

IKOM 2

Semi-formale Methoden bei der Entwicklung verteilter Systeme

Achim D. Brucker

brucker@informatik.uni-freiburg.de

Lehrstuhl für Softwaretechnik
Institut für Informatik
Albert-Ludwigs-Universität Freiburg
<http://www.informatik.uni-freiburg.de/~softtech>

Interactive Objects Software GmbH
Basler Strasse 65
79100 Freiburg
<http://www.io-software.com>

7. Mai 2001

Lehrstuhl für Softwaretechnik



- 🕒 Logische Repräsentationen zur Spezifikation
- 🕒 Formale Methoden
- 🕒 Sicherheitsanalysen
- 🕒 Verifikation
- 🕒 Testdatengenerierung
- 🕒 Spezialist für IT-Architektur und verteilte Systeme
- 🕒 Schwerpunkt CORBA und EJB
- 🕒 Werkzeugentwicklung: ArcStyler - the Architectural IDE
- 🕒 Consulting
- 🕒 Schulungen



Previous Next Start End Back FullScreen Quit

Warum ...

verteilte Systeme?

- 🕒 alle „Internet-Anwendungen“ sind verteilt
- 🕒 erfolgreich Einsatz im Bereich E/M-commerce
- 🕒 Verwendung von erprobter Middleware

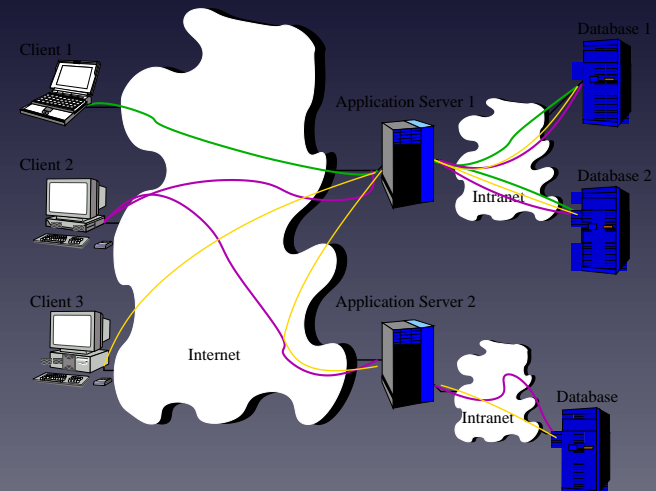
formale Methoden?

- 🕒 mathematische Modelle zur Spezifikation von Software
- 🕒 lange Erfahrung in der Forschung
- 🕒 Erfolge in Sicherheitskritischen Bereichen
- 🕒 Verbesserung der Softwarequalität



Previous Next Start End Back FullScreen Quit

Verteilte Systeme



Previous Next Start End Back FullScreen Quit

(Semi-) formale Methoden

Forschung

- ☉ formale Methoden (z.B. VDM, Z)
- ☉ aufbauend auf Logik / Algebren
- ☉ mathematisch exakt
- ☉ Beweisverfahren
- ☉ Testdatengenerierung



Industrie

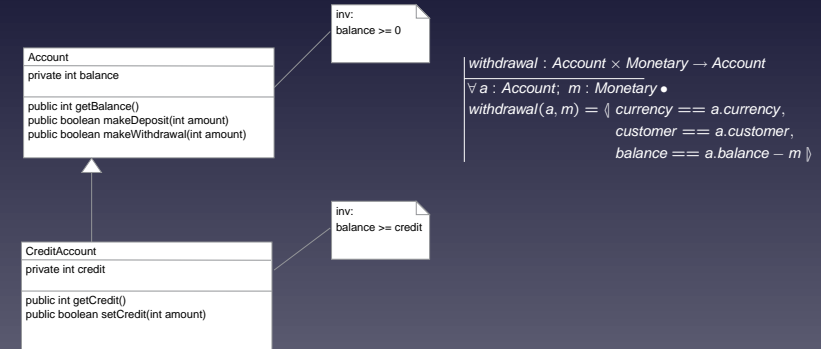
- ☉ im OO-Bereich: UML
- ☉ Erweiterung von UML durch OCL
- ☉ Unterstützung durch CASE Tools
- ☉ Visualisierung
- ☉ Dokumentation
- ☉ „Design by Contract“

Verhaltensbeschreibung um die
Zusicherung von (Sicherheits-)
Eigenschaften nachweisen zu
können.

Systembeschreibung mit dem
Ziel der Dokumentation und
Code-Generierung.



UML/OCL: Beispiel einer semi-formaler Spezifikation



Zielsetzung: Formalere Beschreibung der Geschäftsmodelle mit minimalem Aufwand.



Ergebnisse und Ausblick

- ☺ die Einschränkung auf die Geschäftsmodelle ist sinnvoll
- ☺ Semi-formale Spezifikation verteilter Systeme möglich
- ☺ OCL ermöglicht „Design by Contract“ in verteilten Systemen (Laufzeitchecks)
- ☹ „formale“ Probleme mit OCL
- ☹ Formulierung von OCL Constraints in der Praxis „schwierig“
- ☉ Formale Definition der Semantik von OCL
- ☉ automatische Generierung von Testdaten

