

A comparative study of testing semantic web services

Nooshin Anari¹, Hadi Saboohi^{2,3}, Gholamreza Ahakian⁴

¹Department of Computer, Kerman Islamic Azad University, Kerman, Iran

²Department of Computer, Karaj Islamic Azad University, Karaj, Iran

³Faculty of Computer Science and Information Technology, University of Malaya, Malaysia

⁴Department of Computer, Varamin-Pishva Islamic Azad University, Pishva, Iran

{anari_nooshin,hsaboohi,ahakian}@yahoo.com

Abstract: Web Services provide efficient reusability mechanism, thereby reducing the development time and cost. Mostly the source code of web services is unavailable to other developers who use these services. The manual effort spent by them in testing these web services is very large in order to increase the interoperability. Thus, automated testing needs to be developed for testing these Web services. This paper reviews test cases for Web Services using reduction techniques Pair-Wise Testing (PWT) and Orthogonal Array Testing (OAT) and compares the two techniques with general method. The structure of Web Services is specified using UML diagrams. The pre and post conditions for the service rule are specified using Object Constraint Language (OCL). The framework transforms into WSDL-S specifications. These specifications are parsed and transformed into structured DOM tree. Test data set generated by this framework would satisfy the constraints of the WSDL. The test cases are then developed based on the data generated, documented in XML based test files. The number of test cases required by general testing, PWT, OAT are compared and the better testing technique for testing Web Services is determined.

Keywords: Test case, web service, Reduction, Semantics, Pair wise, Orthogonal Array

1. Introduction

The concept of the Semantic Network Model was introduced by the cognitive scientist Allan M. Collins, linguist M. Ross Quillian and psychologist Elizabeth F. Loftus in different journals in the early 1960s [13,14,15,16,17] as a form to indicate semantically structured knowledge. It expands the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are connected to each other, enabling automated agents to use the Web more intelligently and do tasks instead users. The term "Semantic Web" was first used by Tim Berners-Lee, [12] the inventor of the World Wide Web and director of the World Wide Web Consortium ("W3C"), which administers the development of suggested Semantic Web standards. He explains the Semantic Web as "a web of data that can be processed directly and indirectly by machines".

Many of the technologies suggested by the W3C existed before they were located under the W3C control. These are utilized in different settings, especially those dealing with information that holds a small and defined domain, and where sharing data

is a common need, such as scientific study or data exchange among trades. Moreover, other technologies with the same aims have come up, such as micro formats. The Semantic Web works more on the solution. It has publishing in languages particularly intended for data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML). HTML explains documents and the links between them. RDF, OWL, and XML, by contrast, are able to describe conventional things such as people, meetings, or airplane parts.

In order to explain that supplement or replace the content of Web documents, these technologies are joined. Accordingly, content may be shown as descriptive data stored in Web-accessible databases, [18] or as markup within documents (specially, in Extensible HTML (XHTML) combined with XML, or, more often, merely in XML, with layout or rendering cues stored separately). The machine-readable descriptions allow content managers to give meaning to the content, i.e., to explain the structure of the knowledge we have about that content. In this way, a machine is able to process knowledge, instead of text, utilizing similar processes of that of human deductive reasoning and inference, thereby gaining more meaningful outcomes and aiding computers to do automated information collection and research. Like other scientific fields, specifically computer science, any evaluation needs test data. Test data are data that have been overtly produced or gathered to be utilized in evaluation practices. This is normal for any computer program. Web services as distance uses in a service oriented architecture (SOA) are not exempt from this notion[11] The quality of services is an important issue for making service-based software systems, and testing is needed for assessing the functional correctness,[6,9] performance and reliability of individual also composite services. Nonetheless, the dynamic properties of WS have lot of new problems to old testing methods. First, services are available, bound, raised and integrated at runtime. To integrate into the framework, testing has to be automated consisting automatic test case generation, test performance, test results collection and analysis. Second, WS suggests a fully specification-based process utilizing collections of XMLbased standards, for instance SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and WSFL (Web Services Flow Language). Instead of graphical user interfaces, services are

provided through programmable interfaces, which cannot be seen to end users. Therefore, test cases have to be produced based on the standard properties and test tools need to analyze the specifications to extract necessary information such as interface operations, data structures, and operation semantics. Third, to use and reuse test cases in the open environment and through service evolutions, test cases should be documented following the XML-based standard format [1]. IEEE Std. 610.12-1990 (IEEE, 1990) explains testing as “the process of operating a system or component under specified conditions, watching or recording the outcomes, and making an assessment of some facet of the system or component.” One of the major aims of testing is to cause problems, so that, based on the happening and nature of the failures, software producers are directed in the identification and removal of errors. Despite the fact that there is wide conformity on its significance for software quality assurance, testing is often not performed systematically enough. One probable description is that testing is a cost- and time-intensive process [3]. The purpose of this paper is reviewing of the considerations about test cases for Web Services using reduction techniques Pair-Wise Testing (PWT) and Orthogonal Array Testing (OAT). In the following part, the related studies in this area will be introduced. In part 3, the testing methods will be compared with each other, at the end, there will be a conclusion.

2. Related work

Animesh Chaturvedi et al, recognized, categorized, and capture the web service regression testing needs into three various categories, for instance, changes in WSDL, changes in code, and selective re-testing of web service operations. To capture above three changes they suggested three intermediate forms of WSDL, namely, Difference WSDL (DWSDL), Unit WSDL (UWSDL), and Reduced WSDL (RWSDL), respectively. These intermediate forms of WSDLs are after that joined to form Combined WSDL (CWSDL) which is then utilized for regression testing of the web service. This method is prototyped as an instrument; name Automatic Web Service Change Management (AWSCM), which aids in performing the effective regression testing of web services by opting the relevant test cases to constructing reduced test suite from the old test suite file of SoapUI [4]. William Alton Ballance et al, probed the effectiveness of the pair-wise method for one practically significant case — testing logical expressions in software. Pair-wise effectiveness is experimentally assessed in general terms and separately for particular kinds of faults. Two freely available software tools are used for pair-wise test set generation. Their experiments show that pair-wise testing has a benefit over random testing for every kind of faults. Especially, pair-wise testing is helpful for expressions with 8-10 Boolean variables where its effectiveness can become 70% or more [8]. M. ShabanJokhio et al suggested a novel approach for generating test cases based on user necessities for testing semantic web services. In SWS frameworks like Web Service Modelling Ontology (WSMO), the user needs are presented as a goal setting in form of a state model. They utilize a model checking approach to generate test cases from this state model.

To achieve this, they indicate a set of rules for translation from a goal specification to a formal B abstract state machine. The B representation of the goal specification is given as input to the model checker to generate concrete test cases using the assertion violation property of the model checker. Finally, the suggested framework is assessed using a real world case study based on, the Amazon E-commerce Service, This provides a platform for the web service testing research community, in which they utilize existing B tool support for testing and verification [5]. They have suggested new multi-dimensional criteria for testing the WS Transactions. Their approach generates test cases according to the dependencies between the activities involved in a WS transaction. The proposed criteria elaborate a classification-tree analysis for each kind of dependency in order to recognize the related test condition and test coverage items. Evaluation outcomes indicated that the proposed criteria have the potential to plan efficient test cases for WS transactions and to permit the tester to adjust the approach in form of its effectiveness, test effort and cost-benefit analysis. In addition, it provides the benefits of doing the testing process in a resource-scarce environment. Further the planning of the test cases is automatically produced in to meet the needs of the recognizing features of WS transactions. It decreases the expense of the test design and as well as amending its effectiveness [2]. Dengshan Tian et al, explained a test model for Web use based on layered method and TDD concept. This model classifies the unit test kinds in every layer, can be used in both J2EE and .NET applications, even WAP sites, particularly developing with XP methodology. With this model, the scalability and code quality of Web application can be highly raised [10].

3. Methods test

Orthogonal array testing is a black box testing method. It is a systematic, statistical way of software testing. It is utilized when the number of inputs to the system is nearly low, however, too big to permit for exhaustive testing of every probable input to the systems [19]. It is especially effective in detecting faults related to faulty logic within computer software systems [19]. Orthogonal arrays can be used in user interface testing, system testing, regression testing, configuration testing and performance testing. The permutations of factor levels consisting a single treatment are so chosen that their answers are uncorrelated and therefore each treatment gives a single piece of information. The net effects of organizing the experiment in such treatments is that similar piece of information is collected in the minimum number of experiments. Consider (table1) a system which includes 3 parameters and each of them has 3 values. To test all the probable combinations of these parameters (i.e. exhaustive testing) we will need a set of $3^3 = 27$ test cases. But instead of testing the system for each combination of parameters, we can utilize an orthogonal array to opt just a subset of these combinations. Utilizing orthogonal array testing, we can enhance the test coverage while decreasing the number of test cases to consider [19]. We here suppose that the pair, that

increases interaction between the parameters, will have more deficits and that the method works.

Orthogonal Array

Test case ↓	Parameter 1	Parameter 2	Parameter 3
1	1	1	3
2	1	2	2
3	1	3	1
4	2	1	2
5	2	2	1
6	2	3	3
7	3	1	1
8	3	2	3
9	3	3	2

Table 1- Test cases and differences parameters

Regarding that assumption, the table indicates the set of nine combinations of parameters which are enough to have the fault, considering the interaction of the input parameters, which is highly efficient and economical. The array is orthogonal, due to the fact that all probable pair-wise combinations between parameters happen just once. The given L9 Orthogonal Array assess result of test cases as follows, [19] Single Mode Faults - Single mode faults happens just due to one parameter. For instance, in above orthogonal array if test cases 7, 8 and 9 display error, we can expect that value 3 of parameter 1 is causing the error. In this way, we are able to find in addition to isolate the error.

Double Mode Fault - Double mode fault is produced by the two particular parameters values interacting together. Such an interaction is a destructive interaction between interacting parameters.

Multimode Faults - If more than two interrelating components yield the steady erroneous output, then it is a multimode fault. Orthogonal array spots the multimode faults.

Pair-Wise Testing Technique (PWT)

In computer science all-pairs testing or pair wise testing is a combinatorial method of software testing that, for each pair of input parameters to a system (normally, a software algorithm), examines all possible discrete combinations of those parameters. Implementing carefully selected test vectors, this can be performed much quicker than a tiring search of all combinations of all parameters, by "parallelizing" the tests of parameter pairs. Consider that the test function has N parameters given in a set $\{P_i\}=\{P_1,P_2,\dots,P_N\}$. The range of the parameters are given by $R(P_i)=n_i$ We find out that the set of choices of ranges $x=\{n_i\}$ can be a multi set, due to the fact that there can be multiple parameters having similar number of choices [20].

First we describe Max(s) as one of the maximum of the multi set S. Then, the number of pair-wise test cases on this test function would be: $T=Max(x) \times \sum_{x \in S} Max(x)$

Simply that would mean, if the $n=Max(x)$ و $m=\sum_{x \in S} Max(x)$ after that the number of tests is normally $O(nm)$. Where n and m are the number of probabilities for each of the two parameters with the most choices. In this case the parameters are enabled with choices range of 2, Choice Type with 3, and Category with 4. That would mean: $x= \{2, 3, 4\}$ Hence, $n = 4$, $m = 3$ and number of tests would be 12.

Object Constraint Language (OCL)

A formal language is utilized to define expressions on UML models. These expressions normally specify invariant conditions that must hold for the system being modeled or queries over objects defined in a model [7].

4. Evaluation

This paper reviews different techniques of test cases utilizing both PWT and OAT. Test runs show that for OAT and PWT is the same for different levels for Factor 3. For ex, Factor (2,1) and Level (2,10) means that two parameters takes two different values while the third parameter takes 10 different values. Comparing PWT and OAT runs for various strength shows that OAT runs required for strength 2 is lesser compared to PWT, whereas OAT runs increase with strength.

The number of test cases more in OAT technique than the PWT techniques. OAT considers more combinations than PWT techniques. With the rise in levels, OAT technique needs less number of test cases than the PWT. Based on our consideration OAT is optimal technique for more number of parameters.

5. Conclusion

This paper reviews test cases for Web Services using reduction techniques Pair-Wise Testing (PWT) and Orthogonal Array Testing (OAT) and compares the two techniques with general method. The structure of Web Services is specified utilizing UML diagrams. The pre and post conditions for the service rule are specified using Object Constraint Language (OCL). The framework transforms into WSDL-S specifications.

In this article various test case reduction techniques and determines the better technique for testing the semantic based web services. When there are few parameters, PWT make sense. When there is more number of parameters (factors) taking various values (levels), OAT is better technique. In our consideration indicates very mild differences between the techniques. Number of runs required is highly reduced saving lot of testing attempt in both the techniques, compared to General Technique. If PWT is not giving coverage, the tester can move to OAT technique. Test case reduction is very much useful for application's regression testing.

References

- [1] Xiaoying Bai, Wenli Dong: WSDL-Based Automatic Test Case Generation for Web Services Testing, Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering.
- [2] Rubén Casado a, Muhammad Younas b, Javier Tuya: Multi-dimensional criteria for testing web services, Transactions, Journal of Computer and System Sciences 79 (2013) 1057–1076.
- [3] Abbas Tahir, Davide Tosi, Sandro Morasca: A systematic review on the functional testing of semantic web services, The Journal of Systems and Software 86 (2013) 2877–2889.
- [4] Animesh Chaturvedi, Atul Gupta: A Tool Supported Approach to Perform Efficient Regression Testing of Web Services, 2013 7th IEEE International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)
- [5] M. Shaban Jokhio, Gillian Dobbie, Jing Sun: A Framework for Testing Semantic Web Services Using Model Checking, 2009 Fourth South-East European Workshop on Formal Methods
- [6] Emelie En, Per Ru, Mats Sk: A Systematic Review on Regression Test Selection Techniques, information and software technology 52(2010) 14-30
- [7] A. Askarunisa, A.M. Abirami, S. Madhan Mohan: A Test Case Reduction Method for Semantic Based Web Services, 2010 Second International conference on Computing, Communication and Networking Technologies
- [8] William Alton Ballance, Sergiy Vilkomir, William Jenkins: Effectiveness of Pair-wise Testing for Software with Boolean Inputs, 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation
- [9] Cesare Bartolini, Antonia Bertolino, Sebastian Elbaum, Eda Marchettia: Bringing white-box testing to Service Oriented Architectures through a Service Oriented Approach, The Journal of Systems and Software 84 (2011) 655–668.
- [10] Dengshan Tian, Jiwen Wen, Yin Liu, Ning Ma, Huiping Wei: A Test-Driven Web Application Model Based on Layered Approach
- [11] Hadi Saboohi, Sameem Abdul Kareem: A Resemblance Study of Test Collection for World-altering Semantic Web Services, Proceedings of the International Multi-Conference of Engineers and Computer Scientists 2011, Vol I, March 16-18, 2011, Hong Kong.
- [12] Berners-Lee, Tim; James Hendler; Ora Lassila (May 17, 2001). "The Semantic Web". Scientific American Magazine. Retrieved March 26, 2008.
- [13] Allan M. Collins; M. R. Quillian (1969). "Retrieval time from semantic memory". Journal of verbal learning and verbal behavior 8 (2): 240–247. doi:10.1016/S0022-5371(69)80069-1.
- [14] Allan M. Collins, A; M. Ross Quillian (1970). "Does category size affect categorization time?". Journal of verbal learning and verbal behavior 9 (4): 432–438. doi:10.1016/S0022-5371(70)80084-6.
- [15] Allan M. Collins, Allan M.; Elizabeth F. Loftus (1975). "A spreading-activation theory of semantic processing". Psychological Review 82 (6): 407–428. doi:10.1037/0033-295X.82.6.407.
- [16] Quillian, MR (1967). "Word concepts. A theory and simulation of some basic semantic capabilities". Behavioral Science 12 (5): 410–430. doi:10.1002/bs.3830120511. PMID 6059773.
- [17] Semantic memory |book:Marvin Minsky (editor): Semantic information processing, MIT Press, Cambridge, Mass. 1988.
- [18] Artem Chebotko and Shiyong Lu, "Querying the Semantic Web: An Efficient Approach Using Relational Databases", LAP Lambert Academic Publishing, ISBN 978-3-8383-0264-5, 2009.
- [19] Copeland, Lee. "Object-Oriented Testing." Software Quality Engineering. STAR East 2001. The Rosen Centre Hotel, Orlando, Florida. 14 May 2001
- [20] PRACTICAL COMBINATORIAL TESTING, D. Richard Kuhn, Raghu N. Kacker, Yu Lei, October, 2010