

Repositorygestützte Erkennung von Schnittstellen in einem hochintegrierten Anwendungssystem

Johannes Bach, Max Doppler, Martin Schulze
Debeka-Hauptverwaltung

REPRO-Workshop Software-Migration
Chemnitz, 23. - 24. November 2006

- ◆ Kernsystem der Debeka
- ◆ Software-Repository
- ◆ Teilsystembildung und Brückenvermeidung
- ◆ Fazit

Kernsystem der Debeka

- ◆ Bull-Großrechner / Betriebssystem GCOS8
- ◆ Fachanwendungen einzelner Unternehmen
- ◆ Zentrale Serviceanwendungen
 - ◆ Verwaltung von Mitgliederdaten
 - ◆ Inkasso / Exkasso
 - ◆ Personalverwaltung
- ◆ Datenintegration mit Hilfe von Zugriffsroutinen
- ◆ Hoher Integrationsgrad
 - ◆ Beispiel Partneranwendung ZP1006

JeT - ZP1006 STAMMAUSKUNFT

Datei Bearbeiten Navigation Extras Entwicklung Hilfe

DIBAS Test

ZP1006 STAMMAUSKUNFT UB [] FC [] 17.10.06 16:07

EB 1000003.3 STAND 28.08.2006 ÄART: Pers-01 SB: PERSNR: []

GS: 030 RLT: 001 BZ: 10 GÜLTIG AB: 29.05.2003 ANGS: [] BETREUER: PERSNR: []

ADRESSE: ----- DATENSPERRE (PNR 04) -----

[] tel Chaussures DST: []

[] en TEL: [] 78 P

PERSONEN: --GWG-DATEN (PNR 01)----- GEBDAT----- G-FS-NAT-BST-BGR--OF-A

<input type="checkbox"/>	01	[]	02.03.1945	W	D	W	
<input type="checkbox"/>	07	Sechzigjähriger Test	01.06.1942	M	D	W	
<input type="checkbox"/>	08	Beihilfewegfall Test	20.08.1976	M	D	W	
<input type="checkbox"/>	09	Fünfundanzig Test	20.04.1979	M	D	W	
<input type="checkbox"/>		+					

KONTO: -KTONR-----BLZ-----BANK-----KTOINH-ABWEICH.KONTOINH-----

+04 [] 57051001 Kr Spk Westerwald 02 05 []

VERTRAG: -VERTNR--UART--UN--UP--BEITRAG-Z-KTO-----SUMME-W-WDAT--GRUND

<input type="checkbox"/>	KU	[]		07	07			09.02
<input type="checkbox"/>	KU	[]		07	07			09.02
<input type="checkbox"/>	AH	[]	F	01	01	5,22	M	E 06.04 AKTIU
<input type="checkbox"/>	AH	[]	F	02	02	5,22	M	E 10.06 ANTRG
<input type="checkbox"/>	RS	[]	E	01		218,38	J 01	E 01.07 ÄNDRG
<input type="checkbox"/>	NB	[]	ABN	01		162,40	E 01	100.000 E 05.05 ÄNDRG
<input type="checkbox"/>	NB	[]	ABN	01		0,00	E 01	100.000 E 08.07 ÄNDRG

ABGEGANGENE_VERTRAEGE_(AH_RS_NB)_ZUR_SERVICENUMMER_VORHANDEN

Pos: 10 OK

- ◆ Personaldaten
- ◆ Adressdaten
- ◆ Personendaten
- ◆ Kontodaten
- ◆ Vertragsspiegel

Software-Repository

- ◆ Ziel:
 - ◆ Zentrale Informationsquelle für die Entwickler und für Projekte
 - ◆ Abbildung der Softwarelandschaft
 - ◆ Grundlage für Auftragswesen und Produktionsübernahme
- ◆ Inhalt: ca. 2,8 Mio. Objekte
 - ◆ Software-Module: Programme, Copybooks, Jobs, Masken, permanente Dateien und deren Beziehungen (ca. 110.000 Module, grobe/mittlere Granularität)
 - ◆ Datenbank-Schemata, Datei-Strukturen (feine Granularität)
 - ◆ Entwicklungs-Aufträge, Stammdaten (ca. 70.000 Aufträge)
 - ◆ Quelltexte, Moduldokumentationen (ca. 215.000 Verweise auf externe Dateien und Office-Dokumente)

◆ Technik:

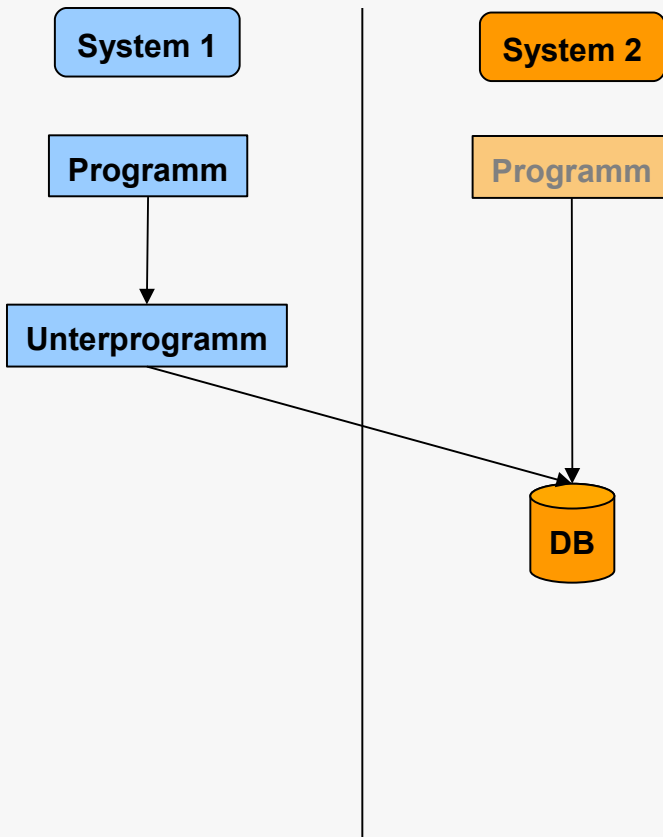
- ◆ Oracle-Datenbank
- ◆ Repository-API (Java): Objekt-Sicht, Konsistenzsicherung, Pfadsprache, kaskadierende Deaktivierung und Löschung von Objekten
- ◆ Java-Applikationen (Eclipse-RCP, Swing, Kommandozeile)

◆ Werkzeuge:

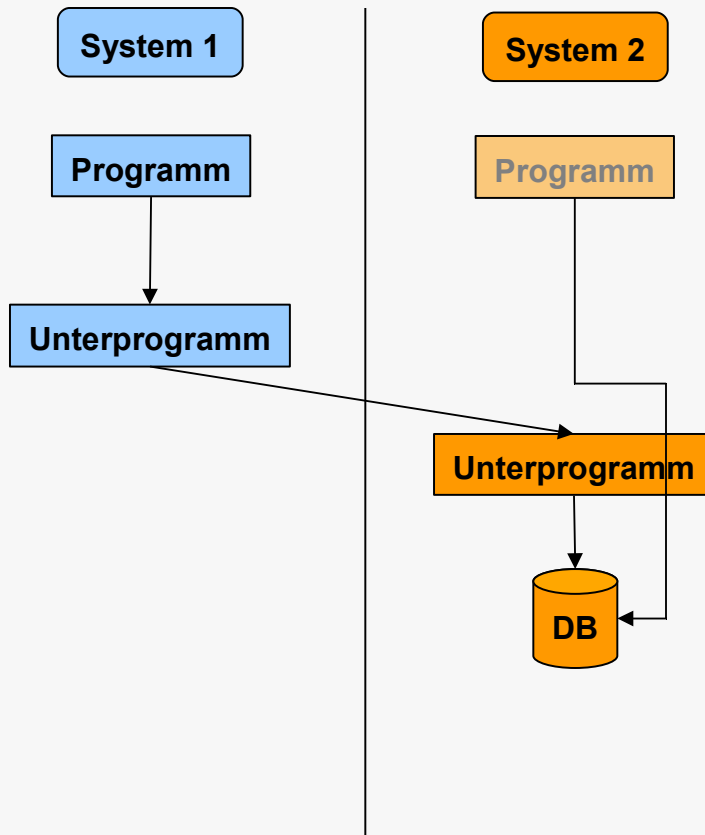
- ◆ ZEDERBrowser (siehe Tooldemo)
- ◆ ZEPRA (Auftragswesen)
- ◆ Scanner, Importer (Befüllung/Update bei der Produktionsübernahme)
- ◆ DB-Admin-Tools
- ◆ Java-API für weitere Anwendungen (z.B. hier zur Brückenanalyse)

Teilsystembildung und Brückenvermeidung

- ◆ Phasenweise Migration des Kernsystems
- ◆ Bilden von Teilsystemen
 - ◆ Namenskonventionen
- ◆ Voraussetzungen
 - ◆ Aus Zielumgebung nur lesender Zugriff
 - ◆ Auf Zielumgebung zusätzlich schreibender Zugriff
 - ◆ Keine plattformübergreifenden Datenbankzugriffe eines Moduls
- ◆ Ziel
 - ◆ Möglichst wenig Brücken
 - ◆ Technisch nicht mögliche Brücken beseitigen



- ◆ **Beschreibung**
 - ◆ Programm eines Teilsystems greift mittels DML-Befehl auf Datenbank eines anderen Teilsystems zu
- ◆ **Probleme**
 - ◆ Technisch nicht möglich
 - ◆ Teilsysteme müssten gleichzeitig migriert werden

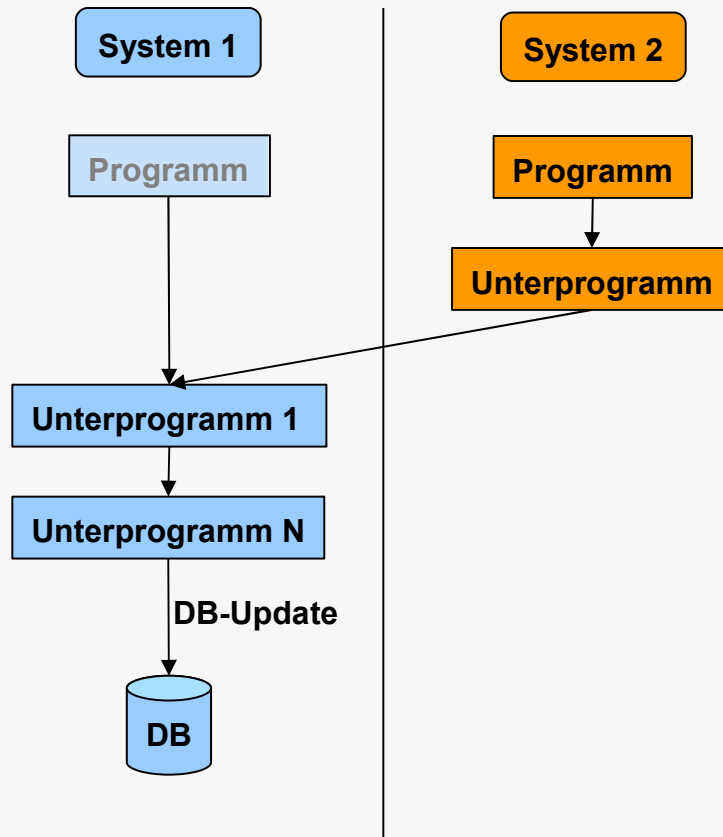


◆ Erkennung

- ◆ Alle Hauptprogramme aus Teilsystem ermitteln
 - ◆ Beispiel: Alle Programme die mit ZP beginnen
- ◆ Alle verwendeten Datenbanken ermitteln, die nicht zu diesem Teilsystem gehören

◆ Beseitigung

- ◆ Datenbankzugriffe durch Schnittstellen ersetzen

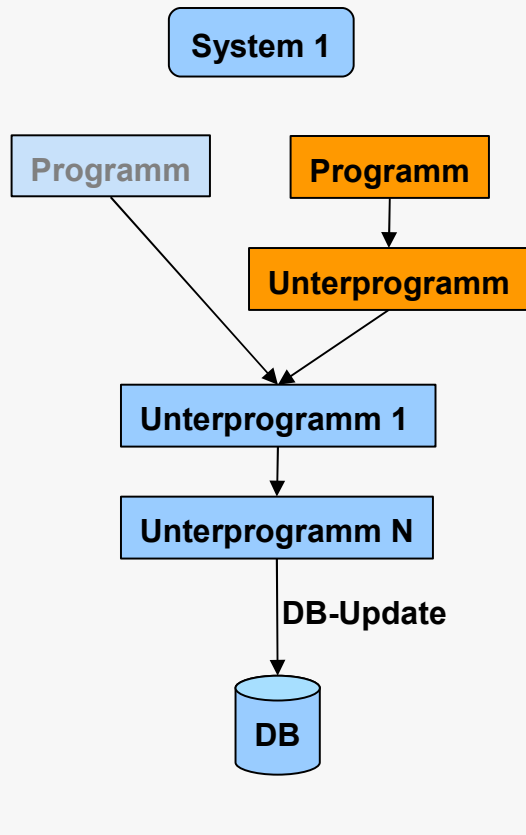


◆ Beschreibung

- ◆ Aufruf von Unterprogramm aus Zielsystem
- ◆ Unterprogramm macht Datenbankupdate

◆ Probleme

- ◆ Technisch nicht möglich
- ◆ Aufrufende Programme müssten gleichzeitig mit Unterprogramm migriert werden

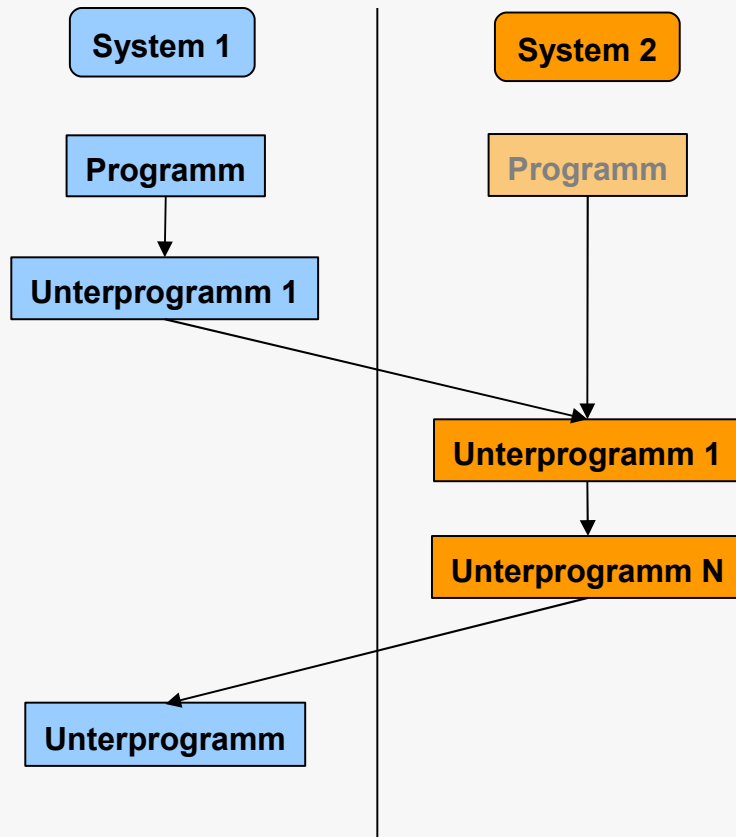


◆ Erkennung

- ◆ Alle Hauptprogramme aus Zielsystem ermitteln
 - ◆ Beispiel: Alle Programme, die mit ZP beginnen
- ◆ Aufgerufene Unterprogramme des anderen Systems ermitteln
 - ◆ Beispiel: Alle Programme, die mit CKV beginnen
- ◆ Kontrolle, ob schreibender DB-Zugriff in Unterprogrammen

◆ Beseitigung

- ◆ Unterprogrammaufruf verhindern (wenn möglich)
- ◆ Haupt- und Unterprogramme später migrieren

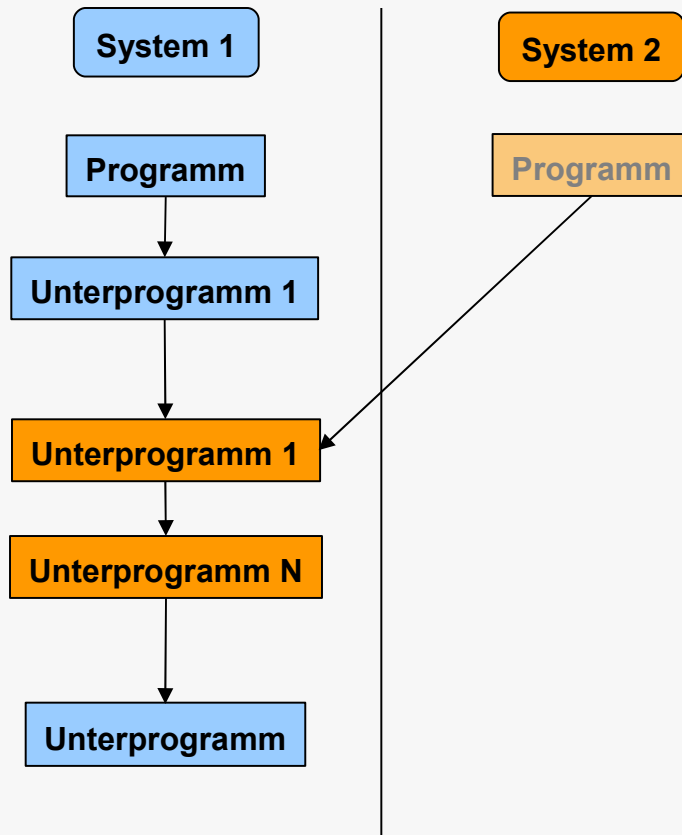


◆ Beschreibung

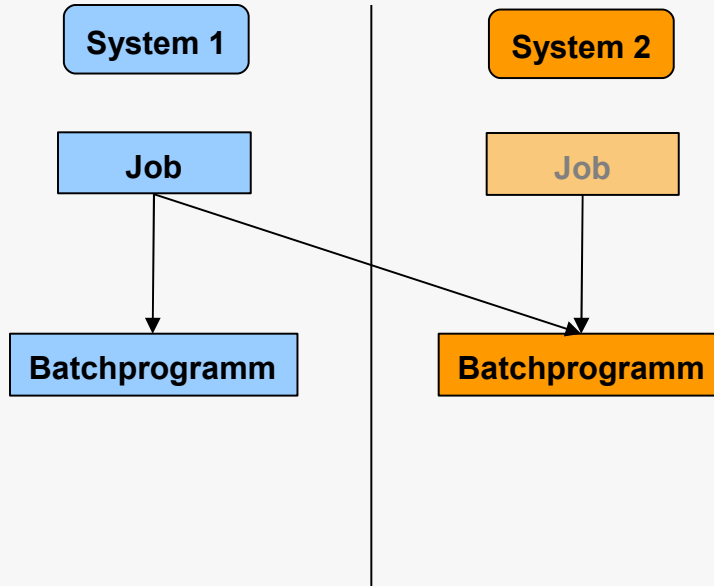
- ◆ Plattform wird während einer Transaktion mehr als einmal gewechselt

◆ Probleme

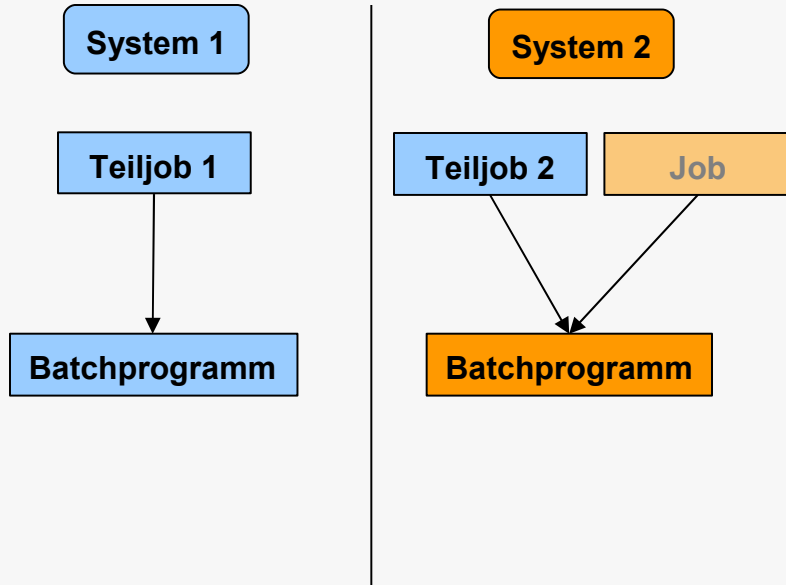
- ◆ Transaktionssicherheit gefährdet
- ◆ Nur Two-Phase-Commit möglich



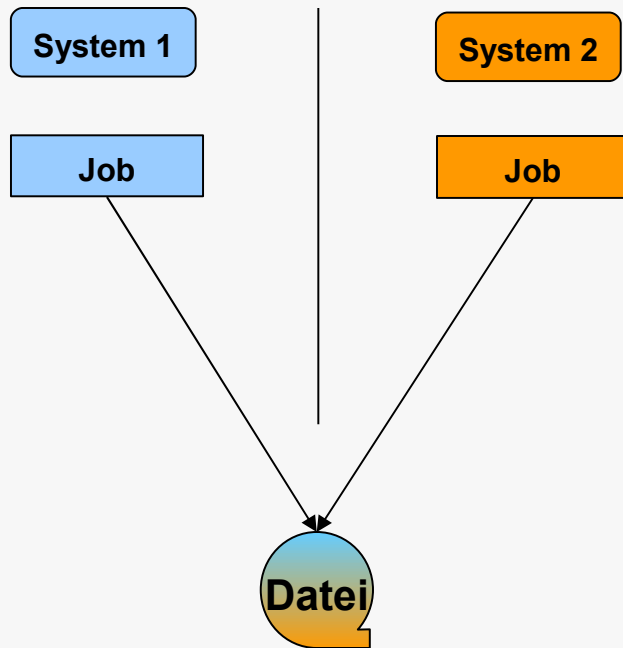
- ◆ **Erkennung**
 - ◆ Aufrufe von ermittelten Brücken bestimmen
 - ◆ Aufrufe in andere Richtung vorhanden?
- ◆ **Beseitigung**
 - ◆ Ausschließen, dass Rücksprünge tatsächlich durchlaufen werden
 - ◆ Module modifizieren, dass Rücksprünge vermieden werden
 - ◆ Teilsystemzugehörigkeit von Modulen verändern



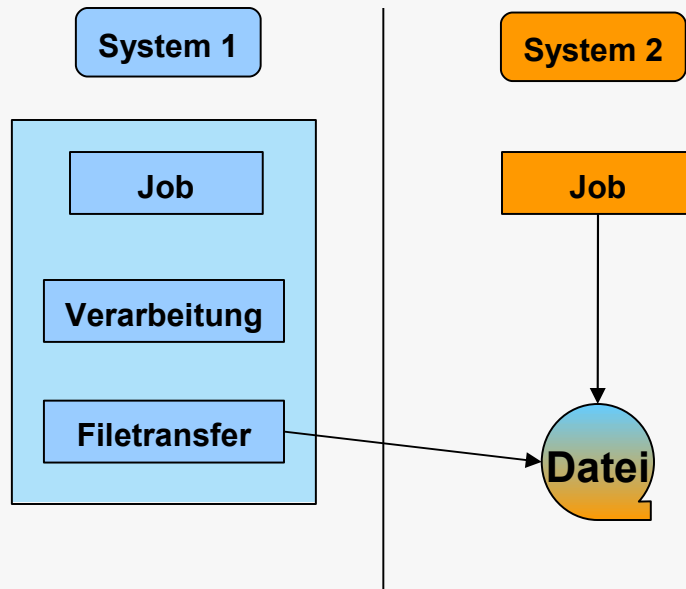
- ◆ **Beschreibung**
 - ◆ Job verwendet Batchprogramm eines anderen Teilsystems
- ◆ **Probleme**
 - ◆ Technisch nicht möglich



- ◆ **Erkennung**
 - ◆ Verwendete Hauptprogramme eines Jobs ermitteln
 - ◆ Kontrolle, ob alle Hauptprogramme im selben Teilsystem
- ◆ **Beseitigung**
 - ◆ Programmzuordnung ändern (wenn möglich)
 - ◆ Job auf beide Systeme aufteilen



- ◆ **Beschreibung**
 - ◆ Jobs aus zwei Teilsystemen greifen auf dieselbe Datei zu
- ◆ **Probleme**
 - ◆ Technisch nicht möglich



- ◆ **Erkennung**
 - ◆ Verwendete Dateien eines Teilsystems ermitteln
 - ◆ Verwendende Jobs aus anderem Teilsystem ermitteln
- ◆ **Beseitigung**
 - ◆ Job mit schreibendem Zugriff wird um FTP erweitert

Fazit

- ◆ Nutzen von Beziehungsinformationen
- ◆ Geeignete Anfragen
- ◆ Bildung möglichst unabhängiger Teilsysteme
- ◆ Detailanalyse
- ◆ Einleitung von Maßnahmen

- ◆ Vorstellung des ZEDERBrowsers bei der Tooldemo
- ◆ Fragen?