TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 12010 - 12017

Comparative Evaluation of State-Of-The-Art Approaches in Web Service Testing

Israr Ghani[1], Wan M.N. Wan-Kadir[2], and Ahmad Mustafa[3]

Article Info
Volume 82
Page Number: 12010 - 12017
Publication Issue:
January-February 2020

Article History
***Article Received**: 18 May 2019*
***Revised**: 14 July 2019*
***Accepted**: 22 December 2019*
***Publication**: 21 February 2020*

Abstract

Abstract: Due to the increasing popularity of service-oriented software that are mainly developed using web service technology, web service testing is considered as an important and the fastest growing technical activity in the software engineering field. Web-service testing is required to detect errors and anomalies, and consequently improve the quality of web service-based applications. Web service-based applications have several dissimilarities in the testing process as compared to other software systems testing procedures. There are many efforts made to introduce effective web service testing approaches. However, the proposed approaches focus on different problems and achieved various degree of success. In this study, we categorize and deliberate the prominent web service testing approaches. The approaches are categorized into model-based, AI-based and semantic-based web service testing approaches. This study also presents a set of criteria used to evaluate and compare the state-of-the-art web service testing approaches and conducts the comparative evaluation. The comparative evaluation results may illuminate the strengths and gaps of the existing approach as well as give insight into the potential areas where the effectiveness of web service testing approaches can be improved.

***Keywords:** Service-oriented software, software testing, and web service-based application testing.*

I. INTRODUCTION

Web services are generally defined as self-explanatory applications which can be promoted, traced, and used through the internet using a set of architectural components such as SOAP, WSDL, and UDDI [1]. Because of the extensive recognition of SOAP and REST as a resolution to interoperation, reprocess, globalization, there has been a large flow of web services deployed over the Web [2]. For that reason, there might be several facility providers competing to deal the similar functionality, but with different quality of service (QoS) such as value and response time. Therefore, QoS is a critical measure for distinguishing between functionally similar web services.

As more businesses depend on web service-based applications, the value and consistency of the web service based applications become important. In Order to meet these requirements, web service testing can determine the majority of the mistakes or errors of the web service-based application. To make sure the requirements have been fulfilled according to the request and given specifications by the application, web service should be tested carefully. Due to web service characteristics of distribution, heterogeneity, platform independence and concurrence, web service testing have become more difficult than common applications. The extensive spread of the Internet has resulted in a significant extension of the requests for web service-based applications with an ever-increasing number of extreme prerequisites.

Web service-based applications have numerous details such as disperse heterogeneous, independent platform and real-time synchronization. Now, because of these features, testing of web service has become harder and more difficult as compared to traditional software testing. Therefore, to discover significant testing approaches for web service-based application, testing has become a new challenge and preference.

Previous research has been conducted with a focus on various testing issues in the web service-based application, and several solutions have been recommended. The functionality, quality of testing and testing approaches used for performance, interoperability, maintainability, reusability, efficiency, and reliability of web service-based applications are an essential testing challenge. These testing challenges have introduced new testing techniques and also provide the new ways to improve the requests for testing models, testing methodologies and testing approaches to progress the testing approaches in the state-of-the-art evaluation. This study concentrates on a review of the state-of-the-art web service testing approaches and provides a comparative evaluation that highlights the gaps and challenges in web service testing. In the next section, We also compare these testing approaches with a set of evaluation criteria in order to introduce the general idea and classification of existing testing approaches for web service-based application. We expect this research will assist researchers to deliver well-built testing approaches for web service-based applications.

II. Web Service Testing

In order to meet the web service testing requirements, it is necessary to make sure that complete APIs visible by the application are functioned with expected outcome after testing. To clarify, Web services testing is the process of testing to make sure that application communicates and accesses functions correctly. Testing of web service-based application also need to make sure by effectively verifying the behavior of web services connected to them with appropriate request and response [3]. Web service contains two types, REST and SOAP, in order to communicate and synchronize records and data with facts over internet protocol to ensure that the QoS tests most common quality attributes in web service testing in order to ultimately to improve the service testing [4].

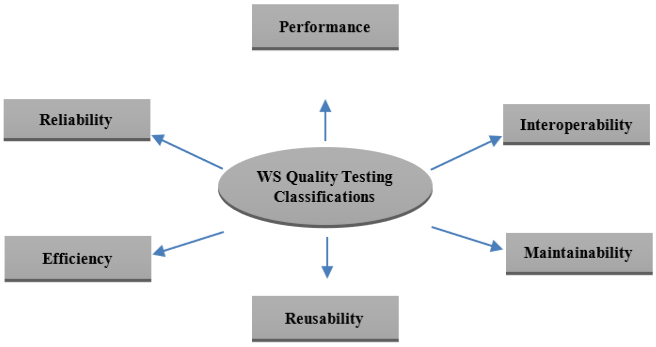

Fig. 1. Web Service (WS) Quality Parameters Classifications

A. Quality of service (QoS)

Quality of Service (QoS) is one of the important challenges currently in the WS testing. In order to mention QoS, nonfunctional properties such as performance, interoperability, maintainability, reusability, efficiency and reliability are proposed. Since a quality based service uses additional services to procedure itself, its value consists of the WSs and its consumption. To be recognized by its clients, a business should try to deliver good quality regarding the customers' desires to a composed WS. As QoS information is assigned to each basic service, performance, interoperability, maintainability, reusability, efficiency, efficiency and reliability were selected.

1) Performance

It specifies the ability of the mechanism to ensure performance quality and also measures how fast a request of service can be completed. Performance also makes sure to test the sensitivity under specific workload and stress situations. There are two important types of performance testing, which are stress and load testing. Stress testing usually works to test the performance of a system under extreme capacity, increasing its maximum load. Load testing is performed by executing the system under explicit predefined load capacity to realize its performance and efficiency [5].

2) Interoperability

It focusses on the capability of the system to interrelate efficiently with different frameworks whenever needed. It also makes sure that testing web services and system are interoperable to implement services between the different development environments. Usually a framework needs to perform data trade through different components to associate with additional frameworks. This is a vital element, as the instrument should have the capacity to collaborate and arrange with outer parts [6].

3) Reusability

This term specifies the ability of the testing mechanism to accomplish reusability web software and web service testing. Under explicit embedding of the same functionality and circumstances, in this study reusability deals with the sensitivity of the web service test. Multiple fluctuating constraints can be reused from the earlier version's test cases of web service. So, testing of reusability can reduce work effort to provide a solution with constraint standards for those variables while testing latest changes in the system and application [7].

4) Efficiency

This attribute shows the capacity of web service and software system to play out and run by using a base measure of assets. Efficiency in software and web service testing will assess if the web service testing makes the most efficient consumption of framework assets like processor perimeter, RAM, transfer speed and so on. An instrument excessively consuming plate space or memory shows a low proficiency. Thus, in testing the model, a framework or technique is being used to make sure the results are prompt while performing efficiency so that a testing technique that does not respond in time to produce the expected results can be considered as inefficient [8].

5) Reliability

This parameter demonstrates the capacity of the application to execute the given assignment by consuming a low volume of assets, which signifies the capacity of a web service testing to execute the given necessary tasks under mentioned circumstances within the respective time period. The purpose of the reliability is to make sure to maintain the overall measures of web service testing and its quality. Reliability will evaluate the usage of system resources like RAM, processor capacity, bandwidth while testing web service in this study [9].

The mentioned testing features and terms are the straightforward and valuable testing options, and thus need to be valued and realized for web service quality testing. Therefore, several testing approaches have been introduced to make the quality testing better in order for the mentioned parameters to improve the service, which is discussed in the state-of-the-art approaches in WS testing section.

B. State of-the art approaches in web service testing

Web Services-based applications have the capability of additional features to their applications as part of implementing quality of service testing aspect in their services. It has been observed that the limitation between following aspects of arrangement are not always stringently well-defined [10]. Several testing approaches, for example, can be also deliberated as a multi dimension web service-based testing approaches. However, these major testing approaches have been categorized as follows.

1) Model-Based Approach

Model-based testing (MBT) approach in a web service that facilitates the behavioral testing of SOAP and REST services, respecting both the service types with the intention of quality assurance of performance, interoperability, maintainability, reusability, efficiency and reliability and providing a more systematic and formal testing. Model-based approach has been proposed to test the model based web service using the UML protocol state machine model in order to ensure the quality of service attributes (QoS)[11]. The starting point of the approach is to create the behavioral model (protocol state machine) of a specific behavior of the SUT.

The behavioral model must then be converted to the XML Metadata Interchange (XMI) standard. The XMI was preferred subsequently due to its good format in existing up to date modelling [12]. The approach uses an adapted UML protocol state machine that conforms to the web service and its type. From theoretical and experimental perspective, the authors consider that this preliminary effort provokes further research on model-based testing of web services. It is important to analyze how the testing approach performs in the wild with real-world and industrial web services with the purpose of determining the effectiveness of the proposed testing approach. To conclude, investigational assessments (e.g. Case studies) should be introduced.

2) Artificial Intelligence (AI)-Based Approach

At present, numerous testing approaches familiar to AI planning have been introduced to ensure the quality of service (QoS) [13, 14]. Most AI testing approaches depend on the ideal of the state-transition system. The point of arranging is to discover which activities to apply to which states so as to accomplish some goal, beginning from some predefined condition.

With the aim of explaining and resolving some objective, starting from any given situation, particularly web service testing issues, Artificial Intelligence (AI) has contributed comprehensively in this area [15, 16]. The AI-based testing approaches differ in web services testing from the approach taken for more traditional software testing techniques. In terms of the design of test cases and the metrics used to measure quality like maintainability, interoperability, performance, reusability, efficiency and reliability, Artificial Intelligence (AI) contributed extensively to make the testing approach and software performance more efficient and reliable. To test and evaluate, the functionality and performance of this concentrated system has been familiarized at both the end-to-end integrated level testing and individual component level testing against quality parameters which have been discussed earlier [17].

Furthermore, we present certain WS testing approaches dependent on AI planning. Incredible overviews of AI- based testing approaches ways to deal with handle the issue of WS quality attributes.

Chakravarty [18] adapt and extend the AI based testing approach to evaluate the performance and accuracy for WSs. Indeed, this testing approach depends on the AI framework, for instance, of stress testing an AI-based web service. The authors address the AI interpretation framework considered, and the methodology utilized for stress testing it is introduced, which are close to correlations with a progressively conventional testing approach. Advantages of using this AI based approach are Predictability of response, Mean Time between Failures (MTBF) and Capacity to handle expected traffic in order to precisely assess separate changes in AI framework reaction brought about by load and changes brought about by enhancements in the AI algorithm as the framework learns over long spans. Executing and Implementing tests with these needs and knowledge have prompted a stress testing framework that has remained helpful and versatile as the WS has advanced to help a developing number of features and language sets.

Anabalon and De Renzis [19] introduced an extension of an earlier AI based testing approach that applied Case-Based Reasoning (CBR) for WS Selection, where WS are interactively tested for conformance of specified essential functionality. It also presented an improvement on our CBR approach in [20] for the problem of service selection. For the most part, the novel testing approach introduces the case validation phase to ensure the appropriateness of a proposed resolution by a dynamic behavioral assessment. The testing-based Behavior Compatibility Analysis, through the applied testing criteria, escalates the consistency and effectiveness level.

Androcec, et al. [21] discuss the Service-Level Interoperability Issues of Platform as a Service. To address interoperability testing complications, the ontology determined statistics intervention is used and tested in this study. This is a more flexible approach than direct mapping and transformation approach used in web service composition languages like BPEL. Testing and validation was executed on a use case where existing Salesforce client is added to data container in content management system used on Google App Engine[22]. The stated use case verifies that ontology driven service data mediation approach and AI planning techniques are proper to find and resolve service-level interoperability complications among PaaS APIs. Using AI test approach both authors on cloud's Web APIs and SaaS offers.

For functionality testing using the AI testing approach, some AI based testing solutions have been discussed for web service testing, especially for component testing, stress testing, test case generation criteria to make the system interpretable.

3) Semantic-Based Approach

The concept of the Semantic Network Model was introduced by the cognitive scientists Sadoski and Paivio [23], Mittal and Jain [24], and psychologist Loftus and Loftus [25] in different journals in the early 1960s as a form to indicate semantically structured knowledge. The semantic web consents to the illustration and synchronization of information and data in a significant approach, simplifying automated procedure of explanations on the web service testing [26]. Interpretations on the semantic web are precise associations between information and data resources on the web services and connect information and data resources to standard terms [27].

These connective configurations are termed ontologies. Since the web service causes time and cost decrease, these components produce the capacity in order to ensure the quality attributes against that ontology to associate information and data assets to recognized terms in web services.

Semantic web service testing that uses decrease approaches Pairwise Testing (PWT) and Orthogonal Array Testing (OAT) [28] and relates the two testing approaches with general procedure has been introduced for such quality attributes as performance, interoperability, maintainability, reusability, efficiency and reliability.

III. Comparative Evaluation

In this segment, we present a comparative evaluation of the most progressive testing approaches for web service that are discussed in the previous section with respect to a set of well-defined criteria. The evaluation criteria will be recognized and depicted then the outcomes and results will be displayed in the wake of applying the evaluation criteria into existing approaches. The aim of this study is to provide a classification and comparison of the most prominent testing approaches with determination to check how far the prevailing testing approaches can support to improve the web service testing. Therefore, those testing approaches can be categorized into four main categories, namely Component based approach, Model based approach, AI based approach, and Semantic based approach. We consider that at this time it is not conceivable to claim that the existing classification is comprehensive.

An evaluation was executed on the existing approaches for web service testing, based on six identified criteria to fulfil the testing requirements as mentioned and discussed in previous section (QoS) such as performance, interoperability, maintainability, reusability, efficiency and reliability. These six quality parameters have been used as evaluation of comparative study, to evaluate the weaknesses and the strengths of the existing testing approaches of web service testing.

A. Evaluation Criteria

This section describes and clarifies concisely the evaluation criteria used in this study. Our objective is that any testing approach to web service must satisfy the subsequent conditions. We considered an approach as a "Good" advantage approach on the off chance that the approach can give all parts of the criteria against maintainability, interoperability, performance, reusability, efficiency and reliability while testing web service.

If an approach can provide some portion of what the quality of service (QoS) criteria adopts, it is reflected as "Average" quality testing approach established on mentioned standard. If test approach does not deliver the expected results according to the (QoS) criteria, it is considered as a "Low" quality approach regarding that criteria. Similarly, if an approach is tool supported, it is considered as "Yes", and if that specific approach is not able to support tool, it is considered as "No". Same case with "Fulfil the quality parameter requirements" such that if an approach satisfied the given criteria according to the mentioned quality parameters, it is considered as "Yes". If discussed technique does not provide criteria expect, it is considered as a "No".

B. Evaluation Results

Quality of service attributes, parameters mentioned and discussed in previous section and evaluation criteria and the results of the presented comparative assessment of existing testing approaches of web service and result can be seen in Table I.

Table I. Results of Comparative Evaluation

Criteria / Approaches	QoS	Performance	Interoperability	Maintainability	Reusability	Efficiency	Reliability	Tool Supported?	Fulfil the quality parameter requirements?
WS-BPEL [11]	Low	Yes	Average	Yes	Low	Low	Low	Yes	No
BLEU [29]	Good	Yes	Low	Yes	Low	Yes	Yes	Yes	Yes
CBR [18]	Good	Low	Low	Low	Low	Yes	Low	No	Yes
SAWSDL [21]	Low	Low	Yes	Yes	Low	Yes	Yes	Yes	No
IOT [26]	Average	Average	Low	Low	Low	Average	Average	Yes	No
RBT [27]	Average	Low	Low	Yes	Low	Low	Average	No	No
WSC [28]	Low	Low	Low	Low	Low	Average	Low	No	No

Our results indicate that most of the existing testing approaches of web services demand additional information packaged with the AI based testing approach to facilitate model based and semantic based testing approaches to simplify the testing process. The outcome also specifies and/or promote structure for reliable use of web service applications. Moreover, based on the result in Table 1, we conclude that each testing approach suffers from some drawbacks and have their own limitations. Based on functional, nonfunctional testing targets to evaluate the quality parameters performance, interoperability, maintainability, reusability, efficiency and reliability, all testing approaches work to resolve the testing problems.

Therefore, our research aim is to introduce a new means of testing web service that efforts to cover furthermost. As this study is in the initial stage, we do not claim that our classification and evaluation can be seen as comprehensive. We summarize that there is a requirement for additional research in the field of web service testing to make the semantic based and model based testing by implementing AI testing techniques and we hope this study will help as preliminary review for the software testers.

As upcoming work, this study would be prolonged to deliver inclusive information which contains a systematic literature review of state-of-the-art in web service testing approaches. We expect the systematic literature review will also help to discover the new testing gaps and future directions in the area of web service testing. The final objective, based on the result of the comparative evaluation, is that these testing approaches will be extended to provide more comprehensive testing approaches in the state of the art testing approaches that address these limitations, and it will be validated using strength case studies.

IV. Conclusion

This study aims at providing an overview which introduced to latest development in web service testing. Based on the review of the existing prominent web service testing approaches, they can be classified into three categories, namely model-based, AI-based and semantic-based testing approaches. However, these testing approaches are not unusually constrained to prove required outcomes and need more concentration to present progressively productive and successful testing ways to deal with get expected outcomes in WS testing. In addition, we provide a comparison of the selected approaches in each category to measure the degree of success of each approach in fulfilling the requirements according to the evaluation criteria. The actual challenge with most of these approaches are the confirmation of performance, interoperability, maintainability, reusability, efficiency and reliability of web service testing and the exploration of quality of service aspects. We conclude that there is a need for further research in the field of web service testing especially in the areas mentioned in our study. For our future work, we aim at improving the web service testing approaches to address these limitations, and validate it using the industrial strength case studies.

ACKNOWLEDGEMENT

The authors would like to precise their deepest gratitude to Universiti Teknologi Malaysia (UTM) and Ministry of Higher Education Malaysia (MOHE) for their financial support under the Research University Grant Scheme (Vot number Q.J130000.2516.19H64).

V. REFERENCES

[1] M. Wynne, A. Hellesoy, and S. Tooke, The cucumber book: behaviour-driven development for testers and developers. Pragmatic Bookshelf, 2017.

[2] S. Jehan, I. Pill, and F. Wotawa, "SOA testing via random paths in BPEL models," in IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops, 2014: ABC.

[3] A. Gandhi, V. Elbert, P. Perry, and G. Wang, "Method, apparatus, and computer program for providing web service testing," ed: Google Patents, 2015.

[4] H. M. Sneed, "Testing web services in the cloud," in Software Testing in the Cloud: Perspectives on an Emerging Discipline: IGI Global, 2013.

[5] T. M. Ahmed and C.-P. Bezemer, "Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web," in Proceedings of the 13th International Conference on Mining Software Repositories, 2016.

[6] S. Hussain, Z. Wang, I. K. Toure, and A. Diop, "Web service testing tools: A comparative study," arXiv preprint arXiv:1306.4063, 2013.

[7] X. Wang, B. Zhou, and W. Li, "Model-based load testing of web applications," 2013.

[8] Y. Du, W. Tan, and M. Zhou, "Timed compatibility analysis of web service composition: A modular approach based on Petri nets," 2013.

[9] W. Ahmed and Y. W. Wu, "A survey on reliability in distributed systems," 2013.

[10] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," 2011.

[11] C.-s. Wu, "The Web Services Composition Testing Based on Extended Finite State Machine and UML Model," Fifth International Conference on Service Science and Innovation, Kaohsiung, Taiwan, 2013.

[12] R. Sungkur and S. Daiboo, "SOREST, a novel framework combining SOAP and REST for implementing web services," 2015.

[13] J. Zhou and X. Yao, "Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service," Applied Soft Computing, 2017.

[14] J. Lemley, S. Bazrafkan, and P. Corcoran, "Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence," IEEE Consumer Electronics Magazine, 2017.

[15] F. Michel and C. F. Zucker, "SPARQL micro-services: lightweight integration of web APIs and linked data," in LDOW Workshop of the World Wide Web Conference, 2018.

[16] J. Bozic and F. Wotawa, "Planning the attack! or how to use ai in security testing?," in IWAISe: First International Workshop on Artificial Intelligence in Security, 2017.

[17] J. Cheng, C. Liu, M. Zhou, Q. Zeng, and A. Ylä-Jääski, "Automatic composition of semantic web services based on fuzzy predicate petri nets," 2014.

[18] A. Chakravarty, "Stress testing an ai based web service: A case study," in Seventh International Conference on Information Technology: New Generations, 2010: IEEE.

[19] D. Anabalon and A. De Renzis, "Testing-supported Case-Based Reasoning for Web Service Selection," in IEEE Biennial Congress of Argentina (ARGENCON), 2016.

[20] S.-H. Ji, J. Ahn, E.-B. Lee, and Y. Kim, "Learning method for knowledge retention in CBR cost models," Automation in Construction, vol. 96, pp. 65-74, 2018.

[21] D. Androcec, N. Vrcek, and P. Küngas, "Service-level interoperability issues of platform as a service," in IEEE World Congress on Services, New York, 2015.

[22] D. Petrova-Antonova, S. Ilieva, and D. Manova, "TASSA: Testing Framework for web service orchestrations," 2015.

[23] M. Sadoski and A. Paivio, Imagery and text: A dual coding theory of reading and writing. 2013.

[24] K. Mittal and A. Jain, "Word sense disambiguation method using semantic similarity measures and owa operator," ICTACT Journal on Soft Computing, 2015.

[25] G. R. Loftus and E. F. Loftus, Human memory: The processing of information. New York, 2019.

[26] S. K. Datta and C. Bonnet, "Approach for Semantic Interoperability Testing in Internet of Things," in Global Internet of Things Summit (GIoTS), 2018.

[27] X. Bai, R. S. Kenett, and W. Yu, "Risk assessment and adaptive group testing of semantic web services," 2012.

[28] F.-G. Liu, "Design and implementation of semantic web service clustering algorithm," in International Conference on Machine Learning and Cybernetics, 2013: IEEE.

[29] A. Chakravarty, "Stress testing an ai based web service: A case study," in 2010 Seventh International Conference on Information Technology: New Generations, 2010, pp. 1004-1008: IEEE.

AUTHORS PROFILE

ISRAR GHANI Ph.D. Student in Department of Software Engineering, School of Computing, Faculty of Engineering Universiti Teknologi Malaysia, Johor Bahru, Malaysia. He received the M.S. degree in computer science from Pakistan in 2010. His research interests include

Email: israrghani@outlook.com

Member of Software Engineering Research Group (SERG)

Dr. Wan M.N. Wan Kadir is an Associate Professor of Software Engineering in the School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM). He received his Ph.D. in the field of Software Engineering (SE) from the University of Manchester. He has been with UTM since 1997. He was the Head of SE Department (2005-2009) and the Deputy Dean of the Faculty of Computing (2010-2014). Currently, he is the Chair of School of Computing. He was invited as the Keynote Speaker of the ICT-ISPC 2013 and MySEC 2014, and Invited Speaker of the InfoTech 2014. He also served a Program Committee of various local and international conferences related to software engineering. His research interest covers various SE knowledge areas based on the motivation to reduce the cost of development and maintenance as well as to improve the quality of large and complex software systems. They include adaptable software architecture, software evolution, service-oriented computing, model-driven development, software testing, and software quality assurance. His research and consultancy work has been supported by numerous grants from government and industries totaling over RM 1 million.

Email: wnasir@utm.my

Member of Software Engineering Research Group (SERG)

AHMAD MUSTAFA Ph.D. Student in Department of Software Engineering, School of Computing, Faculty of Engineering Universiti Teknologi Malaysia, Johor Bahru, Malaysia. His research interest includes Natural Language Requirement Analysis, Application of NLP in requirement Analysis, Software Test case generations.

Email: cs.mustafa.khan@gmail.com

Member of Software Engineering Research Group (SERG)