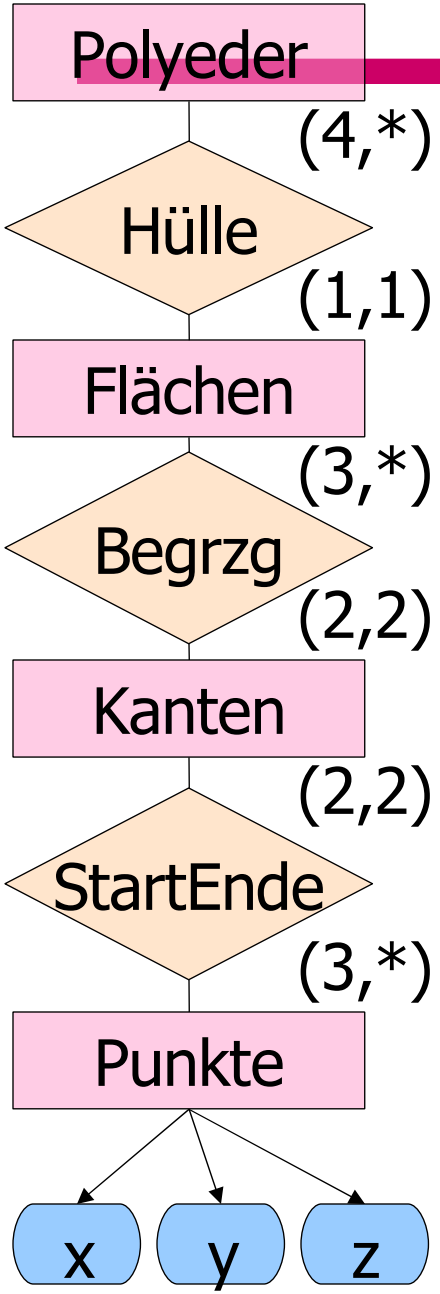
Teil 2: Objektorientierte deduktive Datenbanken (DOOD)

Gerd Stumme
Christoph Schmitz

Wintersemester 2004/05

ENDOWED CHAIR OF THE HERTIE FOUNDATION
Knowledge and Data Engineering
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

Nachteile relationaler Modellierung



Polyeder			
PolyID	Gewicht	Material	. . .
cubo#5	25.765	Eisen	. . .
tetra#7	37.985	Glas	. . .
. . .	. . .	. . .	. . .

Flächen		
FlächenID	PolyID	Oberfläche
f1	cubo#5	. . .
f2	cubo#2	. . .
. . .	. . .	. . .
f6	cubo#5	. . .
f7	tetra#7	. . .

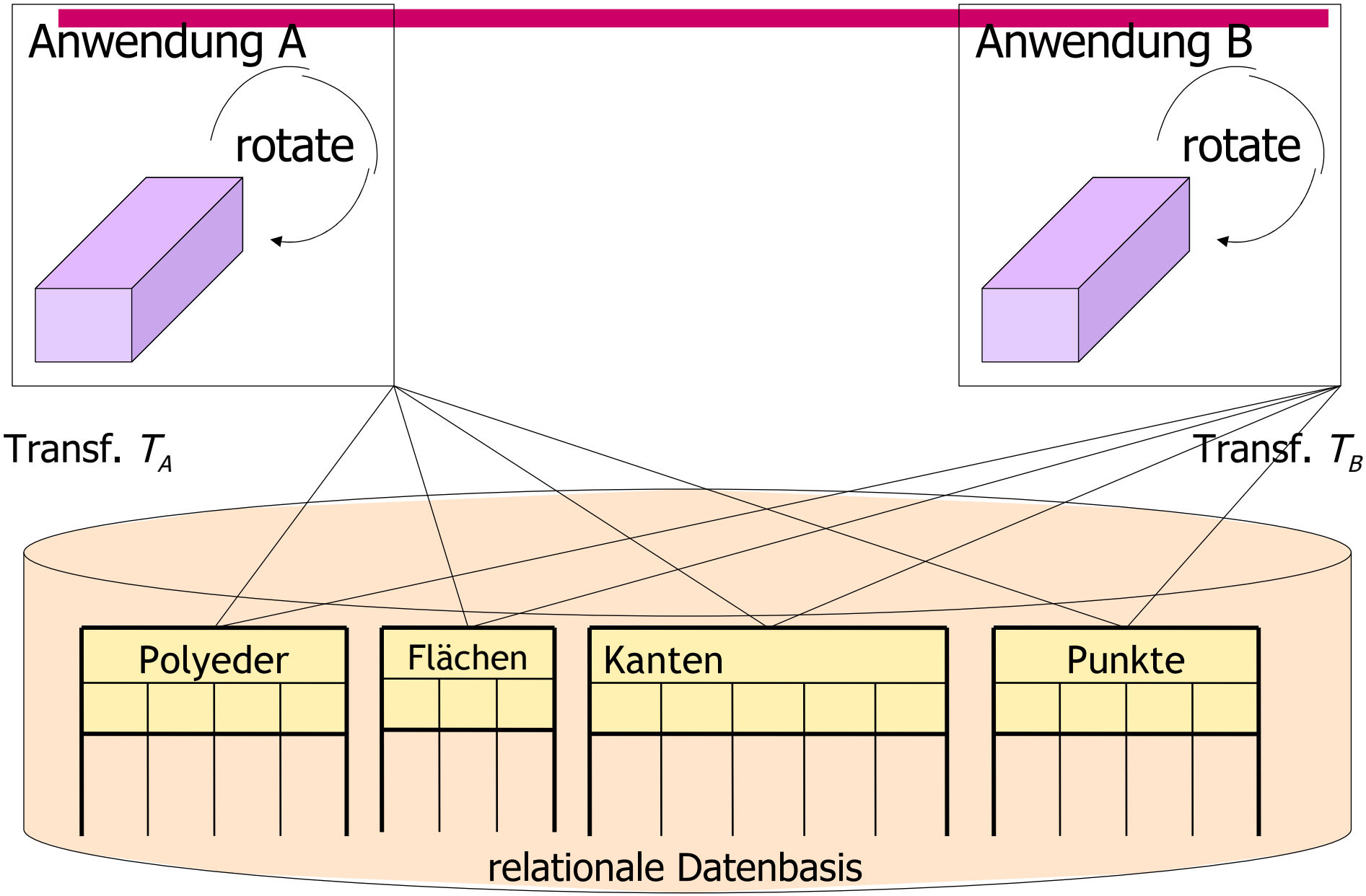
Kanten				
KantenID	F1	F2	P1	P2
k1	f1	f4	p1	p4
k2	f1	f2	p2	p3
. . .	. . .	. . .		

Kanten			
PunktID	X	Y	Z
p1	0.0	0.0	0.0
p2	1.0	0.0	0.0
. . .	. . .	. . .	

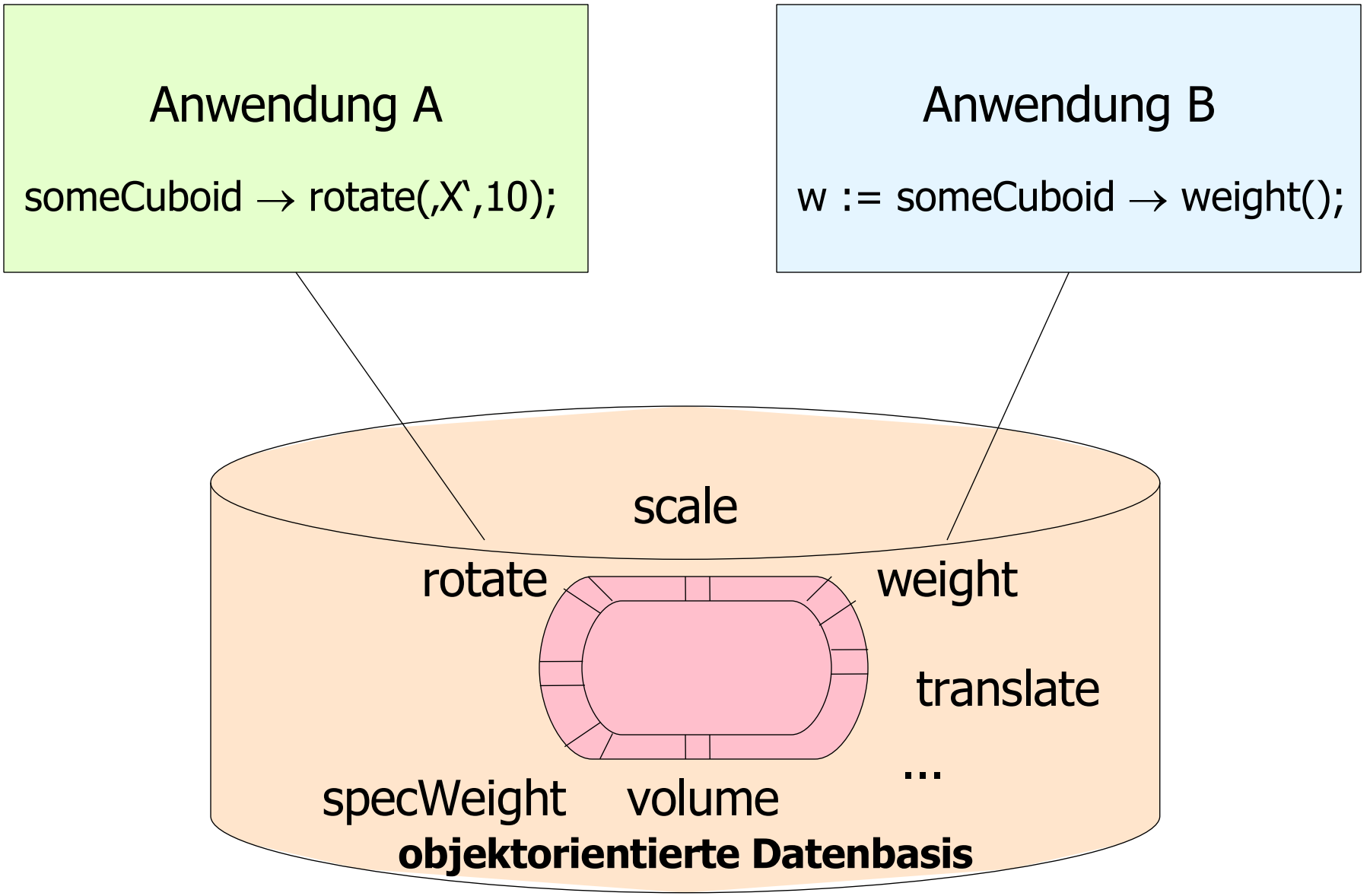
Nachteile relationaler Modellierung

- **Segmentierung**
- **Künstliche Schlüsselattribute**
- **Fehlendes Verhalten**
- **Externe Programmierschnittstelle**

Visualisierung des „Impedance Mismatch“



Vorteile objektorientierter Datenmodellierung



Vorteile objektorientierter Datenmodellierung

- **„information hiding“/Objektkapselung**
- **Wiederverwendbarkeit**
- **Operationen direkt in Sprache des Objektmodells realisiert (kein Impedance Mismatch)**

Objektorientierte Datenbanken

- Standard ODMG
  - 1993: ODMG 1.0
  - 2001: ODMG 3.0, Standardisierungsgruppe beendet Arbeit.
  - Siehe www.odmg.org.
- ca. 12 kommerzielle Produkte: GemStone, Illustra, Itasca, MATISSE, O_2, Objectivity/DB, ObjectStore Ontos, OpenOBD, POET, UniSQL, Statice, Versant (siehe auch http://www.peterindia.net/OODBMSLinks.html)
- OO-Konzepte wurden in Relationale Datenbanken übernommen: *Objekt-Relationale Datenbanken* entstanden als Alternative zu reinen OO-DB.

ODMG-Standardisierung

Beteiligte

- SunSoft (Organisator: R. Cattell)
- Object Design
- Ontos
- O_2Technology
- Versant
- Objectivity

Reviewer

- Hewlett-Packard
- Poet
- Itasca
- intellitic
- DEC
- Servio
- Texas Instruments

Bestandteile des Standards

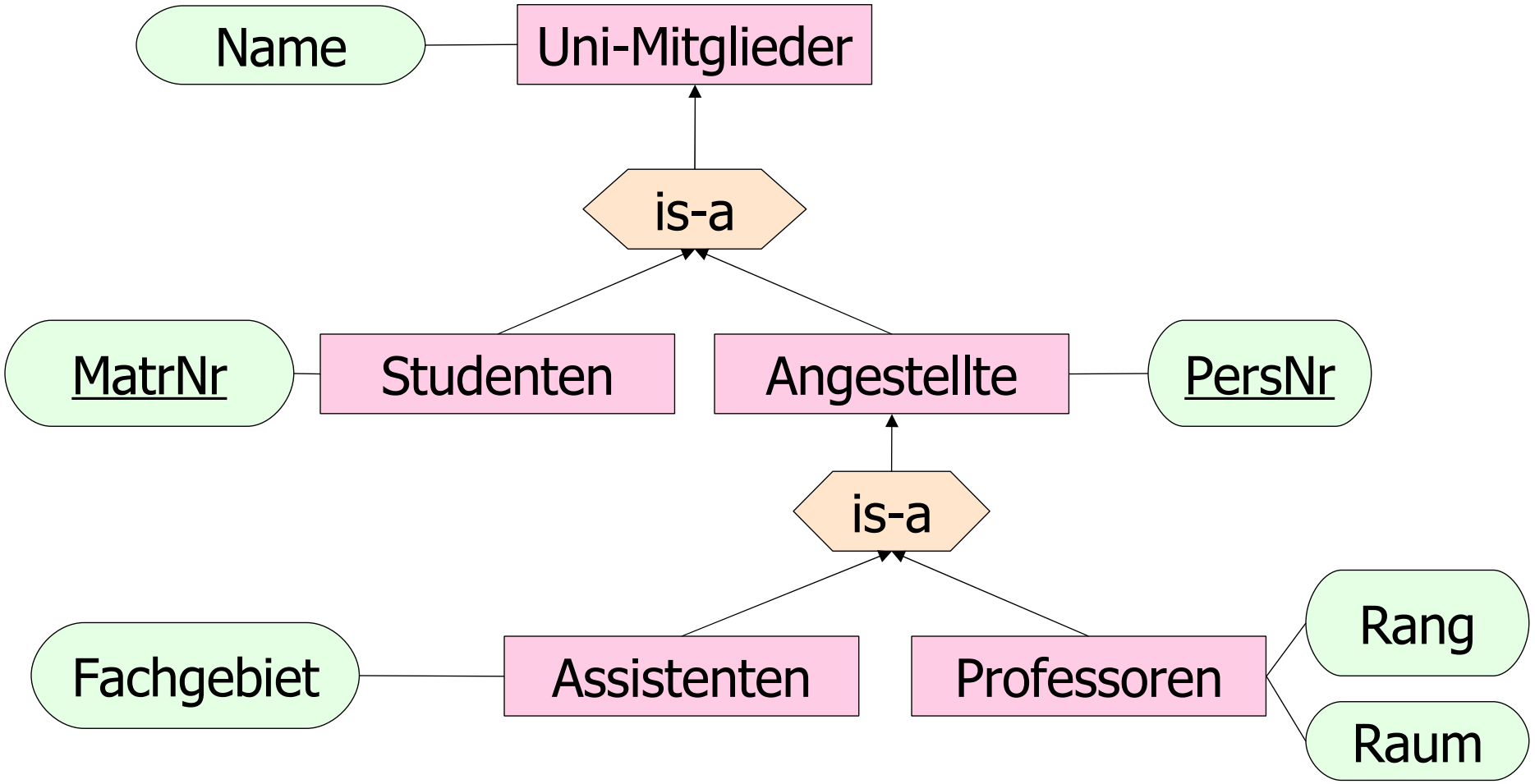
1. Objektmodell
2. Object Defintion Language (ODL)
3. Object Query Language (OQL)
4. C++ Anbindung
5. Smalltalk Anbindung
6. Java Anbindung

Motivation der Standardisierung

- Portabilitäts-Standard
- kein Interoperabilitäts-Standard

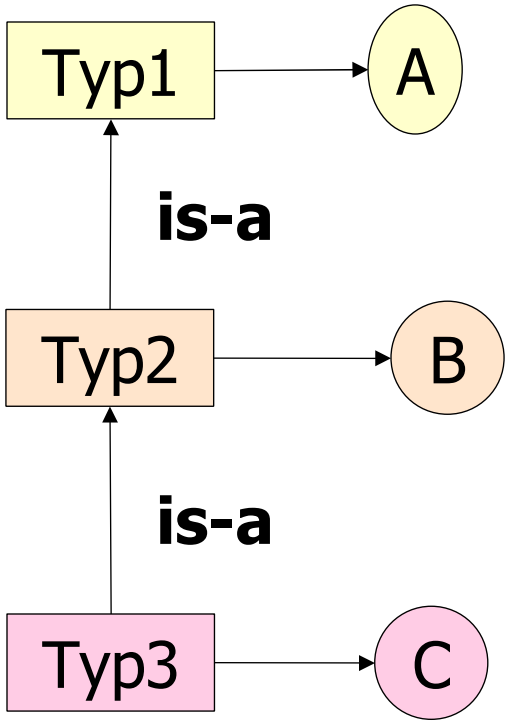
Aber: Eine formale Semantik *aller* Komponenten fehlt.

Vererbung und Subtypisierung

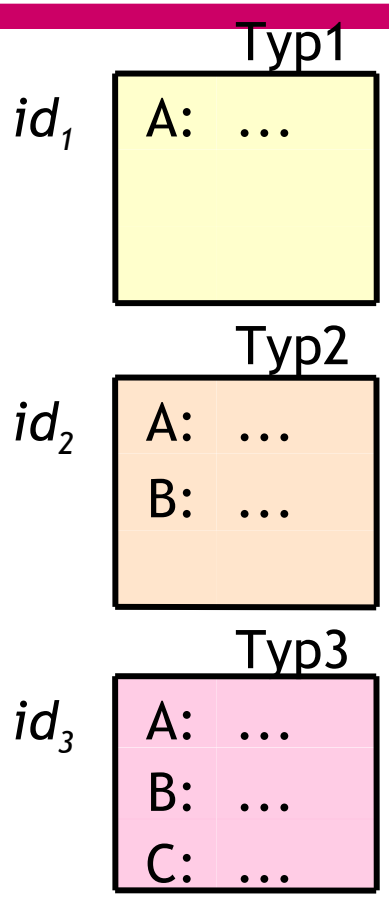


Terminologie

Objekttypen

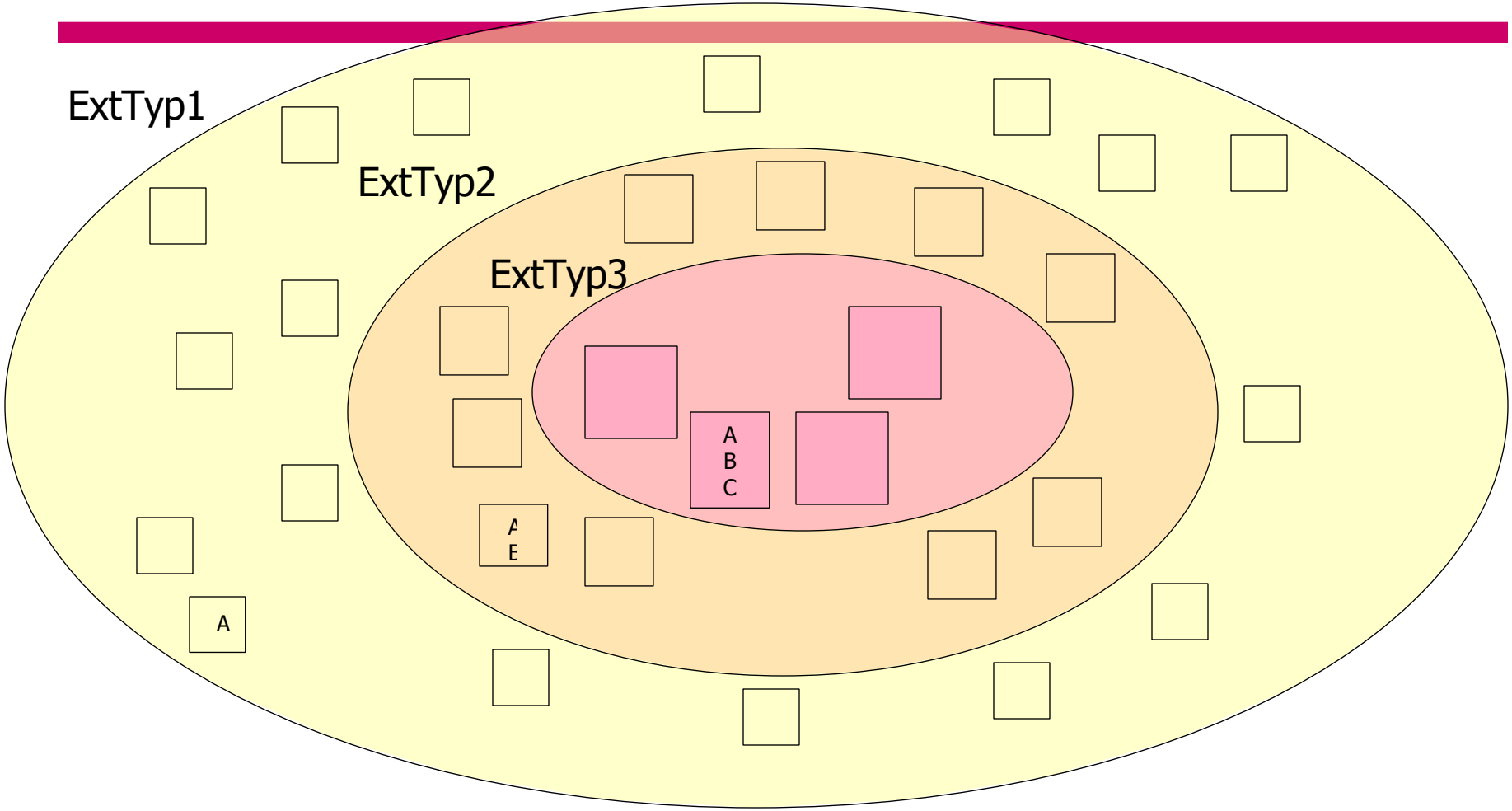


Instanzen



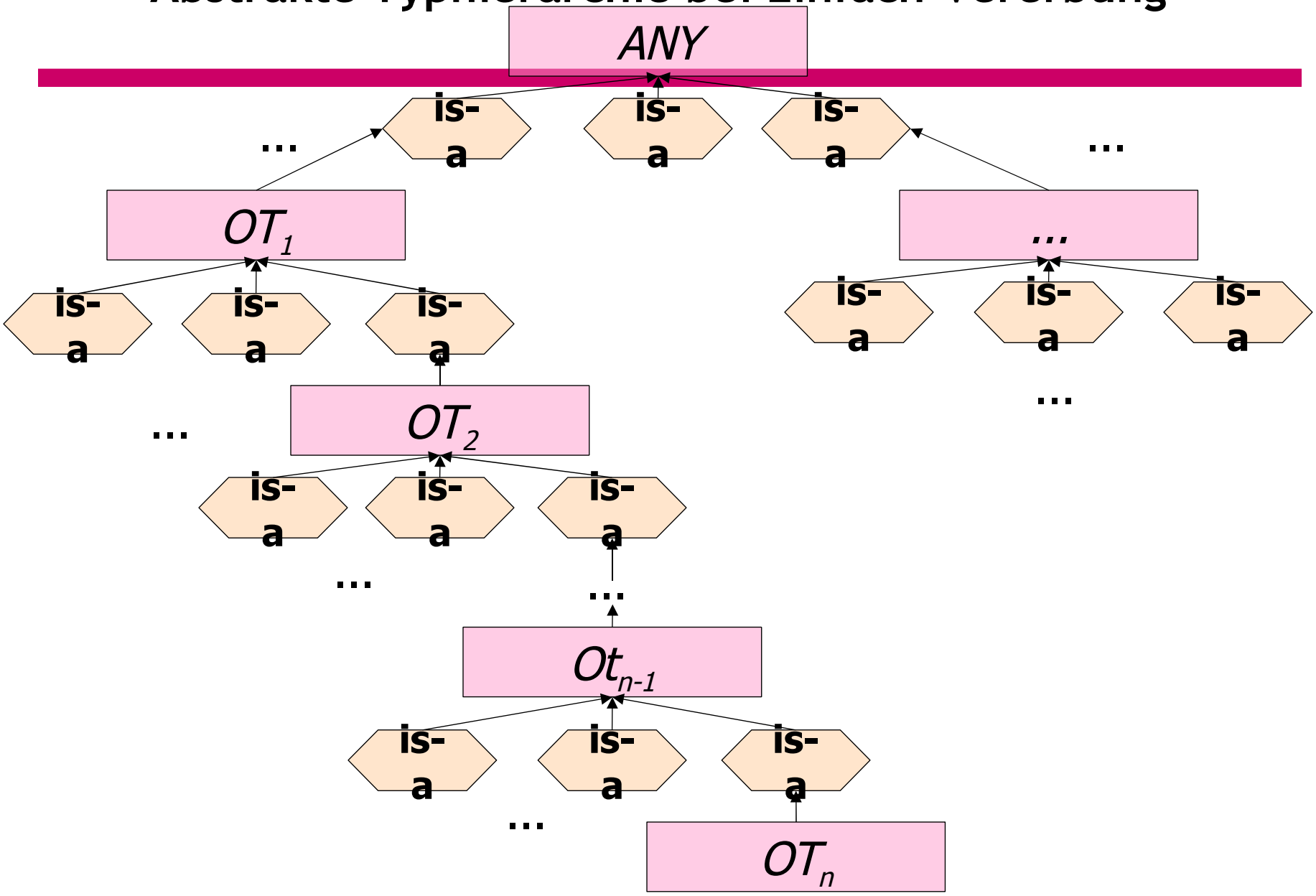
- Untertyp / Obertyp
- Instanz eines Untertyps gehört auch zur Extension des Obertyps
- Vererbung der Eigenschaften eines Obertyps an den Untertyp

Darstellung der Subtypisierung



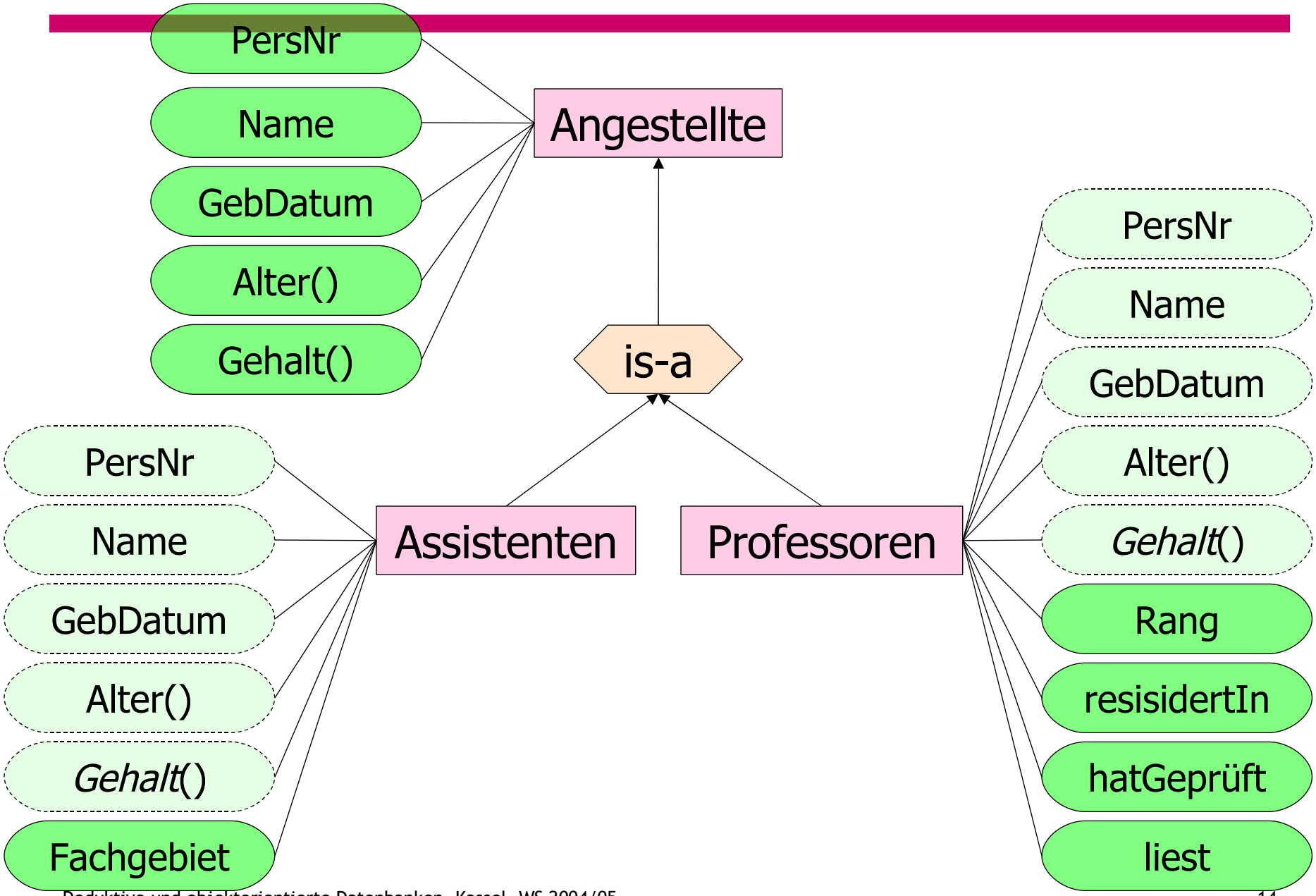
- Inklusionspolymorphismus
- Subtituierbarkeit
  - Eine Untertyp-Instanz ist überall dort einsetzbar, wo eine Obertyp-Instanz gefordert ist.

Abstrakte Typhierarchie bei Einfach-Vererbung



eindeutiger Pfad: $Ot_n \rightarrow Ot_{n-1} \rightarrow ... \rightarrow OT_2 \rightarrow OT_1 \rightarrow ANY$

Vererbung von Eigenschaften



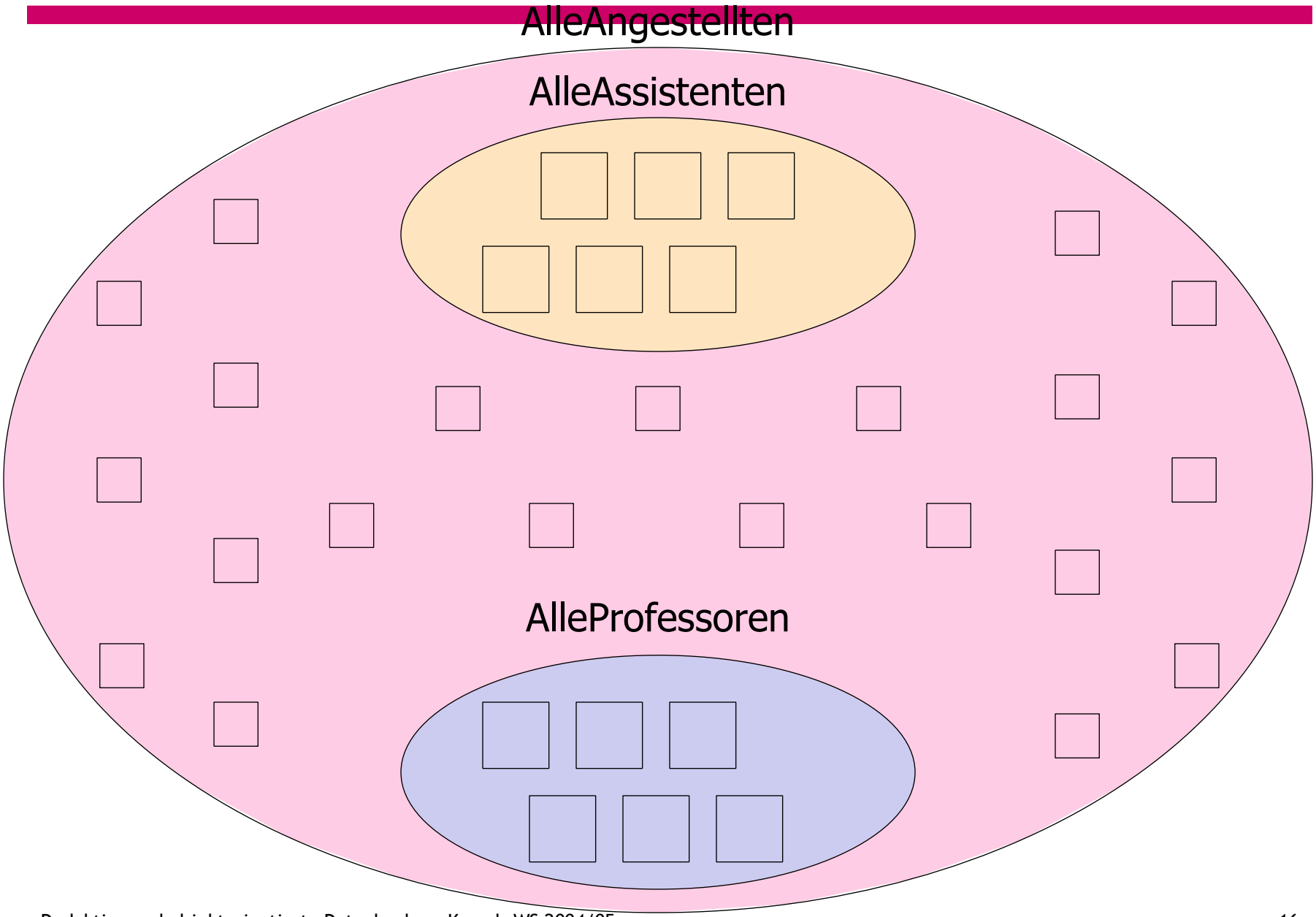
Interface-Definition in ODL

```
class Angestellte (extent AlleAngestellte) {
    attribute long PersNr;
    attribute string Name;
    attribute date GebDatum;
    short Alter();
    long Gehalt();
};

class Assistenten extends Angestellte (extent AlleAssistenten) {
    attribute string Fachgebiet;
};

class Professoren extends Angestellte (extent AlleProfessoren) {
    attribute string Rang;
    relationship Räume residiertIn inverse Räume::beherbergt;
    relationship set(Vorlesungen) liest inverse Vorlesungen::gelesenVon;
    relationship set(Prüfungen) hatGeprüft inverse Prüfungen::Prüfer;
};
```

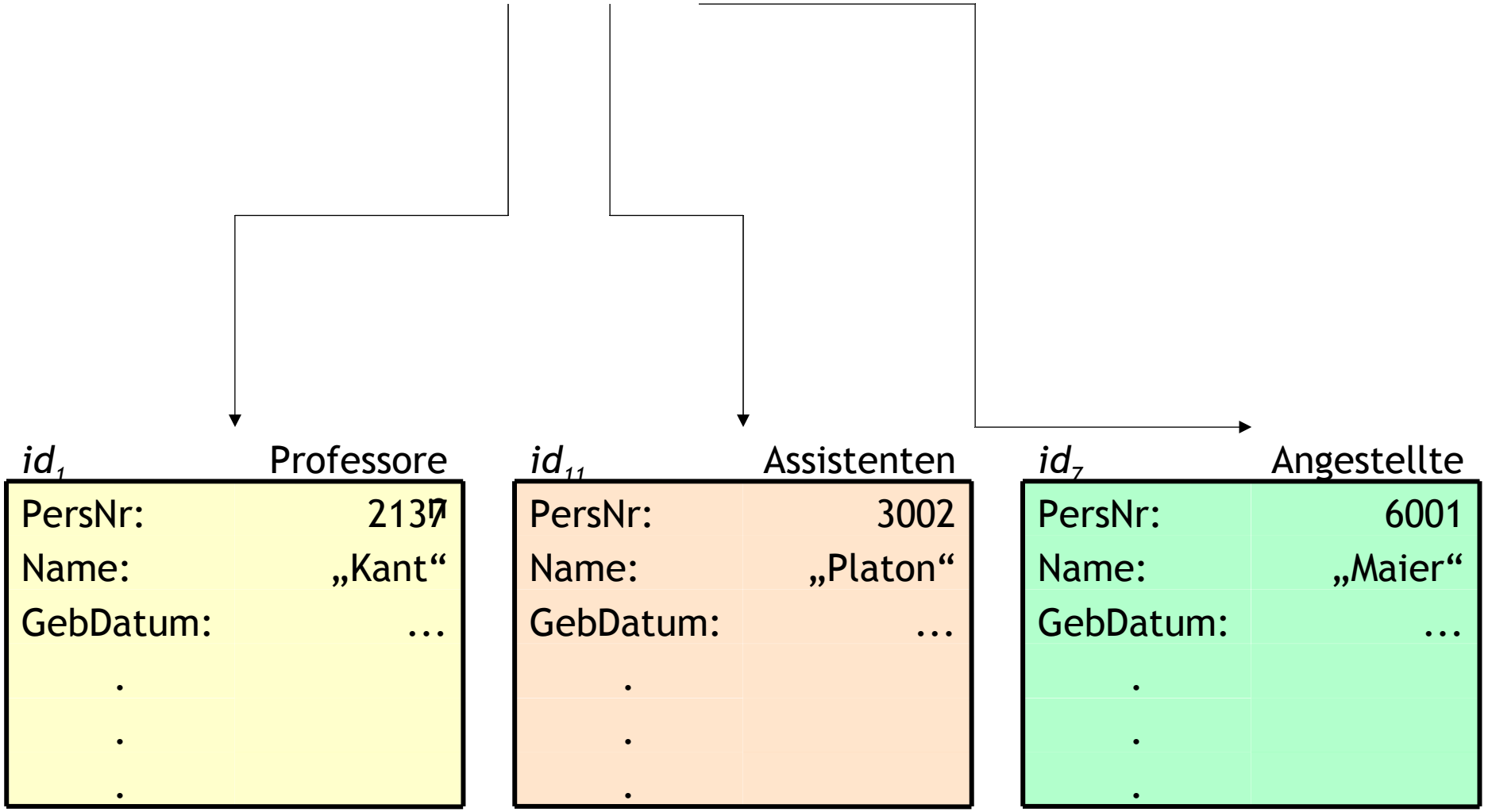
Darstellung der Extensionen



Verfeinerung und spätes Binden

- Die Extension *AlleAngestellten* mit (nur) drei Objekten

  AlleAngestellten: $\{id_1, id_{11}, id_7\}$



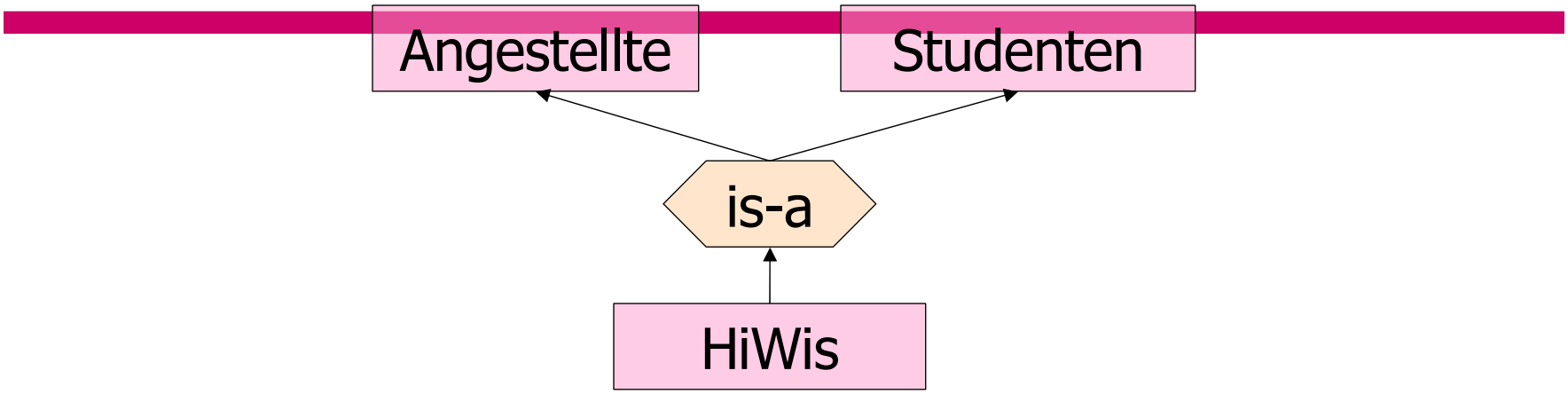
Verfeinerung (Spezialisierung) der Operation Gehalt

- *Angestellte* erhalten: 2000 + (Alter() – 21) * 100
- *Assistenten* bekommen: 2500 + (Alter() – 21) * 125
- *Professoren* erhalten: 3000 + (Alter() – 21) * 150

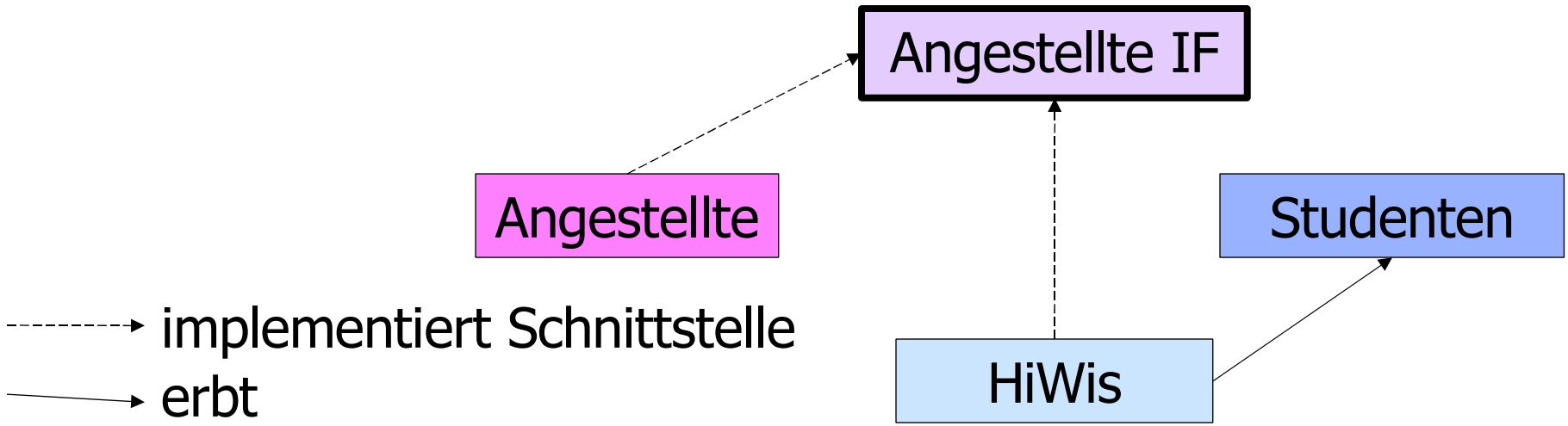
```
select sum(a.Gehalt())
from a in AlleAngestellten
```

- für das Objekt id_1 wird die *Professoren*-spezifische *Gehalts*-Berechnung durchgeführt,
- für das Objekt id_{11} die *Assistenten*-spezifische und
- für das Objekt id_7 die allgemeinste, also *Angestellten*-spezifische Realisierung der Operation *Gehalt* gebunden.

Graphik: Mehrfachvererbung



- geht so in ODMG nicht
- eine Klasse kann nur von einer Klasse erben
- sie kann aber auch mehrere Interfaces implementieren – à la Java



Interface- / Klassendefinition in ODL

```
class HiWis extends Studenten, Angestellte (extent AlleHiWis) {
    attribute short Arbeitsstunden;
    ...
};

interface AngestellteIF {
    short Alter();
    long Gehalt();
};

class HiWis extends Studenten : AngestellteIF (extent AlleHiWis) {
    attribute long PersNr;
    attribute date Gebdatum;
    attribute short Arbeitsstunden;
};
```

Die Anfragesprache OQL

Einfache Anfragen

- finde die Namen der C4-Professoren

```
select p.Name
from p in AlleProfessoren
where p.Rang = „C4“;
```

- Generiere Namen- und Rang-Tupel der C4-Professoren

```
select struct(n: p.Name, r: p.Rang)
from p in AlleProfessoren
where p.Rang = „C4“;
```

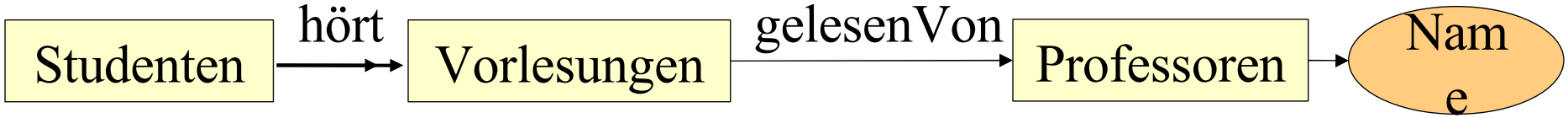
Geschachtelte Anfragen und Partitionierung

```
select struct(n: p.Name, a: sum(select v.SWS from v in p.liest))
from p in AlleProfessoren
where avg(select v.SWS from v in p.liest) > 2;
```

Pfadausdrücke in OQL-Anfragen

```
select s.Name
from s in AlleStudenten, v in s.hört
where v.gelesenVon.Name = „Sokrates“;
```

- Visualisierung des Pfadausdruckes



- ein längerer Pfadausdruck
- *eineVorlesung.gelesenVon.residiertIn.Größe*

  Vorlesungen

  Professoren

  Räume

  float

F-Logic

Wir betrachten in dieser Vorlesung eine spezielle Theorie für *deduktive objekt-orientierte Datenbanken*:

F-Logic ("F" steht für "Frames")

- kombiniert die Vorteile framebasierter Sprachen mit der Ausdrucksstärke, der kompakten Syntax und der wohldefinierten Semantik von Logik.
- formalisiert Signaturen, Objektidentität, komplexe Objekte, Methoden, Klassen und Vererbung.

Wir betrachten hier die Syntax und die Semantik von F-Logic.

Literatur:

- Micheal Kifer, Georg Lausen, James Wu: Logical foundations of object-oriented and frame-based languages. Journal of the ACM (JACM) 42(4), July 1995, 741–843. http://portal.acm.org/citation.cfm?id=210335
- J. Angele, G. Lausen: Ontologies in F-Logic. Ontoprise Whitepaper. http://www.ontoprise.de/members/angele/pubs/ontologyhandbook.pdf
- J. Biskup: Grundlagen von Informationssystemen. Vieweg 1995