Name: Sayyed Sohail Rashid	Course Name: DC-LAB
Class: BE-CO	Batch: 01
Roll no: 18CO48	Experiment No: 09

Aim : Case Study on Common Object Request Broker Architecture(CORBA).

Theory:

What is CORBA?

Common Object Request Broker Architecture: An Object Request Broker (ORB) architecture specification for distributed object systems.
CORBA consists of a language independent RMI consisting 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.

Features:

CORBA RMI Features:

The CORBA RMI consists of the following main components: –An interface definition language (IDL)

1. An architecture (discussed in Structure)
2. The General Inter-ORB Protocol (GIOP)
3. 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.

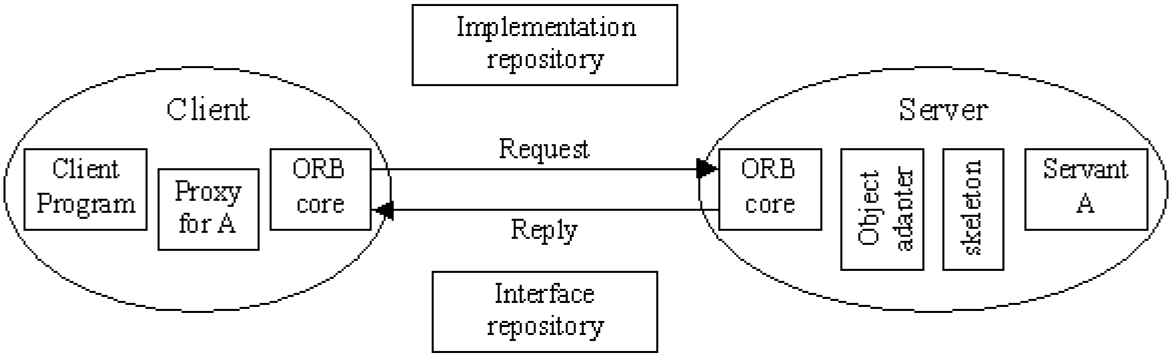
1. Primitive types: short, long, unsigned short, unsigned long, float, double, char, boolean, octet, and any.
2. Constructed types such as arrays and sequences must be defined using typedefs and passed by value. Interfaces and other IDL type definitions can be grouped into logical units called modules.

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:

1. CORBA Naming Service - essential to any ORB
2. CORBA Event Service - define interfaces
3. CORBA Notification Service - extension of event service
4. CORBA Security Service - controls access
5. CORBA Trading Service - allows location by attribute
6. CORBA Transaction and Concurrency Control Service
7. CORBA Persistent Object Service



CORBA RMI Structure

How to use CORBA:

Server must include IDL interfaces in the form of servant classes. An interface compiler generates:

1. the program(Java or C++) interfaces.
2. server skeletons for each IDL interface.
3. proxy classes (or client stubs) for each IDL interface.
4. A Java / C++ class for IDL defined struct.
5. helper classes for each IDL defined type.

Server:-

1. Creates and initializes the ORB
2. Creates an instance of servant class, which is registered with the ORB. Servant class extends the corresponding skeleton class and implementation methods of an IDL interface.
3. Makes a CORBA object Client
4. Creates and initializes the ORB
5. Contacts Naming service to get reference to the server

6. Invokes methods on the server

Applications of CORBA:

1. Used primarily as a remote method invocation of a distributed client – server system.
2. Can communicate between clients and servers on different operating systems and implemented by different programming languages (Java cannot do this).
3. Has many standards and services useful in implementing distributed applications.
4. Process can be both server and client to another server Ideal for a heterogeneous distributed system like the Internet.

Conclusion:

CORBA is 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.