Objekt-Adapter

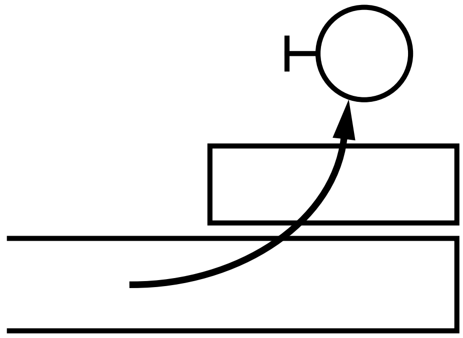
Kay Römer
Institut für Informationssysteme
ETH Zürich

Eidgenössische
Technische Hochschule
Zürich

Überblick

- Einordnung in die CORBA-Architektur
- Motivation
- Terminologie
- Funktionen eines Objekt-Adapters
- Basic Object Adapter (BOA)
- Portable Object Adapter (POA)
- Beispiel: CORBA-Datenbankanbindung

Einordnung



Serverseitige Anbindung von Objekt-Implementierungen an den ORB

Warum Objekt-Adapter?

- Verschiedene Typen von Objekt-Implementierungen (Bibliothek, separate Server, “Datenbank-Objekte”) erfordern spezifische Anbindung an den ORB
- Kleiner und einfacher ORB durch speziell zugeschnittene Objekt-Adapter

Terminologie 1

- **CORBA-Objekt**: Identität + Zustand + Operationen
- **Objekt-Referenz**: “Adresse” eines CORBA-Objektes
- **Objekt-ID**: Identität eines Objektes
- **Servant**: Realisierung eines CORBA-Objektes in einer Programmiersprache
- **Skelett**: verbindet Servant mit Objekt-Adapter (vom IDL-Compiler generiert)

Terminologie 2

- **Aktivierung**: “Einschalten” eines CORBA-Objektes
- **Deaktivierung**: “Ausschalten” eines CORBA-Objektes
- **Verkörperung**: Zuordnen eines Servants zu einem CORBA-Objekt
- **Vergeistigung**: Abtrennung eines Servants von einem CORBA-Objekt

Terminologie: Beispiel

- **CORBA-Objekt**: Bankkonto
  - **Identität**: Kontonummer
  - **Zustand**: Kontostand
  - **Operationen**: Abheben, Einzahlen, ...
- **Servant**: Bankangestellter
- **Deaktivierung**: Konto sperren
- **Aktivierung**: Sperre aufheben
- **Verkörperung**: Bankangestellter kommt zum Schalter und führt Operationen auf dem Konto aus
- **Vergeistigung**: Bankangestellter wendet sich anderen Dingen zu...

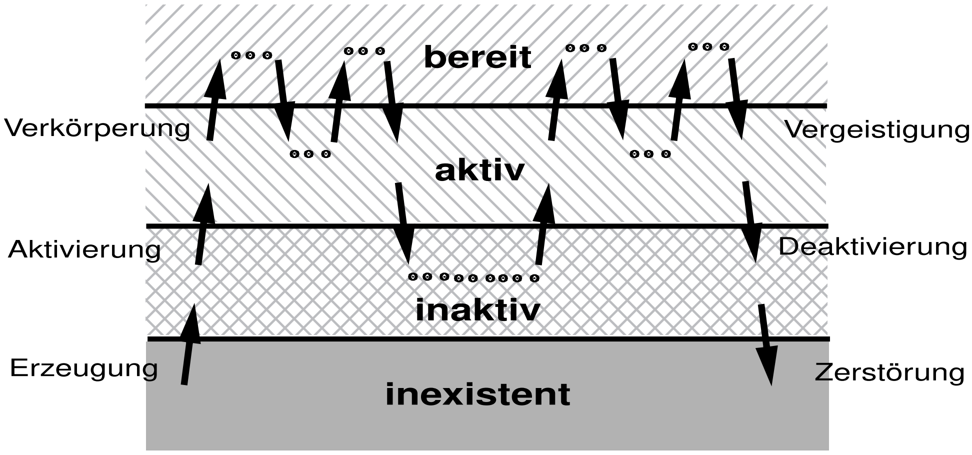
Funktionalität 1

Erzeugung von Objekt-Referenzen:

- In Kooperationen mit dem ORB
- Enthalten folgende Informationen:
  - Objekt-ID (Identität)
  - Adresse (IP-Adresse, TCP-Port)
  - Objekt-Adapter-Name
  - zusätzliche Infos

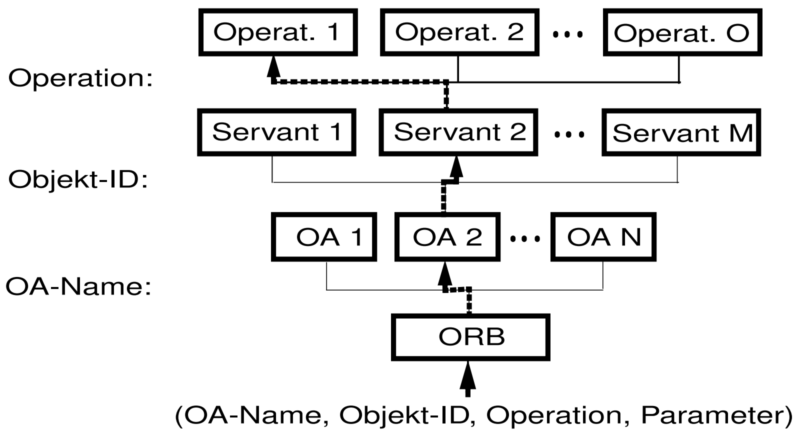
Funktionalität 2

Verwaltung Objekt-Lebenszyklus:



Funktionalität 3

Request Dispatching:

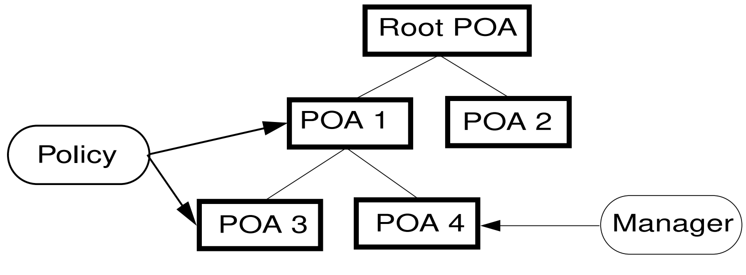


Basic Object Adapter

- “General Purpose” OA
- Automatisches Starten von Serven (verschiedene Modi)
- Viele wichtige Features fehlen (persistente Objekte, mehrere BOA-Instanzen, Konfigurierbarbeit, etc.)
- BOA-Anwendungen nicht portabel, da Spezifikation sehr unvollständig
- Ersetzt durch POA in CORBA 2.2

Portable Object Adapter

- POA-Anwendungen portabel, da Spezifikation detailliert
- Mehrere POAs pro ORB
- Weitgehend konfigurierbar durch *Policies* und *Manager*



POA: Policies

Regeln ein bestimmtes Verhaltensdetail durch Auswahl einer Option aus einer vordefinierten Menge von Alternativen:

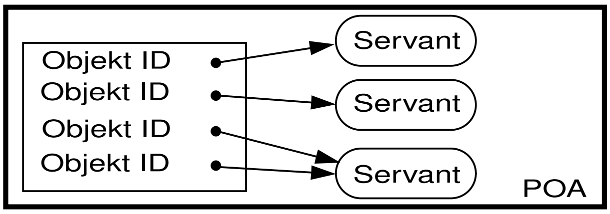
- **Objekt-Lebensdauer** (transient, persistent)
- **Objekt-ID-Erzeugung** (System, Benutzer)
- **Objekt-Aktivierung** (implizit, explizit)
- **Multi-Threading**
- **Servant-Wahl** (Active Object Map, Default, Manager)

POA: Manager

Regeln ein bestimmtes Verhaltensdetail durch Bereitstellung von benutzer-definiertem Code:

- **POA Manager**: Verwaltung von POAs (Erzeugung, Zerstörung, etc.)
- **Servant Manager**: Verwaltung von Servants (Erzeugung, Zerstörung, etc.)

POA: Active Object Map



Zuordnung von Servants zu Objekten:

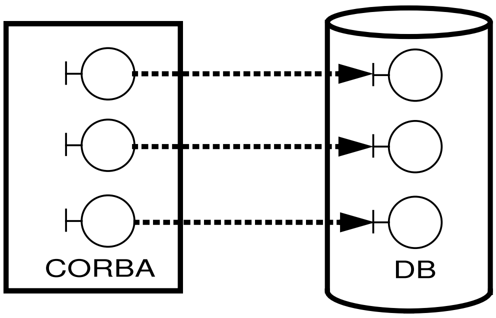
- Ein Servant für eine Objekt-Instanz
- Ein Servant für mehrere Instanzen des gleichen Objekt-Typs
- Ein Servant für Instanzen verschiedener Objekt-Typen

Bsp: Datenbanken

- Aufgabe: Zugriff auf Datenbank-Objekte in einer CORBA-Umgebung
- Hauptproblem: Skalierbarkeit
- Lösungsansätze:
  - Wrapping
  - spezieller DB Objekt-Adapter
  - POA mit Default Servant

Wrapping

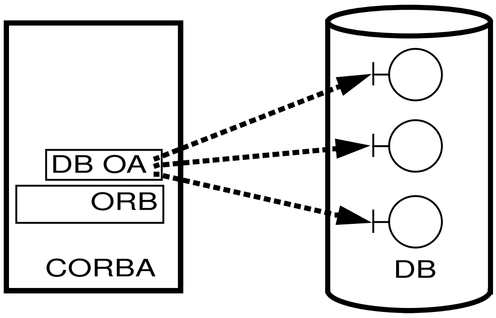
Für jedes Datenbank-Objekt existiert ein Wrapper-Objekt im CORBA System:



Problem: hoher Ressourcenverbrauch, schlechte Skalierbarkeit

DB Objekt-Adapter

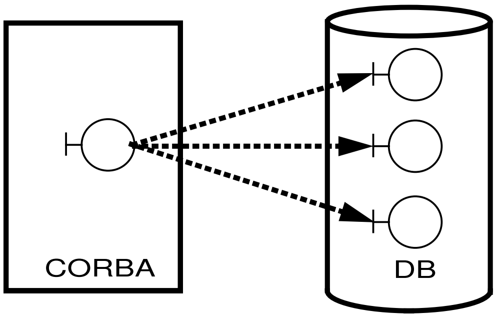
Spezieller Objekt-Adapter greift direkt auf DB Objekte zu:



Problem: benötigt speziellen DB OA, abhängig von ORB und DB

POA mit Def. Servant

Default Servant ist für Zugriff auf alle Datenbank-Objekte zuständig:



Skaliert gut, nur Standard-Software

