

Projekt Erdbebenfrühwarnung im WiSe 2010/11



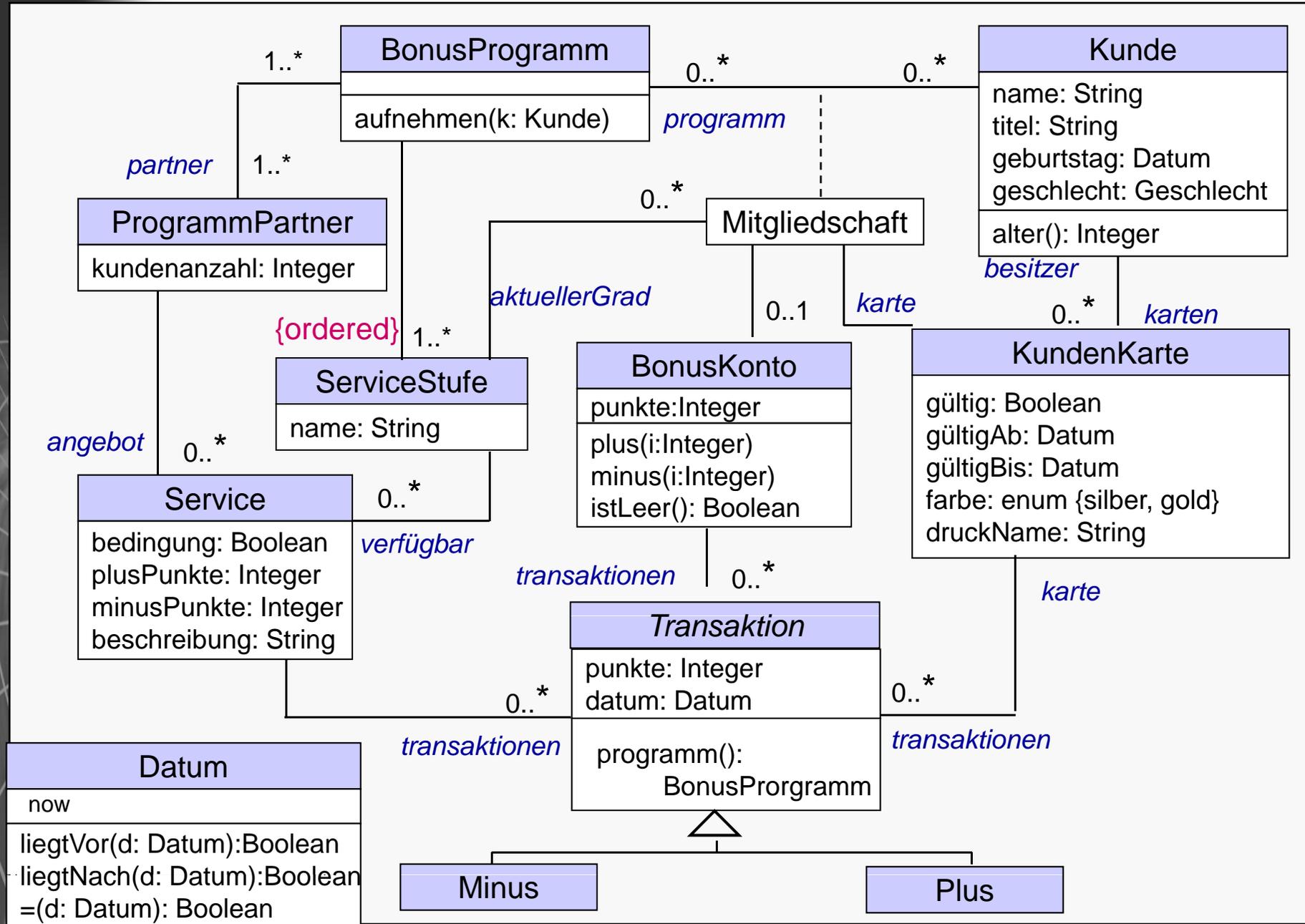
Entwicklung verteilter eingebetteter Systeme

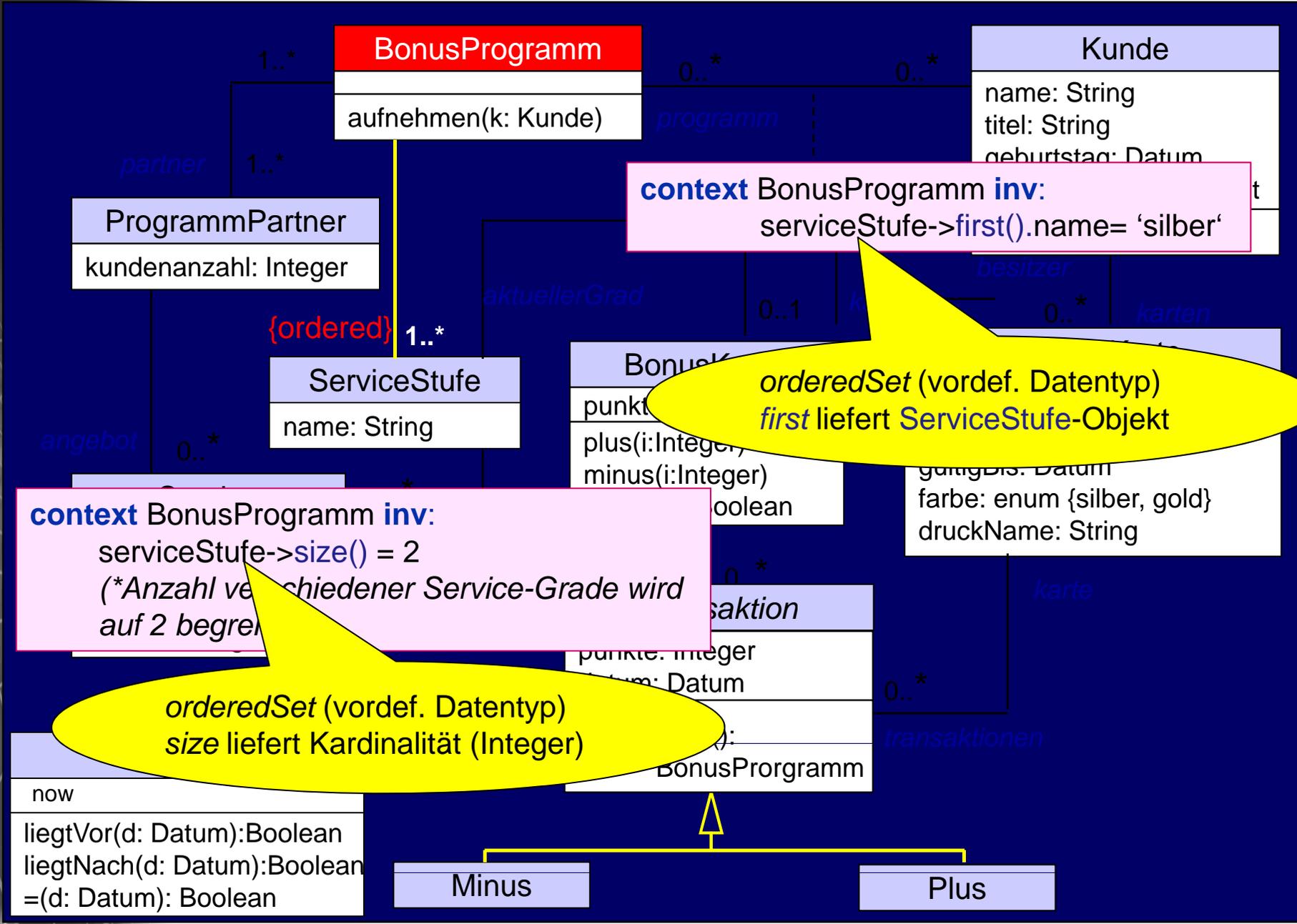
Prof. Dr. Joachim Fischer
Dipl.-Inf. Ingmar Eveslage
Dipl.-Inf. Frank Kühnlenz

fischer|eveslage|kuehnlenz@informatik.hu-berlin.de

4. UML-Überblick

1. Historie von UML
2. Modellierungselemente von UML im Überblick
3. UML-Diagrammarten
4. Diagrammrepräsentationen in UML
5. Zum UML-Standard
6. Beispiel: UML-Klassendiagramm



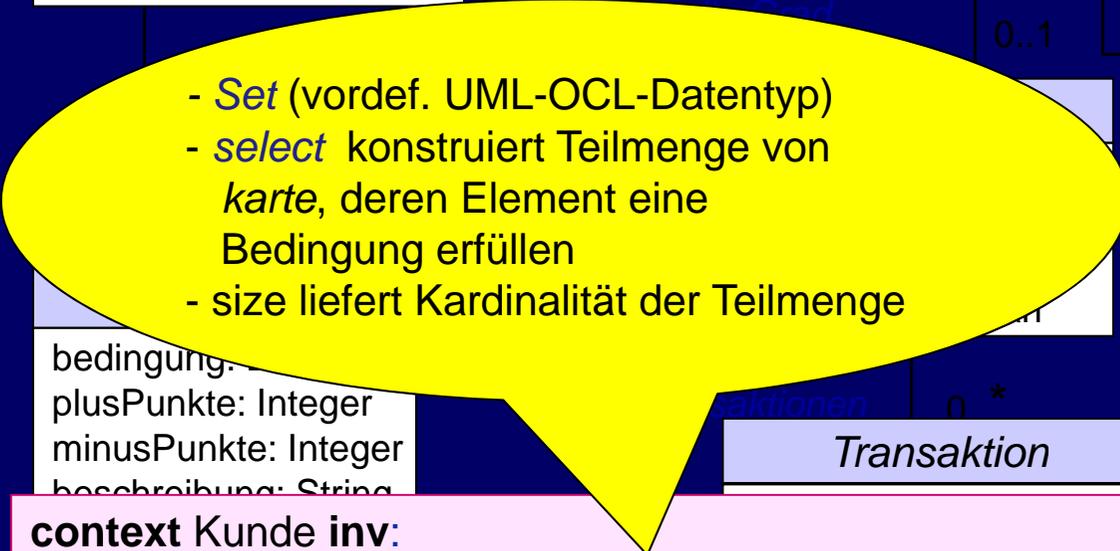


context BonusProgramm **inv:**
 serviceStufe->first().name= 'silber'

context BonusProgramm **inv:**
 serviceStufe->size() = 2
 (*Anzahl verschiedener Service-Grade wird
 auf 2 begrenzt)

orderedSet (vordef. Datentyp)
 first liefert ServiceStufe-Objekt

orderedSet (vordef. Datentyp)
 size liefert Kardinalität (Integer)



Kunde

name: String
 titel: String
 geburtstag: Datum
 geschlecht: Geschlecht

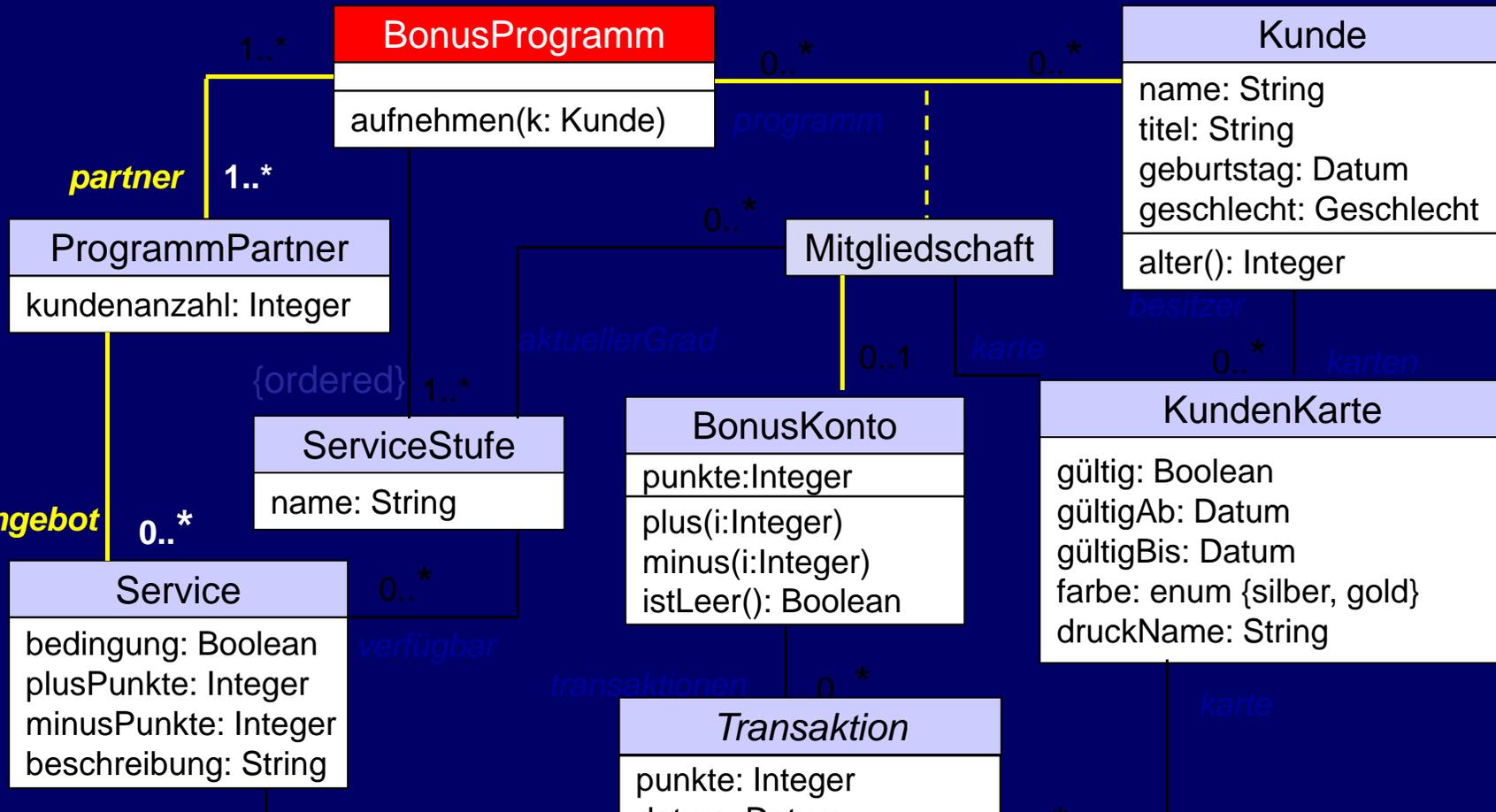
alter(): Integer

KundenKarte

gültig: Boolean
 gültigAb: Datum
 gültigBis: Datum
 farbe: enum {silber, gold}
 druckName: String

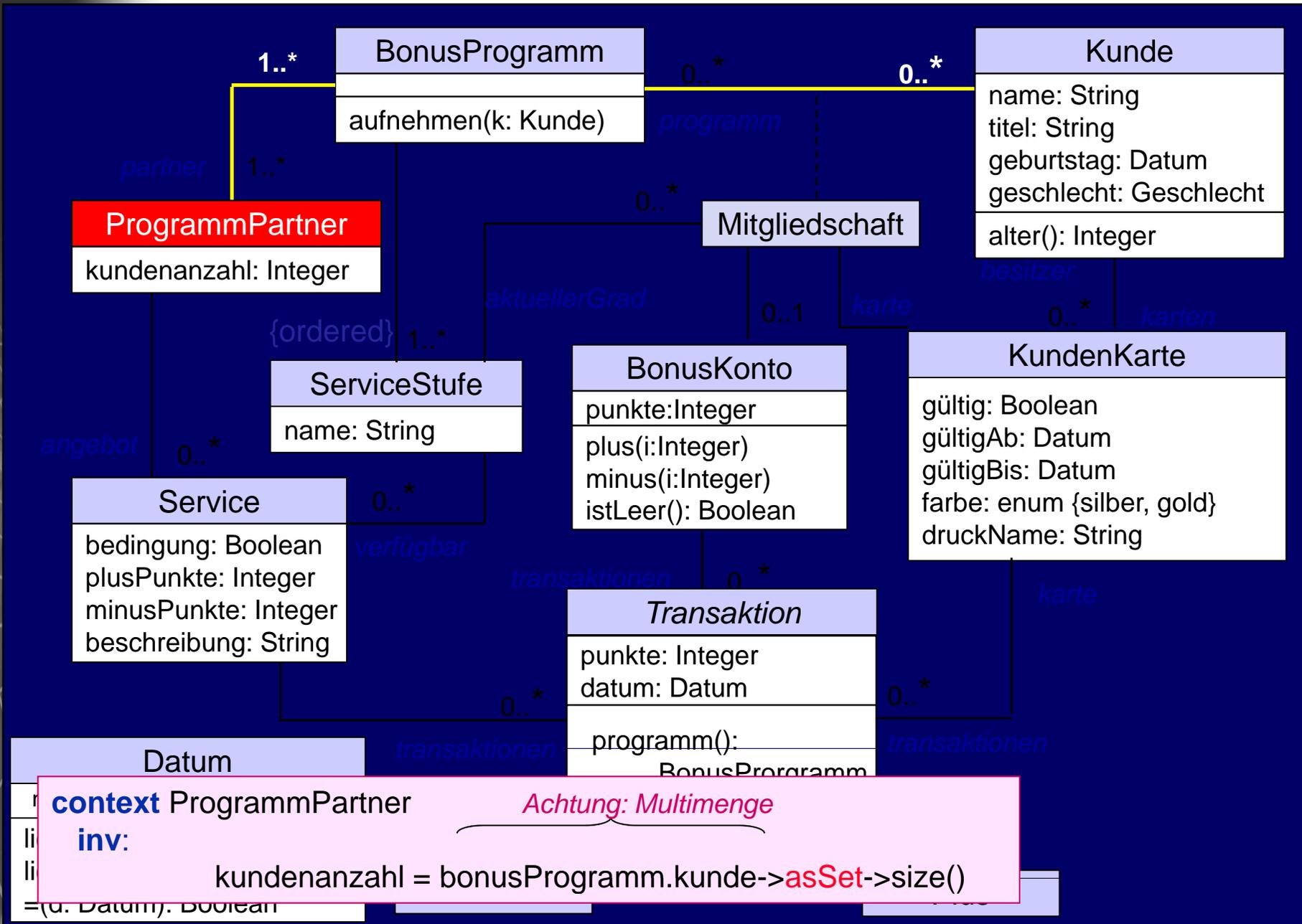
- *Set* (vordef. UML-OCL-Datentyp)
 - *select* konstruiert Teilmenge von *karte*, deren Element eine Bedingung erfüllen
 - *size* liefert Kardinalität der Teilmenge

context Kunde inv:
 programm->size() = karten->select (gültig =true)->size()
Anzahl gültiger Karten eines beliebigen Kunden (kann Null sein) ist gleich der Anzahl von Programmen, an denen dieser Kunde teilnimmt



context BonusProgramm **inv:**
 partner-> anbot-> **forall** (plusPunkte=0 **and** minusPunkte=0)
implies
 mitgliedschaft.bonusKonto->istLeer()

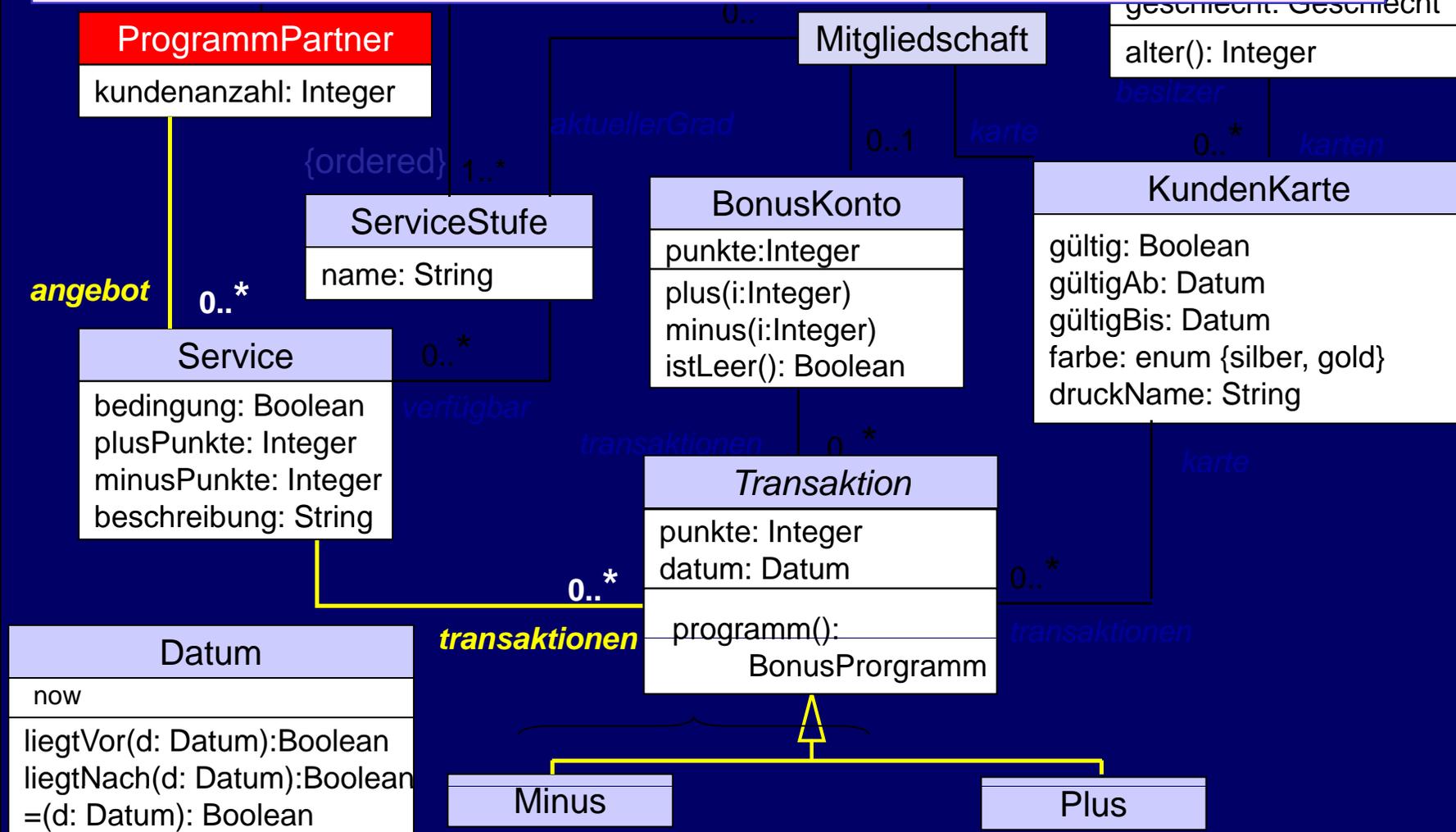




context ProgrammPartner **inv:**

```

self.angebot->transaktionen->select (ocllsTypeOf(Minus))->collect(punkte)->sum()
<=
self.angebot->transaktionen->select (ocllsTypeOf(Plus))->collect(punkte)->sum()
  
```

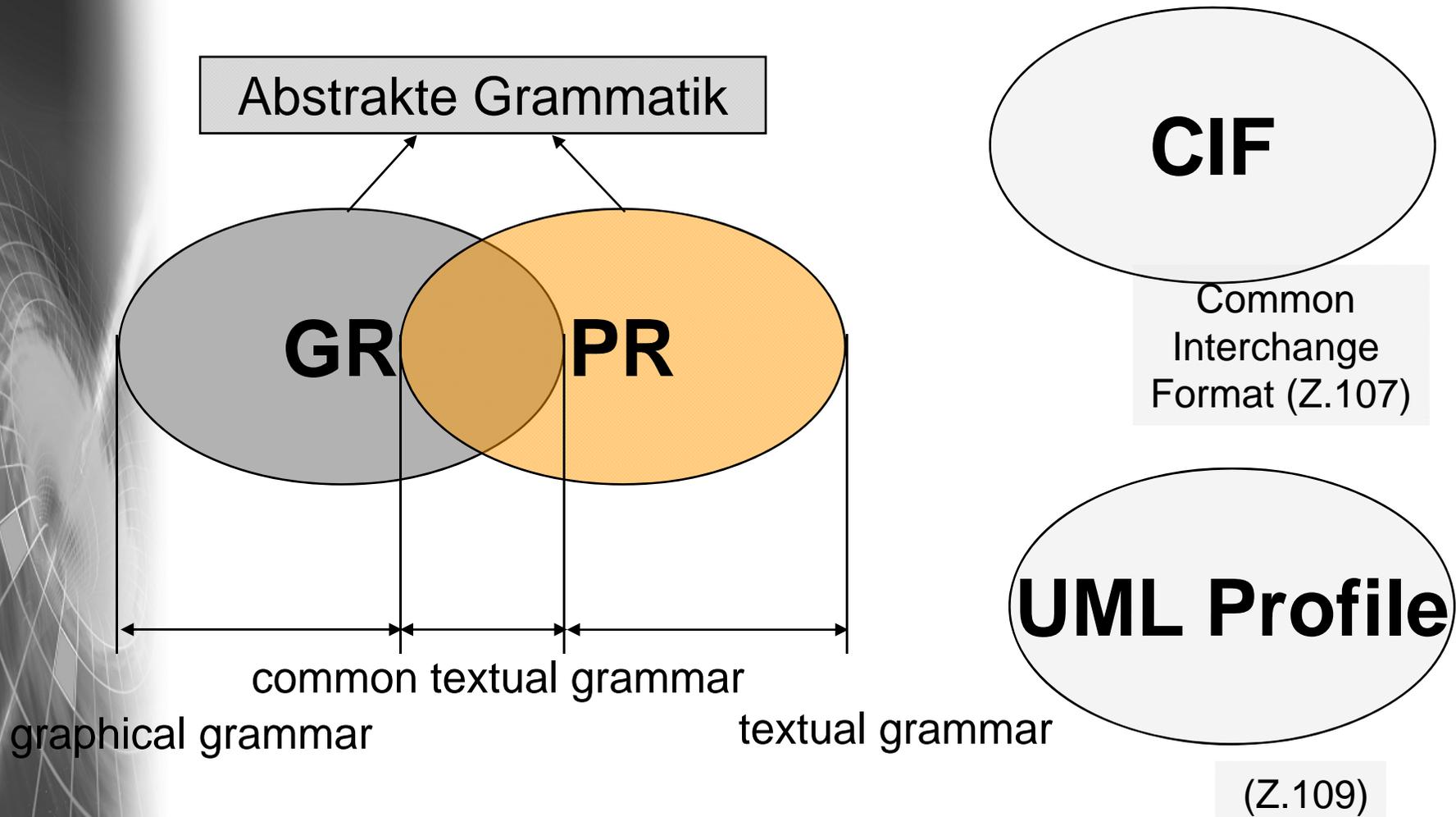


5. SDL

1. ITU-Standard Z.100
2. Werkzeuge
3. SDL-Grundkonzepte
4. Musterbeispiel (in UML-Strukturen)
5. Struktur- und Verhaltensbeschreibung in SDL-RT

Repräsentationsformen

Z.100 (konkrete, abstrakte Syntax, statische u. dynamische Semantik)



Geklärte semantische Variationen in SDL

- Datentypen, Action-Syntax und Action-Semantik (C- Style), SDL erlaubt Werte- und Referenz-Semantik
- Beziehungen zwischen aktiven Klassen und Zustandsautomaten
Agenten sind die Vereinigung von Aktiver Klasse und Zustandsautomat
Mehrfachvererbung für Agenten sind ausgeschlossen
- Agenten besitzen systemweit 1-deutige Referenzen, deren Kenntnis initial nur lokal gegeben ist. *Referenzen werden zur Signaladressierung benutzt.*
- *Keine Aussage zum quantitativen Zeitverhalten der Zustandsübergänge. Übertragungskanäle verbrauchen keine oder nur geringe Zeit*
- Ereignisverwaltung (Signalempfangspuffer) ist präzisiert
- *Remote-Procedure-Call ist über Ersetzungsmodell präzisiert*
- *Systeminstanzen lassen sich definieren*