

Reuse And Component Based Development (CBD)

M. Rizwan Jameel Qureshi
Dept. of Computer Science,
COMSATS Institute of Information Technology
rjamil@ciitlahore.edu.pk
Ph # (92-42-5431602) Cell # (03334492203)

Abstract

Reuse of the SW components concept started in 1960 when engineering and scientific libraries were used in the SW development to reuse the previously developed functions. This concept is now widely used in SW development as component based development (CBD). CBD is the technology that facilitates the reuse of the existing components into the new ones. This paper presents different architectures of CBD such as ActiveX, common object request broker architecture (CORBA), remote method invocation (RMI) and simple object access protocol (SOAP). The overall objective of this paper is to support the practice of CBD by comparing its advantages and disadvantages. This paper also evaluates object oriented (OO) process model and seek to propose a process model for CBD.

Key words: CBD, CORBA, ActiveX, RMI, SOAP

1. Introduction

Reuse of the SW components concept has been taken from the manufacturing industry and civil engineering field such as manufacturing of the vehicles in parts and construction of the building using bricks. The spare parts of a product should be available in the market to make it successful. The best example to understand the previous sentence:

- ❖ Could Honda, Toyota and Suzuki car manufacturing companies be so successful if these companies have not provided the spare parts of their cars?

The software providing companies have used the same concept to develop the software in parts. Those companies have provided the parts with their software, to make themselves successful. The

software parts are shipped with the libraries available in the software. The software parts are called components in the IT and computer science fields. Different people have defined the component in different ways. The most popular definition is “Each reusable binary piece of code is called a component” [1]. The component concept is similar to object concept of Object Oriented Programming. A component is an independent part of the system having complete functionalities. A component is designed to solve a particular purpose, such as command button and text box of VB [2]. The component is like a pattern that forces the developers to use the predefined procedures and meets the specifications to plug it into the new SW components.

Microsoft Corporation and Sun Microsystems are two major software-providing organizations. Their tools are most widely used in the SW market to develop applications. These two companies and other vendors are also providing the components with their tools to make them successful.

Section 2 to 5 describes several architectures, proposed up to now, for CBD. Microsoft Corporation has introduced ActiveX technology. Sun Microsystems has introduced enterprise java beans (EJB) and remote method invocation (RMI). Object Management Group (OMG) has introduced common object request broker architecture (CORBA). Microsoft, IBM and Lotus corporations have introduced simple object access protocol (SOAP).

Section 6 proposes a process model for CBD. Section 7 and 8 describe main advantages and disadvantages of CBD.

2. Microsoft Architecture

Microsoft Corporation ActiveX components have three major categories, based on the type of applications where these are used.

- ❖ Component Object Model (COM)

- ❖ Distributed Component Object Model (DCOM)
- ❖ Object linking and embedding (OLE)

COM was originally developed to create the desktop applications. DCOM is being developed to create remote or distributed applications. OLE is used to integrate one SW tool with another tool seamlessly, e.g., integrating MS Word with MS Excel [3]. COM+ is an extended version of COM. The COM, DCOM and OLE technologies as a whole are called ActiveX technology. Microsoft Corporation has introduced a new set of tools called Dot Net (.NET). DOT NET is a framework that uses common language runtime (CLR) environment, such as VB.NET, C#. NET and ASP.NET. ActiveX technology can be used with the .NET technology. The Microsoft corporation components are saved with the extensions of .dll, .exe and .ocx. The dll stands for dynamic link libraries. The .exe stands for executable files and ocx stands for ActiveX object controls. The Microsoft Corporation and Sun Microsystems have provided the facility, for the programmers, to develop their own components. The components being developed by Microsoft or by programmers can be developed in any tool or language that supports COM. These ActiveX components are being registered in the Windows System 32 folder. The components provided by Microsoft have registration by default but the programmer-defined components have to be registered in the Windows System32 folder.

The ActiveX components can be used with any tool that supports COM. A component developed in VB can be used with ASP, Visual C, Visual JAVA and Visual C++. This means that components developed in different programming languages communicate with each other. This ability of components to intercommunicate is also known as interoperability. COM technology helps the programmer to integrate tools of different vendors. Use of Crystal report with VB or ASP is an example of COM. Word application of one vendor integrating Spreadsheet application of another vendor is another example of COM.

3. Sun Microsystems Architecture

Sun Microsystems components are called *beans*. The beans perform the same operation as the ActiveX components. RMI is used to call the components of a Java application into another Java application over the network. The method to call the objects is the same as is followed by CORBA, which has been explained later. The main limitation of RMI is that it is applicable only for Java applications. The programmer has to use CORBA if he/she wants to

call the component of an application developed in a tool other than JAVA [4].

4. Common Object Request Broker Architecture

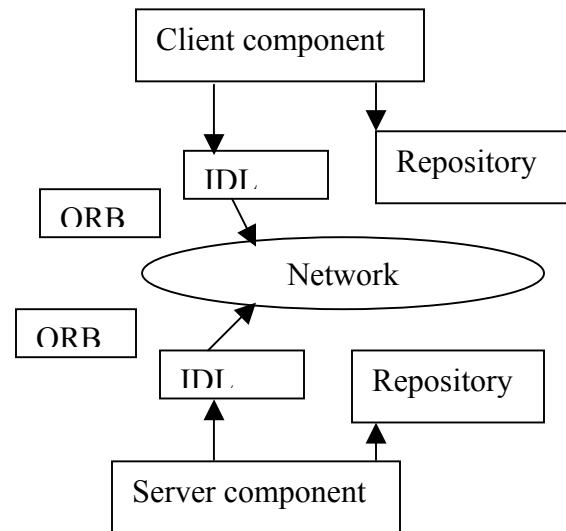


Figure 1 The Architectural Design of CORBA

Object Management Group (OMG) has introduced a standard for the intercommunication of the components known as *common object request broker architecture (CORBA)*. The CORBA divides the architecture of different programming languages developed by different vendors, e.g. VB procedure can be called to C or C++ and vice versa. The language used for intercommunication among the components is known as interface definition language (IDL). The procedure used to call the remote objects is known as remote procedure call (RPC).

The service that is used to communicate one component with another is called object request broker (ORB) [5]. The components used by the CORBA need to be registered in a repository. The architectural design of CORBA is shown in figure 1. The client component request is passed to IDL stub and then to client operating system. The client operating system passes that request to server operating system. The server OS sends that message to server IDL stub and then to the server component. The server component replies by providing the necessary data to IDL stub. The IDL server stub sends that message to the server OS. The server OS then sends that reply to the client OS. The client OS

sends the reply to the client IDL stub and then to the client object.

There are two main limitations of the CORBA architecture: the programmer needs to learn a new programming language, i.e. IDL; CORBA supports a limited number of data types using JAVA.

5. Simple Object Access Protocol (SOAP)

Microsoft, IBM and Lotus Corporation have developed SOAP protocol, which is a combination of HTTP and XML. XML is based on the document object model (DOM). Unlike the HTML, that is used only to display the data, XML is used to manipulate the data. It is based on the object-oriented concept. XML document has parent and child objects that are also called as nodes. The procedure used to call the object of one application into another application is the RPC as is used by CORBA [6].

6. Process Model of CBD

The object oriented process model is the only process model that indicates the reuse of existing SW parts [7].

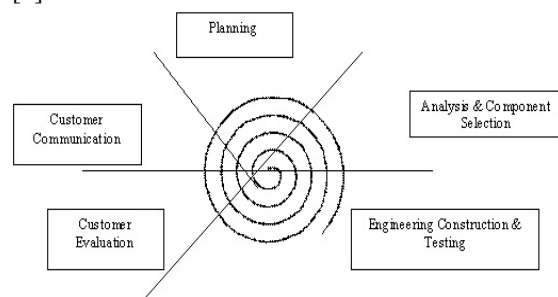


Figure 2 CBD Process Model

That model can be modified to implement the reuse of component-based development. The main phases of process model are: Customer Communication; Planning; Analysis; Engineering, Construction and Testing; and Customer Evaluation. The engineering, construction and testing phase reflect the reuse of existing classes. The main phases of CBD process model are shown, in figure 2.

- Customer Communication
- Planning
- Analysis and component selection
- Engineering, construction and testing
- Customer Evaluation

The analysis phase of the OO process model can be modified according to the CBD process model. The new name of analysis phase can be 'analysis and

component selection' phase. This is the phase where an analyst gathers the detailed requirements and tries to identify and select those components that can be reused. The relationships among the components are identified. The properties and behaviors of the components are identified as well [8]. The core objective of this phase is to reuse maximum components, rather than reinventing the wheel.

Engineering, construction and testing phase of the OO model matches the requirements of CBD process model. The newly components are designed, developed and tested. The integration and system tests of newly developed as well as of the reused components are performed. IDL is coded to integrate the components if the programmer is using CORBA or RMI architecture. The customer evaluation phase of OO process model fits to the requirements of the CBD process model.

7. Advantages of CBD Approach

There are many advantages of developing the applications using the CBD.

- I. Reusable
- II. Interoperable
- III. Up-Gradable
- IV. Saving the programmers from complexity
- V. Time effective
- VI. Cost effective
- VII. Makes programmers Efficient
- VIII. Reliable
- IX. Improved Quality

Reusable: It is an important advantage of developing the applications. Developing a mail transfer web application system in ASP can be an example application using the CBD. CDONTS component has been provided by Microsoft Corporation to develop e-mail systems. This component can be used in similar applications. The built-in components within a particular tool also reflect the benefits of reusability approach such as Microsoft VB 6.0 text box and command button objects. The programmer needs intensive coding to develop a text box with its complete functionality using the C language whereas a text box with its complete functionality in VB 6.0 is a component and can be used frequently without writing a single line of code. That makes the VB 6.0 the most widely used Rapid Application Development (RAD) tool. Reusability helps us to concentrate on adding more complex functionalities in our applications rather than focusing on developing basic components.

CBD architectures allow the components to intercommunicate with each other and this makes them *interoperable*. The interoperability facilitates the programmers to integrate one application with other applications. The information systems used by ATM machines of different banks are integrated with each other. One link system is the benefit of interoperability provided by CBD.

Up gradable: Up gradation of the web applications is easy if some new component has been introduced during the life of application. The programmer has just to replace that previous component with the new component in the application without a change in the front-end code of the application. The client-server architecture is used for the development of distributed applications. There are three types of client-server architecture. Single tier architecture is composed of integrating presentation, application logic and data layers into single SW application such as MS Excel and Access SW. Two-tier architecture integrates presentation and application logic layers into one SW application and data layer is handled by second SW application. This can be explained by taking an example of a business application that is coded by using VB or ASP to handle the presentation and application logic layers where data layer is handled by MS Access or SQL server database. The three tier architecture is also called multiple or n-tier architecture. The presentation, application logic and data layers are handled by three SW applications. The front-end tool such as VB or ASP handles the presentation layer; the COM or COM+ component handles business logic layer while MS Access or SQL server database handles data layer.

These days the programmers prefer to use three-tier architecture to develop business applications. The business logics are continuously changing due to current business environment. Therefore, it is very easy for the programmer to replace the COM component without changing the front-end VB or ASP code and back-end MS Access or SQL server database.

The programmer does not need to understand how any component is working. He should know how to integrate that component with the application. This is similar to a car driver who does not need to know how a Honda Car engine operates. The programmer can concentrate more on main features of the system rather than wasting time on designing the basic components of the application such as text box and command button. CBD saves the programmers from the *complex programming* of designing the development environment. CBD helps to reuse the components again and again in the similar

applications that result in *time* and **cost saving**. It also makes the programmers *efficient*.

The reusable components are already thoroughly tested and maintained when these were used for the first time. Therefore the reusable components are *reliable* in terms of usage as compared to the newly developed components. As stated by Wayne C. Lim, the defect rate for reuse code is 0.9 defects per kilo line of code (KLOC) where as it is 4.1 defects per KLOC for new code [9]. He made a comparison between two applications, one was developed with reuse and other was developed without reuse. The improvement in the development time was 51% for the application that was developed from the reusable components.

The *quality* of the application developed from reusable components is also significantly improved. The reusable components are highly tested and maintained. The newly developed components need a lot of testing and maintenance and still there are some bugs left that may be reported when they are integrated with other components. The quality of the SW is affected because of such maintenance. As stated by Henry and Faller [10], the quality of the SW is 35 % improved because of the reusable components.

8. Disadvantages of CBD Approach

Following are the main disadvantages of component-based development in terms of reuse:

- I. Customization
- II. Adaptability
- III. Integration
- IV. Security
- V. Efficiency

Customization of an already developed component according to the requirements of new application is a major issue in CBD. The developers also face a problem to adapt a component to a new platform if it were not developed for that platform. The integration of a reusable component into new component is also a major problem faced by most of the developers. Security is another major concern for the developers who reuse the components available over the Internet. There may be a virus inside that component and may pass all the information of the business organization to attacker, who uses such an application. Efficiency of the SW applications developed using CBD is also debatable. The component to be reused may have extra functionalities that may be a requirement when it was developed. The new application that does not require extra functionalities becomes less efficient because of the loading time of those functions.

9. Conclusion

This paper explored the main architectures (ActiveX, EJB, CORBA, RMI and SOAP) of CBD. A process model is also proposed for the Component Based SW Engineering (CBSE). Main advantages and disadvantages of CBD are also described. From these discussions it is concluded that CBD is more cost effective, time saving and productive for the SW development, according to the state of art tools, and meets the tight deadlines of the market.

10. References

- [1] Deitel & Deitel. "Visual Basic 6.0 How to program". Prentice Hall, 1999.
- [2] Roger Lee "An architecture Model for dynamically converting components into web services" [ASPEC 04]
- [3] Microsoft Corporation, COM Technologies, <http://www.microsoft.com/com/default.msp> x, (2005).
- [4] Deitel & Deitel. "Java How to program". Prentice Hall, 2001.
- [5] Roger S. Pressman. "Software Engineering". McGraw Hill, 2005.
- [6] Deitel & Deitel. "E-Business & E-Commerce". Prentice Hall, 2001.
- [7] Luiz Fernando Capetz, Miriam A.M. Capretz, Dahai Li, "Component-Based Software Development," IEEE Industrial Electronics Society, 2001, pp. 1834-1837.
- [8] Jim Q. Ning, "A Component-Based Software Development Model," IEEE Software, September 1996, pp. 390-391.
- [9] Lim, W.C., "Effects Of Reuse On Quality, Productivity And Economics," IEEE Software, September 1999, pp. 23-30.
- [10] Henry, E., and B. Faller, "Large Scale Industrial Reuse To Reduce Cost And Cycle Time," IEEE Software, September 1995, pp. 47-53.