


L. Kern-Bausch

Fachhochschule Augsburg

M. Jeckle

Daimler-Benz Forschung Ulm

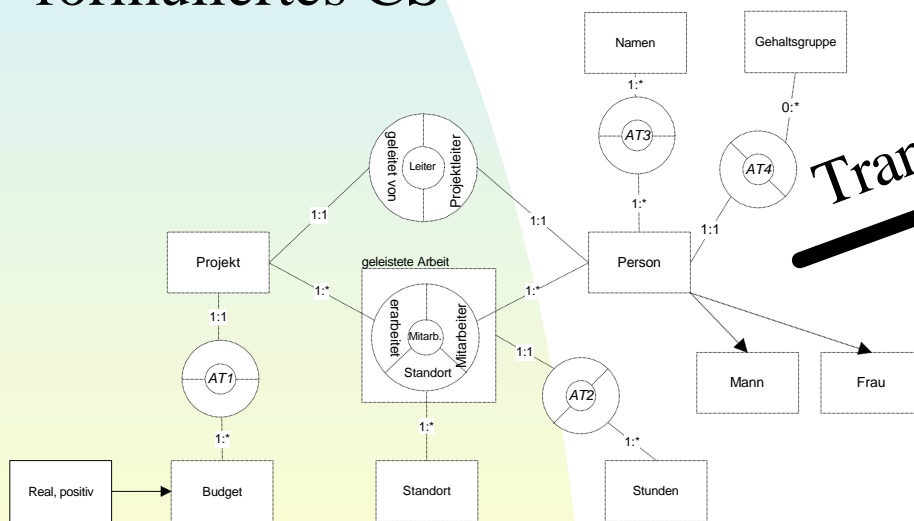
---



# Ableitung eines UML- Modells aus einem semantisch irreduzibel formulierten konzeptuellen Schema

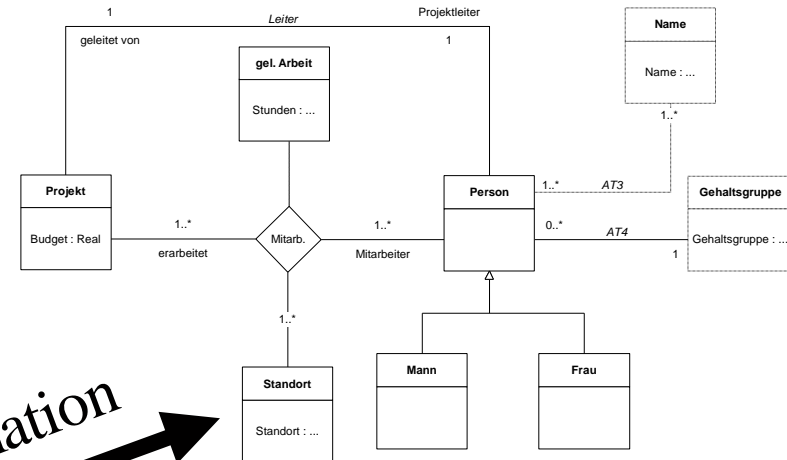
# Vom konzeptuellen Schema (CS) zum UML-Klassenmodell

Ergebnis der Informationsanalyse (IA):  
Semantisch irreduzibel  
formuliertes CS



Beschreibung der Informations-  
struktur der Applikation

Transformation



- Klassen
- Attribute
- Operationen
- Assoziationen, Dependencies
- Stereotypes, Constraints
- Konsistenzbedingungen (OCL)

# *Warum CS, und nicht gleich UML?*

- Enthält die gesamte relevante Information, insbesondere Regeln
- Ist ausgehend vom Informationsbereich sukzessive verfeinerbar
- IA-Beschreibungsmittel orientieren sich an der natürlichen Sprache
- IA integriert anerkannte Modellierungsmethoden (wie ER, EER, NIAM, ORM)
- aus CS  $\Rightarrow$ 
  - Klassenstrukturdiagramm und
  - konsistenzgarantierende dynamische Konstrukte
  - ... algorithmisch ableitbar

# Konzepte der Informationsanalyse

Semantisch irreduzible Formulierung des konzeptuellen Schemas

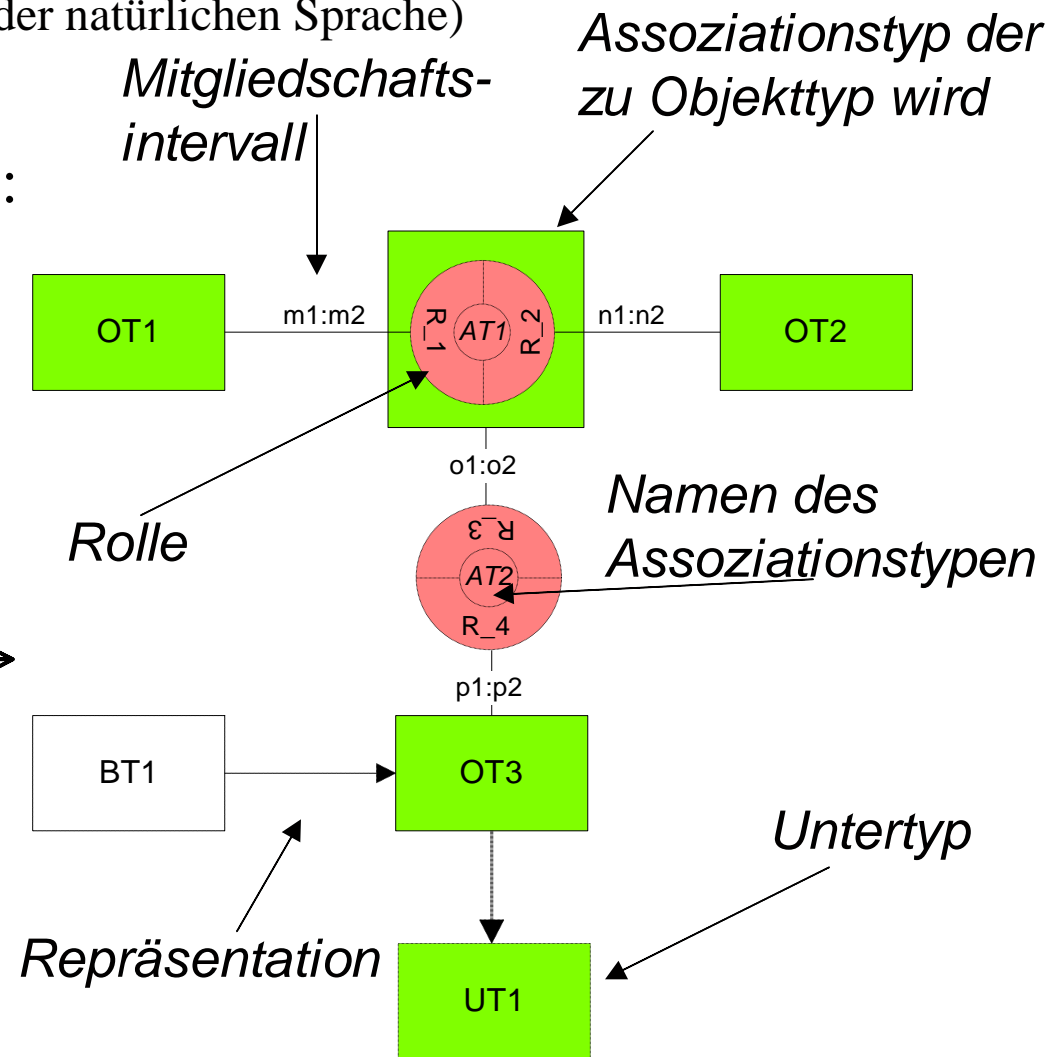
(Orientiert an der natürlichen Sprache)

Strukturelle Konzepte,  
Grundeinheiten der Information:

- Objekttypen (OT)
- Assoziationstypen (AT)
- Dependenztypen

Graphische Visualisierung →  
und  
Sprache (ISDL):

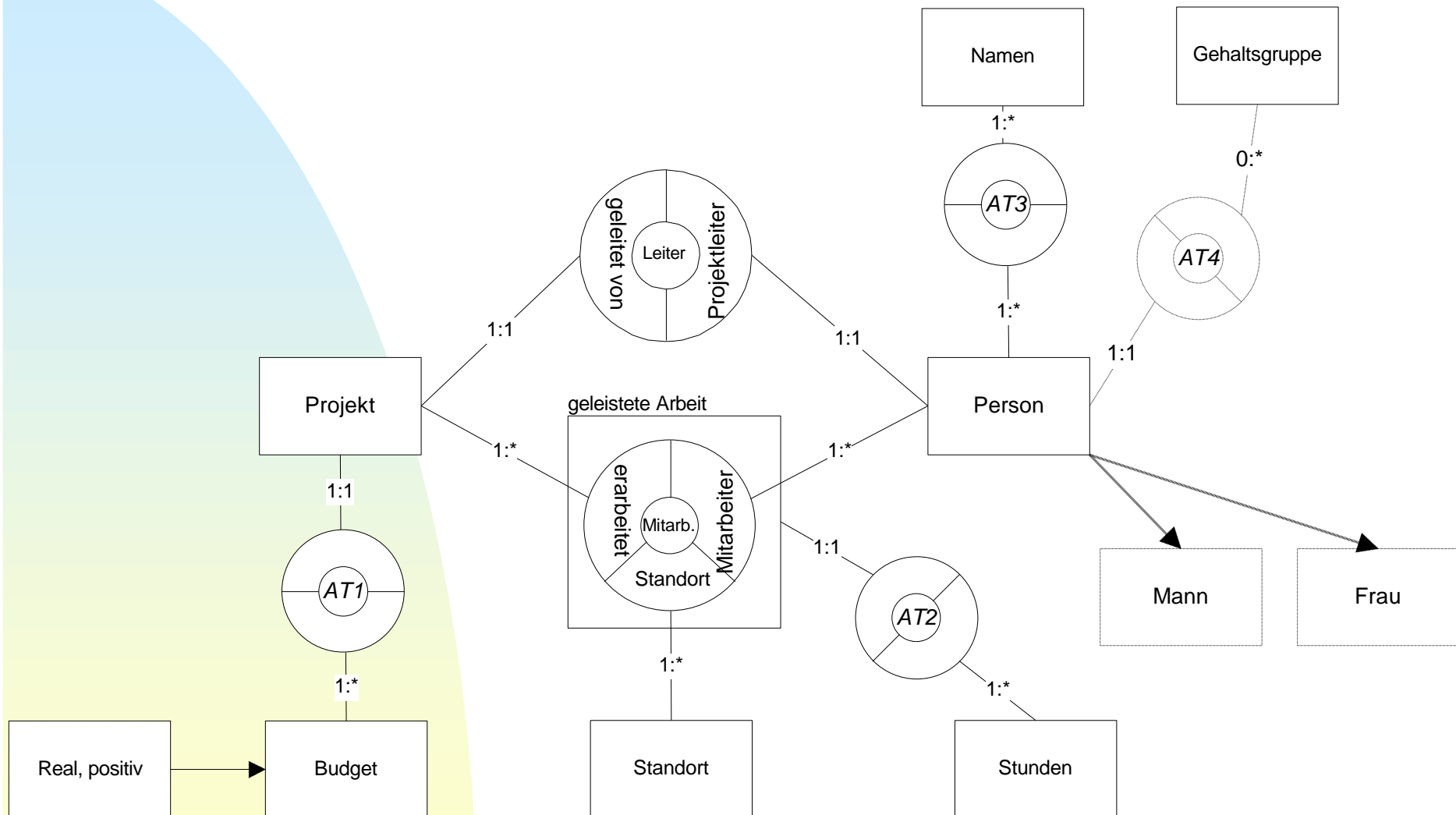
```
OBJECT (TYPE OT3, REPRESENTED_BY BT1);  
ASSOC ( TYPE AT2 ,  
        COMP ( OT3 R_4 INTERVAL (p1,p2)),  
        COMP ( AT1 R_3 INTERVAL (o1,o2)) );
```



# Information Structure Description Language (ISDL)

```
Schema Header {
    CONCEPTUAL_SCHEMA ( Firma / Alt_Firma, 0.0 );
    #doc
        Beschreibung
        =====
    doc#
Objekt- typ {
    OBJECT ( TYPE Person , REPRESENTED_BY String );
    #doc
        *** Standard Description Object ***
    doc#
Objekt- typ {
    OBJECT ( TYPE Gehaltsgruppe, REPRESENTED BY REAL(5,2));
    #doc
        *** Standard Beschreibung Objekt ***
    doc#
Assoz.- typ {
    ASSOC ( TYPE Bezahlung,
            COMP ( Gehaltsgruppe INTERVAL (0,*)),
            COMP ( Person INTERVAL (1,1)) );
    #doc
        *** Standard Beschreibung Assoziation ***
    doc#
    END_SCHEMA ;
```

## Schritt 1

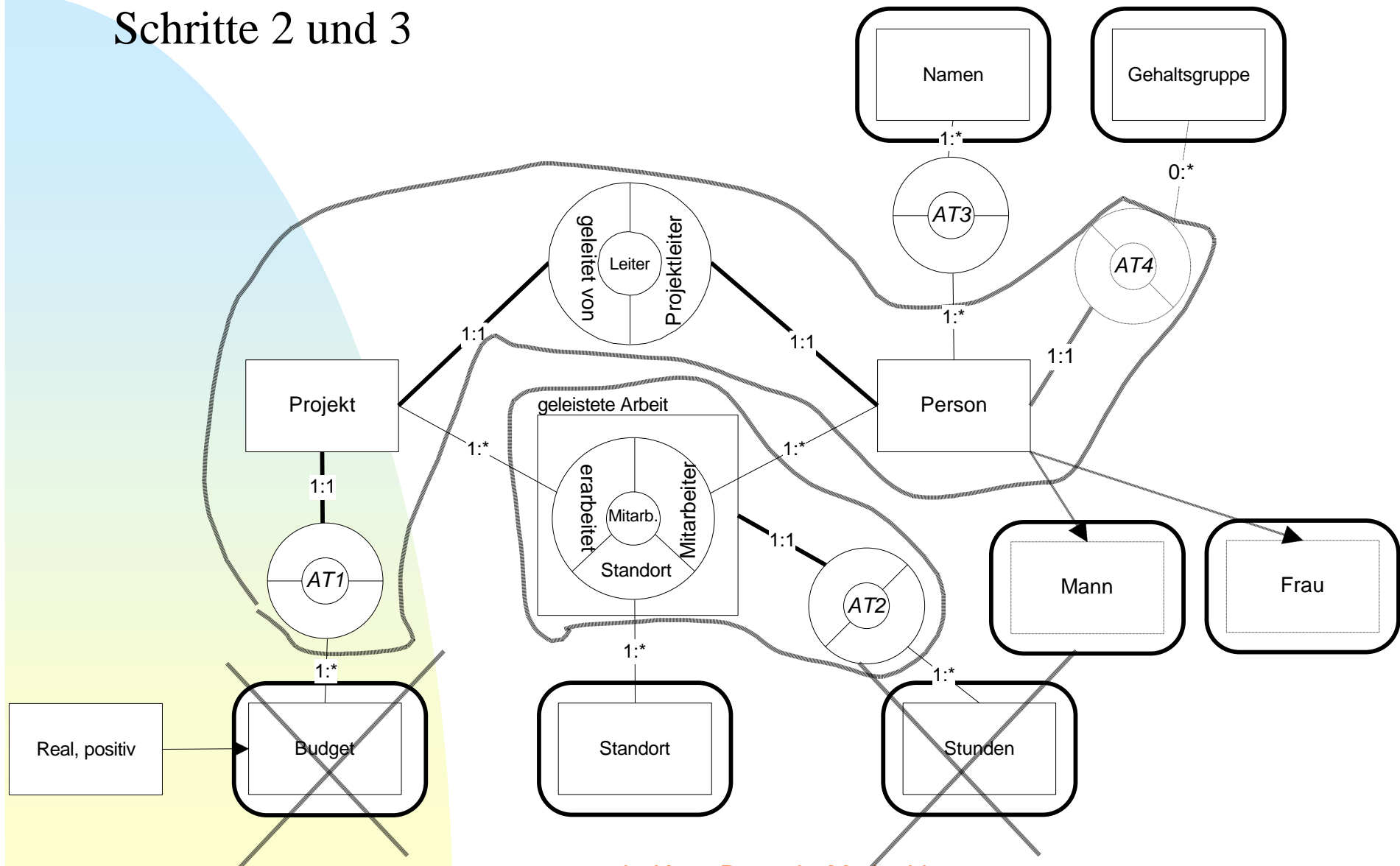


# Warum semantisch irreduzible Formulierung?

- Unterstützt Vollständigkeit und Widerspruchsfreiheit
- erleichtert Modellierung komplexer Zusammenhänge
- ist stabil hinsichtlich Schema-Änderungen
- ist notwendig zur Ableitung relationaler DB-Strukturen
- Stabilität hinsichtlich Schema-Änderungen (Änderbarkeit)
- ermöglicht die Gewinnung eines optimalen Klassenmodells

# Der Transformationsalgorithmus

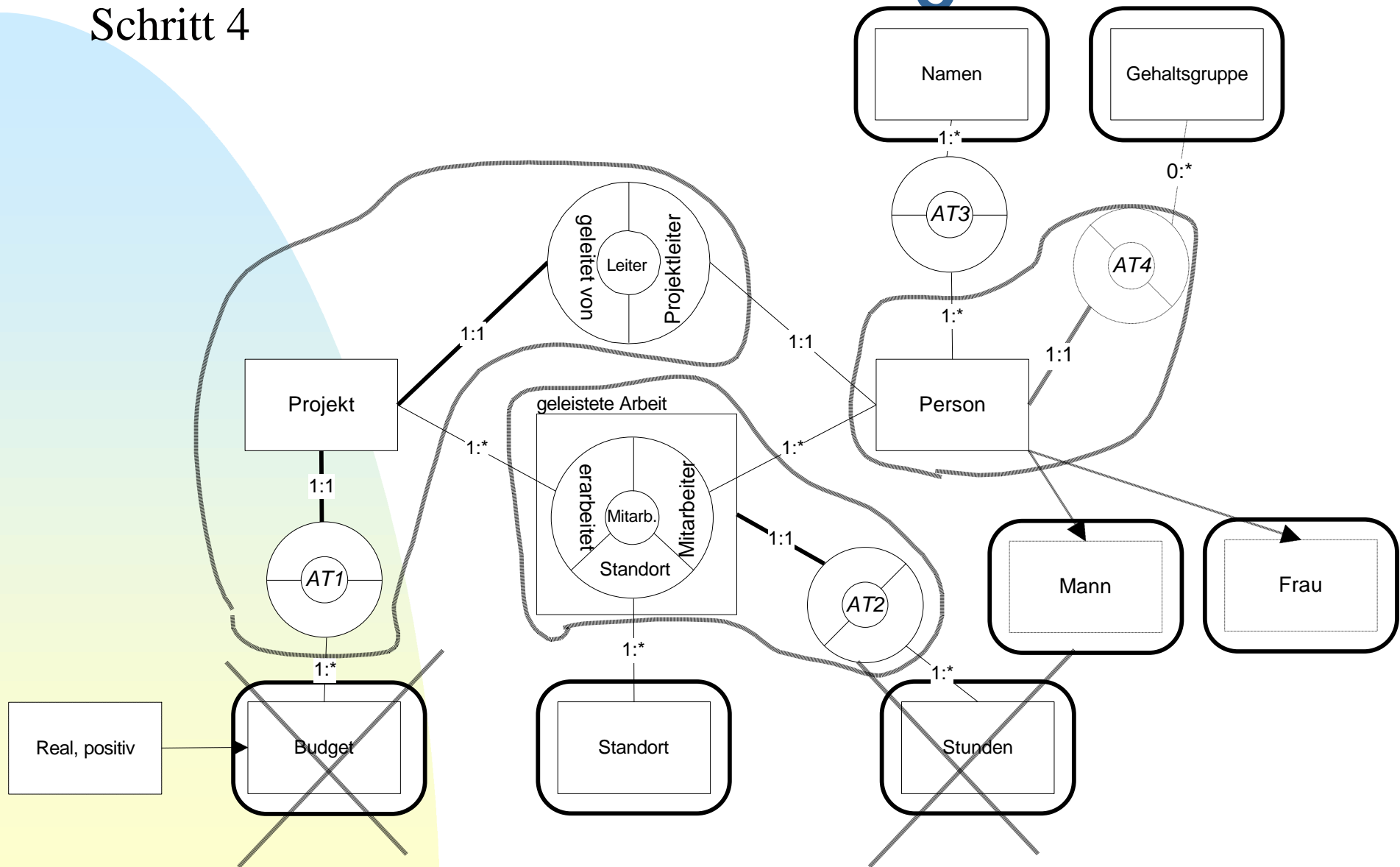
## Schritte 2 und 3





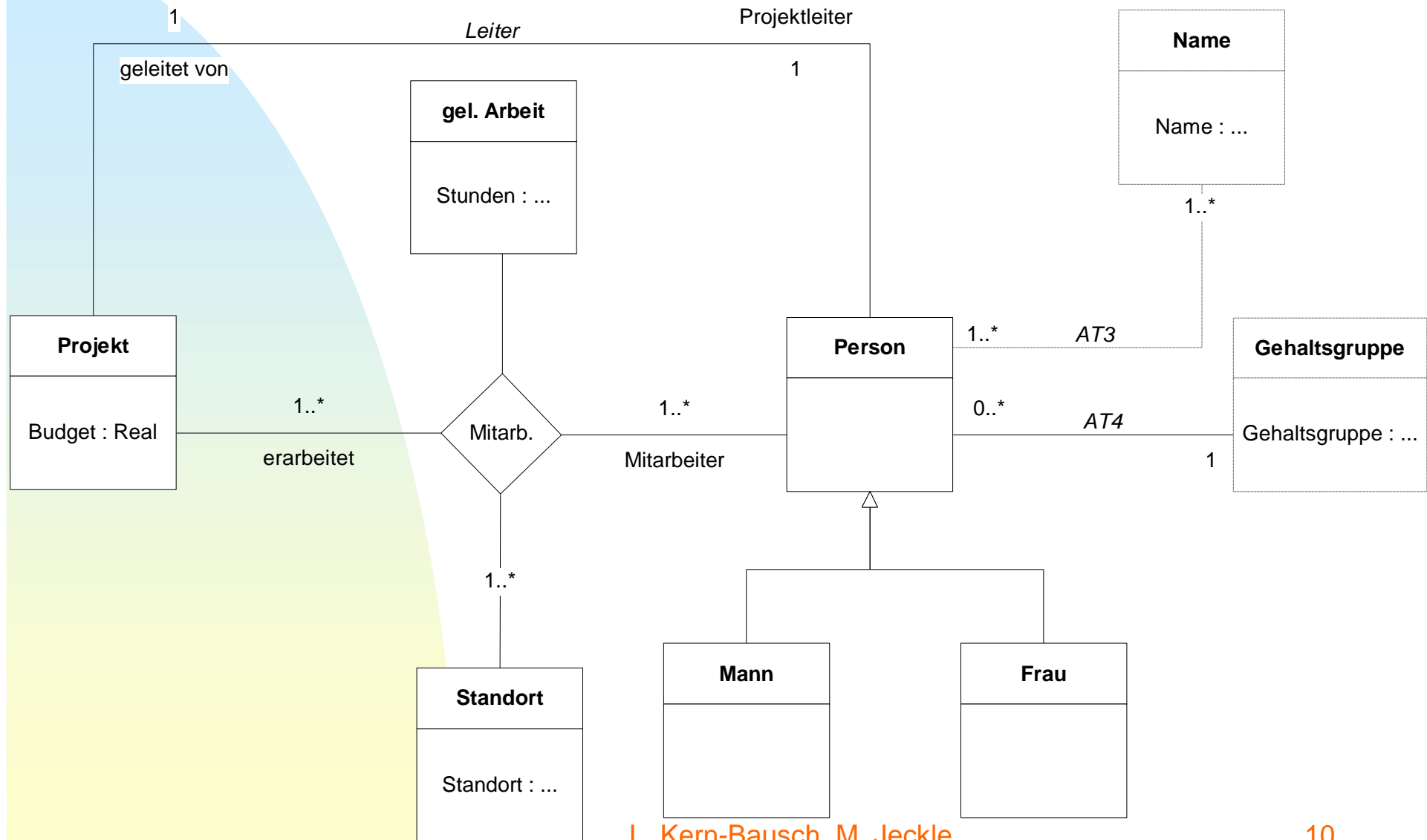
# Der Transformationsalgorithmus

## Schritt 4



# Der Transformationsalgorithmus

## Schritte 5, 6 und 7



# Semantisch irreduzible Formulierung

- Unterstützt Vollständigkeit und Widerspruchsfreiheit
- erleichtert Modellierung komplexer Zusammenhänge
- notwendig für relationale Ableitung
- Stabilität hinsichtlich Schema-Änderungen (Änderbarkeit)
- hilfreich bei Klassen-Design (Zur Gewinnung eines optimalen Klassenmodells)

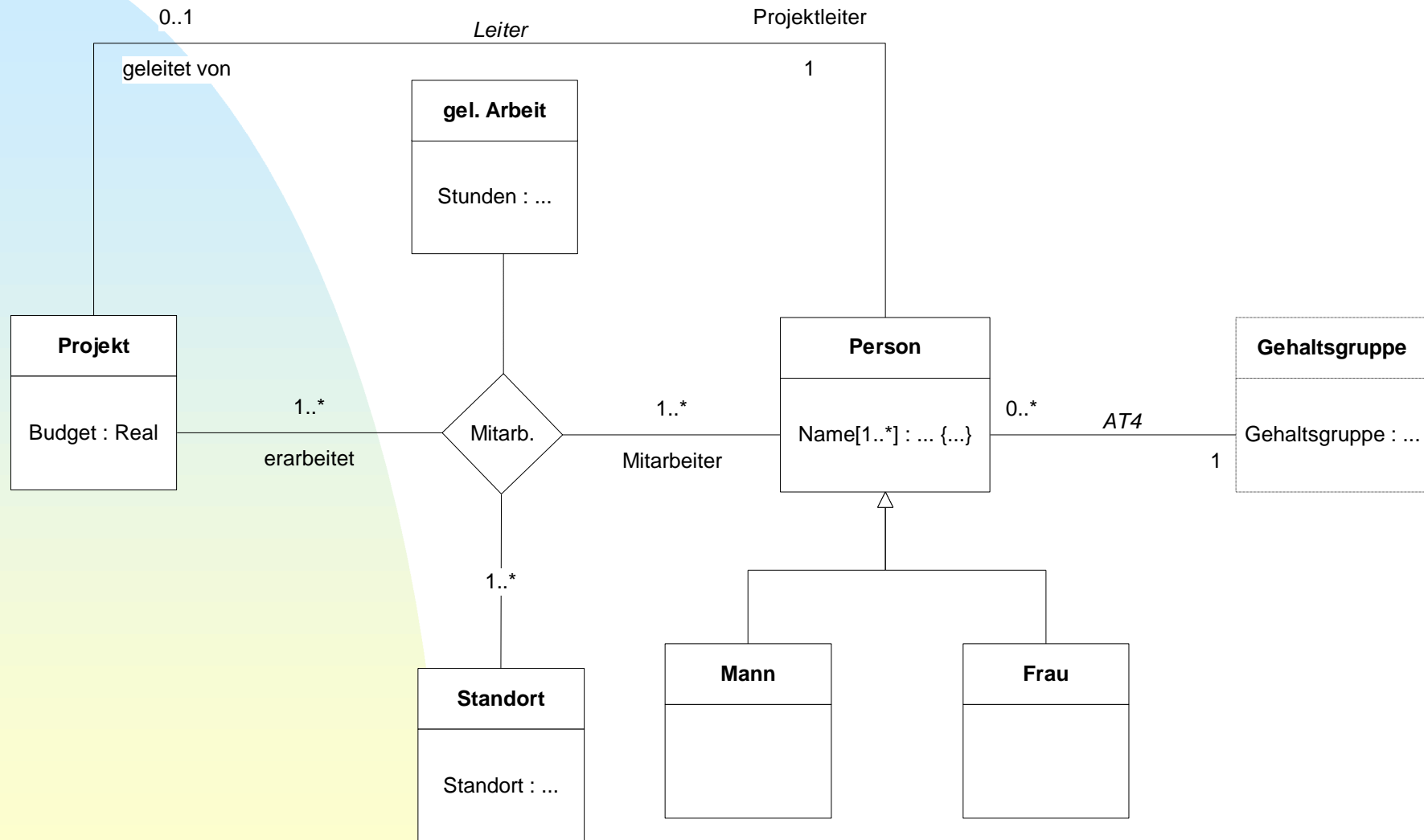


- Erlaubt spätes Einbringung der objektorientierten Applikationssicht



# Der Transformationsalgorithmus

## Schritt 8: Einbringung einer Applikationssicht

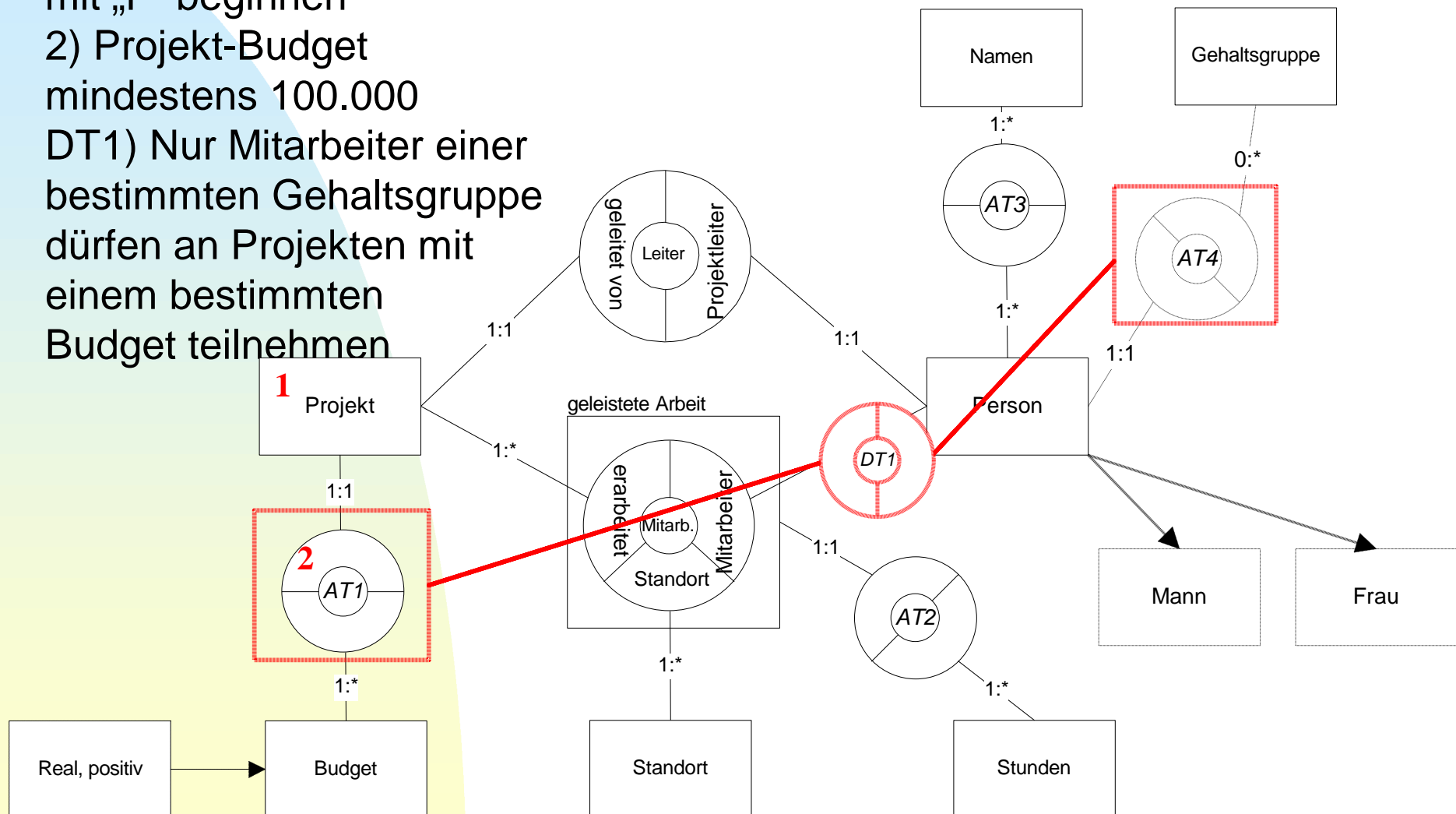


# Constraints (expl.) - Dependenztypen

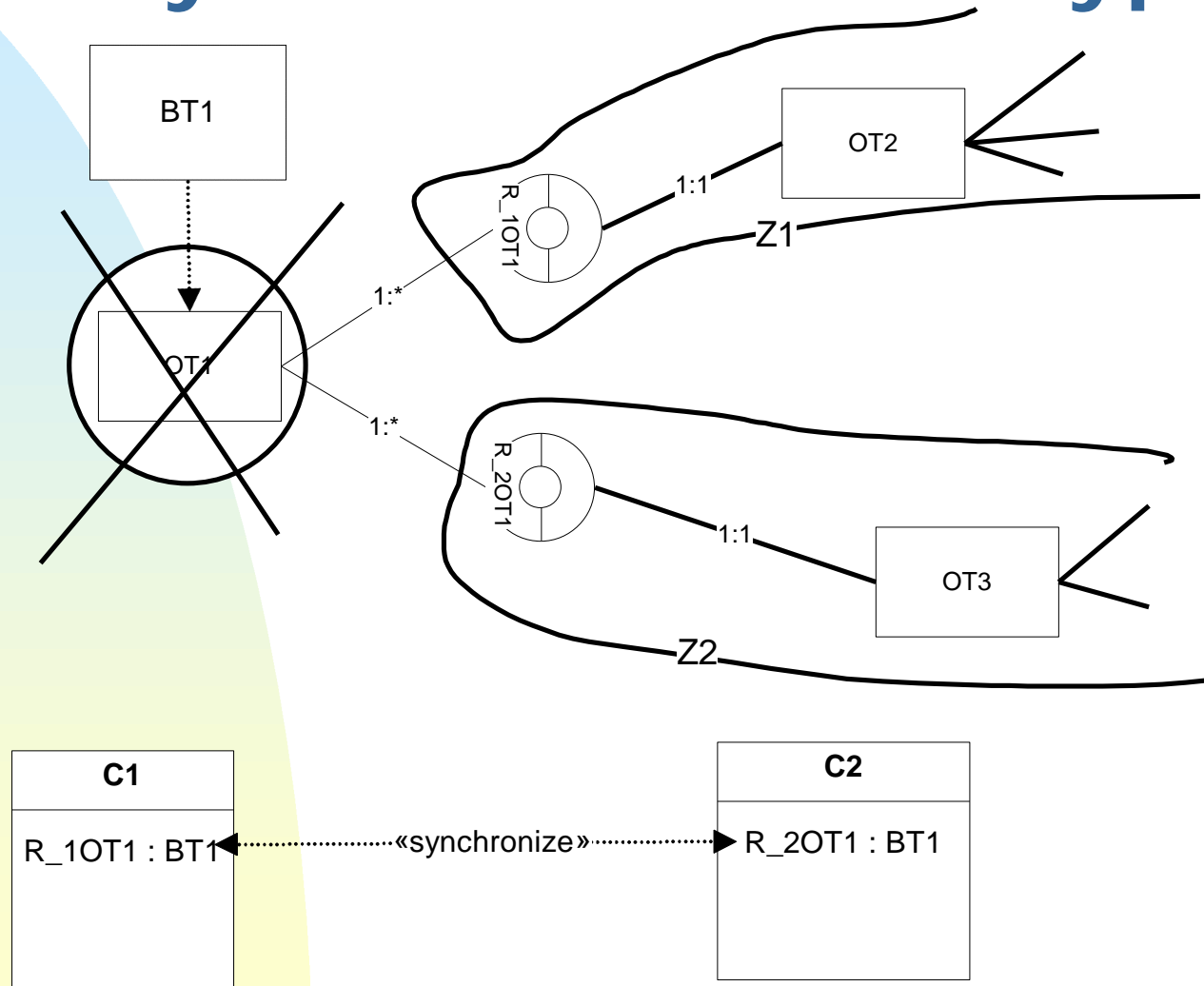
1) Projektnamen müssen mit „F“ beginnen

2) Projekt-Budget mindestens 100.000

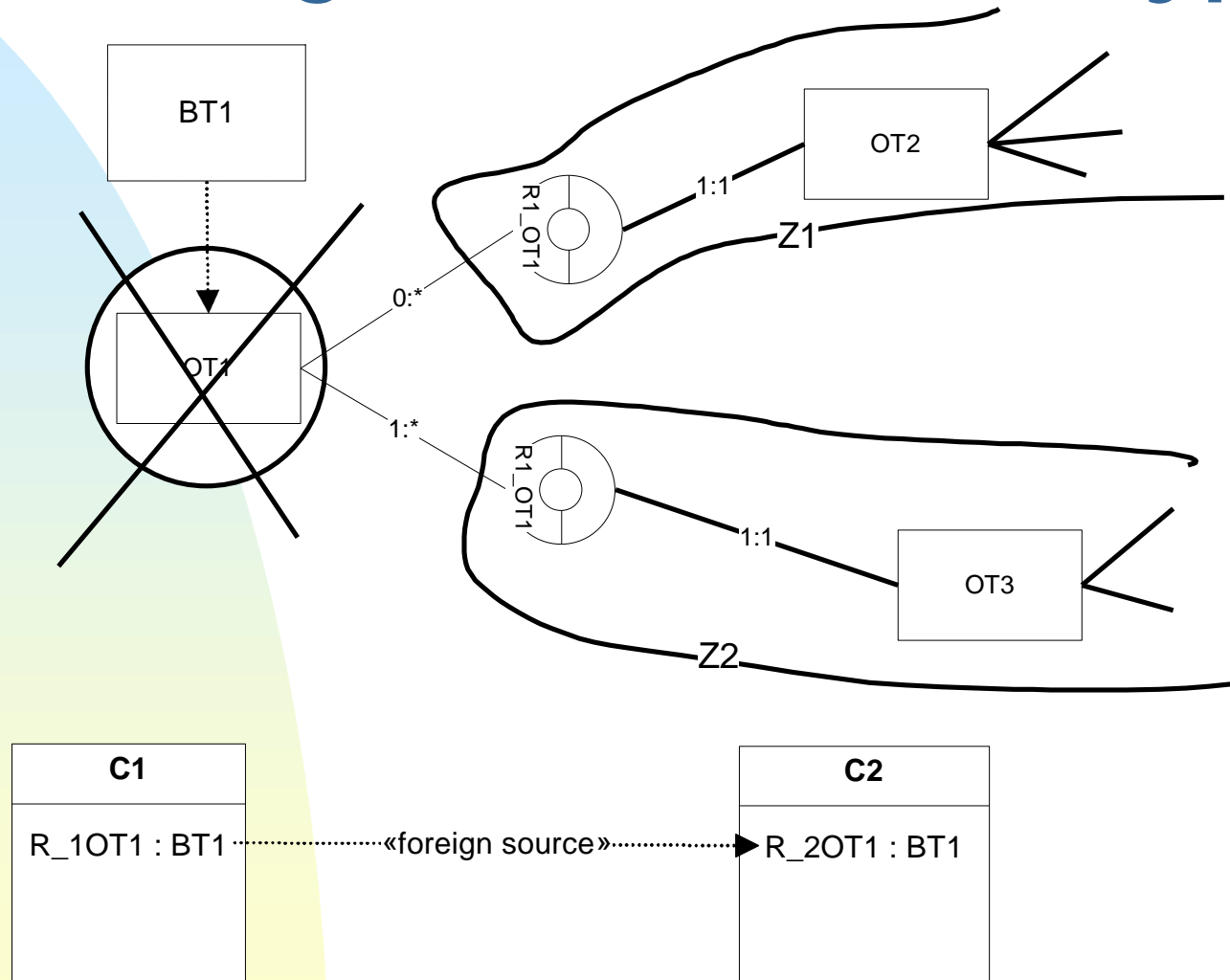
DT1) Nur Mitarbeiter einer bestimmten Gehaltsgruppe dürfen an Projekten mit einem bestimmten Budget teilnehmen



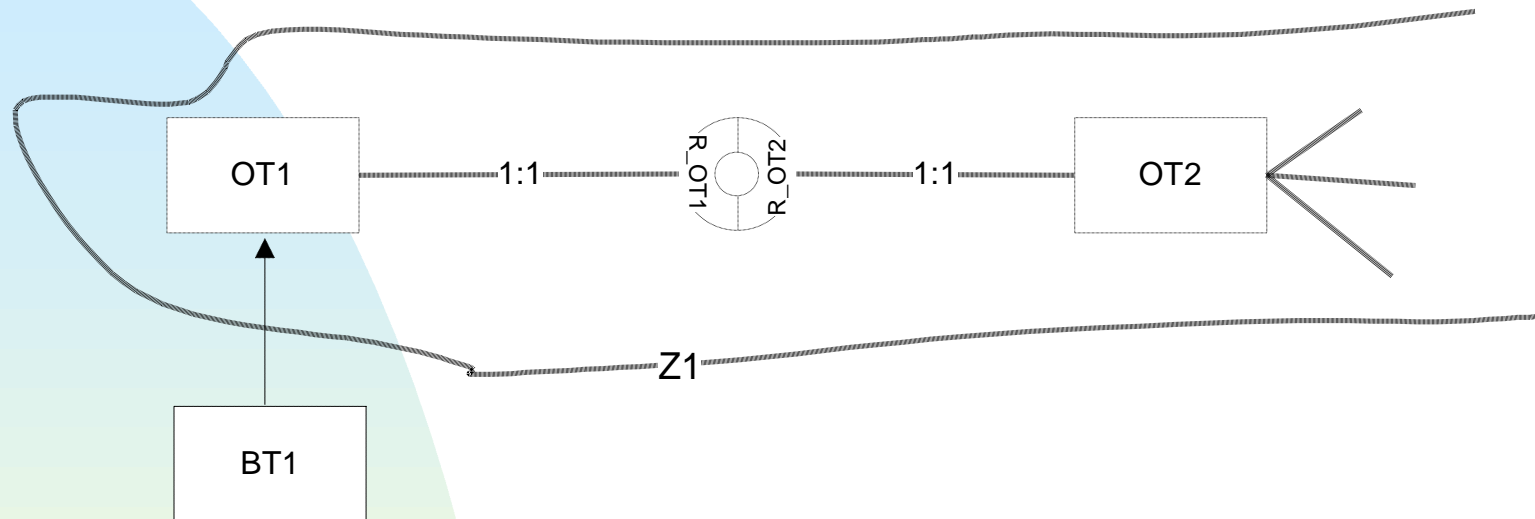
# Constraints (implizit I) synchronize-Stereotyp



# Constraints (implizit II) foreign source-Stereotyp



# Constraints (implizit III) unique-Attribut



C1
R_OT1 : BT1 {unique}

C1

self.allInstances->forAll(c1, c2|c1<>c2 implies c1.OT1<>c2.OT1);