International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Integrated Framework for Functional and Data Regression Testing of Web Based Applications

Gyanendra Kumar Dwivedi[1], Amit Agnihotri[2]

[1]Computer Science & IT Department, JRD State University, Chitrakoot - 210204
Email: *gyanendra.mtech[at]gmail.com*

[2]Computer Science & IT Department, JRD State University, Chitrakoot - 210204
Email: *amit.agnihotri[at]gmail.com*

Abstract: ***Web services are software technologies based on Service-Oriented Architecture (SOA) that enable the delivery of business functionalities over the web. Therefore, ensuring the quality and maintainability of web services is essential. Software maintenance is widely regarded as one of the costliest activities in the software development life cycle. Regression testing is a critical component of software maintenance and is performed whenever modifications are made to an existing software system. However, regression testing for web services is particularly challenging and time-consuming due to their inherently distributed, heterogeneous, and dynamic characteristics. Consequently, reducing the effort required for regression testing is important for minimizing overall software maintenance costs. This paper presents an efficient approach for performing regression testing of web-based application systems whenever changes are introduced, with the objective of reducing testing effort while maintaining the effectiveness of the testing process.***

Keywords: Software Maintenance, Web Applications, Web Services, Regression Testing, Automated Testing

1. Introduction

A web service is a software technology based on the principles of Service-Oriented Architecture (SOA) and is widely used across various application domains. Web services employ XML for information representation and tagging, WSDL for describing service interfaces and behavior, SOAP for communication among different parties, and UDDI for publishing and discovering available services. One of the primary advantages of web services is that they enable organizations using heterogeneous IT infrastructures to communicate and exchange information efficiently [1]. This technology has significantly transformed the way software applications are designed and developed, enabling organizations to achieve greater productivity, interoperability, and efficiency.

Before the emergence of web services, applications were primarily developed as static and relatively self-contained systems. With the introduction of web service technology, organizations can develop reusable business functionalities that can be made available whenever required and can be integrated or composed with other existing services to provide new functionality. Another important advantage of web services is their ability to adapt to changing business requirements. As business conditions evolve or new constraints are introduced, web services can be modified and integrated effectively to accommodate these changes [2]. Consequently, web services continuously evolve in response to changing organizational requirements and operational environments.

Because web services often implement critical business functionalities, ensuring their quality, reliability, and maintainability is essential. Effective regression testing is one of the key approaches for achieving this objective. To improve the efficiency and productivity of web-service-based applications, it is necessary to address the challenges encountered by software testers during regression testing [3]. These challenges arise from several inherent characteristics of web services. Web services are distributed and heterogeneous in nature [4] and may operate across different hardware, software, and architectural platforms, making their testing, particularly regression testing, a complex task. Furthermore, the dynamic nature of web services allows services to be added, removed, modified, or composed with other services as requirements change [4]. Such continuous changes further increase the complexity and cost of regression testing. Therefore, there is a need for an efficient regression testing framework capable of addressing the challenges associated with the distributed, heterogeneous, and dynamic characteristics of web services.

In software engineering, software maintenance is generally regarded as a costly activity [5]. Maintenance is required whenever software needs to be modified, whether to correct defects, add or remove existing functionality, or adapt the system to new platforms, architectures, or operating environments. Regression testing is an important component of software maintenance and involves re-executing previously developed test cases to verify that software changes have not adversely affected existing functionality. However, regression testing can itself be expensive [6], as it consumes significant resources in terms of time, effort, and budget.

Consequently, reducing the effort and cost associated with regression testing is essential for minimizing overall software maintenance costs. One of the most effective approaches for achieving this objective is selective retesting, commonly referred to as regression test selection (RTS). Several regression testing strategies have been proposed in the literature; however, regression testing of web-service-based applications remains a comparatively less explored research area.

This paper presents an automated approach for regression testing of web-service-based applications. To demonstrate and evaluate the proposed approach, an online survey reporting system is used as a case study.

2. Literature Review

Regression testing involves the execution of previously developed test cases to ensure that modifications made to a software system have not introduced new defects or adversely affected existing functionality. Generally, two major approaches are used for regression testing: **retest-all** and **selective retest** strategies [7]. In the retest-all approach, the entire previously developed test suite is re-executed on the modified program. Although this approach is considered safe because it provides comprehensive coverage of the modifications introduced into the system, it is often impractical for large-scale software systems due to the considerable time and cost involved. In contrast, the selective retest approach involves identifying a subset of test cases from the existing test suite and executing them only on the relevant portions of the modified software. This strategy can significantly reduce the time and computational resources required for regression testing.

Rothermel and Harrold [7] identified two major challenges associated with selective regression test selection. The first challenge is determining an appropriate strategy for selecting the necessary test cases from the existing test suite. The second involves determining when and where additional test cases should be introduced to augment the existing test suite. In [8, 9], Rothermel and Harrold proposed a framework of metrics for evaluating the effectiveness of different regression test selection strategies. Over the years, several regression test selection techniques have been developed for different software paradigms. However, regression test selection for web services remains a relatively recent area of research.

Chaturvedi and Gupta [10] categorized the requirements of web service regression testing into three major areas: changes in WSDL, changes in source code, and selective retesting of web service operations. To address these requirements, the authors introduced three types of WSDL specifications: **Difference WSDL**, which represents changes in the WSDL; **Unit WSDL**, which addresses changes in the source code; and **Reduced WSDL**, which supports selective retesting.

Claudio et al. [11] proposed an efficient regression testing technique based on keyword analysis. In [12, 13], Ruth M. et al. presented a framework for automating regression test selection for web services through service monitoring. The authors also evaluated the proposed framework by comparing the cost of automated regression test selection with that of regression testing without test selection.

Masood et al. [14, 15] proposed a fully automated and safe regression testing strategy based on the comparison of original and modified WSDL specifications. Li et al. [16] introduced an approach for selecting test cases for regression testing of different versions of BPEL (Business Process Execution Language) composite services when modifications are made between versions. Ruth M. [17] further conducted empirical studies on privacy-preserving regression test selection techniques for web services.

Alsmadi [18] investigated the activities and challenges associated with regression testing of web services. According to the author, reducing the number of test cases is particularly important in web service testing, whereas such reduction may be less critical in traditional programming paradigms. The primary objective is to optimize test execution and make efficient use of available resources. To achieve this, two strategies were proposed. First, a pre-test execution component can be introduced to evaluate the generated test cases and subsequently reduce the number of test cases selected for execution. Second, historical service usage sessions can be analyzed to further minimize the effort required for test case selection.

Bhuyan et al. [19] addressed UML-based regression testing for service-oriented architecture (SOA) using UML use case and activity diagrams to generate test cases. Tarhini et al. [20] modeled a web application using a two-level abstract model based on Timed Labeled Transition Systems (TLTS). They also proposed a regression testing algorithm for web applications and demonstrated that the approach is safe.

Mohanty et al. [21] proposed a control-flow-graph (CFG)-based approach for implementing a safe regression test selection strategy for service-oriented programs in an end-to-end manner. A regression test selection technique is considered safe if it selects all test cases that are capable of revealing the effects of modifications introduced into the software. To validate their approach, the authors used a navigational subsystem consisting of three web services. Ruth et al. [22, 23] and Lin et al. [24] also proposed safe regression test selection techniques based on control-flow analysis.

Ruth et al. [22, 23] introduced a grey-box technique for regression test selection. A white-box approach may not always be feasible in web service environments because the source code of individual service components may not be accessible to the web service developer or tester. The proposed technique addresses this limitation by representing each process within a web service using a CFG at the developer side. The CFGs obtained from individual components are then integrated to construct a global CFG representing the complete system. Subsequently, potentially affected or "dangerous" edges are identified, and test cases that traverse these edges are selected for regression testing.

Lin et al. [24] proposed another technique based on code transformation. Their approach models the Java code executed on both the client and server sides as a single program component. The control-flow-based techniques presented in [22, 23, 24] have both advantages and limitations, which can be compared with traditional control-flow-based regression testing techniques developed for procedure-oriented programs.

Khan et al. [25] proposed a model-based regression testing strategy in which service interfaces are represented using visual contracts consisting of preconditions and postconditions expressed as graph transformation rules. By analyzing the dependencies among these rules, the potential impact of a modification can be estimated, thereby enabling the selection of appropriate test cases for regression testing.

3. Proposed Methodology

The This paper presents an effective and automated regression-testing approach for web-based applications. To demonstrate the proposed approach, an Online Survey Reporting System, as illustrated in Fig. 1, is used as a case study.

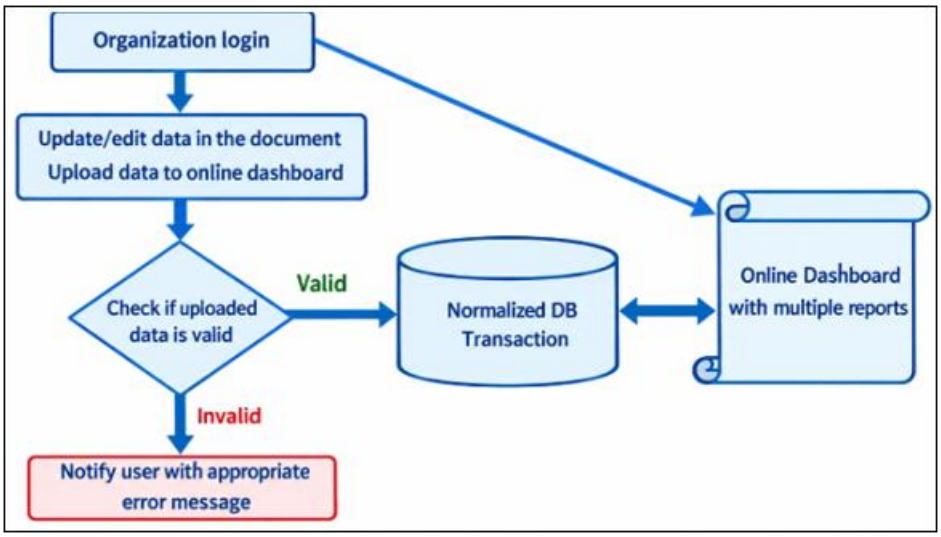

Figure 1: Online Survey Reporting System

The Online Survey Reporting System is designed to facilitate the analysis and reporting of surveys conducted for movie trailers. Previously, historical survey data and key findings were maintained in Microsoft Excel files. The proposed web-based system replicates the core functionality of these Excel-based reports while providing a more interactive and user-friendly graphical interface. The system incorporates GUI-enabled drop-down selections, multi-select filters, and data-comparison capabilities through charts and graphs, thereby enabling users to analyze survey results more efficiently.

The survey data is initially stored in SPSS or other database formats rather than directly in SQL databases. Therefore, the data is first migrated from the source database into an SQL-based Online Transaction Processing (OLTP) database. Subsequently, the data is transformed and processed into an Online Analytical Processing (OLAP) structure to support efficient analytical queries. Based on the filters selected by the user, the corresponding OLAP result set is retrieved and presented through the User Interface (UI) of the web-based application using web services. The resulting data can also be exported from the UI to Microsoft Excel for further analysis and reporting.

Whenever a new survey is introduced or existing survey-tracker data is updated, the complete data-processing pipeline must be validated. This includes testing the Extract, Transform, and Load (ETL) process, validating the OLTP-to-OLAP data transformation, and verifying the conversion of OLAP results into the web-based UI. Traditionally, these activities require the execution of a large number of test cases manually each time new data is added. Consequently, the regression-testing process is time-consuming, labor-intensive, and prone to human error.

To address these limitations, this paper proposes an automated regression-testing approach that significantly reduces the manual effort required to validate the web-based application. The proposed approach begins by comparing the frequency of responses for each question and its associated variables between the source database and the OLTP database. The validated OLTP data is then processed into OLAP form based on the required filtering criteria.

For automated result validation, database connectivity to the analytical services is established in Microsoft Excel using appropriate authentication credentials. A Pivot Service is configured in Excel to generate analytical results based on the same filter criteria used by the web application. A macro-based service is developed to transfer the selected filter values simultaneously to both the web application and the Pivot Service. In this way, identical input parameters are supplied to both systems, and their corresponding outputs can be compared objectively.

The proposed automation framework uses the selenium testing tool to execute the regression tests. The required filter inputs are maintained in an Excel worksheet and are simultaneously supplied to Selenium and the Pivot Service. Since both systems receive identical inputs, their outputs are expected to be equivalent. The macro service exports the data generated by the web application's UI into one Excel worksheet, while the corresponding Pivot results are exported into another worksheet.

Finally, the two Excel worksheets containing the UI-generated data and Pivot-generated data are compared cell by cell. If all corresponding values match, the test result is recorded as **"True"** by appending the status to the Excel sheet name. If any discrepancy is detected between the two result sets, the test result is recorded as **"False."** This automated comparison mechanism provides an efficient means of validating the application's data-processing pipeline and substantially reduces the time and manual effort required for regression testing.

Overall, the proposed approach establishes an automated, data-driven mechanism for regression testing of web-based

survey-reporting applications. By using identical inputs and independently generated outputs for comparison, the approach enables systematic validation of data consistency across the source database, OLTP layer, OLAP layer, and web-based user interface.

To validate our proposed approach, we implemented it using an online, web-based application that utilizes a web service for an online survey reporting system. The results obtained from the implementation are presented below. Figure 2 illustrates a snapshot of the pivot services developed in Excel.

4. Results

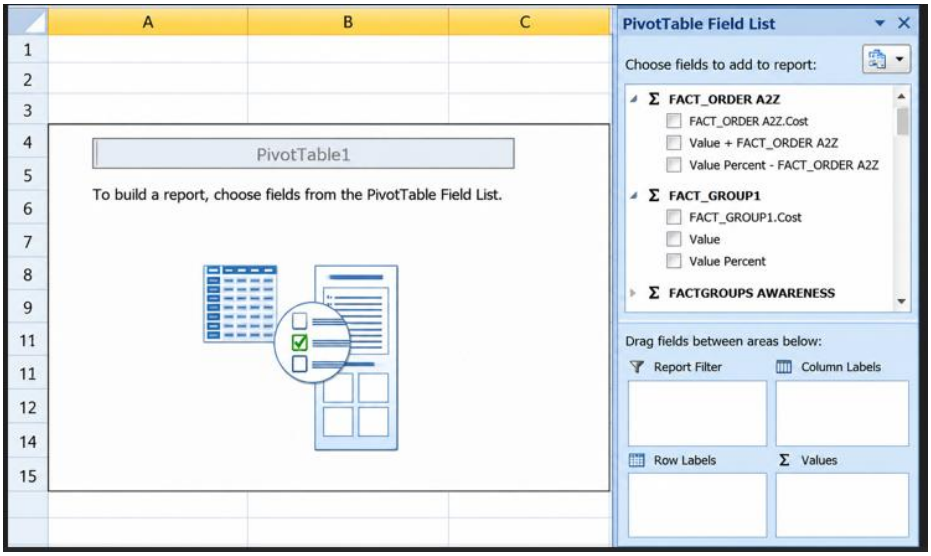

Figure 2: Pivot of OLAP data in excel

A Selenium or *custom* service is made which is used to pass the filters to application system as well to pivot as shown in Fig. 3.

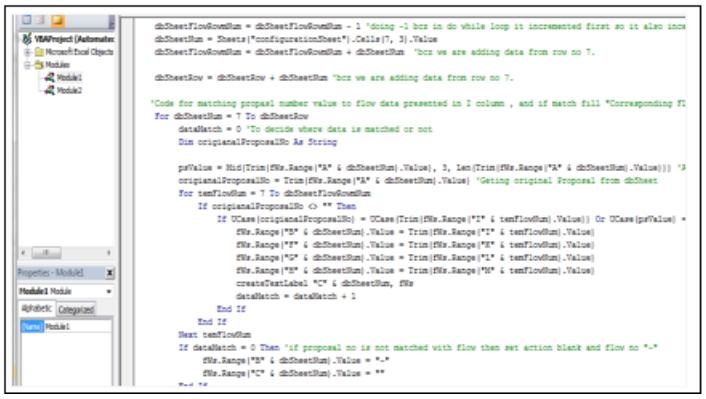

Figure 3: Macro Service or any customize application to pass the filters

To automate the functional and regression testing, we have used Selenium testing tool which is used to read the filter data kept in excel, pass the same filter data to web application and export the result set in excel. Fig. 4 shows the snapshot of Selenium tool.

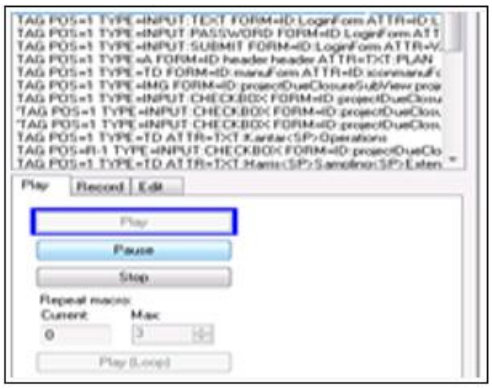

Figure 4: Selenium or any custom testing Tool with script

Macro service matches data from the application excel with pivot excel data and appends "True" or "False" on comparison sheet which is another sheet in excel. Fig. 5 shows the result after comparison.

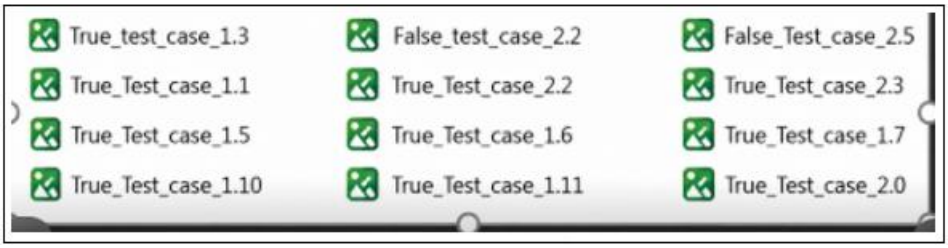

Figure 5: Result after comparison of application excel and pivot excel

Thus, with this approach, we attempted to reduce the regression testing of web- based application system containing web services as whenever a new job or new survey data is added to the application, we need not to test the full application manually. Instead, we automated the whole approach to save manual time and effort.

5. Conclusion

In this paper, we analyze the issues and challenges associated with regression testing of web services that form an integral part of web-based application systems. To address these challenges, we propose a novel approach for maintaining the reliability and quality of such systems. The proposed approach is evaluated using an online web-based application system, and the results of the implementation are presented and analyzed. The findings demonstrate that the approach can help ensure software quality while reducing the maintenance costs associated with continuously evolving web services. Furthermore, the proposed approach is platform-independent and enables fast and efficient data processing for testing purposes. However, a major limitation of the proposed approach is that it has been implemented only for web-based applications that use structured databases. Extending the approach to support unstructured databases remains an area for future research.

References

[1] http://www.webopedia.com/TERM/W/Web_Services
[2] Minder Chen, Andrew N. K. Chen, Benjamin B. M. Shao, "The Implications and Impacts of Web Services to Electronic Commerce Research and Practices", Journal of Electronic Commerce Research, VOL. 4, NO. 4, 2003
[3] Rajani Kanta Mohanty, Binod Kumar Pattanayak and Durga Prasad Mohapatra, "UML Based Web Service Regression Testing Using Test Cases: A Case Study", ARPN Journal of Engineering and Applied Sciences Vol. 7, No. 11, Nov. 2012 ISSN 1819-6608
[4] Youssef Bassil, "Distributed, Cross-Platform, and Regression Testing Architecture for Service-Oriented Architecture", Advances in Computer Science and its Applications (ACSA), ISSN: 2166-2924, Vol. 1, No. 1, March 2012
[5] Seacord R.C., Plakosh D. and Lewis G.A., "Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices", Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
[6] H. Leung and L. White, "Insights into regression testing", Proceedings of the Conference on Software Maintenance, pages 60–69, 1989.
[7] Gregg Rothermel and Mary Jean Harrold,"A safe, efficient regression test selection technique", ACM Transactions on Software Engineering Methodology (TOSEM), 6(2):173{210, 1997)
[8] G. Rothermel and M. Harrold, "Analyzing regression test selection techniques", IEEE Transactions on Software Engineering, 22(8):529–551, August 1996
[9] E. Engström, P. Runeson, and M. Skoglund,"A systematic review on regression test selection techniques", Information and Software Technology,52(1):14–30, January 2010.
[10] Chaturvedi, A. and Gupta, A., "A tool supported approach to perform efficient regression testing of web services", In Proceedings of IEEE 7th International Symposium on Maintenance and Evolution of Service Oriented and Cloud Based System,2013
[11] Cláudio Magalhaes, Flávia Barros, Alexandre Mota, Eliot Maia "Automatic Selection of Test Cases for Regression Testing", Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing, ACM 2016
[12] Michael E. Ruth, **"**Concurrency in a decentralized automatic regression test selection framework for web services**"**, Proceedings of the 15th ACM Mardi Gras conference: From lightweight mash-ups to lambda grids, ACM 2008
[13] Michael E. Ruth, Shengru Tu, "Empirical studies of a decentralized regression test selection framework for web services", Proceedings of the workshop on Testing, analysis, and verification of web services and applications, Pages 8-14, ACM,2008
[14] Masood T., Nadeem A., Ali, "An automated approach to regression testing of web services based on WSDL operation changes", In Proceedings of IEEE 9th International Conference on
[15] Masood T., Nadeem A., Lee, "A Safe Regression Testing Technique for Web Services Based on WSDL Specification", Software Engineering, Business Continuity, and Education Communications in Computer and Information Science Volume 257, 2011, pp 108- 119, Springer, 2011
[16] Bixin Li a, Dong Qiua, Hareton Leungb, Di Wanga, "Automatic test case selection for regression testing of composite service based on extensible BPEL flow graph", The Journal of Systems and Software, (2012) 1300– 1324 Science Direct © 2012 Elsevier Inc.
[17] Ruth M., "Empirical Studies of Privacy-Preserving Regression Test Selection Techniques for Web

Services", Proceedings of the IEEE International Conference on Software Testing, Verification, and Validation Workshops Pages 322-331,2014

[18] Izzat Alsmadi, Sascha Alda, "Test Cases Reduction and Selection Optimization in Testing Web Services", I.J. Information Engineering and Electronic Business, 2012, 5, 1-8

[19] Prachet Bhuyan, Abhishek Kumar, "Model Based Regression Testing Approach of Service Oriented Architecture (SOA) Based Application: A Case Study", International Journal of Computer Science and Informatics, ISSN (PRINT): 2231 –5292, Volume3, Issue2, 2013

[20] A. Tarhini, H. Fouchal, and N. Mansour, "Regression Testing Web Services-based Applications", ACS/IEEE Int. Conf. on Computer Systems and Applications, 2006, pp. 163-170.

[21] Rajani Kanta Mohanty, Binod Kumar Pattanayak and Durga Prasad Mohapatra, "A Regression Test Selection Technique for SOA Based Applications", International Journal of Software Engineering and Its Applications Vol.8, No.3 (2014), pp.65-72

[22] M. Ruth and S. Tu., "A safe regression test selection technique for web services", In Proceedings of the Second International Conference on Internet and Web Applications and Services, pages 47–. IEEE Computer Society, 2007.

[23] M. Ruth, S. Oh, A. Loup, B. Horton, O. Gallet, M. Mata, and S. Tu. "Towards automatic regression test selection for web services", In Proceedings of the 31st Annual International Computer Software and Applications Conference - Volume 02, pages 729–736. IEEE Computer Society, 2007.

[24] Tamim Ahmed Khan and Reiko Heckel, "On Model-Based Regression Testing of Web-Services Using Dependency Analysis of Visual Contracts", FASE 2011, LNCS 6603, pp. 341–355, 2011. Springer Verlag Berlin Heidelberg 2011