

History

- The CORBA (Common Object Request Broker Architecture) specification was developed in 1991 by the Object Management Group (OMG).
- The OMG was founded by eleven corporations to develop CORBA.
- CORBA 2.0 specification (introduced in 1996) permit communications between implementations made by different developers.

Goal

- The OMG's goal was to adopt distributed object systems that utilize object-oriented programming for distributed systems.
- Systems to be built on heterogeneous hardware, networks, operating systems and programming languages.
- The distributed objects would be implemented in various programming languages and still be able to communicate with each other.

Definitions

- CDR – Common Data Representation: A external data representation of all data types that can be used as arguments or return values in remote invocations.
- CORBA – Common Object Request Broker Architecture: An Object Request Broker (ORB) architecture specification for distributed object systems.
- GIOP – General Inter-ORB Protocol: CORBA 2.0 standard that enables different developer implementations to communicate with each other.
- IDL – Interface Definition Language: An interfacing programming language that that allow communication across a heterogeneous distributed system.
- IIOP – Internet Inter-ORB Protocol: Internet version of GIOP that use the TCP/IP protocol.
- Marshalling: The process of taking a collection of data items and assembling them into a form suitable for transmission in a message.
- Middleware: A software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems, and programming languages.
- OMG – Object Management Group: The organization that developed and maintains CORBA
- POS – Persistent Object Service: A persistent object store for CORBA where objects that live between processes activations can be stored.

Features

- CORBA consists of a language independent RMI
- Consists of a set of generic services useful for distributed applications.
- The CORBA RMI acts as a “universal translator” that permits client processes to invoke a method or process that may reside on a different operating system or hardware, or implemented via a different programming language.

CORBA RMI Features

- The CORBA RMI consists of the following main components:
 - An interface definition language (IDL)
 - An architecture (discussed in **Structure**)
 - The General Inter-ORB Protocol (GIOP)
 - The Internet Inter-ORB Protocol (IIOP)

CORBA IDL Features

- Provides an interface consisting of a name and a set of methods that a client can request.
- IDL supports fifteen primitive types, constructed types and a special type called *Object*.
 - Primitive types: *short, long, unsigned short, unsigned long, float, double, char, boolean, octet, and any*.
 - Constructed types such as arrays and sequences must be defined using *typedefs* and passed by value.
- Interfaces and other IDL type definitions can be grouped into logical units called modules.

GIOP and IIOP Features

- GIOP: General Inter-ORB Protocol are the standards (included in CORBA 2.0), which enable implementations to communicate with each other regardless of who developed it.
- IIOP: Internet Inter-ORB Protocol is an implementation of GIOP that uses the TCP/IP protocol for the Internet.

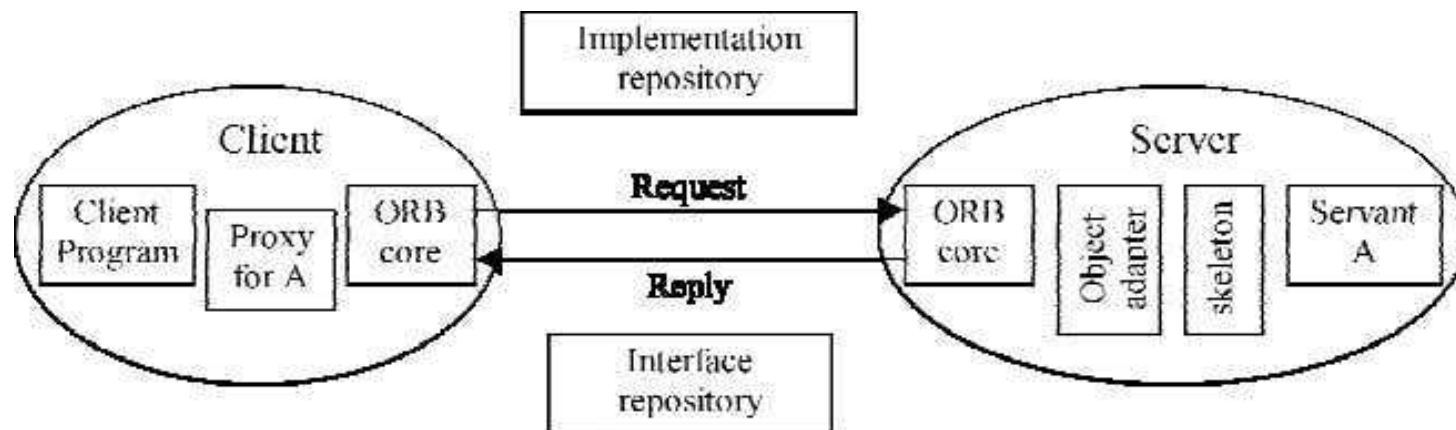
CORBA Services

- Set of generic service specifications useful for distributed applications. Each discussed in Chapter 17.3 and documentation provided @ OMG site.

- CORBA Naming Service - essential to any ORB
- CORBA Event Service - define interfaces
- CORBA Notification Service - extension of event service
- CORBA Security Service - controls access
- CORBA Trading Service - allows location by attribute
- CORBA Transaction and Concurrency Control Service
- CORBA Persistent Object Service

CORBA RMI Structure

CORBA RMI Structure



- Process can be both server and client to another server
- Ideal for a heterogeneous distributed system like the Internet.

Significant Points

- Harder to use than a single language RMI such as JAVA.
- Compatible with various languages and operating systems.
- CORBA's IDL is the crucial part because it permits communication on a heterogeneous distributed system.
- The OMG consists of several companies that maintain compatibility and standardization of the CORBA system.

Summary

- The OMG developed CORBA for distributed object systems that will work on a heterogeneous distributed system.
- CORBA consists of many parts and services that are useful for distributed applications.
- The IDL portion is the crucial part that makes CORBA a language independent middleware.
- CORBA includes generic service specifications useful in implementing distributed applications.