Broker Architectural Style CORBA

Presented by: Dr. R. BENTRCIA

Department of Computer Science, M'sila University

CORBA RUNNING STEPS

- The basic steps for CORBA development include:
  - Define an interface in IDL
    - The IDL provides the operating system and programming language independent interfaces to all services and components that are linked to the ORB. The IDL specifies a description of any services a server component exposes to the client.
  - Map the IDL interface to Java (done automatically):
    - The JDK1.3+ comes with the idlj compiler, which is used to map IDL definitions into Java declarations and statements.
  - Implement the interface
  - Develop the server
  - Develop the client
  - Run the naming service, the server, and the client.

CORBA RUNNING STEPS

- To develop the server:
  - Initialize the ORB
  - Create a the remote object (servant)
  - Register the object in the CORBA Naming Service (COS Naming)
  - Notify the CORBA environment that it is ready to receive client requests
  - Wait for incoming client requests
  - Two important packages:
    - import org.omg.CosNaming.*;
    - import org.omg.CORBA.*;

CORBA RUNNING STEPS

- To develop the client:
  - Once a reference to the naming service has been obtained, it can be used to access the naming service and invoke the methods,
- To run the application:
  - Run the CORBA naming service using the command *tnameserv*
  - Start the server
  - Generate stubs for the client
  - Start the client

RMI vs. CORBA

Characteristics	CORBA	RMI
Life cycle	Objects are not garbage collected	Objects are garbage collected automatically
Language Support	Supports multiple programming languages	Supports only Java
Architecture	Distributed object middleware based on the object request broker (ORB)	Client-server architecture based on remote objects
Complexity	More complex and requires more setup and configuration.	Simpler and requires less setup and configuration.
Security	Provides security services such as authentication, authorization, and encryption.	Provides security services such as authentication and encryption.
Performance	Generally slower due to the additional overhead of the ORB.	Generally faster due to the simpler architecture
Usage	Suitable for complex distributed systems with heterogeneous components	Suitable for simpler distributed systems built entirely in Java
Full Form	Common Object Request Broker Architecture (CORBA)	Remote Method Invocation (RMI)