An Observed Model Approach for Testing GUI Model Driven Interface in Water-Flow Model

Charusheela Vasantrao Torawane
M.Tech Student
Department of Computer Science and Engineering
Manakula Vinayagar Institute of Technology,
Pondicherry University, Pondicherry, India

K. Poonkavithai, Ph.D.
Assistant Professor
Department of Computer Science and Engineering
Manakula Vinayagar Institute of Technology,
Pondicherry University, Pondicherry, India

ABSTRACT
Graphical-user interface (GUI) -based software applications are with the job of verifying that these tasks can be performed using the software; and that the software does not "behave badly". A set of use cases with high-level descriptions is also given to the testers. The tester executes these high-level steps by using GUI widgets on which events can be performed. GUI testers are both automated and manual working with undetermined input spaces. The testers unknowingly miss the event sequences navigated by the GUI, and fail to realize its implementation which may allow the execution of some disallowed sequences. This thesis proposes Preemptive Regression Testing (PRT) an adaptive taxing move to address this challenge. Whenever a change in the exposure of any service artifact is detected, PRT recursively preempts the current setting of regression test and creates a sub-session of the current test session to assure lately identified changes in coverage by adjusting the precedence of the test cases in the test suite. Then, the execution will resume the sub session from the balanced location of the test cases. PRT terminates only when each test case in the test suite has been executed at least once without any preemption activated in between any test case executions. The new result confirms that testing water flow method based web service in the phase of such changes is very difficult and one of the PRT-enriched techniques will overcome the challenge.

Keywords
PRT Technique, APFD ,BPEL,WSDL,XML.

1. INTRODUCTION
A workflow-based service communicates with other web services to implement the required functionality. Web service uses a standardized XML messaging system where XML is used to encode all communications to a web service. All the web service communication is in XML, they are not tied to any one operating system or programming languages. Web services are XML-based information exchange systems which include messages, documents programs or objects and use the Internet for direct application-to-application interaction. The web service is a collection of standards is used for exchanging data between systems or applications. In software applications various programming languages are written which can use web services to exchange data over computer networks like the Internet. A web service paradigm is a programmable Web application that is accessible through standard Internet protocols [1]. Software services are exposing its means of the Internet and make them accessible via standard programmatic interfaces.

Web services also offer a promising way to facilitate Business-to-Business (B2B) collaboration. For distributed computing and resource sharing over the Internet Web service technology provides a uniform framework to increase cross-language and cross-platform interoperability. Web service paradigm opens a new cost-effective way to quickly develop and deploy Web applications by dynamically integrating other published Web services to conduct new business transactions. Workflow-based service should be fully tested before its deployment, otherwise the changes made in the system to enforce the external service of a workflow-based service remains unchanged during a test session. Thus, testing will be re-conducted. Regression testing [40] is carried out in two ways: (a) it guards against regression faults. (b) it verifies whether a web service working with external services behaves as expected or not even though it has not been modified since the last test session. The majority of existing regression testing research for web services only considers the regression faults and Verification of regression faults is still inadequately explored. Web services can be assessed by the set of classic software "elites," such as security, usability, adaptability, maintainability, availability, reliability, scalability, efficiency.

All the components and relationships of the software systems are pre-decided before the software runs. So that, each component can be thoroughly tested and fully examined before the system starts to execute. By providing a more flexible approach Web service extend this paradigm to dynamically locate and assemble distributed Web services in an Internet-scale setting. Whenever a system requires a web service component, the system first search a public registry where Web service providers publish their services, choose the optimal Web service fulfills the requirements, binds and invokes the Web service. Techniques have been developed to generate test cases from syntax definitions of WS in WSDL, business process and behavioral models in BPEL, ontology based descriptions and other formal models of WS such as finite state machines and labeled transition systems, grammar graphs. Various WS specific issues have been addressed using these techniques, such as the robustness in dealing with invalid inputs and errors of the fault tolerance of other services and security of the environment that is vulnerable to malicious attacks. When changes are made to the existing software regression testing is performed.

The main purpose of regression testing is to provide confidence that the newly introduced changes do not obstruct the behaviors of the unchanged and existing parts of the software. This complex procedure is more challenging because of some of the recent trends in software development paradigms. There are three major branches of the regression testing, which include test case selection, test suite minimization, and test case prioritization. Test suite minimization is a process which first identifies and then eliminates the redundant test cases from the test suite. Test case selection is concerned with the dealing problem of selecting a subset of the test cases which is used to test the

changed parts of the software. Test case priority identifies the fault detection order of the test cases which maximizes the desired properties.

2. LITERATURE SURVEY

M. Harman et al.[18] has studied Regression testing which describes the work that is closely related to our study, in the context of code coverage, external services, execution monitoring, test oracles, revision identification, test case prioritization, code coverage, external services, stopping criteria, execution monitoring, test oracles, cloud-based service testing, and service environment evolution. The use of a stopping criterion is the main feature of PRT strategies . W.K. Chan et al.[13] studied new dataflow-based test adequacy criteria to test WS-BPEL web services. Grochtmann et al. [7],[8] proposed a classification-tree method to help software testers to construct test suites. Also defined classifications of the criteria for partitioning the input domain of the program.

Specification, classifications and classes known as classification tree are arranged in the form of a hierarchical structure from which a test suite is generated .PRT strategies does not check whether the service under test undergoes any revision or not. If the details of service revision are available, further it will refine a criterion by only picking test cases that affects the changed part of the service under test . R.Casado et al.[2] focuses on the testing aspect of WS transactions which composes distributed and autonomous services and it ensures that their execution is consistent and correct. The proposed criteria have the potential to capture the behaviour of WS transactions and to analyze the possible (failure) situations which affects the execution of such transactions. Li et al.[10],M.E. Ruth et al. [16] and Tarhini et al. [17] conducted an analysis on web service implementations to identify revised frag-ments of code in a service to compare the flow graph of the new version with that of the previous version. Li et al. [10] explored the use of messages in selecting paths for regression testing to obtain a comprehensive view of the service behavior. Z.Chen et. al[1] proposed a Novel Hybrid Coverage for dynamic web applications which counts both the code executed and the HTML elements to represent both client-side and server **s**ide conditions. Two web applications Statement coverage and element coverage used to compare hybrid coverage with coverage criteria. The data which is collected is used to identify the accessed HTML elements in the web User Interface model and compute the element coverage. Liu et al. [11] studied the changes in the concurrency control in BPEL process executions. Tarhini et al. [17] exploited impact analysis for Quite a number of test case prioritization techniques have been proposed from a model-based perspective. Hou et al. [9] observed the need to test service-oriented applications with external services. Their techniques were used to constrain the total number of requests for specific web services. Mei et al. considered both Grey-box coverage[12][13] and black-box coverage [14] in test case prioritization. They have not considered the need for dynamic changes in test case ordering , the feedback collected from the service under regression test. Nguyen et al. [15] integrated test case prioritization with audit testing to control resource consumption. B.Jiang et al. [19] used the dynamic features of service selection to reduce the service invocation cost and they have also studied [20] different diversity strategies to reorder the test cases. Therefore, the evolution of external web service was not considered in the thesis above. This thesis uses dynamic coverage data of the BPEL process achieved by test cases against both the original and the evolved web services to determine adequacy. The closest related work in this aspect is Z.Chen et al. [1] conducted a study on integration of coverage of program statements and HTML elements for testing dynamic web applications. Their work has not studied whether the test cases applied to the evolved version of the web service produces compatible results as the original service or not. Becker et al. [6] contucted a study to describe a technique to check whether a service description is backward-compatible. P. Krause et al.[4] describes BPEL (Business Process Execution Language) a semi-formal flow language with complex features such as concurrency and hierarchy. For complex modelling languages test cases designed are time-consuming, error prone and tedious**.** J. Luo [5] has introduced a novel test method for XML based communication.XML-based applications can receive messages from arbitrary applications until the protocol is defined by the schema. J. Tuya et al.[3]studied a novel abstract model for dynamically modeling distinct web services transaction standards and test their reliability in terms of failures. Web services provide a distributed computing environment where in service providers and consumers can dynamically interact and cooperate on various tasks in different domains such as business, education, government and healthcare.

3. STRATEGIES IN PREEMPTIVE REGRESSION TESTING

In the Preemptive Regression approach there are three testing strategies that is fix, reschedule and fix and reschedules which is a hybrid approach.

Fix (Strategy 1) Let us consider a test case t misses at least one coverage item which has been covered in its last execution. We consider that the *F* be the number of missed items of *t.* A sequence U is selected by the strategy in test cases T such that all the test cases in U of last execution can cover all the missed coverage items in F. The coverage is achieved by many un-executed test cases in T in following last executions may cover some items in f, to construct U and run these test cases strategy 1 adopts the following criteria. Strategy 1 executes and chooses one test case among the unexecuted test cases in T in a round robin fashion (from position of t in T) in descending order of the number of items covered by each of the test cases for every missed coverage item in F.

An additional coverage item that has been missed may be discovered by executing such replacement test case. In such cases strategy 1 prevents its current session and starts a new session. The new session will adjust the priorized test cases, in which the execution is resumed by pre-emption point and will be removed from f of current session of those coverage items which has already been covered by recursively invoked sessions of strategy 1.

Reschedule (strategy 2) When a new item(s) is covered by a test case which is not been covered in its last execution the additional coverage items achieved by the test case is recorded by strategy and reprioritizes the un-executed test cases according to the coverage technique of additional items.

Fix-and-Reschedule (strategy 3) It is a hybrid strategy of 1 and 2, where a test case does not cover an item(s) is been covered in their last execution. The strategy 1 is invoked first and even after completing strategy 1 there are any additional coverage items which is not covered in the last execution then strategy 2 is been invoked.

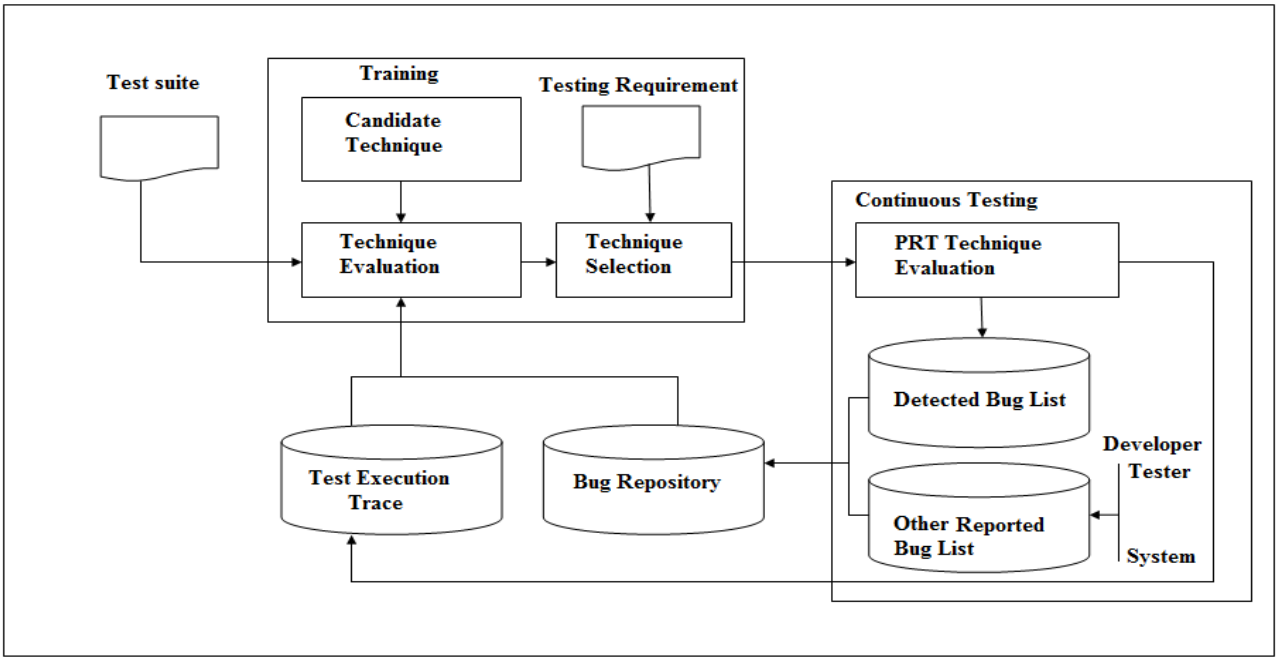


Fig.1 Workflow of PRT Techniques

4. EXPERIMENTAL EVALUATION

The PRT technique system can be used only by the administrator. Thus any user with administrator status could use the tool. User authentication is done whenever the user is trying to connect to a remote system Users are supposed to provide the user name and password. This avoids any unauthorized user to access the remote machines. At the same time when accessing the server, the IP Address along with proper authentication is provided to make sure only authorized users accessing the tool.

4.1 Input and Output design

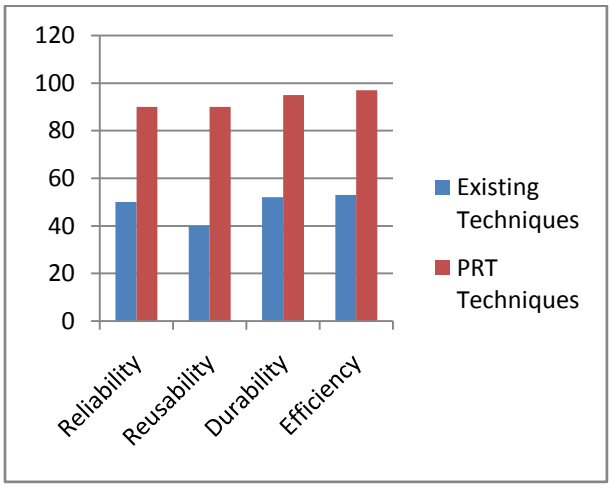
The input design is the link between the information system and the user. The developing specification and procedures for data preparation are compromised and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The input design focuses on controlling the amount of input required, and the errors, which avoid delay, and extra steps keeping the process simple. Provides security and ease of use with retaining the privacy which design the required input.

A quality output is one, which meets the requirements of the end user and presents the information clearly. Processing results of the system are communicated to the users and to other system through outputs. How the information is to be displaced for immediate need is determined in the output design of the system. This information is most important and direct source to the user. Intelligent and Efficient output design improves the system's relationship to help the user for decision-making.

Table 1. Performance Evaluation of PRT Technique

Quality Of Services	Existing Techniques	PRT Techniques
Reliability	Low	High
Reusability	Very Low	High
Durability	Low	Very High
Efficiency	Low	Very High

The output form of an information system should accomplish one or more objectives which convey information about current status, past activities, or the future projections. The design is achieved by creating user-friendly screens which handle the large volume of data for the data entry. The goal makes the data entry easier and free from errors with the designed input. To perform all the data manipulates the data entry screen is designed which also provides record viewing facilities. As the data entered it will check the validity. Data can be entered with the help of screens. When needed by the user appropriate messages are provided at instant. Create an input layout which is easy to follow is the objective of input design. The graphical representation of PRT Technique is given.



Graph 1. Representation of PRT Technique

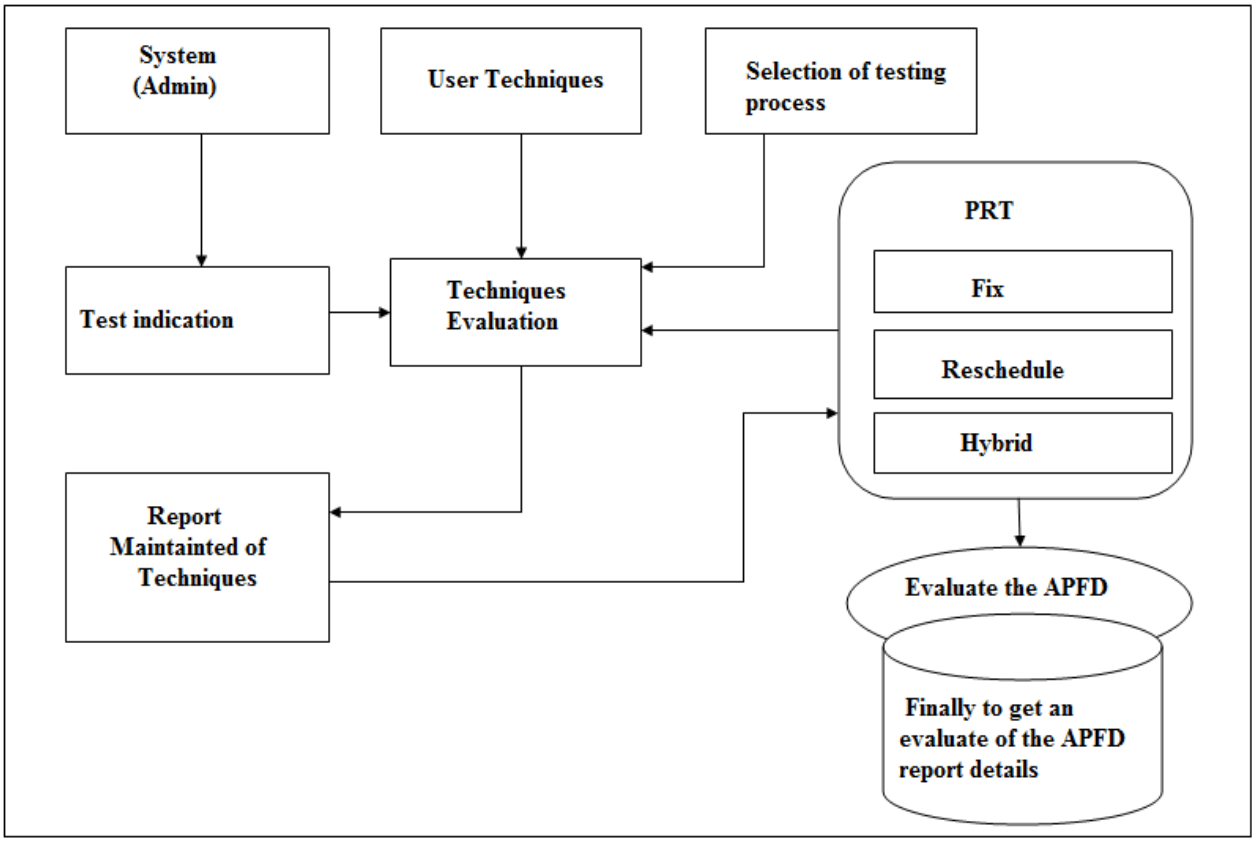


Fig.2 Architecture diagram of proposed work

In the architecture diagram figure 2 above, the overall working of the proposed work is explained. When the system admin starts system test indications, the user predefined techniques are evaluated using technique evaluations. The report of test indication is also based on evaluation techniques. Technique evaluation maintains the report of test techniques as well as the selection testing process. Based on the report maintained by the technique evaluation our proposed PRT technique is maintained using APFD (Average percentage of fault detection) finally, evaluation report from APFD is obtained.

5. CONCLUSION

As software systems have grown increasingly complex, our testers are tasked with verifying that these systems function correctly; but the testers do not fully understand these systems' input spaces. This problem is severely compounded in GUIs that have immense, even infinite, input spaces. GUI testers routinely miss allowable event sequences, any of which may cause failures once the software is fielded. And the tester may fail to discover that the software's implementation allows the execution of some disallowed sequences. We will enhance the PRT technique for security purpose and integrate it Quality of Service(QoS) testing.

6. REFERENCES

[1] Z. Chen Y. Zou, C. Feng,, X. Zhang, and Z. Zhao, "A hybrid coverage criterion for dynamic web testing," http://software.nju.edu. cn/zychen/paper/2013SEKEa.pdf.

[2] R. Casado, M. Younas, and J. Tuya, "Multi-dimensional criteria for testing web services transactions," Journal of Computer and System Sciences, vol. 79, no. 7, pp. 1057–1076, 2013.

[3] R.Casado, M. Younas, and J. Tuya,"Testing the reliability of web services transactions in cooperative applications," Information and Software Technology ,2012, pp. 743-748.

[4] Y. Zheng, J. Zhou, and P. Krause, "An automatic test case generation framework for web services," Journal of Software, vol. 2, no. 3, pp. 64–77, 2007.

[5] W. Xu, J. Offutt, and J. Luo, "Testing web services by XML perturbation," Proceedings of the 16th International Symposium on Software Reliability Engineering (ISSRE '05), pp. 257–266, 2005.

[6] K. Becker, J. Pruyne, S. Singhal, A. Lopes, and D. Milojicic, "Automatic determination of compatibility in evolving services," International Journal of Web Services Research, vol. 8, no. 1, pp. 21–40, 2011 SSS.

[7] Grochtmann, M., And Grimm, "Classification trees for partition testing", Softw. Testing, Verification and Reliability, 1993, 3, (2), pp. 63–82

[8] Grochtmann, M., Wegener, J., And Grimm, "Test case design using classification trees and the classification-tree editor CTE'", Proceedings of the 8th International Software Quality Week, QW '95.

[9] S.-S. Hou, L. Zhang, T. Xie, and J.-S. Sun, "Quota-constrained test-case prioritization for regression testing of service-centric systems," Proceedings of the IEEE International Conference on Software Maintenance (ICSM '08), pp. 257–266, 2008.

[10] B. Li, D. Qiu, H. Leung, and D. Wang, "Automatic test case selection for regression testing of composite service based on extensible BPEL flow graph," Journal of

Systems and Software, vol. 85, no. 6, pp. 1300– 1324, 2012.

[11] H. Liu, Z. Li, J. Zhu, and H. Tan, "Business process regression testing," Proceedings of the 5th International Conference on Service-Oriented Computing (ICSOC '07), pp. 157–168, 2007.

[12] L. Mei, Y. Cai, C. Jia, B. Jiang, and W.K. Chan, "Test pair selection for test case prioritization in regression testing for WS-BPEL programs," International Journal of Web Services Research, vol. 10, no. 1, pp. 73–102, 2013.

[13] L. Mei, W.K. Chan, and T.H. Tse, "Data flow testing of service-oriented workflow applications," Proceedings of the 30th Interna-tional Conference on Software Engineering (ICSE '08), pp. 371–380, 2008.

[14] L. Mei, W.K. Chan, T.H. Tse, and R.G. Merkel, "XML-manipulating test case prioritization for XML-manipulating services," Journal of Systems and Software, vol. 84, no. 4, pp. 603–619, 2011.

[15] C.D. Nguyen, A. Marchetto, and P. Tonella, "Test case prioritization for audit testing of evolving web services using information retrieval techniques," Proceedings of the 2011 IEEE International Conference on Web Services (ICWS '11), pp. 636–643, 2011.

[16] M.E. Ruth and S. Tu, "Towards automating regression test selection for web services," Proceedings of the 16th International Conference on World Wide Web (WWW '07), pp. 1265–1266, 2007.

[17] A. Tarhini, H. Fouchal, and N. Mansour, "Regression testing web services-based applications," Proceedings of the IEEE International Conference on Computer Systems and Applications (AICCSA '06), pp. 163–170, 2006.

[18] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," Software Testing, Verifica-tion and Reliability,

[19] vol. 22, no. 2, pp. 67–120, 2012.

[20] K. Zhai, B. Jiang, and W.K. Chan, "Prioritizing test cases for regression testing of location-based services: metrics, tech-niques and case study," IEEE Transactions on Services Compu-ting, vol. 7, no. 1, pp. 54–67, 2014.

[21] K. Zhai, B. Jiang, W.K. Chan, and T.H. Tse, "Taking advantage of service selection: a study on the testing of location-based web services through test case prioritization," Proceedings of the IEEE International Conference on Web Services (ICWS '10), pp. 211–218, 2010.