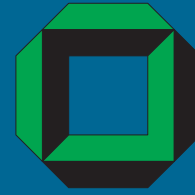


Software aus Komponenten



Prof. Dr. Gerhard Goos
Fakultät für Informatik
Universität Karlsruhe

Karlsruhe, Germany 2004

© Gerhard Goos, Dirk
Heuzeroth,
Elke Pulvermüller, Volker
Kuttruff
2004

[http://www.info.
uni-karlsruhe.de/](http://www.info.uni-karlsruhe.de/)



- Vorlesung:
 - ▶ 1. Dienstag 14:00 – 15:30, -101, wöchentlich
 - ▶ 2. Mittwoch 14:00 – 15:30, -101, vierzehntägig
- Übungen:
 - ▶ Mittwoch 14:00 – 15:30, -101, vierzehntägig, erstmals 28.04.2004
- Übungsleiter: Dr. Dirk Heuzeroth, Mamdouh Abu-Sakran, Volker Kuttruff
- Adressen:
 - ▶ Allgemeines Verfügungsgebäude, 2. OG
 - ▶ ggoos@ipd.info.uni-karlsruhe.de
 - ▶ heuzer@ipd.info.uni-karlsruhe.de, msakran@ipd.info.uni-karlsruhe.de, kuttruff@fzi.de
- Sprechstunden:
 - ▶ Goos: Dienstags 16:00 – 17:00
 - ▶ Heuzeroth, Abu-Sakran, Kuttruff: Mittwochs 16:00 – 17:00
- Vorlesungsseite:
<http://www.info.uni-karlsruhe.de/lehre/2004SS/swk>

- Unterschiedliche Basiskommunikation der Systeme,
 - ▶ z.B. können Corba Komponenten nicht direkt COM Komponenten verwenden
 - ▶ Wrapping mit zusätzlicher Indirektion oder
 - ▶ Invasive Änderung der Komponente nötig (unmöglich bei Binärlösung)
- Keine formale Definition von Protokollen
 - ▶ Komponenten mit gleichen Schnittstellen i.a. nicht wiederverwendbar
 - ▶ Lösung Standardisierung:
 - Schnittstellenidentifikation ist eindeutig (COM)
 - Reflexion

- Aspekttrennung schafft Voraussetzungen
 - ▶ Trennung von Schnittstellen und Implementierungen
 - ▶ Mehrere Schnittstellen pro Komponente
 - ▶ Trennung von Persistenz-, Transaktions-, Verteilungsaspekt
- Reflexion erlaubt das Erkennen der Notwendigkeit
- Kommunikationsanpassung — Brücken zwischen den Welten
 - ▶ EJB - Corba
 - ▶ Java - DCOM
 - ▶ Corba - DCOM
- Konzepte zur Anpassung
 - ▶ Explizites Adaptieren und Delegieren
 - ▶ IDL Generatoren erzeugen Code für Sprach- und Verteilungsanpassungen
 - ▶ siehe Konzepte zur Komposition
- .NET: Vereinheitlichung
 - ▶ Brücke zu (D)COM

- Daten, Funktionen, Kommunikation werden **explizit** und **extern** angepaßt.
Ausnahme: Datenanpassung bei XML über XSLT oder andere Transformatoren
- Synchronisation und Protokollanpassungen entsprechen Anpassungen von Transaktionen und Sessions (siehe vorige Folien)
- Keine Konzepte zur globalen Lebendigkeit
 - ▶ Synchronisationskonzepte unter Komposition
 - ▶ Lebendigkeitstests

- + Komponenten- und Verdrahtungsstandard
- + Transparenz bzgl. Verteilung, Sprache, Plattform, Persistenz
- + Transaktionskonzepte
- + Vordefinierte Dienste, Komponenten, Anwendungen
 - Komplex, hoher Einarbeitungsaufwand
 - hoher Kommunikationsaufwand, da (außer COM, .NET) für verteilte Anwendungen vorgesehen
 - Flexibilität führt zur hoher Komplexität auch im Produktionscode (Laufzeitprobleme)
 - Objektorientierte Entwurfskonzepte nicht auf Komponentenebene umgesetzt
 - Kaum Unterstützung für Anpassungen der Komponenten