Component--based development

Component--based development

- Component--based software engineering (CBSE) is an approach to software development that relies on software reuse.
- It emerged from the failure of object--oriented development to support effective reuse. Single object classes are too detailed and specific.
- Components are more abstract than object classes and can be considered to be stand--alone service providers.
- Independent components specified by their interfaces.
- Component standards to facilitate component integration.
- Middleware that provides support for component inter--operability.
- A development process that is geared to reuse.

- Definitions:
- •• A software component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard.
- •• A software component is a unit of composition with contractually A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third--parties.

Component models

- A component model is a definition of standards for component implementation, documentation and deployment.
- Examples of component models
  - EJB model (Enterprise Java Beans)
  - COM+ model (.NET model)
  - Corba Component Mode

Common Object Request Broker Architecture (CORBA)

- The Common Object Request Broker Architecture (CORBA) is a specification of a standard architecture for middleware.
- Using a CORBA implementation, a client can transparently invoke a method on a server object, which can be on the same machine or across a network.

- The middleware takes the call, and is responsible for finding an object that can implement the request, passing it the parameters, invoking its method, and returning the results of the invocation.
- The client does not have to be aware of where the object is located, its programming language, its operating system or any other aspects that are not part of an object’s interface.

CORBA reference model

- The CORBA reference model called Object Management Architecture (OMA)
- The OMA is itself a specification (actually, a collection of related specifications) that defines a broad range of services for building distributed client-server applications.

Functions of Object Request Broker (ORB)

- ORB is the central component of the CORBA architecture.
- The main responsibility of ORB is to transmit the client request to the server and get the response back to the client.
- ORB abstracts out many procedures involved in service invocation and makes service invocation by client seamless and easy.
- The main responsibilities of ORB are the following:
  - • Server location
  - • Server state management
  - • Communication between clients and servers

- The ORB must support a large number of functions in order to operate consistently and effectively.
- The ORB implements much of these functionality as pluggable modules to simplify the design and implementation of ORB and to make it efficient.
- ORB allows objects to hide their implementation details from clients.
- This can include programming language, operating system, host hardware, and object location.

Commercial ORBs

- There are several ORBs that are commercially available.
  - • **Visigenic:** This is probably most popular one. Netscape browser supports Visigenic. CORBA applications can be run using Netscape web browser. In other words, Netscape browser can act as client for CORBA applications. Netscape is extremely popular and there are several millions of copies installed on desktops across the world.
  - • **IONa**
  - • **Orbix**
  - • **Java IDL**

Component Object Model (COM)

- The main idea in the Component Object Model (COM) is that:
  - • Different vendors can sell binary components.
  - • Application can be developed by integrating off-the-shelf and proprietary components.
- COM runs on a single computer.
- The concepts used are very similar to CORBA.
- The components are known as binary objects.

Distributed Component Object Model (DCOM)

- Distributed Component Object Model (DCOM) is the extension of the Component Object Model (COM).
- The restriction that clients and servers reside in the same computer is released here.
- So, DCOM and CORBA both operate on networked computers.
- Here development is much easier as compared to CORBA development.
- Many of the things are transparent to the programmer such as proxy generation, service invocation etc.

CORBA vs. COM/DCOM

- If it is the case that all applications reside on PCs and are to run fully on Microsoft platforms then it will be better to use COM/DCOM because development would be much easier here.
  - • If an application is to be developed for a heterogeneous environment then it will be better to use CORBA.
  - • Microsoft is very strong on desktop applications i.e. GUI-based applications. Whereas, CORBA-based development is stronger on server side. Java Beans promise to overcome the shortcoming on desktop side i.e. the client part.