

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 9, September 2014, pg.573 – 581

RESEARCH ARTICLE



Improvised Software Testing Tool

Tanuj Wala¹, Aman Kumar Sharma²

¹Research Scholar, Department of Computer Science, Himachal Pradesh University Shimla, India

²Associate Professor, Department of Computer Science, Himachal Pradesh University Shimla, India

¹tanu.thakur52@gmail.com; ²sharmaas1@gmail.com

ABSTRACT:-- *In current scenario, where e-business is emerging and expanding field, web services form the basic building block. Reusability mechanism is efficiently provided by web services thereby reducing the development time and cost. With the changing trends and technology the web is evolving and expanding continuously and so are its services. Day by day web services are getting more complex but publically accessible and usable. The benefits and challenges in web services is for those services to be offered in a very useful, flexible, effective and secure way. This paper investigate open source web service testing tool wizdl for lack of parameters such as response time, number of data processed in terms of bytes and throughput which are then added to it through the modification of the tool.*

Keywords:-- *Web services, e-business, reusability, open source software, web service testing tool, performance*

I. INTRODUCTION

Software testing is one of the crucial phases of the software development life cycle. It is important as it helps the user to measure different parameters of quality like correctness, robustness, usability, integrity and security. Software testing is a set of activities conducted with the intent of finding errors in software. It also verifies and validate whether the program is working correctly without bugs or errors. Testing is the phase where the errors remaining from all the previous phases must be detected. Hence testing performs a very critical role for quality assurance and for ensuring the reliability of software.

Software testing is a process used to measure the quality of the software. Quality of the product depends upon various parameters such as response time; throughput etc. Response time is the time between the request and first response that is received by the user

and throughput is the amount of work that can be performed by a system or component in a given period of time. Throughput depends on transactions/sec or byte/sec [1]. To find out the problems the verification, error detection and validation is the main aim of testing. Testing use the words of successful and unsuccessful in the results of test cases. If the test case did not find an error then it's called a successful test run whereas if it discovers an error then it's called an unsuccessful test run [2]. Manual and automation are the two ways of testing. Manual testing is also called a static testing. It is carried out by the tester. Walk through, informal review; technical review and inspection are the techniques of manual testing [3]. But the problems are it is very time consuming process and requires more effort. So, automation testing is used to solve these problems. Automation testing is also called a dynamic testing. Automated testing is further classified into four types such as correctness testing, performance testing, reliability testing and security testing. It automates the steps of manual testing using automation tools [4]. These tools test the process and targeted to specific test environment. The environment may be functional, performance or exceptional testing etc.

Functional testing is also called black box testing. It is based on requirement and functionality. The test cases for functional testing are based on the requirement and specification. Basically it is the method of generating test cases that are independent of software internal structure. Black box testing occur throughout the software development life cycle and software testing life cycle i.e. in regression testing, acceptance testing, unit testing, integration testing and system stage[5]. Increase in the use of software testing tools has lead to new revolution. Varieties of software testing tool available have multiple usage areas. Few of them can be named as test management tools, functional testing tools and load testing tools etc. In the market various functional testing tools are available for testing the web services automatically. These tools are also used to test the application software and web applications such as web sites [6].

In a business environment, where hundreds of web services working together makes testing of web services is essential to achieve the goal of robust and successful web services. A web service is a relatively new concept which can be used to call methods located on a server. It is an interface that describes a collection of operations. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. Basically web service is a self-contained, self-describing, modular application that can be published, located and invoked by other software systems across the web. This communication is done by using a standardized protocol called Simple Object Access Protocol (SOAP) which is build up by using an Extended Markup Language (XML) schema. Web services work using HTTP and XML which makes the web services technology in heterogeneous in nature because HTTP is transfer independent and XML is data independent.

The heterogeneous nature of web services offers various advantages like interoperability, usability, use of standardized communication protocol and deployability [7]. The web service architecture contains four main parts such as XML, SOAP, Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) [8].

XML:-The web service architecture is built up using XML. XML is a meta language that has a well defined syntax and semantics. The syntax and semantics "self describing" features of XML make it a simple, powerful mechanism for capturing and exchanging data between different applications [9].

SOAP:-Soap is a protocol based on XML. It is used for communication between a Web Services provider application and a client application. A SOAP message can be either a response or a request. These SOAP messages are based XML. So it will be easy to communicate with SOAP within a heterogeneous network, where web services can be found at various internet web servers. A Soap message contains four parts envelope, header, body and fault. Figure 1 shows the structure of SOAP message. In the structure the "Envelope" defines the start and the end of the message. The "Header" contains any optional attributes of the message. The "Body" tag is the place where the request and response data are stored. Within the Body tag a "Fault" tag may exist that provides information about errors that occurred while processing the message.

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope"
SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<SOAP-ENV:Header>
...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
...
<SOAP-ENV:Fault>
...
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP_ENV:Envelope>
```

Figure 1: A SOAP Message Structure [10]

WSDL:-Web service interface is written in WSDL. WSDL document contains the information about the location of the web service and specifications of the method that are present. The method specification includes parameters and the return type of the method [11]. When a web service is published it is actually the WSDL-documents that are published, not the functionality of the web service. It specifies the location of the service, the operation or methods. The WSDL document structure contains the elements such as type, binding, port type, services, message and operations. Figure 2 shows the structure of the WSDL document.

```
< definitions>
< types>
  definition of types.....
</ types>

< message>
  definition of a message....
</ message>

< portType>
  definition of a port.....
</ portType>

< binding>
  definition of a binding....
</ binding>

</ definitions>
```

Figure 2: WSDL Structure [12]

In document the “types” provides information about any complex data types. The “messages” define abstract description of the data being exchanged, “port types” define abstract collections of operations and “binding” defines concrete protocol and data format specification for a particular port type.

UDDI: - UDDI is a registry where web services are stored or the yellow page of the web services. It is an XML based standard for describing, publishing, and finding web services. Registry and a set of WSDL port type definitions for manipulating and searching that registry are the two main parts of the UDDI [13].

Web Services Tools:-Web services are very important in the field of software testing. So, different tools have been developed and designed to enhance the use of web services. Test tool enables the testers to create, manage and execute test for a particular environment and maintained a specific test for a particular application. Various open source web service testing tools are available in the market with different features and functionality. These test tools are free for the users to use. The tools can be downloaded from the internet or can be obtained by the vendor without any charges. The study in this paper is based on the wizdl open source web service testing tool.

The paper can be studied under the following sections: Section I, gives the brief introduction in which software testing, software testing tools, architecture of web services and web service testing tools has been explained. Section II provides the brief overview of wizdl web service testing tool. In Section III, Modification in wizdl according to desired parameters has been given. Section IV gives the result and discussion and Section V concludes the paper along with scope for future work.

II. wizdl: - Brief Overview

There is a huge amount of shift towards web services technology in various sector of IT industry. It can perform various operations from simple request to complex one. Web service testing is a challenging activity which has many characteristics that can be exploited fruitfully for testing purposes [14]. Effective testing produces high quality software. Selection of the tools based on the project requirement. Wrong selection of a tool can lead to various problems like lower efficiency, less effectiveness etc.

There are many web service testing tools are available in software market. A good software testing tools contains various features such as efficiency, maintainability, functionality, usability and adaptability. Though testing is becoming more vast and complex but still accuracy and reliability are the basic needs. So choosing a good software testing tool for any of product will increase the performance of software, improve the communication between the developers, produce the reliable and secure software and it will raise the customer satisfaction. For this research wizdl open source web service testing tool has been used to evaluate the weather web service.

wizdl: - wizdl is a .NET utility written in C# that allows faster import and test web services within the comfort of a windows forms GUI. It supports calling complex web services that take arrays and deeply nested objects as parameters. Being pointed at web service it dynamically creates and interface where the required data can be entered and execution of service can be done [15]. The tool allows saving the data as an XML file which can be used later for regression testing. The tool can be downloaded from <http://wizdl.codeplex.com>. wizdl is licensed by GPLv2. The study shows that the wizdl is simple and easy to use. For this study weather web service is used. Firstly add the wsdl file for weather web service. To perform a test the study needs to provide the input parameters for methods. The valid zip code e.g. 60601 provided as an input to the method for checking the weather report of the city. The result of the web service call along with the SOAP response is returned to the user. The testing tool provide the weather report in the form of city, state, weather city, weather ID, temperature, humidity, wind and pressure. The Figure 3 shows the result of the weather web service.

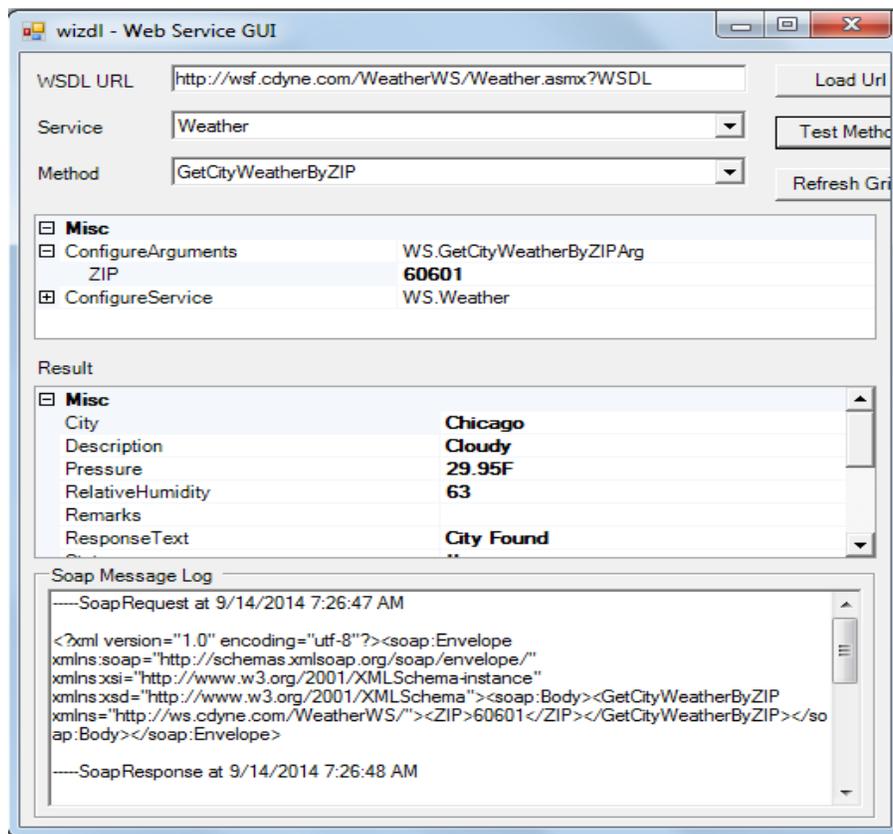


Figure 3: Weather Web Service Result.

Initially the tool was tested for weather web service and the results are shown in Figure 3. It can be seen that the parameters i.e. response time, bytes, throughput etc. which are important for efficient web service testing are absent in wizdl tool. Tool provides only the weather details of the city whose zip code is provided as the input to the tool. So the initiative has been taken to accommodate these important above mentioned parameters into the tool.

III. Modification in wizdl

This section represents efforts done to modify the initial tool to accommodate changes.

1) System Requirements

The tool was run on an Intel Core i3 2.40 GHz processor machine with 3GB RAM, Microsoft Windows 7 Home Basic, and 2Mbps Internet connection. Testing of the tool requires configuration which in turn includes installation, setting up test environment, data collection, selection of parameters and analytical survey. The sample web service i.e. weather forecast is tested on the selected configure tool.

2) Approach Followed

For this research practical approach has been used. wizdl weather web service tool is used in the study. Experiments were conducted based on the desired parameters. Thus observed results were collected and compiled for the analysis and evaluation to determine the efficiency of the tool. Weather web service was tested using wizdl tool.

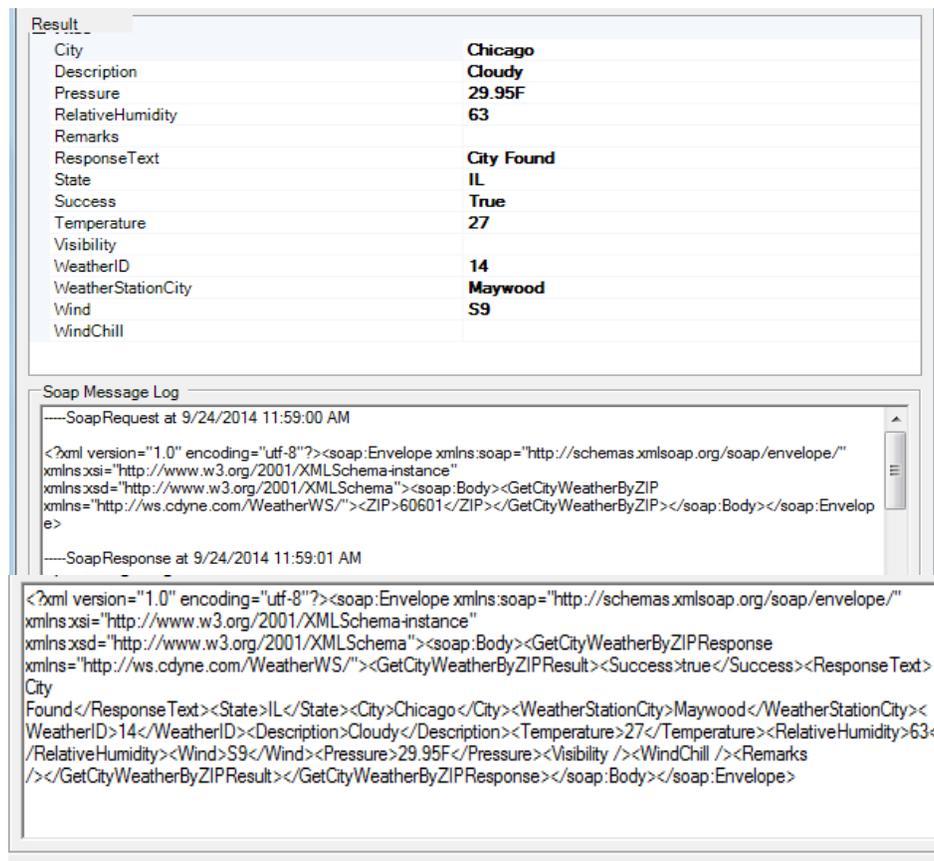


Figure 4:-Results without desired parameters.

Tool shows the complete detail of weather of the city whose zip code is provided as the input to the tool. Efficiency of the tools is based on some basic parameters that are 1) response time 2) throughput 3) number of data processed in terms of bytes.

Initially wizdl tool whose results are shown in Figure 4 do not show these three important parameters which are important for efficient web service testing tools. So the initiative has been taken to accommodate these important above mentioned parameters into web service tool. Respective formulas and coding have been incorporated in the initial source code of wizdl tool to calculate response time, throughput and bytes. After modification in wizdl tool and providing the same input in the form of zip code same output and three additional parameters response time, throughput and bytes are obtained as shown in Figure 5.

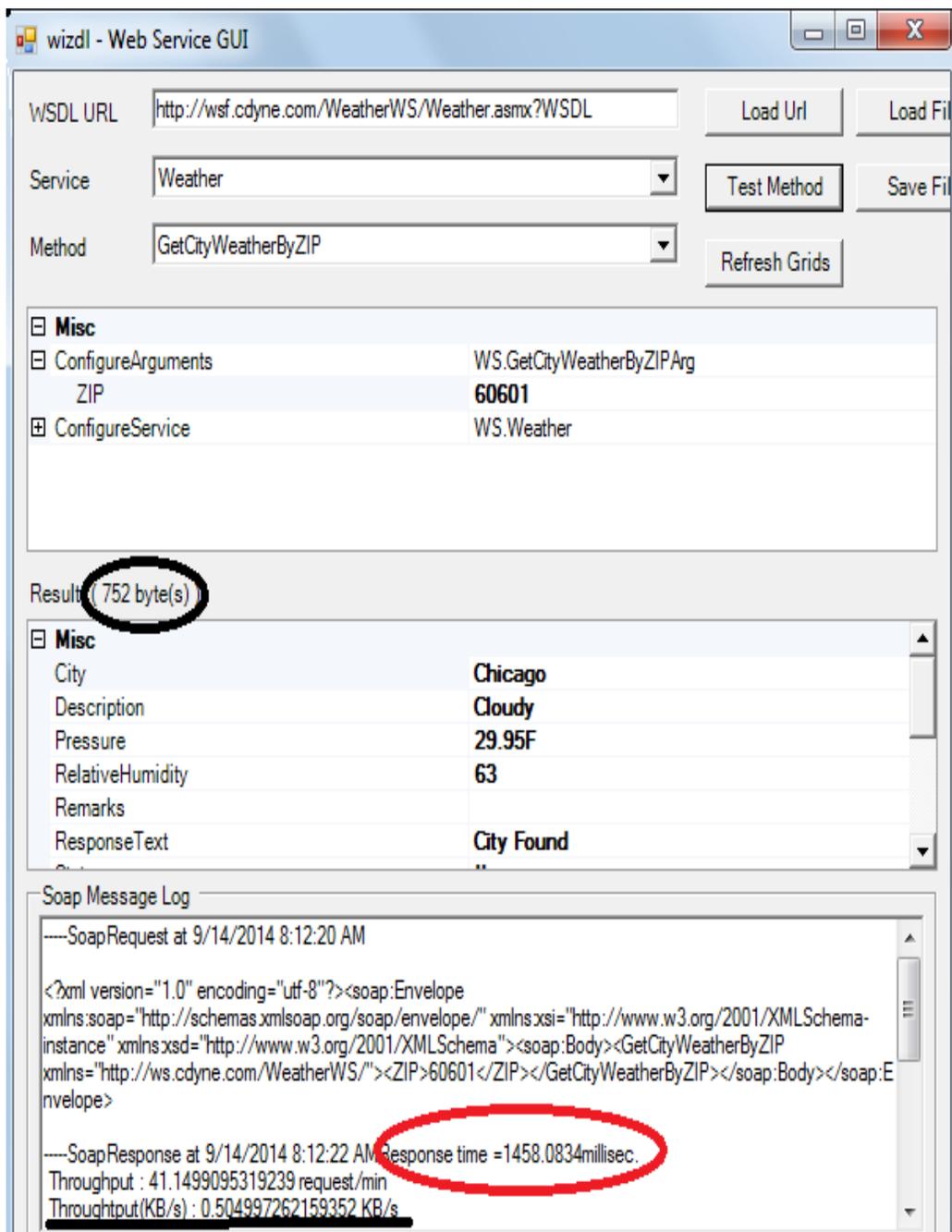


Figure 5: Enhanced Weather Web Service Results

From the above results it can be seen that the modified tool provides weather details of the city in terms of response text, relative humidity, description, pressure etc. and give three additional results in terms of response time, throughput and bytes. This can be visualized as the marked field in the Figure 5 and the results are summarized below.

IV. Results and Discussion

The observed results of both the initial and modified version of the wizdl has been shown in Table 1 and in Table 2 respectively.

Table 1:-Output without parameters (response time, bytes and throughput) in wizdl

Input	Results										
Zip Code	S.	R. Text	State	City	W. City	W. ID	Des.	Temp	Wind	Humidity	Pressure
60601	True	City Found	IL	Chicago	May wood	14	Cloudy	27	S9	63	29.95F

Note: - S denotes Success, R denotes Response and W denotes Weather.

Table 1 shows the evaluated results of the wizdl for respective input in the form of zip code.60601 is the valid zip code of Chicago.So tool provides the weather details of Chicago city. The first output parameter of the result is Success and its value is true means that the input is correct. Other parameters of the result provide weather report in the form of Response text i.e. city found, IL state, city Chicago, weather city Maywood, weather id 14,description of the weather i.e. cloudy,27 temperature, wind S9,humidity 63 and pressure 29.95F.

Table 2:- Output after integrating parameters (response time, bytes and throughput) in wizdl

Input	Results											Output			
Zip Code	S	R. Text	State	City	W. City	W. ID	D	T	Wind	H	P	R. Time	B	Th/m in	Kb/s
60601	True	City Found	IL	Chicago	May wood	14	Cloudy	27	S9	63	29.95F	1458.09ms	752	41.15	0.5

Note: - T: denotes temperature, H denotes humidity, P denotes pressure, D denotes description, Th denotes throughput and B denotes bytes.

For weather web service the zip code i.e. 60601 was given as common input to both the initial and modified version of the wizdl. After comparing the two tables the study shows that the initial version of wizdl gives output without any parameters. Later when the tool was modified and the output was re-evaluated the result included response time, bytes and throughput. After parameter integration the tool gives output with response time 1458.09ms, 752 bytes, 41.15 throughput/min and 0.5 in Kb/s.

V. Conclusions and Future Work

Testing of web service is a challenging activity which includes many characteristics such as runtime discovery, multi organization integration, throughput, calculation of bytes and much more. The paper evaluated the *wizdl* for few of these parameters named as response time, bytes and throughput and it was found that the tool does not supported them. Effort was made to integrate the calculation of these parameters into the tool and the result was re-evaluated. The difference was found in the prior results which can be seen in the Figure 5. Nevertheless there are still some limitations to current work that can be fruitfully improved and exploited for future work. The modified version of the tool calculated throughput for the single user which may limit its efficiency and reliability. So, the research work can be extended for the facilitation of throughput calculation for multiple users and by running the concurrent web services for the same input, results can be checked and validated.

References

- [1] Ahlawat Pooja and Tyagi Sanjay, "A Comparative Analysis of Load Testing Tools Using Optimal Response Time", *International Journal of Advanced Research In Computer Science and Software Engineering (IJCSSE)*, May 2013.
- [2] Myers Glenford J., "The art of software testing", New York, 2nd Edition 2004.
- [3] Sawant Abhijit A., Bari Pranit H. and Chawan P. M., "Software Testing Techniques and Strategies", *International Journal of Engineering Research and Applications (IJERA)*, May 2012.
- [4] Kaur Manjit and Kumari Raj, "Comparative Study of Automated Testing Tools: TestComplete and QuickTest Pro", *International Journal of Computer Applications (IJCA)*, June 2011.
- [5] Khan Mohd. Ehmer, "Different Approaches to Black Box Testing Technique for Finding Errors", *International Journal of Software Engineering and Applications (IJSEA)*, October 2011.
- [6] Prasad K.V.K.K.K., "Software Testing Tools", Dreamtech, 1st Edition 2006.
- [7] Hussain Shariq, Wang Zhaoshun, Toure Ibrahima Kalil and Diop Abdoulaye, "Web Service Testing Tools: A Comparative Study", *International Journal of Computer Science (IJCS)*, January 2013.
- [8] Larsson Andreas and Ekelund Marcus, "Web Services- A Comparative Study", *Vaxjo University at School of Mathematics and Systems Engineering (MSI)*, November 2004.
- [9] www.developer.com/services/article.php/145091, "Introduction to Web Services Part 2: Architecture" Retrieved on: 14 September, 2014.
- [10] www.tutorialspoint.com/soap/soap_message_structure.htm, "SOAP Message Structure", Retrieved on: 14 September, 2014.
- [11] <http://www.w3.org/TR/wsdl>, "Web Services Description Language (WSDL) 1.1", Retrieved on: 15 September, 2014.
- [12] http://www.tutorialspoint.com/wsdl/wsdl_elements.htm, "WSDL Elements", Retrieved on: 15 September, 2014.
- [13] http://www.tutorialspoint.com/uddi/uddi_overview.htm, "UDDI Overview" Retrieved on: 15 September, 2014.
- [14] Wala Tanuj and Sharma Aman Kumar, "A Comparative Study of Web Service Testing Tools", *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, February 2014.
- [15] <http://wizdl.codeplex.com/>, "Wizdl", Retrieved on: 15 September, 2014.