International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2015): 3.791

Techniques of Test Case Prioritization

Swati Puri[1], Sheena Singh[2]

[1]Student, Department of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab, India

[2]Assistant Professor, Department of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab, India

Abstract: ***The testing is a technique which will test the system software to detect faults. In today's time, various versions of the software are launched time to time which leads to the improvement in the software. The Regression testing is a type of testing which is applied to test the changes made in the software. In the regression testing, technique of test case prioritization is applied to detect maximum number of errors from the software. In this paper, BPEL technique of regression testing is reviewed which detects maximum number of the errors from the software.***

Keywords: Test case prioritization; models; Regression Testing; BPEL

1. Introduction

Software design and development radically changed in the last decade. Software systems were traditionally designed to operate in a completely known and immutable environment [1]. Whenever software had to be changed, to improve its quality or to meet new requirements, a maintenance lifecycle {design, development, and deployment} of a new Version of the system had to be planned. This approach led to costly maintenance Activities and an unsatisfactory time-to-market [1]. Several consolidated testing approaches, applied for years to traditional systems, apply to service-centric systems as well. Primarily, the idea that a combination of unit, integration, system, and regression testing is needed to gain confidence that a system will deliver the expected functionality [2].

Software maintenance is becoming important and expensive day by day. When the software is modified during maintenance phases, retesting is performed. This process of retesting the software is known as regression testing [3]. Regression testing, aimed at detecting potential faults caused by software changes, is the de facto approach. It reruns test cases from existing test suites to ensure that no previously working function has failed as a result of the modification. Although many researchers point out that frequent executions of regression test are crucial in successful application development, rerunning the regression test suite for large and complex systems may take days and even weeks, which is time-consuming [3].

Service-oriented computing (SOC) can bring unprecedented flexibility both in the way software is built and in the way it is structured. A web service is composed of activities" whose execution performs tasks of interest, and “Messages" that enable the service to participate in a more complex Web service [5]. A conversation protocol is a finite state automaton which specifies the desired set of conversations of a composite web service [5]. In service-oriented computing, a business process may invoke external web services, which may incur charges. To reduce costs, it is desirable to detect failures as soon as possible when executing the test suite. The use of effective regression testing techniques is, therefore, crucial. Thus, test case prioritization as shown in [fig1.] is important in regression testing. It schedules the test cases in a regression test suite with a view to maximizing certain objectives (such as revealing faults earlier), which help reduce the time and cost required to maintain service-oriented business applications. Existing regression testing techniques for such applications focus on testing individual services or workflow programs. Surprisingly, to the best of our knowledge, the integration complexity raised by non-imperative artifacts such as XPath and WSDL among workflow steps has been inadequately addressed in regression testing research. Proper test case prioritization techniques increase the fault detection rate of a test suite and the chance of executing test cases with higher rates of fault detection earlier [6].

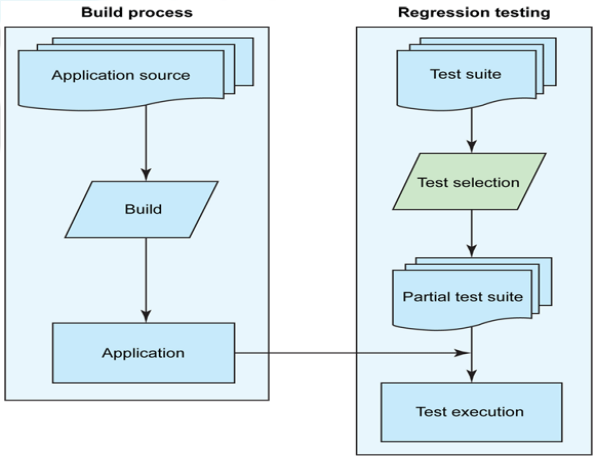

Figure 1: Test case prioritization in regression testing

2. Related Work

In this paper, Epifani, I et al. addresses the problem of identifying changes concerning the non-functional behavior of software services managed by external organizations, and therefore considered as black-box artifacts. The author defines the concept of change-point and provides a statistical technique aimed at identifying and gives an

execution trace produced by client invocations. Author also implemented a tool supporting change-point analysis as part of the KAMI framework {a toolset} and used to validate the method using simulations. As a result, the relation between the length of the trace and the range of values of probabilities appearing in the models, and the relation between the length of the trace and the distance between different change-points (in a multiple change-point setting) is executed [1].

Many prioritization techniques order test cases according to their particular coverage of program statements. On the other hand, industrial service-oriented business applications are typically written in orchestration languages such as WS-BPEL and integrated with workflow steps and web services using XPath and WSDL. Faults in artifacts may cause the application to extract wrong data from messages, leading to failures in service compositions. L. Mei et al. proposes a multilevel coverage model to capture the business process XPath, and WSDL from the perspective of regression testing and develop a family of test case prioritization techniques atop the model. Empirical results show that the techniques can achieve significantly higher rates of fault detection than existing techniques [2].

Regression testing is a very costly process performed mainly as a software maintenance activity. A regression test selection technique selects an appropriate number of test cases from a test suite that might expose a fault in the modified program. In this paper, Ruchika Malhotra et al. propose both a regression test selection and prioritization technique. This technique identifies and locates errors, help in preserving the quality and reliability of the software in the modified program. The results show that the technique may significantly reduce the number of test cases and thus the cost and resources for performing regression testing on modified software [3].

Testing of Service Oriented Architectures (SOA) plays a critical role in ensuring a successful deployment in any enterprise and cover functional and non-functional aspects. SOA is a combination of features like run-time discovery and ultra-late binding entail that the actual configuration of a system is known only during the execution, and makes the existing testing techniques inadequate. Canfora, G et al. has survey the recent research achievements related to SOA testing and explores ways to improve the testability of SOA. Challenges are analyzed from the viewpoints of different stakeholders and solutions are presented at different levels of testing include unit, integration, and regression testing [4].

Fu et al. presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. Author model the interactions of composite web services as conversations, the global sequence of messages exchanged by the web services and present a framework where BPEL specifications of web services are translated to an intermediate representation. As the target verification language author uses Promela, input language of the model checker SPIN, which is a finite-state verification tool. Author proposes the concept of synchronizability to address this problem. Based on synchronizability results, a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN [5].

M. E. Ruth et al. reports a safe regression test selection (RTS) approach that is designed for verifying Web services in an end-to-end manner. The Safe RTS technique has been integrated into a systematic method that monitors distributed code modifications and automates the RTS and RT processes [6].

Web Services and Service-Oriented Architecture (SOA), quality assurance of SOA applications, such as testing, has become a research focus. Programs implemented by the Business Process Execution Language for Web Services (WS-BPEL), which can be used to compose partner Web Services into composite Web Services, are one popular kind of SOA applications. In this paper, Yitao Ni et al. presents a novel methodology to generate effective message sequences for testing WS-BPEL programs. To capture the order relationship in a message sequence and WS-BPEL's routing mechanism models the WS-BPEL program under test as a message-sequence graph (MSG). The results show that the message sequences generated by using our method can effectively expose faults in the WS-BPEL programs [7].

The rate of fault detection can provide earlier feedback to system developers, improving fault fixing activity and ultimately software delivery. In this paper, Shifa-e-Zehra Haidry et al. presents a family of test case prioritization techniques that use the dependency information from a test suite to prioritize that test suite. The hypothesis is that dependencies between tests are representative of interactions in the system under test, and executing complex interactions earlier is likely to increase the fault detection rate. Empirical evaluations on six systems built towards industry use demonstrate that these techniques increase the rate of fault detection compared to the rates achieved by the untreated order, random orders, and test suites ordered using existing “coarse-grained” techniques based on function coverage [8].

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the file “MSW_USltr_format”.

A. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

3. Modification Impact Analysis based Test Case Prioritization for Regression Testing of Service-Oriented Workflow Applications

Test case prioritization for regression testing is an approach that schedules test cases in a specific order to detect faults early, which is well known as an effective technology to ensure the quality of modified service oriented workflow applications. Service-oriented workflow applications orchestrate web services to provide value-added service and typically are long-running and time consuming processes. Therefore, these applications need more precise prioritization to execute earlier those test cases that may detect failures.

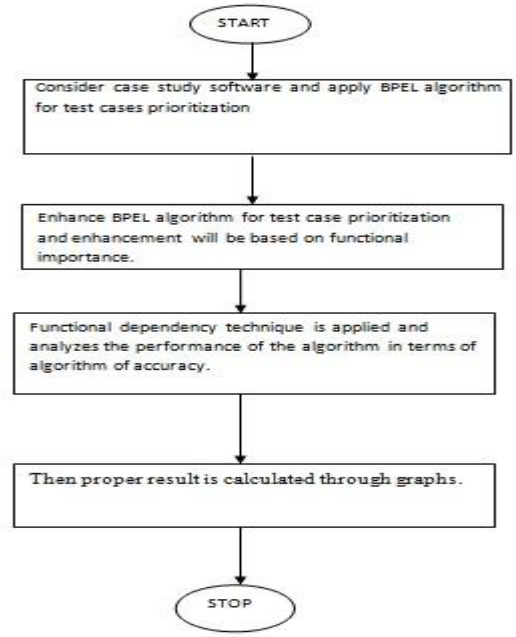

Figure 2: Flow Chart

This paper considers the internal structure and fault propagation behavior of activity in service-oriented workflow applications. The faults of software are usually caused by the propagation of defects with the internal structure of software and thus the design of software internal structure has an important influence on software quality. Therefore, study on the test case prioritization of service-oriented workflow applications should take not only coverage information but also the internal structure information into consideration. Surprisingly, to the best of our knowledge, the internal structure information of service-oriented workflow applications has been inadequately considered in existing regression test prioritization research. The internal structure of service-oriented workflow applications can be seen as the interaction of activities to realizing the expected target. The interaction of activities includes performing process logic, exchanging message, invoking external web services etc. The modification or fault in one activity will definitely propagate to other activities that directly or indirectly dependent on it. All kinds of BPEL dependencies between activities should be analyzed.

Our prioritization approach schedules test cases based on dependence analysis of internal activities in service oriented workflow applications.

For test case prioritization BPEL algorithm had been used to increase efficiency of test cases prioritization. The enhancement will be based on function importance and number of functions associated. The crossover value will be taken on function importance.

4. Conclusion

In this work, it is been concluded that regression testing is the type of testing which is applied to detect errors from the software which are raised due to changes made in the software. The various test case prioritization techniques has been reviewed and it is been analyzed that BPEL algorithm is the most efficient algorithm for prioritization. In future improvement will be proposed in BPEL algorithm to improve error detection rate in regression testing.

References

[1] Epifani, I., Ghezzi, C., & Tamburrelli, G. (2010, November). “Change-point detection for black-box services”. In Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering (pp. 227-236). ACM.
[2] L. Mei, Z. Zhang, W. K. Chan, and T. H. Tse. “Test case prioritization for regression testing of service-oriented business applications”. In Proceedings of the 18th International Conference on World Wide Web (WWW 2009), pages 901–910. ACM, New York, NY, 2009.
[3] Ruchika Malhotra, Arvinder Kaur and Yogesh Singh. “A Regression Test Selection and Prioritization Technique”. In Proceedings of the Journal of Information Processing Systems, Vol.6, No.2, June 2010, doi:10.3745/JIPS.2010.6.2.235.
[4] Canfora, G., & Di Penta, M. (2009). “Service-oriented architectures testing: A survey”. In Software Engineering (pp. 78-105). Springer Berlin Heidelberg.
[5] Fu, X., Bultan, T., & Su, J. (2004, May). “Analysis of interacting BPEL web services”. In Proceedings of the 13th international conference on World Wide Web (pp. 621-630). ACM.
[6] M. E. Ruth and S. Tu. “Towards automating regression test selection for Web services”. In Proceedings of the 16th International Conference on World Wide Web (WWW 2007), pages 1265–1266. 2007.
[7] Yitao Ni, Shan-Shan Hou, Lu Zhang, Jun Zhu, Zhong Jie Li, Qian Lan, Hong Mei, Jia-Su Sun, "Effective Message-Sequence Generation for Testing BPEL Programs", IEEE Transactions on Services Computing, vol.6, no. 1, pp. 7-19, First Quarter 2013, doi:10.1109/TSC.2011.22.
[8] Shifa-e-Zehra Haidry, Tim Miller, "Using Dependency Structures for Prioritization of Functional Test Suites", IEEE Transactions on Software Engineering, vol.39, no. 2, pp. 258-275, Feb. 2013, doi:10.1109/TSE.2012.26