

TransBS

Transformation monolithischer
Business-Softwaresysteme in verteilte,
workflowbasierte Client-Server-Architekturen

Sascha Hunold

`hunold@uni-bayreuth.de`

Universität Bayreuth

Dagstuhl, 15. Februar 2007

The logo for TransBS, featuring the word "Trans" in blue, "BS" in red, and a blue arrow pointing from the "s" to the "B".

- **Universität Bayreuth**
Lehrstuhl parallele und verteilte Systeme
Prof. Dr. Thomas Rauber
- **Berndt & Brungs Software GmbH**
Dipl.-Math. Jürgen Berndt
- **TU Chemnitz**
Professur Praktische Informatik
Prof. Dr. Gudula Rünger (Koordinator)
- BMBF Forschungsoffensive “Software Engineering 2006”,
Förderperiode: 2006–2009
- Themenfeld: **Pflege und Wiederverwendung von
Anwendungssoftware-Systemen**

Legacy-Software

- über Jahre gewachsenes Softwaresystem
- ständiger Wandel der Hard- und Softwaretechnologien

Problem des Managements

- **Überführung** der **erprobten** monolithischen Software in
- *kostengünstigere/effektivere* Software (Dokumentation, Modularität),
- mit *modernerer* Architektur (verteilt, Service-orientiert).

Vorhaben

- Transformation der monolithischen Legacy-Business-Software
 - funktions-, korrektheits- und merkmalerhaltend
- in Client-Server-Architektur mit folgenden Eigenschaften
 - komponentenbasierend
 - verteilt und skalierbar
 - Verwendung von Open-Source-Software

Kriterien

- Transformation: logische Struktur (Business-Software) erhalten
- konfigurierbare Workflows
- heterogen, verteilt

TU Chemnitz

- Funktionsserhaltendes **Transformationswerkzeug**

BBS GmbH

- exemplarisches **Business-Softwaresystem GBware** (General Businessware) → GBwareD – verteiltes Softwaresystem auf Client-Server-Basis

Universität Bayreuth

- offenes **Client-Server-Framework CBFrame**

Modellgetriebene Transformation

- **schrittweise** Überführung in verteilte Softwarekomponenten
- plattformunabhängiges Modell (PIM) → plattformspezifisches Modell (PSM) (domänenspezifische Sprachen (DSL), Xtext)
- Erzeugung von **Programmcode** mittels MDA-Werkzeugen (Template-Sprachen, Xpand)

Verteilte Workflowbearbeitung

- Anwendung des **Workflow-Referenzmodells** (WfMC)
- **Verteilte Realisierung**
- Einbeziehung bei der Erstellung verteilter Komponenten

Verteiltes Komponentenframework

- **offenes**, erweiterbares **Client-Server-Framework**
- Probleme der **Enterprise Application Integration** (CORBA, EJB)
- Sicherheitsaspekte
- **Effiziente** und korrekte **Umsetzung**
 - ☞ **Skalierbarkeit** und **Ausfallsicherheit**
- **Kompatibilität** von Transformationsausgabe und Komponentencontainern des Frameworks