Regression Testing Techniques for SOA

T. Manaswi
M. Tech.,
Computer Science Department,
Gitam University.

Received Nov. 24, 2015 Accepted Dec. 10, 2015

ABSTRACT

Service-Oriented Architecture (SOA) has changed the way business enterprises get aligned with technology with a very fast pace keeping the demand of re-alignment time very short. Testing of services becomes equally important in SOA implementation in the context of enterprise architecture, business re-alignment and so on maintaining quality. Unit and integration testing though holds a key testing strategy but regression test becomes inevitable when services are orchestrated, choreographed and bound dynamically as per the change in requirement as well as when services are upgraded. In this paper a survey work has been done for regression testing method for SOA. Here the changed scenario of implementation of SOA enabled applications has been considered.

Key words: regression, SOA.

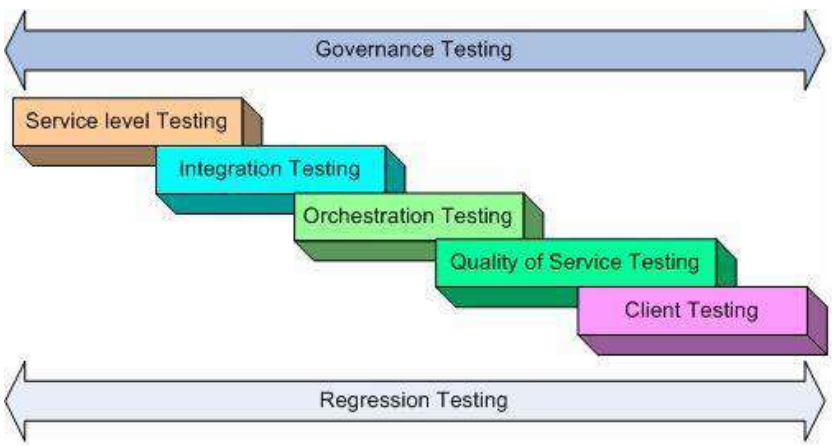
Introduction

Regression Testing in SOA is one of the major matters of concern because of the dynamic nature of the Service binding and the adaptability to accommodate the changed requirement. The actual configuration of the service is known only at run time, so It becomes very complex to verify whether the changes made in earlier version of the system are correct or not and it does not affect the functionality and performance of the existing system. Regression testing is essential for ensuring software quality. It is the process of validating modified software to provide confidence that the changed parts of the software behave as intended and that the unchanged parts of the software have not been adversely affected by the modification. Service-Oriented Architecture (SOA) has changed the way business enterprises get aligned with technology with a very fast pace keeping the demand of re-alignment time very short. Testing of services becomes equally important in SOA implementation in the context of enterprise architecture, business re-alignment and so on maintaining quality. Unit and integration testing though holds a key testing strategy but regression test becomes inevitable when services are orchestrated, choreographed and bound dynamically as per the change in requirement as well as when services are upgraded. In this paper a survey work has been done for regression testing method for SOA. Here the changed scenario of implementation of SOA enabled applications has been considered.

BASIC APPROACH OF REGRESSION TESTING

An issue that makes service-centric systems very different from traditional applications is the lack of control a system integrator has over the services one is using. System integrators select services to be integrated in their systems and assume that such services will maintain their functional and non-functional characteristics while being used. However, this is not the case: a system exposed as a service undergoes-as any other system-maintenance and evolution activities. Maintenance and evolution strategies are out of the system integrators control, and any changes to a service may impact all the

systems using it. This makes service-centric systems different from component-based systems: when a component evolves, this does not affect systems that use previous versions of the component itself . In the SOA Testing Life cycle depicts that Regression and Governance Testing are the activities that span all over the testing life cycle of the SOA Application. While performing other testing activity sequentially, Regression Testing is performed concurrently at different phases along with other testing activity . Testing SOA applications is different from traditional testing approaches. Besides regular functional or regression testing, we also need to ensure Service Level Testing, Integration Testing, Orchestration Testing, Quality of Service Testing and Client



SOA Testing Life Cycle

COMPARISON OF VARIOUS REGRESSION TESTING APPROACH

Comparison of various Regression Testing Approach is given below in Table on the basis of various factors and the merits and demerits are also listed in the table:

Selective Regression Testing

Selective Regression testing is performed when the changes to the system covers only some part of the system. Suppose n number of test cases are there to completely test the system then we will select subset of the tests and repeat that sub set of the test cases again to ensure that the changes in the system does not affect the functionality of the other part of the system. Selective regression testing takes less time than the time required to retest the system. A control flow graph is used to select the test subset; the most critical issue in the subset selection is creating the subset of tests from the test suite.

Code Based Regression Testing

In code based regression testing, modified code and the code associated with that are considered and testing is performed to ensure whether the changes in code is working properly or not. The code based regression testing does not support scalability but it is good for Unit Testing as the code module is considered as a unit and that can be tested individually without any problem.

Risk Based Regression Testing

In Risk Based Regression Testing the approach is based on the selection of a risk area of test suite. A risk area is identified which is considered as the most coverage area that get affected by the changes made in the system. Though the Risk Based Regression Testing does not guarantee the success of the Test. Level of Security is also less in Risk Based Regression Testing.

State Based Regression Testing

A set of coverage criteria is proposed based on control and data flow in UML state diagram and it is shown how to generate test cases satisfying these criteria from UML state diagrams. First control flow is identified by transforming UML state diagram in to Extended Finite State Machines(EFSM)

Model Based Regression Testing

Model based regression testing is applied when the code is unavailable. In model-centric approaches, modifications are first done to the models, rather than to code. In model based regression testing test cases are selected if they traverse the modified model parts. UML class and sequence diagrams are used to implement Model Based Regression Testing but in this approach the identification of the modification is a challenge, which can be marked at towards a stable methodology for SOA Regression Testing.

Observation

Various approaches have addressed the need and methodologies of regression testing in traditional and Object- Oriented software development. SOA based system development caters to the agility of business strategies. These strategies put a pressure of quick re-alignment of business functionalities which are mapped to services of SOA. Most of the traditional regression testing approach surveyed lacks in handling the demand for SOA-based Regression Testing. SOA-based system development challenges its modeling, dependencies of services and CFG generation etc. Generation of test cases for SOA-based application poses difficulties as user interfaces to services are not user friendly.

Deriving dependencies either from SOA modeling or from services becomes difficult because UML (Unified Modeling Language) has shortfall in addressing complex SOA-based modeling, as services are independent business functionalities which are loosely coupled. Hence identifying dependencies among services is a tedious task.

In code based approach, regression testing is done for code changes. Also it can cater to the need for SOA-based regression testing when a service is upgraded with new versions. A combination of various UML diagrams can help us to get the dependency graph like component and state chart diagram. More appropriate will be to use SoaML or Microsoft's Oslo specifically for SOA based application modeling.

CHALLENGES IN TESTING OF SOA

Traditional testing strategies that were used earlier may not be adequate to test these SOA-based systems. Even though they may not be sufficient, there are some open source and commercial tools, some of which are listed in the subsequent section to aid some part of testing SOA-based applications. Various issues related to the testing of SOA-based application are test case management, testing tool requirements and evaluation criteria, testing the underlying implementation (e.g., Web services), testing quality attributes, evaluating the applicability of traditional testing techniques to new problems .

Service-oriented computing promises greater flexibility and efficiency in application development by enabling applications to be composed using third-party services; however, such applications can be difficult to test. Several factors, such as multiple runtime configurations, remote hosting of services, lack of access to service source code, and unanticipated changes in service semantics, present challenges in testing service-oriented

applications] There are various challenges in testing of SOA that are as follows.

A. Lack of Observability of Service code and structure: For users and system integrators services are just interfaces, and this prevents white-box testing approaches that require knowledge of the structure of code and flow of data.

B. Dynamism and Adaptiveness: For traditional systems one is always able to determine the component invoked in a given call-site, or, at least, the set of possible targets. This is not true anymore for SOA, where a system can be described by means of a work flow of abstract services that are automatically bound to concrete services retrieved by one or more registries during the execution of a work flow instance.

C. Lack of control: While components/ libraries are physically integrated in a software system, this is not the case for services, which run on an independent infrastructure and evolve under the sole control of the provider. The combination of these two characteristics implies that system integrators cannot decide the strategy to migrate to a new version of a service and, consequently, to regression testing the system.

D. Cost of Testing: Invoking actual services on the provider's machine has effects on the cost of testing, too, if services are charged on a peruse basis. Also, service providers could experience denial-of-service phenomena in case of massive testing, and repeated invocation of a service for testing may not be applicable whenever the service produces side effects other than a response, as in the case of a hotel booking service.

Rajani Kanta Mohanty et. al are working on model based Regression Testing of SOA. In their paper they have also indicated that regression testing of SOA-based applications impose many challenges. Currently the SOA adoption is increasing in the industry due to various applications aligning with SOA particularly for cloud based application and other areas. This throws open the challenges to quickly build and upgrade the services to meet the market demands. The up gradation of the services demands regression testing, which is an important task to be addressed.

Another important issue that challenge the testing of SOA based Application, is the lack of user interfaces of various Services in SOA context. So there can be many such criteria which can be worked on that will give us a stable methodology for SOA Regression Testing.

Table Different Approach of Regression Testing (RT)

Approach of RT	Ref.	Key Features	Merits	Demerits
Selective	[14]	By using Algorithm to construct Control Flow graph.	1. Takes less than the retest time.	1. Problem of selecting test from existing Test Suite. 2. Problem of determining where additional test may be required.
Code Based	[5]	Regression is done for code changes	1. Good for Unit Testing	1. Traverse only code modification 2. Don't support scalability
Risk Based	[5]	Focuses on test cases which test the risky area	1. Supports Scalability 2. Easy to implement	1. Level of safety is less

State Based	[16]	Uses Program Dependence Graph and Control Dependence Graph	1. BFS and DFS can be applied for traversing and path visibility.	1. No of edges and nodes will change as per the dependence graph.
Model Based	[12,16]	UML class and sequence diagrams are used.	1. Less effort is required 2. Testing activity can be started in the early phase of software development. 3. Better complexity Management	1. Identification of modification is challenging due interdependency

TESTING TOOLS USED

The following list of products, by no means comprehensive, provides an initial roadmap to SOA testing technology. The products cover unit, integration, and system testing and include both commercial and open source tools [3].

ANTS: Load supports testing Web services behavior and performance under the stress of a multiple user load.

E-Test: Suite for Web services provides ways to generate Web services test scripts, validate XML responses, and identify performance bottlenecks by server-side monitoring.

J-Blitz: carries out stress, performance, and functionality testing by generating different loading levels and records anomalies as they occur.

SOAP Scope: Supports testing SOAP transactions by monitoring communications among SOAP endpoints and analyzing Web Services Description Language (WSDL) and SOAP messages against industry standards, such as Web Services- interoperability.

SOA Test: Supports WSDL validation and functionality, performance, and scalability testing. It features a collaborative work flow in which engineers create test cases that the quality assurance team can leverage into scenario based testing.

Web Service Tester: Is an integrated testing suite for functionality, performance, and regression testing Web services.

SITT: Service Integration Testing Tool (SITT) is used for automatic testing of SOA and used for monitoring purpose also [6].

IBM Rational Tester for SOA Quality: A testing tool which is used for the various types of testing of SOA based application. The testing of the application is done through generating the script for various actions. This tool significantly reduces the time and effort required for the SOA based application. A functional and regression testing tool that enables code-free testing of GUI-less web services

Apart from the above listed tools there are some others such as Push to Test, SoapUI and JUnit which are also used for testing SOA-based system.

CONCLUSION

Regression Testing in Service-Oriented Architecture is one of the important testing activity which requires a lot of effort to test the system because of the dynamic nature of the system. In this paper an effort has been made to put the different issues involved with Regression Testing and the comparison among different approaches which will be helpful for the automation of the SOA based Regression Testing

REFERENCES

1. IBM rational tester for soa quality. IBM developerWorks, Mar 2007. http://www.ibm.com/developerworks/.
2. Canfora, G., and Penta, M. D. Soa: testing and self-checking. In In Proceedings of the International Workshop on Web Services: Modeling and Testing (WSMaTe 2006 (2006), pp. 3-12.
3. Canfora, G., and Penta, M. D. Testing services and service-centric systems: Challenges and opportunities. IT Professional 8 (2006), 10-17.
4. Canfora, G., and Penta, M. D. Service-oriented architectures testing: A survey. In Software Engineering, International Summer Schools, ISSSE 2006-2008,
5. Salerno, Italy, Revised Tutorial Lectures (2008), A. D. Lucia and F. Ferrucci, Eds., vol. 5413 of Lecture Notes in Computer Science, Springer, pp. 78-105.
6. Chen, Y., and Probert, R. L. A Risk-based Regression Test Selection Strategy. In ISSRE 2003: Proceeding of the 14th IEEE International Symposium on Software Reliability Engineering (Nov. 2003), pp. 305-306.
7. Dustdar, S., and Haslinger, S. Testing of service-oriented architectures - a practical approach. In Object-Oriented and Internet-Based Technologies, 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a NetworkedWorld, Net.ObjectDays 2004, Erfurt, Germany, September 27-30, 2004, Proceedi (2004),
8. M. Weske and P. Liggesmeyer, Eds., vol. 3263 of Lecture Notes in Computer Science, Springer, pp. 97-109.
9. Endo, A. T., Linschulte, M., da Silva Simao, A., and do Rocio Senger de Souza, S. Event- and coverage-based testing of web services. Secure Software Integration and Reliability Improvement Companion, IEEE International Conference on 0 (2010), 62-69.
10. Kontogiannis, K., Lewis, G. A., and Smith, D. B. A research agenda for service-oriented architecture. In Proceedings of the 2nd international workshop on Systems development in SOA environments (New York, NY, USA, 2008), SDSOA '08, ACM, pp. 1-6.
11. Mani, S., Sinha, V. S., Sinha, S., Dhoolia, P., Mukherjee, D., and Chakraborty, S. Efficient testing of service-oriented applications using semantic service stubs. 2009 IEEE International Conference on Web Services (2009), 197-204.
12. Mohanty, R. K., Pattanayak, B. K., Puthal, B., and Mohapatra, D. P. A roadmap to regression testing of service-oriented architecture(soa) based applications. Journal of Theoretical and Applied Information Technology (2012).
13. Moorthy, G., Perumal, H., and Unnikrishnan, S. Test automation framwork for soa application. Tech. rep., Infosys, 2010.
14. Naslavsky, L., Ziv, H., and Richardson, D. J. A model-based regression test selection technique. 2009 IEEE International Conference on Software Maintenance (2009), 515-518.
15. Parveen, T., and Tilley, S. A Research Agenda for Testing SOA-Based Systems. In Systems Conference, 2008 2nd Annual IEEE (Apr. 2008), IEEE, pp. 1-6.
16. Rothermel, G., and Harrold, M. J. A safe, efficient regression test selection technique. ACM Transactions on Software Engineering and Methodology 6, 2 (1997), 173-210.
17. Tilley, S. R., Bai, X., and Lewis, G. A. First international workshop on service-oriented architecture testing (soat 2009). In ICSM'09 (2009), pp. 583-584.
18. Ul-ann Farooq, Q., Iqbal, M. Z. Z., Malik, Z. I., and Riebisch, M. A model-based regression testing approach for evolving software systems with flexible tool support. Engineering of Computer-Based Systems, IEEE International Conference on the 0 (2010), 41-49