|  |  |
| --- | --- |
| **Name:** SOHAIL RASHID SAYYED | **Roll No:** 18CO48 |
| **Class/Batch:** BECO/ I | **Course Name:** DC Lab |

**Experiment No: 10**

**Aim:** Case Study on Common Object Request Broker Architecture i.e. CORBA.

**Theory:**

The Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) designed to facilitate the communication of systems that are deployed on diverse platforms. CORBA enables collaboration between systems on different operating systems, programming languages, and computing hardware. CORBA uses an object-oriented model although the systems that use the CORBA do not have to be object-oriented. CORBA is an example of the distributed object paradigm.

**CORBA**

* CORBA stands for Common Object Request Broker Architecture.
* It is a type of middleware, client-server software development model.
* CORBA is an open standard for distributed objects.
* In this application directly communicates with the distributed object that is actually performing the operations.
* CORBA provides various important benefits like wide platform and language support, open standard, industry-standard, scalability, maturity, efficiency, and many more.

Open Standard -

* As a CORBA is an open standard therefore users can choose an implementation from a variety of CORBA vendors or any of the freeware implementations.
* This also increases the high degree of interpretability among CORBA-based applications.

Industry Standard -

* CORBA supports both distribution, portability between implementations, and Object Orientation.
* This creates competition among vendors and ensures that quality implementations exist.

Wide Platform Support -

* CORBA implementations are available for a wide variety of computers, including IBM OS/390 and Fujitsu Global Server mainframes.
* Numerous variants of UNIX (including Linux), Windows, AS/400, Open VMS, Apple’s OS X, and several embedded operating systems also support CORBA.

Wide Language Support -

* CORBA supports many existing languages.
* CORBA also supports mixing these languages within a single distributed application.

Like C, C++, Java, Smalltalk, Ada, COBOL, PL/I, LISP, Python and IDLScript.

Maturity -

* CORBA is extremely feature-rich, supporting many programming languages, operating systems, and a diverse range of capabilities.
* Like transactions, security, naming, trading services, messaging, and publish-subscribe services that are essential for many enterprise-level applications.

Scalability -

* The flexible, server-side infrastructure of CORBA makes it feasible to develop servers that can scale from handling a small number of objects up to handle a virtually unlimited number of objects.
* The real-world projects demonstrated with a CORBA server can scale to handle not just a huge amount of server-side data, but also high communication loads from thousands of client applications.

Efficiency -

* The on-the-wire protocol infrastructure of CORBA guarantees that messages between clients and servers are transmitted in a compact representation.
* Most CORBA implement marshal data (that is, convert data from programming-language types into a binary buffer that can be transmitted) efficiently.

Multi-Disciplinary Nature -

* CORBA is used everywhere from billing systems to multi-media news delivery to airport runway illumination, aircraft radio control, and the Hubble space telescope.
* Most of the world’s telephone systems, as well as the truly mission-critical systems operated by the world’s biggest banks, are built on CORBA.
* In short, CORBA is being used successfully in many industries, including aerospace, consulting, education, e-commerce, finance, government, health-care, human resources, insurance, ISVs, manufacturing, military, petrochemical, publishing, real estate, research, retail, telecommunications, and utilitie.

**Conclusion:**

CORBA can be implemented in Distributed System, which improved efficiency, platform independence, interoperability, scalability, and cost-effectiveness. However, it also requires a significant amount of expertise and resources. By following a structured methodology, we were able to design, implement, and test a distributed system that met our objectives and provided us with a more efficient and scalable solution.