

# **SOME INTEROPERABILITY ISSUES IN THE DESIGNING OF WEB SERVICES : CASE STUDY ON CREDIT CARD**

Kalpana Johari

Senior Lecturer  
School of Information Technology  
Centre for Development of Advanced Computing ,NOIDA

Arvinder Kaur

Associate Professor  
University School of Information and Communication Technology  
GGSIP University Dwarka , Delhi

## ***ABSTRACT***

*In today's environment most of the commercial web based project developed in the industry as well numerous number of funded project/and studies taken as part of research oriented initiatives in the academia suffer from major technical issues as to how design, develop and deploy the Web Services that can run in variety of heterogeneous environments. In this paper we address the issues of interoperability between Web Services, the metrics which can be used to measure the interoperability and simulate the Online shopping application by developing the Credit Card Verification Software using Luhn's Mod 10 algorithm having Java Client written in NetBeans and the BankWebService in C#.NET.*

## ***GENERAL TERMS***

*Algorithms, Design, Security.*

## ***KEYWORDS***

*RMI, CORBA, SOAP, WSDL, UDDI, OTP.*

## **1. INTRODUCTION**

Interoperability primarily refers to the seamless flow of the data and information across multiple Web Services hosted on single or multiple platform in a heterogeneous environments. One of the most pertinent question that often popup is to how to defined a "Web Service". A Few definition's have been suggested in [1], [2],[3] but it can be defined in more simpler terms as "Service that caters to other Services" that is hosting of multiple applications comprising of humongous data and information on a single platform on the Web using open standards like XML (Extensible Markup Language) [18] , SOAP (Simple

Object Access Protocol) [19] and WSDL(Web Services Description Language) [20] and UDDI(Universal Description Discovery and Integration)[21]. As per W3C [World Wide Web Consortium] the Web Service can be defined as “A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards”. One of the most important things to be kept in mind while understanding the concept of Web services is that Web Service is never going to be used directly by the human being(s).In fact all the information available through the web service is primarily meant for the software as it is the software that directly communicates with the web service and not the human beings. To our knowledge and understanding it is first such paper that not only addresses the issue of interoperability but also design Web Service in ASP.NET that caters to the query received from JAVA Client.

## 2. RELATED WORK

Chidamber and Kemerer [4] introduced the Response for Class (RFC) as a measure of the number of functions or methods that can potentially be executed in response to a message received by an object of that class. Choi and Lee [5] proposed a dynamic coupling metrics which can be used to accurately measure the coupling between the classes. Perepletchikov *et al.*[6] suggested a set of metrics for quantifying the structural coupling of design artifacts in service-oriented systems. Qian *et al.* [7] proposed a practical guide for evaluating decoupling between service- oriented components in the service composition such as Business Process Execution Language (BEPL). Quynh and Thang [8] proposed a collection of metrics to measure service’s quality according to its usability of coupling. In their paper they discussed how the coupling metrics can be used to accurately measure the maintainability, reliability, testability and reusability of services. Li and Henry [9] illustrated the Message Passing Coupling (MPC) as the count of the number of send statements that are present in methods of one class linked to other set of classes. They emphasized that Coupling between Object Classes should be viewed as the count of the number of classes to which it is coupled.

## 3. INTEROPERABILITY IN EXISTING TECHNOLOGIES

The Web Services should be interoperable to an extent that the classes, Arrays and Structures designed and developed using J2SE (Java 2 Standard Edition), or J2ME (Java 2 Mobile Edition), or J2EE(Java 2 Enterprise Edition) architecture should be able to communicate with the Array, Classes and Structures designed in the C#(C sharp).NET application. In the same way their should be seamless flow of data between Applets designed in J2SE to Midlets designed in J2ME to Servlets designed in J2ME. The Same data or information if needed should be made available to the application designed in VB.NET or ASP.NET. J2EE offers two architecture’s for the web services designed to exploit the concepts of distributed computing viz. RMI(Remote Method Invocation) and CORBA (Common Object Request Broker Architecture).RMI is primarily use for carrying out the JAVA – JAVA communication that is at both the ends of the application, the programs are written in JAVA so no issues of the interoperability’s arise between the applications.RMI uses the concept of Remote Interfaces, Remote Objects, Stubs and Skeletons to carry out the communication between Client and server Programs written in JAVA. However the CORBA architecture which is a standard defined by the Object Management Group is primarily used for carrying out communication between the Java application and the non-Java application such as those application designed in the COBOL, Pascal, FORTRAN,ALGOL,C++, and C#.NET etc. CORBA uses special files

called as interface definition language (IDL) to specify the interfaces which objects present to the outer world applications . CORBA then specifies a mapping from IDL to a specific implementation language like C++ or JAVA. Standard mappings exist for C,C++, Ada, Simula, Smalltalk, Python, Ruby on Rails. So effort is always to go in for the seamless connectivity between JAVA and C#.NET application.

#### 4. ISSUES IN THE INTEROPERABILITY OF WEB SERVICES

In our paper we raise one of the biggest problem related to the interoperability of the Web Services at the implementation level is the incompatibilities in terms of the data types supported by the programming languages like JAVA and C#. The Problem of incompatibilities arises in various ends due to the incompatibilities not only in terms of data types but also in terms of following listed incompatibilities between following three programming languages namely C++, JAVA and C#

Table 1. Incompatibilities between the programming languages :-

S.No	Entity Name	C++	JAVA	C#
1	DYNAMIC ARRAYS	NO	Yes	Yes
2	VECTORS	No	Yes	No
3	STRUCTURE	YES	NO	NO
4	POINTERS	YES	NO	NO
5	MULTIPLE INHERITANCE	YES	NO	NO
6	INTERFACES	NO	YES	YES
7	WRAPPER CLASSES	NO	YES	NO
8	APPLETS	NO	YES	NO
9	FINAL CLASSES	NO	FINAL	SEALED
10	DELEGATES	NO	NO	CLASS
11	MULTI- THREADING	NO	YES	YES
12	EXCEPTION HANDLING	YES (try, catch and throw)	YES (try, catch, throw, throws, finally)	YES(try, catch, Throw, throws, finally)
13	PERSISTENCE	NO	YES using the Synchronization keyword	YES
14	FRIEND CLASSES	YES	NO	NO
15	FRIEND FUNCTION	YES	NO	NO
16	DATABASE CONNECTIVITY	YES (through DB-Libraries)	YES (thru JDBC and HIBERNATE)	YES( thru ADO. NET)
17	PREPROCE SSOR DIRECTIVE	YES	YES(thru Packages)	Yes(thru Name spaces)

18	PURE VIRTUAL FUNCTIONS	YES	NO	NO
19	OPERATOR OVERLOADING	YES	NO	NO
20	STANDARD TEMPLATE LIBRARY	YES	NO	NO
21	LINKING TYPE	STATIC	RUNTIME /DYNAMIC	RUNTIME /DYNAMIC
22	CHARACTER SET	ANSI/ASCII	UNICODE	UNICODE/ UCS( Universal Character Set)
23	WEB SERVICES SUPPORT	NO	YES(thru SERVLET S and JSP)	YES (thru ASP. NET)
24	SOCKET PROGRAMMING	YES(Possible through UNIX libraries)	YES(through Socket and ServerSocket classes)	YES(through In-built DLL)
25	ARBITRARY SIZED INTEGERS	NO	Reference type : No Operator	YES
26	ARBITRARY SIZED DECIMALS	NO	Reference type : No Operator	NO

## 5. METRICS FOR INTEROPERABILITY OF WEB SERVICES

### 5.1 Adherence to the standards

The effort should always be laid on designing a Web Service that strictly conforms to norms and standards of WS-I (Web Services Interoperability Organization norms and W3C (World Wide Web) Consortium.

If we want to measure how interoperable a web service is that it should be measured in terms of it's adherence or deviation to the existing set of norms and standards. If a web service closely follows these standard set of Guidelines and norms then it is one which is highly interoperable and if does not then it is considered to be loosely interoperable.

### 5.2 Number of User – Defined Data types

While designing a Web Service efforts should be made in the direction of designing a Web service that uses maximum number of the fundamental data types and least number of user defined or derived data types. The Web Service which is designed using minimum number of derived data types then it is considered to be highly interoperable.

### 5.3 Nesting of Web Services :-

While designing the Web Service every effort should be made that all the information required by the said web service should be provided at one place under single umbrella that is all the functionality be defined under the functions defined in once class and less number of inner classes, anonymous inner classes (as in JAVA),abstract classes and nesting of classes should be used.

#### **5.4 Request-Response Paradigm :-**

Number of request response messages being exchanged between the Service Subscriber (the client Software requesting Program) and the Web Service Publisher that is the Web Service, hosted on the Web Server and designed to handle multiple client request in a typical n-Tier Client Service application.

#### **5.5 Accessibility:**

A Web service should be accessible to one and all and it should be designed in such a manner that it strictly conforms to the Web Content Accessibility Guidelines 1.0 [12] and the standards for defined for Web Accessibility Initiative [13].

#### **5.6 Availability:-**

The Web Service should be designed to host millions of request around the clock on 24x7x365 basis. This availability is the directly linked to the probability that the system is up and related to reliability [1]. An example of this metric is the Time-to-Repair that calculates the time it takes to repair the Web Service and to bring it back in operational state.

#### **5.7 Security:**

A Web Service needs to be secure from the on-site and on line attacks of hackers and crackers. Effort can be made to make a Web Service secure and less vulnerable by incorporating the light weight cryptographic algorithms.

### **6. APPLICATIONS OF WEB SERVICE :-**

We can design the Web Service for variety of applications few of which can be described as follows :-

- 1) In Retail Stores :- In Retail Stores where in we can design the POS (Point of Sales) Application Software in JAVA and another application designed in VB.NET or C#.NET that interfaces with Bar Code Reader to read the Bar Code Number from the Items (such as Cold Drink and NoteBooks ) etc.
- 2) QR Codes (Quick Response Code) consisting of Application designed in JAVA and the actual reading/fetching and storage of the information is coded in VB.NET or ASP.NET.
- 3) ReCharge WebSite :- The Application of the WebSite that accepts the mobile number can be designed in JAVA and the Application that does the task of recharge and the credit card verification can be designed in ASP.NET.
- 4) Online Portals and Vortals catering to different services like online music store and gaming sites et al.

### **7. DESIGNING OF WEB SERVICE**

In order to depict the interoperability between the Web Services, We design a Web Service based application called Credit Card Verification Process that uses the Luhn's Mod 10 Algorithm[15]. We designed two classes one called as Credit Card class that serves as JAVA Client written in Net Beans[16] and other the Bank Class designed as Bank Web Service in C#/ ASP.NET[17].The detailed steps performed in designing of the web service are listed in table 2 and the subsequent snapshots/images designed using JAVA Programming language in Net Beans and C#.NET are schematically represented through figures 1-7.

Table 2: Steps in the designing of Credit Card based Service

S. No	Steps	Validation	Description / Purpose	JAVA	C# NET
CUSTOMER APPLICATION REACHES BANK					
1	Customer Applies to bank	Customer Details need to be verified for Example thru : Valid PAN Card number or Driving License number or Voter ID	Customer details are accepted and stored in the customer Table of the DB		YES
2	Customer applies for Internet Banking Option	Bank class issues 6 digit Unique MCSC (Master Card Secure Code) or VBV(Verified by VISA code) to the customer	VBV(Verified by VISA) and MCSC (MASTER CARD SECURED CODE) need to be stored in the customer table of the Database		Yes
CUSTOMER GOES FOR ONLINE SHOPPING					
3	CC(Credit Card) Class accepts the Credit Card No	Checks whether the length of a Card number is 16 digit or not		Yes	
4	Next CC Class checks whether the number Valid or Invalid	To be verified by applying the Luhn's Mod 10 algorithm		Yes	
5	Next CC Class prompts the user to enter CVV (Credit Card Verification Value Number)	To Check the length of the number :- it should be non-zero and length should not be greater then 3		Yes	
6	Next the CC enters the user to enter the expiry date.	To ensure that an the expiry date should not be less the or equal to current date.		Yes	
7	Next the CC Class enters the user to enter the Amount for carrying out the transaction for doing on line shopping.	The Amount should be non- zero		Yes	
8	The CC number now travels safely thru the Internet to reach the Bank Class		To make sure the information is safe it is encry pted thru DES ALGORIT HM/ AES ALG ORIT HM	Yes	

9	A Verified Credit Card No is send as input to Bank class where check is made to check that if customer is have sufficient funds to carry out the transaction	The Amount should be non- zero.		NO	Bank class in C#.NET
10	If the Customer have sufficient funds then the Bank class prompts the customer to enter MCSC(Master Card Secure Code) or VBV(Verified by VISA code)	This Code should match with the code issued in Step 2.		NO	Bank class in C#.NET
11	Next the Bank class generates a unique number called OTP(One Time Password) which is send back to the CC Class	The number should be non- zero and length between 1-6.		NO	Bank class in C#.NET
12	CC Class receives the OTP and allows the user to carry out transaction				

## 7. SNAPSHOTS OF WEB SERVICE



Fig 1 : Launching of Bank Application

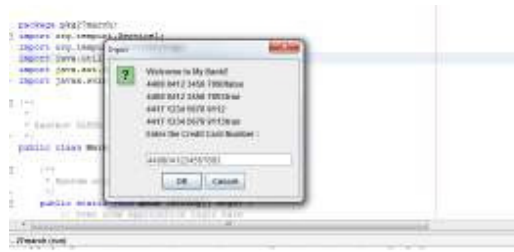


Fig 2 : Prompting the User to enter Credit Card Number

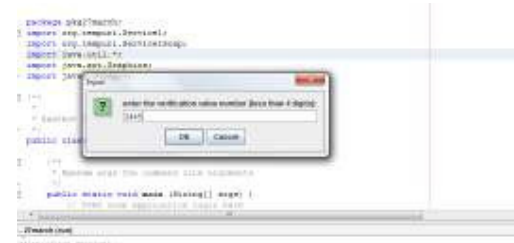


Fig 3 : Prompting the user to enter CVV number

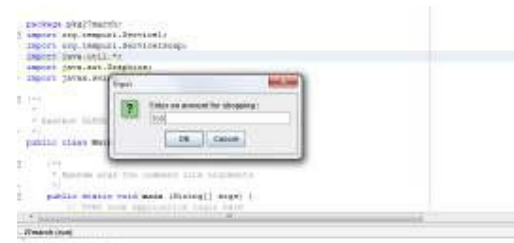


Fig 4 : Prompting the user to enter Shopping amount



Fig 5 : Prompting the user to enter Shopping amount



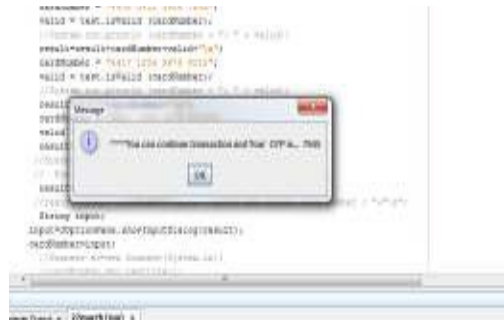


Fig 6 : Generation of OTP(One time Password)



Fig 7 : Message Box for successful transaction

## 8. INDUSTRY SUPPORT : WS-I INITIATIVES

Today Web services are packaged in the WRF (Web Services Resource Framework) [Microsoft document] at includes the XML, SOAP and the WSDL and UDDI (Universal Description Discovery and Integration) et al. In order to achieve high level of interoperability between the different platforms and technologies an organization by the name of Web Service Interoperability [WS-I] [14] has been founded .WS-I is an open industry organization chartered to establish Best Practices for Web services interoperability, for selected groups of Web services standards, across multiple platforms, operating systems and programming languages. The WS-I Basic Profile establishes core Web services specifications (SOAP, WSDL, UDDI,XML Schema, HTTPS) that should be used together to develop interoperable Web services. To date, WS-I has produced the Basic Profile 1.0 and 1.1. [14]

## 9. ACKNOWLEDGEMENT

The author wishes to express their sincere gratitude to the administration of Centre for Development of Advanced Computing, NOIDA and Guru Gobind Singh Indraprastha University, DELHI for providing the academic environment to pursue research activities.

## 10. FUTURE WORK

We propose to develop many more web services and identify more important metrics which can be used to measure the interoperability between the Web Services. We also propose to develop secure and robust Web services which are not prone to the various Active and Passive Attacks .The security is enhanced by using various Symmetric and As symmetric Cryptographic Algorithms such as DES,AES,RSA and RC4 algorithm provided as part of

Java Cryptography Architecture and Java Secure Socket Extension.

**REFERENCES**

- [1] Mohamad Ibrahim Ladan.: 2011. *Web Services Metrics: A Survey and A Classification* International Conference on Network and Electronics Engineering IPCSIT vol.11 (2011) © (2011) IACSIT Press, Singapore.
- [2] Sujala D Shetty , Dr S Vadivel.2009 *Interoperability issues seen in Web Services* In IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August 2009
- [3] Nishith Pathak : Pro WCF 4: Practical SOA Implementation, (2011) Second Edition. ACTA Press
- [4] S. R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Trans. Softw. Eng.*, vol. 20,
- [5] Choi, M. and J. Lee. 2007. A dynamic coupling for reusable and efficient software systems. Proceedings IEEE International Conference on Software Engineering Research, Management and Applications, Busan, South Korea, pp 720-726.
- [6] Perepletchikov *et al.*2007. Coupling metrics for predicting maintainability in service-oriented designs. Proceedings of the 18th Australian Software Engineering Conference Melbourne, VIC Australia, pp 329-340.
- [7] Qian *et al.* 2006. Decoupling metrics for services composition. Proceedings of the 1st IEEE/ACIS int'l conference on computer and information science, Honolulu, HI USA, pp 44-47.
- [8] Quynh, P. T., and H. Q. Thang.2009. Dynamic coupling metrics for service oriented software. Int'l Journal on Computer Science Engineering, 3 pp 282-287.
- [9] Li, W. and S. Henry.1994. Object oriented metrics that predict maintainability. Journal of System Software, 23, 1993, pp111-122.pp. 476-493
- [10] <http://www.w3.org/DesignIssues/WebServices.html>
- [11] <http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html>
- [12] [www.w3.org/TR/WCAG10](http://www.w3.org/TR/WCAG10)
- [13] [www.w3.org/WAI](http://www.w3.org/WAI)
- [14] [http://www.ws-i.org/Docs/ws-i\\_faq\\_01.pdf](http://www.ws-i.org/Docs/ws-i_faq_01.pdf)
- [15] <http://www.beachnet.com/~hstiles/cardtype.html>
- [16] <http://netbeans.org/>
- [17] <http://www.asp.net/>
- [18] <http://netbeans.org/>
- [19] <http://www.w3.org/TR/SOAP>
- [20] <http://www.w3.org/TR/WSDL>
- [21] <http://www.w3.org/TR/UDDI>