

TransBS

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

Gudula Rünger

Fakultät für Informatik



TECHNISCHE UNIVERSITÄT
CHEMNITZ

BMBF-Statuskonferenz, 27.Juni 2006, Leipzig

Outline

- 1 Projekteinordnung**
 - Einleitung
- 2 Thematische Einordnung
 - Softwareansätze
- 3 Teilprojekte
 - Überblick
 - Transformation
 - Business Software
 - Verteiltheit
- 4 Kontakt

Projektpartner

[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"
Förderkennzeichen: 01 IS F10 A / B / C
Förderperiode: 2006–2009
- Themenfeld: **Pflege und Wiederverwendung von Anwendungssoftware-Systemen**

Fragestellung

Legacy-Problematik:

- existierendes, über Jahre gewachsenes Softwaresystem mit geschäftsnotwendigem Expertenwissen
- ständiger Wandel der Hard- und Softwaretechnologien

Häufige Fragestellung im **Management- und Software-Engineering**bereich:

- **kostengünstige und effektive** Überführung/Analyse/Dokumentation
- **erprobter** monolithischer Unternehmenssoftware (Geschäftslogik)
- in **moderne**, verteilte Systeme (service-orientiert)

Ansatz von TransBS

- 1 Funktions-, korrektheits- und merkmalerhaltende **Transformation**
 - 2 sich im Einsatz befindender, **monolithischer** Unternehmens-Softwaresysteme
 - 3 in eine moderne, komponentenbasierte, verteilte, skalierbare, auf Open-Source-Produkten basierende **Client-Server**-Architektur
- Transformation: Berücksichtigung der internen modularen und **logischen Struktur** der gegebenen **Business-Softwaresysteme**
 - Transformierte **Business-Softwaresysteme**: **konfigurierbare Workflows**
 - Unterstützung **heterogener**, verteilter Plattformen.

Softwaretechnische Unterteilung

- 1 Funktions-, korrektheits- und merkmalerhaltendes **Transformationswerkzeug** TransBS – Transformation von Business-Softwaresystemen → **TU Chemnitz**
- 2 exemplarisches Business-Softwaresystem **GBware** (**General Businessware**) → **BBS GmbH**
Transformation in GBwareD – verteiltes Softwaresystem auf Client-Server-Basis
- 3 offenes **Client-Server-Framework** CBFrame – Client-Server-Komponente für Business-Software
→ **Uni Bayreuth**

Outline

- 1 Projekteinordnung
 - Einleitung
- 2 Thematische Einordnung**
 - Softwareansätze
- 3 Teilprojekte
 - Überblick
 - Transformation
 - Business Software
 - Verteiltheit
- 4 Kontakt

Transformation von Altsoftware

Existierende Ansätze:

- sprach- und paradigmenpezifische Einzelansätze zur Modularisierung
- Überführung in Objektorientierung
- Erkennen der Geschäftslogik

Neuere Ansätze:

- Aspekte der Verteiltheit
- Middleware-Lösungen zur Datenintegration

Problematik:

- Vielzahl verschiedener, inkompatibler verteilter Plattformen (CORBA oder EJB)
- führt zu Neuimplementierungen derselben Geschäftslogik für verschiedene Plattformen

Model Driven Architecture (MDA)

modellbasierter Ansatz zur **schrittweisen** Erzeugung verteilter komponentenbasierter Software.

- 1 UML-Spezifikation statischer Schnittstellen und dynamischen Verhaltens der Komponenten — **plattformunabhängigen Modell** (PIM)
- 2 Erzeugung eines **plattformspezifischen Modells** (PSM)
Abbildungen in Form von UML-Erweiterungen, etwa CORBA- oder Enterprise Applications Integration (EAI) Profiles
- 3 MDA-Werkzeuge erzeugen **Programmcode** für die spezifische Plattform

Die Behandlung von **Legacy-Software** wurde bisher bei diesen **Transformationsschritten** nicht berücksichtigt.

→ [Beitrag von TransBS](#)

Verteilte Workflowbearbeitung

- bisher meist keine verteilten Workflows
- **Standard der WfMC** – Softwareerstellung für parallele und verteilte Prozesse und Workflowbearbeitung
- Implementierung im Rahmen von **RAFEG** (BMBF-Projekt: **Petersohn,NHConsult / Scheer,DFKI / Runger,TUC**)

Beitrag von TransBS:

- Transparenz der Verteiltheit:
explizites Wissen uber kostenintensive **Remote**-Anfragen
- direkte Einbeziehung von **Workflows** in Technologien **verteilter Komponenten**
- Kombination von Workflow-Technologie und **Legacy-Software**

Zusammenführung

verschiedener Forschungsgebiete:

- Modernisierung von **Legacy-Software**
- **heterogene** , verteilte Plattformen
- explizite **Verteiltheit** (Kommunikation)
- **Workflow**-Technologie (parallel und verteilt)
- **inkrementeller** Transformationsprozess
- compilerbasiertes **Transformationswerkzeug**
- verteilte Softwareentwicklung
- **Korrektheit**

Outline

- 1 Projekteinordnung
 - Einleitung
- 2 Thematische Einordnung
 - Softwareansätze
- 3 Teilprojekte**
 - Überblick
 - Transformation
 - Business Software
 - Verteiltheit
- 4 Kontakt

Wissenschaftliche und technische Ziele

1 TransBS:

- **Konzeption** der Transformationsmethodik
- Entwurf der TransBS-Architektur mit Schnittstellen für Daten- und Programmformate
- Prototypische **Realisierung**

2 GBware/GBwareD:

- **Analyse** der gegebenen Software
- Modularisierung der gegebenen Software
- Anforderungskatalog an die neu zu erstellende Software GBwareD und **Erzeugung**

3 CBFrame:

- **Konzeption** eines offenen Framework
- Entwurf der Architektur
- **Realisierung** eines Prototyps

Grundprinzipien des Transformationswerkzeuges

- **sprachorientierte Spezifikation** für interagierende Module (Spezifikationskomponenten für Softwarearchitekturen und deren Feinstrukturierung);
- automatisierter **inkrementeller** Transformationsprozess der Koordinationsebene als **mehrstufiger Prozess** bestehend aus
 - **Grob- und Feinstrukturabbildung der Softwarearchitekturen**
 - **flexibler Filtermechanismus** zur Konkretisierung der **Komponentenverteilung**
- Spezifikationskonzept zur Beschreibung von konkreten **Transformationsszenarien**;
- Nutzung von Methoden des **Übersetzerbaus**
- **Korrektheits-** und Bewertungsmechanismen

GBware (General Businessware)

- erfolgreich sich im Einsatz befindendes, **komplexes Unternehmens-Softwaresystem** der **Berndt & Brungs Software GmbH**
- Weiterentwicklung von GBware zu GBwareD (distributed):
 - Client-Server-Architektur, Services
 - Möglichkeiten der kundengesteuerten Konfigurierbarkeit
 - Einbeziehung aktueller Software-Engineering-Gesichtspunkte (Komponentenorientierung oder Sicherheit)

Anforderungen

an die zu erzeugende Software:

- Beibehaltung der **bisherigen Funktionalität** (Teilsysteme)
- flexible **verteilte Abarbeitungsmöglichkeit** auf heterogenen Plattformen
- Integration von Verteiltheit und **Crosscutting- Komponenten** (Zugriffsstrukturen, Synchronisation, Koordination)
- Beibehaltung der **Softwaremerkmale** (Flexibilität, Modularität, Erweiterbarkeit)
- **Skalierbarkeit** mit dynamischer Erweiterbarkeit und Anpassung an kundenspezifische Anforderungen;
- **Ausfallsicherheit** durch redundante Datenhaltung mit Datenkonsistenz bei verteilter Manipulation.

Client-Server-Framework CFrame

- **offenes**, erweiterbares Client-Server-**Framework**
- **Software-Komponente** der zu erzeugenden Unternehmenssoftware im Client-Server-Stil
- Statische Rahmen und **konfigurierbare** Komponenten
- alle wesentlichen Aspekte der Verteiltheit und Sicherheit sind im Rahmen CFrame enthalten
- **effiziente hardwarenahe** Realisierung
- **Entkopplung** der **Transformationssoftware** von gegebener Software (z.B. **GBware**) und Zielsoftware (z.B. **GBwareD** mit **CFrame**)
- **Schnittstelle und Interaktion** des **CFrames** mit dem **Transformationswerkzeug**
- Zusammenspiel mit der zu **transformierenden Software**

Outline

- 1 Projekteinordnung
 - Einleitung
- 2 Thematische Einordnung
 - Softwareansätze
- 3 Teilprojekte
 - Überblick
 - Transformation
 - Business Software
 - Verteiltheit
- 4 Kontakt**

Projektwebseite

<http://www.transbs.de>

<http://www.tu-chemnitz.de/informatik/PI/transbs>