Regression Test Selection Framework for Web Services

Divya Rohatgi, Dr. Tulika Pandey

Abstract—Web service is a class of software based on open standards to provide business functionality on the web. Since business functionalities are prone to change frequently so these inherent services are also undergoing frequent changes. Due to these changes, maintenance of this class of software become tedious and costly affair. Maintenance activity is assumed to be the most expensive activity in software development and is performed throughout the life cycle of the software. Regression testing is a part of maintenance to rerun entire test suite every time whenever a change is made to the software. Regression testing is challenging and time consuming activity in web services because they are inherently distributed, heterogeneous and dynamic in nature. Thus in order to reduce software maintenance and ensure proper quality, regression testing has to be minimized. In this paper we have given an efficient approach by which we can effectively carry out regression testing of a web based application system whenever any change is made to the system.

Index Terms—Automated testing, Regression Testing, Regression test selection, Software Engineering, Software Maintenance, Web Services, Web applications.

1 INTRODUCTION

The web service is a software technology which is based on Service Oriented Architecture and is used in various domains of applications. They use XML for tagging of information, WSDL for describing the service behavior, SOAP for communication between various parties and UDDI to enumerate services present for use. The main advantage of using web service is that despite of using different IT infrastructures, companies can communicate between them easily and efficiently with a web service [1]. They have revolutionized the approach in which applications are developed and with the help of which applications can have better productivity and efficiency across organizations. Prior to web services only static applications were built but now with this technology, organizations can develop business functionality which in the time of need can be made available and can also be combined with other available services in order to give a new functionality. Web Services have an additional advantage that depending upon the changing business conditions or addition of new business constraints we can easily and effectively adapt to changed scenarios [2]. Thus according to changing business needs and conditions these services are also constantly evolving. So it is necessary to ensure quality and proper maintenance of web services as they represent essential business functionalities. This can be achieved through effective regression testing. In order to have better efficiency and productivity of business applications, it is required to remove the obstacles which are faced by software testers in the process of regression testing of such applications [3].

These problems are due to several reasons. Web services are distributed and heterogeneous in nature [4] operating upon different architectural platforms both in terms of hardware and software due to which testing particularly regression testing becomes difficult. Also due to dynamic nature of these services [4] we can always add, remove or collaborate with other services. This also makes regression testing a difficult and challenging process. Thus there is a need to have an efficient regression testing framework which can overcome the challenges posed by the inherent nature of web services. In software engineering, Software Maintenance is considered as an expensive activity [5] and is performed every time we need to change the software either by fixing bugs or by adding or deleting preexisting functions or adapting the software to new platforms or architectures. Regression testing is a part of maintenance which aims at running the old test cases to ensure correctness of the program after changes applied. It is considered as a costly activity [6] as it encompasses considerable fraction of resources in terms of budget, schedule and effort. Thus in order to cut short maintenance cost it is a need to reduce regression testing. One of the best approaches to reduce regression testing effort is selective retesting or regression test selection. There are different regression testing strategies proposed in literature but in case of web service it is a relatively less researched area.

This paper presents an automated approach for regression testing of web services based applications and to illustrate the approach we have used online survey reporting system.

2 LITERATURE REVIEW

Regression testing is considered with running of all old test cases to ensure that new changes made to software had not introduced any new errors in software. There are two approaches or strategies to perform regression testing: retest all and selective retest strategy [7]. In retest all approach, we have to rerun entire earlier developed test suite on the modified program. This approach is safe as with this approach we can check entire modifications in the changed component. However it is not advisable to follow this practice for large software as it incurs large amount of time and budget. Rothermel and Harrold [7] proposed two challenges in the
selective retest techniques. First challenge is concerned with
the strategy to select required test cases from previous test
suite and secondly how to find when and where more test
cases are needed to augment the existing one. In [8, 9],
Rothermel and Harrold have given a set of metrics framework
to measure the effectiveness of different regression test
selection strategies. In literature for different types of software
paradigms there are different regression test selection
strategies proposed. Regression test selection for web services
is a relatively new research area. In [10] Chaturvedi and Gupta
have captured the web service regression testing needs into
three different categories, namely, changes in WSDL, changes
in code, and selective re-testing of web service operations and
for this authors have presented three types of WSDL:
Difference WSDL to incorporate changes in WSDL, Unit
WSDL for changes in code and Reduced WSDL for selective
retest. In [11] Lijun Mei et al. have exploited structural
similarity of XML-based artifacts based on which they have
provided a set of test case prioritization strategies by which
pair of test cases can be selected. In [12, 13] Ruth M. et al. have
presented a framework which automates web service
regression test selection by monitoring of service. Also authors
have evaluated the framework for comparing the cost of using
automated approach of regression test selection with the cost
al. have proposed regression testing strategy which is fully
automated and is safe based on original and changed WSDL.
Li. et al. [16] have presented an approach which choses test
cases for regression testing of different versions of BPEL
(business process execution language) composite service
where these changes are involved. In [17] Ruth M. has
performed empirical studies of Privacy-Preserving Regression
Test Selection Techniques in case of Web Services. In [18] Izzat
Alsmadi has evaluated the activities and challenges which
arises in regression testing of web services. According to the
author, it is very important to reduce test cases in case of web
service whereas there is not so much urgency for reducing test
suite in traditional programming paradigms. Most important
is to optimize test execution in comparison to other processes
as resources can be made available to them. For this, two
strategies were proposed. First and foremost requirement is to
produce a pretest execution component by which one can
evaluate generated test cases and then minimize test case
selection based on generated test cases. Secondly with the help
of historical usage sessions we can minimize the process of
test case selection. P. Bhuyan et al. [19] have addressed the
UML based regression testing method using UML use case
diagram and UML activity diagram to generate a test case of
SOA. In [20] Tarhini et al. modeled a web application as two
level. abstract model provided through a Timed Labeled
Transition Systems TLTS. They have also proposed an
algorithm for regression testing of web applications which is
proved to be safe. In [21] Mohanty et al. have proposed a
control flow graph based approach with the help of which we
can easily use safe regression test selection strategy to
programs based on service oriented architecture in end to end
manner. A given approach of test selection is said to be safe if
it selects all test cases which are capable of revealing
modifications performed. To verify the strategy they have
used a navigational subsystem comprising of three web
services. In [22,23] Ruth et al. and in [24] Lin et al. have
proposed safe regression test selection strategies which
utilizes analysis of control flow models. Ruth et al. [22, 23]
have presented a grey box technique for regression test
selection as it is sometimes problematic to follow white box
approach as sometimes source code for components are not
provided to the developer of the web service. The approach is
based on the concept that each process in a web service is
given by a CFG at the developer side. After this CFG from all
other components are summed up to have a global CFG for
the entire system. Then identify the dangerous edges and
select those test cases that pass through these dangerous
edges. Another technique is presented by Lin et al. [24] which
is based on code transformation which models both the client
side and server side Java code as a single program component.
The techniques based on flow of control in program in [22, 23,
and 24] have pros and cons which can be compared to
traditional control flow based techniques for procedure based
programs. In [25] Khan et al. have presented a strategy based
on model based approach where service interfaces are defined
by visual contracts, i.e .pre and post conditions illustrated as
graph transformation rules. With the study of dependencies
of these rules we can estimate the effect of a change and thereby
decide the selection of test cases as a strategy for regression
test selection.

3 PROPOSED APPROACH
The proposed system is based on branch coverage regression
testing. It is a testing method where each of the possible
branch from every decision point is selected and executed
minimum once. This makes sure that all reachable code is
traversed and executed by taking every branch true or false.

In the proposed system to reduce the no of test cases i.e to
achieve regression test selection with respect to retest all
approach which is based on branch coverage as given above,
hybrid model is implemented. The hybrid model is a
collection of data driven and keyword driven approach.
Hybrid test framework combines the advantages of both data
driven and keyword driven approaches

The Proposed RTS strategy is based on the hybrid model. The
strategy implements the hybrid model in the following steps –

1. If the application is taking data from excel then it is data
   driven approach
2. If application is selecting input based on keywords like
   single select, multi select etc. then it is implementing keyword
   driven. For demonstration we have used an online survey
   reporting application system. The online survey reporting
   system is used to have surveys of movie trailers. An online
   survey is a questionnaire that the target audience can
   complete over the Internet. Online surveys are usually created
   as Web forms with a database to store the answers and
   statistical software to provide analytics. The input to the tool is
   taken as an online survey containing a web service. The
   changes can be done on the survey like change in component
   and component behavior. To retest the web based application
after changes, regression testing has to be carried out. If all the test cases with test input data have to be covered then it is the retest all approach based on branch coverage. For test suite reduction for regression testing the hybrid approach for test case selection has been implemented. Figure 1 below shows the framework for proposed approach based on hybrid model.

4 RESULTS AND DISCUSSION

To test the performance of proposed system three surveys were created for sample given as follows:

Survey 1 comprising 20 questions having 4 categories of questions each comprising different components. Four categories are single select questions, multi select questions, numeric punch questions and open text questions.

Survey 2 comprising 25 questions having 5 categories of questions each comprising different components. Five categories are single select questions, multi select questions, numeric punch questions, open text questions and grid (scale) questions.

Survey 3 comprising 30 questions having 4 categories of questions each comprising different components. Four categories are single select questions, multi select questions, numeric punch questions, open text questions, grid (scale) questions and heat map questions. Figure 2 below gives the comparison between change request and test case reduction.

Fig. 1 Hybrid Framework of proposed approach

The test cases are selected on the basis of the test input data. Once the automated tool will execute, then only those test cases will be selected which will have test input data. The tool will ignore all the test cases where test input data is not available. Hence it is implementing the data driven approach.

The data driven approach integrates with the keyword driven approach also in which based on the keyword the test input data is provided to the tool. For instance, if input is provided to multi select question in which multi select serves as a keyword then only those test cases will be selected where keyword is multi select and test input data is data driven.

Fig. 2 Performance analysis in test case reduction

The proposed system is having inclusiveness which measures the capabilities of techniques to select test cases that will cause the modified program to give a different output than the certified program. The proposed system is safe as it selects all the test cases which are modification revealing.
The generality of a regression test selection technique is its ability to be used in a wide and practical range of situations. The proposed approach is applicable to web applications having transactional and reporting system. Any application which one can access through the web browser can use this proposed approach.

If we compare with existing manual system which one is purely human based analysis and testing system, then due to human intervention there may occur many human errors and problems and some scenarios may get omitted. This manual system is error prone approach and time consuming. We can’t cover all possible permutation and combination using manual approach for regression testing. The approach presented is platform independent and provides fast processing of data for testing. The major disadvantage of the approach is that it is implemented for only those web based application containing structured databases. For unstructured databases, implementation is left as a future scope.

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 have analyzed the issues and challenges related to regression testing of web services which are a part of web based application system. Also a novel approach of maintaining reliability and quality of such systems is proposed. The approach is evaluated on an online web based application system and results are reported thereby. Thus we can ensure to have quality and reduce maintenance costs of constant evolving web services. The approach presented is platform independent and provides fast processing of data for testing. The major disadvantage of the approach is that it is implemented for only those web based application containing structured databases. For unstructured databases, implementation is left as a future scope.

REFERENCES


ISSN 2277-8616

INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 03, MARCH 2020