verstehen, die durch natürlichsprachliche Äußerungen beschrieben werden können




*„Ein Angestellter erstellt einen Auftrag auf Basis eines Angebots“*

```
component OrderManagement {
     ...
     createOrder
          (in offer: Offer,
           out order: Order) {
          ...
     }
}
```

*„OrderManagement erstellt einen Auftrag auf Basis eines Angebots“*

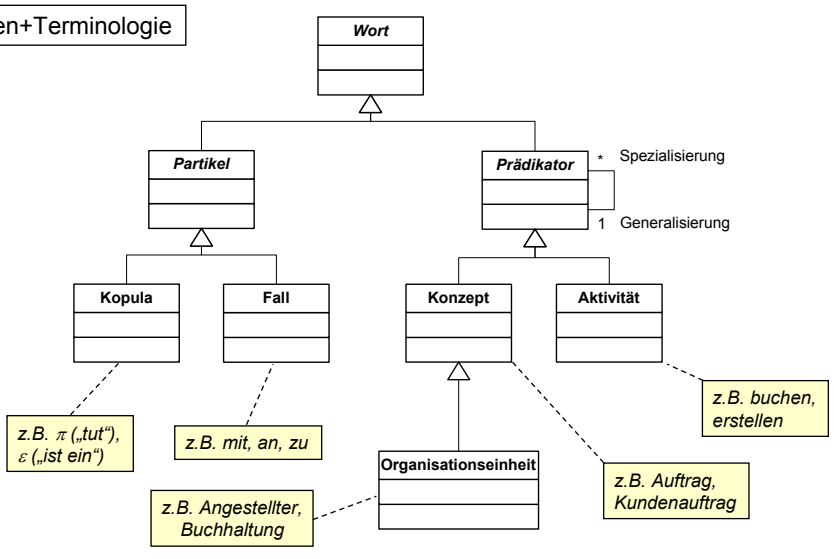
## Semantikebene: Normsprachliche Spezifikation

- *Normsprachen* (*Orthosprachen*):
  - Ansatz für den fachlichen Systementwurf [Wedekind 92, Ortner 97]
  - Ursprung in konstruktiver Wissenschaftstheorie: Charakterisiert durch methodische Rekonstruktion der in Anwendungsbereichen verwendeten „Gebrauchssprache“
  - Beseitigung von „Sprachdefekten“ natürlicher Sprache wie z.B. Unschärfen, Mehrdeutigkeiten oder Widersprüchen
  - formaler Teil: Wortarten + Satzbaupläne (Grammatik)
  - materialer Teil: Terminologie (Lexikon)
- Ansatz:
  Formulierung der Semantik von Geschäftsprozessmodellen (Aktivitäten) und Komponenten (Methoden) durch einfache normsprachliche Sätze über domänenspezifischer Terminologie



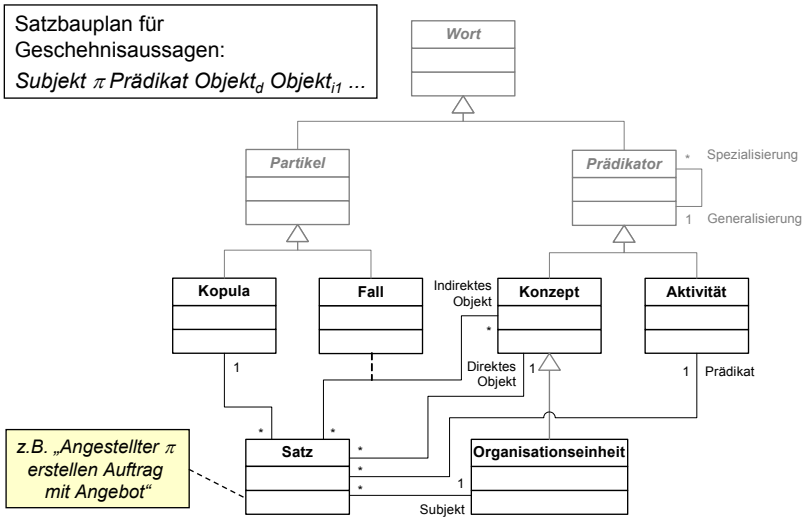
## Semantikebene: Normsprache

Wortarten+Terminologie



## Semantikebene: Normsprache

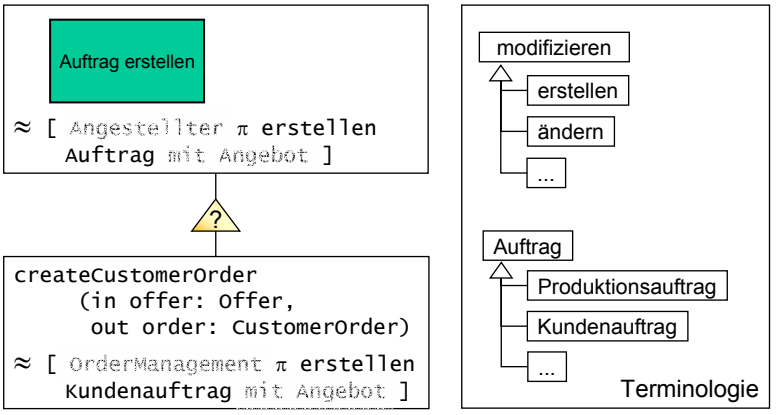
Satzbauplan für
Geschehnisaussagen:
*Subjekt π Prädikat Objekt$_d$ Objekt$_{i1}$ ...*



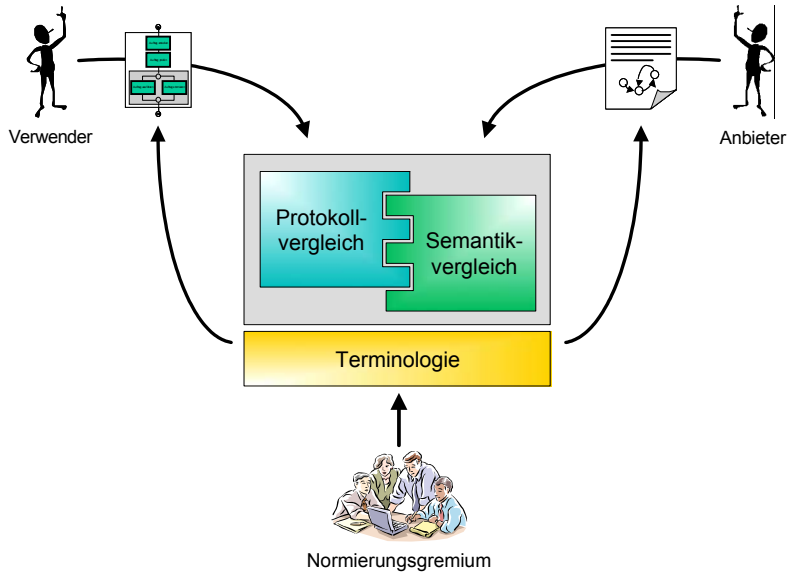



## Semantikvergleich: Spezialisierungsrelation

„Subtyp“-Beziehung auf semantischer Ebene:
Spezialisierungsbeziehung zwischen normsprachlichen Sätzen



## Geschäftsprozessorientierte Komponentensuche im Überblick






## Zusammenfassung

- Ziel von KOSOBAR:
  Entwicklung von Sprachen, Methoden und Werkzeugen zur Unterstützung komponentenbasierter Softwareentwicklungs-prozesse über internetbasierte Komponentenmärkte
- Prototypische Implementierung:
  - Komponentenrepository auf Basis der UML
  - Werkzeug zur Komponentenverwaltung
  - Werkzeug zum Terminologiemanagement
  - Komponenten-Broker
- Evaluierung:
  - Fallstudie: Kommunale Datenverarbeitung in der Kfz-Zulassung
  - Vergleich von Geschäftsprozessmodellen mit den Komponenten, die für ein entsprechendes Anwendungssystem entwickelt wurden