Reusable-Code-for-CSOA-Services-Handling-Data-Coupling-and-Content-Coupling

Dr. Muhammad Zubair Asghar
Reuseable Code for CSOA-Services: Handling Data Coupling and Content Coupling

Shakeel Ahmad1, Sheikh Muhammad Saqib1, Muhammad Ahmad Jan1, Muhammad Zubair Asghar1 and Bashir Ahmad1
1Institute of Computing and Information Technology Gomal University, D.I.Khan, Pakistan

Abstract— ‘Customized-SOA services are loosely coupled’; this should not be a slogan but be physically present. For adopting such features, code for each service should be independent. Common coupling, external coupling, control coupling can be eliminated but data coupling and content coupling may not be removed. But these two coupling types can be handled in such a way that code for each CSOA-service can be treated as a single unit. Here author proposed a strategy to design a code for Customized-SOA service with handling data coupling and content coupling. Such type of code for a CSOA-service will be very useful for reusing service in another CSOA-application. And also such code can be easily changed according to needs of users.

Keywords- Customized-SOA, Loosely Coupling, Data Coupling, Content Coupling, Reusability.

I. INTRODUCTION

Loosely coupling task is not easier, without familiar to understanding the principles it can not be achieved at first step because code is ‘hard to trace’. For writing code, depend on only abstraction not on concrete classes then combined with dependency injection principle, true loosely coupling can be achieved [1]. Interpreted languages often include dynamic execution model and support for functions as first-class values and closures, these features can be very useful to build different coordination mechanisms for distributed asynchronous computing and loosely coupled interactions often supplied by event-oriented application models [3].

SOA model plays a vital role in software design & architecture and through dynamic discovery, it shares different components. There are different methods which enables complex application to be combined with services. Selection of these services depends on cost and quality. So there will be a mechanism which can easily select the services for an application [6]. If services are loosely coupled then their cost will be minimum and quality will be maximum.

Customized-SOA is a services based application i.e. software which has only logical interaction with user and there is no interface for physical entry” is called CSOA Customized SOA application. It provides only services to an organization according to their needs. Business Oriented Services Model for CSOA made up of SOMA and RUP which is Consist of four parts:

BOSI (Business Oriented Service Identification): BOSI presents service identification for service decomposition and provides an approach for finding completed services before development and reusing these services in other CSOA. BOSS (Business Oriented Service Specification): Goal of BOSS is to specify those services which should be implemented during particular iteration functional Components & technical components. BOSR (Business Oriented Service Realization): BOSR includes managing resources, controlling operations to optimize costs, schedules, and quality. BOSD (Business Oriented Service Deployment): This includes the deployment of the services into the productive environment and user acceptance tests [4]. Since services are loosely coupled in nature, but there may be some problems which can be occurred when service of one application is going to be incorporated in other application. Because some classes uses the features of some other classes. But when services are created at design time with all future references or parameters then these problems can be reduced.

In BOSM, BSOI also discover either developed service can be used in single organization or other organizations with percentage. Here BOSS is very important because functional and technical components contain different classes with their attributes and events. Here author proposed a technique how to implement a service for COSA with its loosely coupled features. So it can be easily reused in another CSOA application.

Different types of coupling are in practice such as common coupling (common coupling exists when classes in the system share a global data), external coupling (dependency of the class on third party classes), control coupling (control coupling is when one class controls the logic of another class), content coupling (exists between two classes when one class relies on the internal working of another class, i.e., one class is using the variables of another class), data coupling (one class passes simple data to another as an argument) [2].

Since in customized-SOA service is consumed by the end user and does not require any further processing and also it is self defined. So there will be little bit or no chance of common coupling, external coupling, control coupling. But content coupling and data coupling can not be avoided for designing the classes of CSOA-services. Here author define a strategy that these content coupling and data coupling will not affect the service when it is going to be reused.

II. STRUCTURE FOR SERVICE CODE

Customized-SOA service is just like a separate unit so avoid inheritance, level-of-inheritance and multiple-
inheritance. Now first of all separate the common features of all services, as information about each service must be stored in database, so each service requires connection variable and queries. These queries may be executable (returns records from database) or non-executable (only make changes in database and returns only integer). Hence there may be two classes 1st class (method for establishing connection) relate to database connection and 2nd class (will contain methods for queries) relate to these executable and non executable queries. 3rd class will be for developed service can be extracted from technical and functional components based on "story cards practice of BOSM in BOSI" [4]. Technical and functional components will contain all possible attributes and events which will be used in current and future application. Description of each event will be useful for creating algorithm about service. Then this algorithm will show the shadow of actual code for service. This target class can use the methods of 1st class and 2nd class with their objects. 1st and 2nd classes will be mandatory for each service so we can not avoid content coupling and data coupling. When 3rd class will be reused in another application, then content coupling and data coupling can be easily removed or skipped.

III. IMPLEMENTATION OF SERVICE CODE

DOA is a CSOA based application developed in .Net framework which can be used in any office and can assist all employees. Each user can perform different activities of DOA after authentication [5]. Attendance is a global service for all organization, now we construct attendance service for DOA, then its code will be loosely coupled for other services.

Table-1: FUNCTIONAL COMPONENT FOR ATTENDANCE SERVICE

<table>
<thead>
<tr>
<th>Service Component name</th>
<th>Request Messages</th>
<th>Response Messages</th>
<th>Inputs</th>
<th>Inputs Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clsAttendance</td>
<td>Attendance Marking</td>
<td>Attendance</td>
<td>intUIDS</td>
<td>Sender ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is at time or Not</td>
<td>tmAttendance</td>
<td>Attendance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dtAttendance</td>
<td>Time</td>
</tr>
</tbody>
</table>

This table contains all possible attributes which can be used in all organization such as any user can make attendance on each day at particular time.

Table-2: TECHNICAL COMPONENTS FOR ATTENDANCE SERVICE

<table>
<thead>
<tr>
<th>Service Component name</th>
<th>Actions</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service-1</td>
<td>This event will check the closing time of office and will return false if time is up.</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td>This event will save the marking attendance of user in database.</td>
<td></td>
</tr>
</tbody>
</table>

This table shows the different events related to the attendance service with passing generalized parameters. From above Table-2 class diagram can be easily created.

Hence from Fig-1, it is clear that two classes will be mandatory for each new service. Now we are creating 1st, 2nd and 3rd class of Fig-1 in VB.Net framework. Fig-3 is showing the code for connection class. This class contains only connection with database.

```
Public Class clsConnection
    Public cn As System.Data.SqlClient.SqlConnection
    Sub New()
        cn = New System.Data.SqlClient.SqlConnection("Integrated security=srong;Persist Security Info=true;Initial Catalog=dboA;"
    End Sub

End Class
```

Each service will have executable and non-executable queries, so 2nd class will be beneficial for creating each new service.
Above Fig-4 is using the object of connection class and these two classes will be common among all classes of service. Now from Fig-2, we can easily create a reusable code for service attendance i.e. 3rd class.

In Fig-5, Line-A, Line-B are executable queries and Line-C is non-executable query. In these three lines we are creating just an object of class ‘clsQueries’ and then calling relevant query event. Database connection depends on class ‘clsQueries’, how it is establishing the connection with database. Class in Fig-5 must have knowledge about class clsQueries’ that what will be the internal execution in this class, this is content coupling. Since, we are using object of another class (clsQueries) in clsAttendance class, so here is data coupling. When this service will be plugged in another CSOA type application, then there will be little bit changes required for handling this content coupling and data coupling i.e. Line-A, Line-B and Line-C. Since each application has its own scheme of code for database connections and queries. So this service will be reused only by redesigning lines A, B, C.
IV. CONCLUSION

Some time already developed code can not be used in another application due to some sort of coupling. As major characteristic of SOA and CSOA, services are ‘loosely coupled’ because services are independent in nature. But at the time of reusing such services in another application, code for each class may be hard to use for incorporating such services in another application. Proposed technique for coding of such services is handling data coupling and content coupling, so those services will be easily reusable or changeable for other applications.

REFERENCES


About Author
Dr. Shakeel Ahmad received his B.Sc. with distinction from Gomal University, Pakistan (1986) and M.Sc. (Computer Science) from Qauid-e-Azam University, Pakistan (1990). He served for 10 years as a lecturer in Institute of Computing and Information Technology (ICIT), Gomal University Pakistan. Now he is serving as an Assistant Professor in ICIT, Gomal University Pakistan since 2001. He is among a senior faculty member of ICIT. Mr. Shakeel Ahmad received his PhD degree (2007) in Performance Analysis of Finite Capacity Queue under Complex Buffer Management Scheme. Mr. Shakeel’s research has mainly focused on developing cost effective analytical models for measuring the performance of complex queuing networks with finite capacities. His research interest includes Performance modeling, Optimization of congestion control techniques, Software Engineering, Software Refactoring, Network security, Routing Protocols and Electronic learning. He has produced many publications in Journal of international repute and also presented papers in International conferences.