



Übung zur Vorlesung

# Internetanwendungstechnik

im Sommersemester 2008

S. Bissell, G. Mühl

## Übungsblatt 8

Ausgabe: 11. Juni 2008 – Abgabe: 25. Juni 2008

### Hinweise

Die Bearbeitung der Aufgaben erfolgt in Gruppen zu *maximal* 5 Personen. Die Lösungen sind als *PDF-Dokument* oder standardkonforme *HTML E-Mail* an [bissell@cs.tu-berlin.de](mailto:bissell@cs.tu-berlin.de) mit Betreff „[IAT08] Abgabe <Gruppennummer>-<Blattnummer>“ bis 23:59 Uhr des obigen Abgabetermins zu senden! (Z.B. lautet für Gruppe 01 der Betreff „[IAT08] Abgabe 01-8“.) Bitte führen Sie die Namen und Matrikelnummern *aller* Gruppenmitglieder *im* Abgabedokument auf und fügen Ihrer Abgabe ein *tar.gz* Archiv mit allen Quelldateien und Ressourcen an! Verspätete Abgaben werden *nicht* gewertet!

### Aufgabe 1 – CORBA Methodenaufrufe

CORBA bietet das Dynamic Invocation Interface (DII) an, um zur Laufzeit Aufrufe für beliebige Schnittstellen zu erzeugen. Das DII ist Teil der OMG Spezifikation. Die von den CORBA IDL Compilern generierten Stubs und Skeletons nutzen aber meistens ein so genanntes Static Invocation Interface (SII). Beide Schnittstellen, DII und SII (zusammen mit CORBA AMI), unterstützen sowohl synchrone als auch asynchrone Methodenaufrufe.

- Was ist der Unterschied zwischen dem SII und dem DII?
- Kann ein ORB auch ohne SII (also nur mit dem DII) auskommen? Begründen Sie Ihre Antwort!
- Interpretierte Sprachen wie Python oder TCL erlauben auch den Zugriff auf CORBA. Theoretisch kommen diese Sprachen ohne Stubs und Skeletons aus. Wie ist das möglich?
- Was ist der Unterschied zwischen synchronen und asynchronen Methodenaufrufen?
- Welche Möglichkeiten gibt es asynchrone Aufrufe mit dem SII und dem DII zu realisieren? Worin unterscheiden sie sich die einzelnen Realisierungsmöglichkeiten – was haben sie gemeinsam?

### Aufgabe 2 – CORBA Event Service

Betrachten Sie den CORBA Event Service und die vier möglichen Kombinationen von Push/Pull beim Erzeuger (Produzenten) und Verbraucher (Konsumenten) von Ereignissen.

- Charakterisieren Sie für jede Kombination die jeweilige Verarbeitungsweise!
- Geben Sie jeweils ein Beispiel eines typischen Anwendungsszenarios an!

### Aufgabe 3 – Chat-Applikation

Implementieren Sie eine *einfache* Chat-Applikation – bestehend aus Chat-Client und Chat-Relay – mit CORBA!

Ein Chat-Client kann sich bei einem Chat-Relay an- und abmelden. Nachrichten werden vom Client an das Relay geschickt, welches diese an alle angemeldeten Clients weiterleitet.

- a) Beschreiben Sie in CORBA IDL die benötigten Schnittstellen! Kommentieren Sie Ihre Schnittstellendefinition!
- b) Implementieren Sie die Chat-Applikation!
- c) Erweitern Sie Ihre Implementation aus b), so dass Sie mindestens einen CORBA Service (z.B. Naming) nutzen, der in der Vorlesung vorgestellt wurde!

*Hinweise:* JacORB (<http://www.jacorb.org>) ist eine freie Java-Implementierung des CORBA Standards. Die JacORB-Distribution enthält u.a. eine Dokumentation zur Installation sowie ein Tutorial mit vielen kommentierten Codebeispielen. JacORB ist auch im KBS-Pool installiert.

Eine elegante Lösung der Teilaufgabe c) ist die Realisierung des Chat-Relays mit Hilfe des CORBA Event Service. Beachten Sie, dass dies größere Auswirkungen auf Ihre Schnittstellendefinitionen haben kann.