CORBA

- **C**ommon **O**bject **R**equest **B**roker **A**rchitecture

  - erste Version 1991: CORBA 1.1
  - 1994: CORBA 2.0 (Interoperabilität: IIOP)
  - derzeit aktuell: CORBA 3.0 (aber: konforme Produkte “dauern”)

  eine Architektur, kein Produkt!

- OMG (Object Management Group)

  - herstellerübergreifendes Konsortium
  - gegründet 1989 von 11 Firmen
  - jetzt mehr als 700 Mitglieder (wer eigentlich nicht? Selbst Microsoft ist dabei); vgl. http://www.omg.org/cgi-bin/membersearch.pl
  - Ziel: Bereitstellung von Konzepten für die Entwicklung verteilter Anwendungen mit objektorientierten Modellen
  - genauer: Definition und Entwicklung einer Architektur für kooperierende objektorientierte Softwarebausteine in verteilten heterogenen Systemen (--> “Middleware”)

- CORBA beruht i.w. auf der Idee der Realisierung von Softwaresystemen aus interagierenden Komponenten

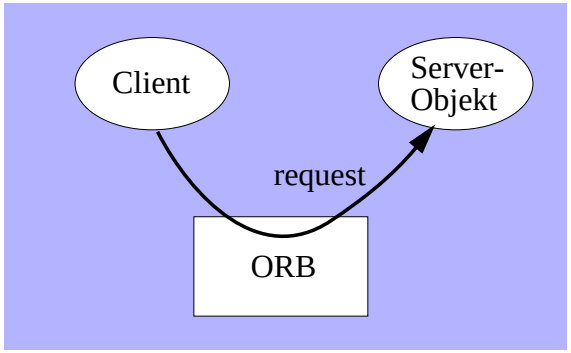
  + Services
  + Objektorientierung

Beachte: Objektorientierung selbst ist eigentlich ein “altes” Konzept

- Mitte der 60er-Jahre (Programmiersprache “Simula”)
- damals bereits fast alle Aspekte der Objektorientierung (Klassen-hierarchien, virtuelle Klassen, Polymorphismus...)

CORBA - Übersicht

- Objektmodell (mit Aufrufsemantik etc.)
- IDL mit entspr. Generatoren und Compilern
- Object Request Broker (ORB) als Vermittlungs-infrastruktur

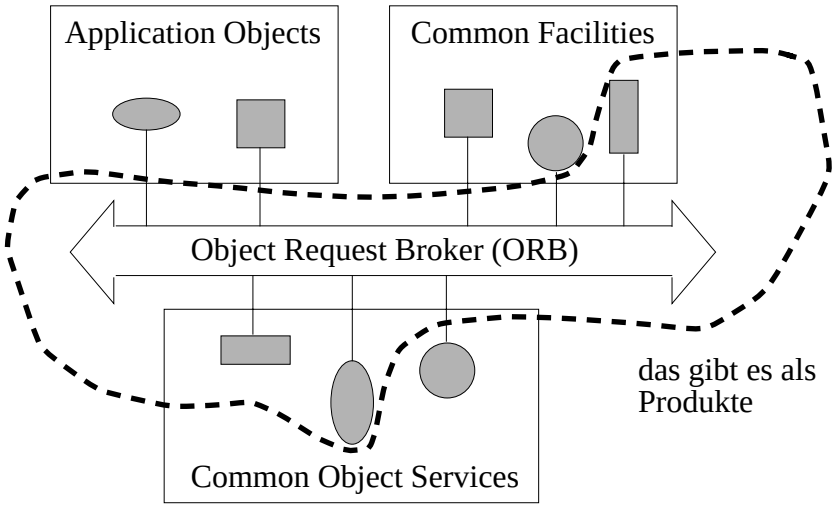


- Systemfunktionen in Form von Object Services
- Unterstützung von Anwendungen durch Common Facilities / Application Domains
- Konventionen bezüglich Schnittstellen, Protokollen etc.

- CORBA ist also eine Infrastruktur und unterstützt die Ausführung verteilter objektorientierter Systeme
- Die Entwurfs- und Spezifikationsphase solcher Systeme werden mit anderen Konzepten unterstützt, z.B. UML (“Unified Modeling Language”), mit der u.a. Diagrammnotationen standardisiert werden

Object Management Architecture

- “OMA” ist die Referenzarchitektur der OMG, die die wesentlichen Bestandteile einer Plattform für verteilte objektorientierte Applikationen definiert



- Application Objects: Objekte der eigentl. Anwendung
  - gehören damit nicht zur CORBA-Infrastruktur
- ORB: Vermittlung zwischen verschiedenen Objekten; Weiterleitung von Methodenaufrufen etc.
  --> Ortstransparenz, Kommunikation...
- Object Services: objektorienteirte Schnittstelle zu standardisierten wichtigen Diensten
- Common Facilities: allgemein nützliche Dienste
  - nicht notw. Teil aller CORBA-Implementierungen

Object Services (1)

- Basisdienste als systemweite Infrastruktur
  - Dienstleistungskomponenten besitzen eine objektorientierte Schnittstelle
- COSS (Common Object Services Specification)
  - Realisierung ist für CORBA-konforme Produkte verpflichtend
  - noch nicht alle Services sind derzeit vollständig spezifiziert

---

1) Ereignismeldung
   - Weiterleitung asynchroner Ereignisse an Ereigniskonsumenten
   - Einrichten von “event channels” mit Operationen wie push, pull...
2) Persistenz
   - Dauerhaftes Speichern von Objekten auf externen Medien
3) Naming
   - Erzeugung von Namensräumen
   - Abbildung von Namen auf Objektreferenzen
   - Lokalisierung von Objekten
4) Lifecycle
   - erzeugen, löschen, kopieren, verlagern... von Objekten
5) Concurrency
   - Semaphore, Locks,...
6) Externalization
   - Export von Objekten in “normale” Dateien

Object Services (2)

7) Transactions
- 2-Phasen-Commit etc.

8) Time
- Uhrensynchronisation etc.

9) Security

10) Licensing
- Management von Lizenzdiensten für Komponenten

11) Trading
- Matching von Services zu einer Service-Beschreibung eines Klienten

12) Replikation
- Sicherstellung der Konsistenz replizierter Objekte in einer verteilten Umgebung

- es gibt noch einige weitere Services...

Common Facilities

- Höherwertige Dienste für ein breites Spektrum von Anwendungsbereichen

- Bereitstellung allgemein interessanter Funktionalität
  - analog zu grossen Klassenbibliotheken

- Horizontal" Common Facilities
  - Basisfunktionalität, die für verschiedene Anwendungsbereiche von Nutzen ist
    - secure time
    - printing
    - mobile agents
    - internationalization ("...will enable developers to use an application in their own language using their own cultural conventions...will allow the developer to use a culture's numeric and currency conventions...)
    - user interface
    - information management (z.B. Speicherung komplexer Objektstrukturen, Formatkonvertierungen,...)
    - systems management (z.B. Installation, Konfiguration... von Objekten)
    - task management (Workflow, lange Transaktionen...)

- Vertikale Common Facilities ("Domain Facilities")
  - Basisfunktionalität für diverse Marktsegmente, z.B. Banken, Finanzdienste, Gesundheitswesen...
  - vgl. "application frameworks", "business objects" etc.

OMA Component Definitions

Object Request Broker - commercially known as CORBA, the ORB is the communications heart of the standard. It provides an infrastructure allowing objects to converse, independent of the specific platforms and techniques used to implement the objects. Compliance with the Object Request Broker standard guarantees portability and interoperability of objects over a network of heterogeneous systems.

Object Services - these components standardize the life-cycle management of objects. Interfaces are provided to create objects, to control access to objects, to keep track of relocated objects, and to control the relationship between styles of objects (class management). Also provided are the generic environments in which single objects can perform their tasks. Object Services provide for application consistency and help to increase programmer productivity.

Common Facilities - Common Facilities provide a set of generic application functions that can be configured to the specific requirements of a particular configuration. These are facilities that sit closer to the user, such as printing, document management, database, and electronic mail facilities. Standardization leads to uniformity in generic operations and to better options for end users for configuring their working environments.

Domain Interfaces - Domain Interfaces represent vertical areas that provide functionality of direct interest to end-users in particular application domains. Domain interfaces may combine some common facilities and object services, but are designed to perform particular tasks for users within a certain vertical market or industry.

Quelle: http://www.omg.org/omaov.htm