

Middleware-Systeme für Dienstqualität

Franz J. Hauck

Lehrstuhl für Betriebssysteme, Universität Erlangen-Nürnberg

fjh@cs.fau.de

<http://www4.cs.fau.de/~fjh/>

1 Überblick

- Middleware
 - ◆ Definitionen
 - ◆ Aufgaben
- CORBA
 - ◆ Architektur
- Dienstqualität
- Vergleichsfragen
- Zusammenfassung

2 Middleware

■ Was ist Middleware?

◆ **www.whatis.com:**

- ... term for any programming that serves to "glue together," mediate between, or enhance two separate and usually already existing programs.
- ...

◆ **Zitat in der Computer Zeitung:**

- ... Kommunikationsschnittstelle zwischen unterschiedlichen Technologien, Standards und Protokollen. ...

◆ **OODS I:**

- ... a piece of software between application and operating system. ...

2 Middleware (2)

■ Unser Fokus:

- ◆ Middleware zwischen Betriebssystem und Anwendung
- ◆ Verbindung zwischen Rechnern: Überwindung der Verteilung
- ◆ Überwindung verschiedener Betriebssysteme
- ◆ Überwindung verschiedener Rechnerplattformen
- ◆ Überwindung verschiedener Programmiersprachen

2 Middleware (3)

- Ausprägungen von Middleware-Systemen:
 - ◆ TP-Monitore
 - verteiltes Transaktionskonzept überwacht durch den TP-Monitor; Wiederholung und Abbruch von Transaktionen
 - ◆ Message-Oriented Middleware (MOM)
 - Versand von Nachrichten zwischen verteilten Anwendungskomponenten
 - ◆ RPC-basierte Middleware
 - entfernter Prozeduraufruf als Mittel zur Überbrückung der Verteilung
 - ◆ DB-basierte Middleware
 - (zentrale) Datenbank als Bindeglied für verteilte Anwendungen; Zugang über durch Middleware definierte allgemeine Schnittstelle
 - ◆ objektbasierte Middleware
 - verteilte Objekte und entfernter Methodenaufruf als Mittel zur Überbrückung der Verteilung

3 Aufgaben von Middleware-Systemen

- Programmiermodell für verteilte Anwendungen
 - ◆ gleich auf allen Plattformen, Betriebssystemen und für alle unterstützten Programmiersprachen
 - ◆ Programmierereinheiten, die verteilt werden und entfernt angesprochen werden können (z.B. verteilte Objekte, zentrale Datenbank)
 - ◆ Dienste zur Lokalisierung, Benennung und Verwaltung dieser Einheiten

4 CORBA

- Common-Object-Request-Broker-Architecture
 - ◆ Standard der OMG (Object Management Group)

- Verteilte Objekte als Verteilungseinheiten
 - ◆ **Verteilungstransparenz**
Objekte können lokal oder verteilt sein
 - ◆ **Ortstransparenz**
für den Aufrufer ist der Ort des Objekts nicht relevant
 - ◆ **Relokationstransparenz**
Objekt kann während der Benutzung seinen Ort wechseln (migrieren)

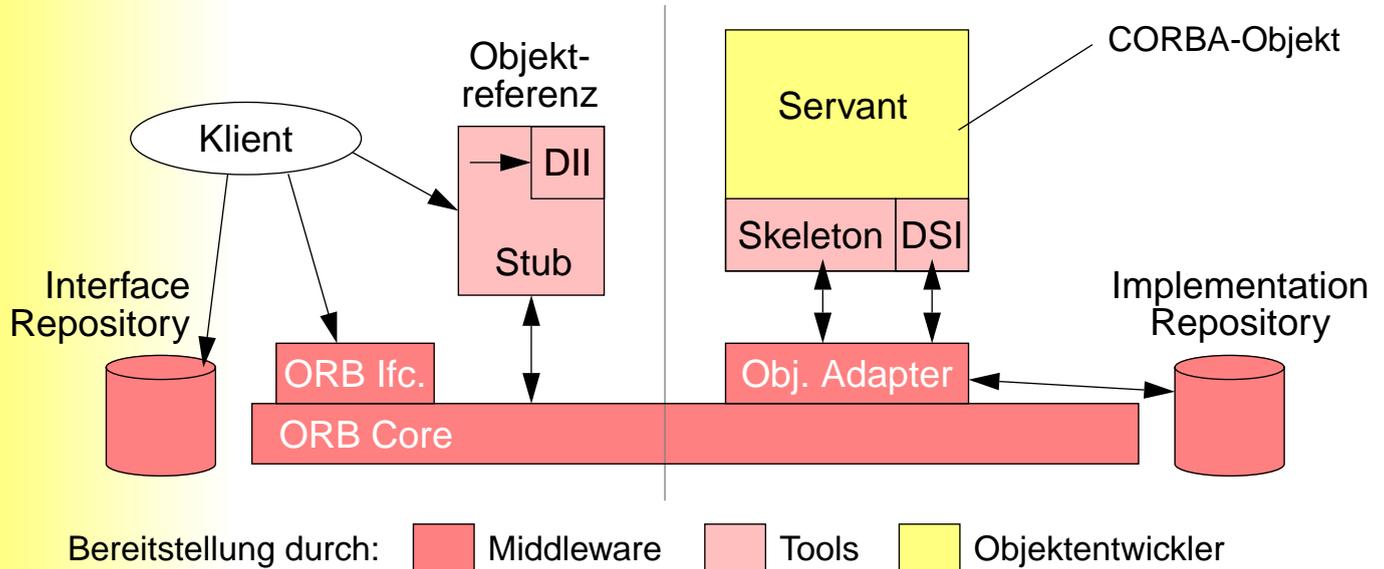
4 CORBA (2)

- Dienste
 - ◆ Namensdienst (Benennung)
 - ◆ Interface-Repository (Bereitstellung von Metainformationen über Schnittst.)
 - ◆ Implementation-Repository (Migrations- und Wartungsunterstützung)

 - ◆ Persistenz
 - ◆ Transaktionen
 - ◆ etc.

4 CORBA (3)

Architektur



5 Dienstqualität

- Ursprung im Netzwerkbereich
 - ◆ Bandbreite, Fehlerrate, Latenz, Jitter als typische Merkmale

- Weitere „Qualitäten“
 - ◆ Fehlertoleranz
 - ◆ Vertraulichkeit
 - ◆ Redundanz
 - ◆ Aktualität von Daten
 - ◆ Genauigkeit von Daten
 - ◆ Rechtzeitigkeit von Daten
 - ◆ Berechnungsdauer

- ★ Dienstqualität weitgehend unabhängig von der funktionalen Diensteschnittstelle

5 Dienstqualität (2)

- Anforderungen des Klienten an einen Dienst
 - ◆ Vorgaben
 - ◆ Verhandlung
 - ◆ Einigung

- Implementierung der Anforderungen
 - ◆ Best-effort (so gut wie möglich)
 - ◆ Statistische Garantie
 - ◆ Deterministische Garantie

 - ◆ Reservierung von Betriebsmittel
 - ◆ Implementierung spezifischer Algorithmen

5 Dienstqualität (3)

- Unerfüllbare Anforderungen
 - ◆ **statisch**: Abweisung der Anforderung
 - ◆ **dynamisch**: Benachrichtigung des Klienten, erneute Verhandlung o. Ä.

- Betrachtung verschiedener Middleware-Systeme, die Dienstqualitäten unterstützen
 - ◆ Unterstützung für Klienten:
 - Spezifikation von Qualitätsmerkmalen, Verhandlung
 - ◆ Unterstützung von Dienstentwicklern:
 - Implementierung von qualitätsbewussten Komponenten

6 Vergleichsfragen

- ★ Welche Kategorien von Dienstqualität werden unterstützt?
 - ◆ z.B. Echtzeitgarantien, Redundanz, Datenaktualität

- ★ Gibt es eine dedizierte Schnittstelle für eine Qualitätsvereinbarung?
 - ◆ Stellt das System eine solche Schnittstelle bereit oder ist das Sache der Anwendung?

- ★ Welche Instanzen sind von einer Vereinbarung betroffen?
 - ◆ Worauf wirkt sich die Vereinbarung aus? Welche Granularität ist vorgesehen?

- ★ Rückmeldung über Nichterfüllbarkeit von Vereinbarungen?
 - ◆ Bekommt der Klient eine Rückmeldung und wie erfolgt sie?

6 Vergleichsfragen (2)

- ★ Automatische interne Adaptierung?
 - ◆ Kann der Dienst die Anforderungen nicht mehr erfüllen, kann er dann interne Maßnahmen vornehmen, um dennoch den Anforderungen gerecht zu werden?

- ★ Automatische Adaptierung bei Nichterfüllbarkeit?
 - ◆ Kann der Dienst die Anforderungen nicht mehr erfüllen, weiß er dann welche noch erfüllbaren Anforderungen dem Klienten am „nächstliebsten“ sind?

7 Zusammenfassung

- Middleware als Plattform für verteilte Anwendungen
- CORBA als objektbasierte Middleware
- Dienstqualität
- Vergleichsfragen