



wirtschaftsinformatik  
managementinformationssysteme

# system architectures

Übung 4

Wintersemester 2009/2010

Arbeitsgruppe Wirtschaftsinformatik

– Managementinformationssysteme –



# verhaltensebene

- **Zweck**
  - Nähere Beschreibung des Verhaltens der Fachkomponente
  - Spezifikation von **Invarianten - Bedingungen, die immer erfüllt sein müssen**
    - Meldebestand eines Bestandskontos muss immer größer sein als der Sicherheitsbestand
  - Spezifikation von **Vor- und Nachbedingungen - Bedingungen, die vor bzw. nach Ausführung von Diensten der Fachkomponente erfüllt sein müssen**
  - Auch Spezifikation der Bedingungen, die die Fachkomponente an Dienste stellt, die von Außen nachgefragt werden (externe Dienste)



- **Object Constraint Language (OCL)**
  - Von der OMG vorgeschlagen
  - Als Ergänzung der IDL zur Spezifikation auf der Verhaltensebene
- **Ergänzung**
  - Alle Bedingungen auch in natürlichsprachlicher Form
  - Verbesserte Verständlichkeit für bestimmte Adressaten
  - Geringere Präzision

# notationselemente (I)

- Kontext auf den sich die Zusicherung bezieht:

- Bestandskontenverwaltung

`self.Konto->forAll(k:Konto | k.Sicherheitsbestand >= 0)`



Bezug auf Kontext-Instanz

- Invariante

- Bedingung muss immer (während der Laufzeit) erfüllt sein
- z.B.: `self.Konto->forAll(k:Konto | k.Sicherheitsbestand >= 0)`



# notationselemente (2)

- Vor- und Nachbedingungen können festgelegt werden
  - die angegebenen Bedingungen müssen für eine Operation entsprechend erfüllt sein
  - Beispiel:
- Bestandskontenverwaltung::BerechneBestandZum(n:KontoNr,z:Zeitpunkt):Bestand  
pre : self.Konto->exists(k:Konto | k.KontoNr = n)  
post: result = self.Buchungssatz->iterate(b:Buchungssatz; r:Bestand = 0 |  
    if b.KontoNr = n and  
    b.Zeitpunkt <= z  
    then  
    r + b.Menge  
    endif  
)

- weitere Konstrukte:
  - ***forall*** – es muss für alle Elemente einer Sammlung (collection) gelten
  - ***exists*** – es gibt mindestens ein Element, dass der nachfolgenden Bedingung entspricht
  - ***iterate*** – über alle Elemente, die den Bedingungen entsprechen, wird ein Wert aufsummiert.
  - ***select*** – nimmt eine Auswahl aller Elemente einer Sammlung (collection) vor, die den Bedingungen entsprechen

# quellen:

- Turowski, K. (Hrsg.) (2002): **Vereinheitlichte Spezifikation von Fachkomponenten. Gesellschaft für Informatik, Arbeitskreis 5.10.3**

Link: <http://www.wi2.info/fachkomponenten/docs/MEMO/Memorandum-final-2-44-mit-literatur-Web.pdf> (11.11.2009)

- Pousttchi, K. (2009): **System Architectures – Standardizing. Skript zur Vorlesung „System Architectures“**

- **OMG (2001): Object Constraint Language – OMG Available Specification. Object Management Group**

Link: <http://www.omg.org/spec/OCL/2.0> (11.11.2009)

