International Journal of Information Technology and Management

Vol. VIII, Issue No. XII, May-2015, ISSN 2249-4510

A NOVEL APPROACH TO TEST SUITE REDUCTION USING DATA MINING APPROACH

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

www.ignited.in

International Journal of Information Technology and Management
Vol. VIII, Issue No. XII, May-2015, ISSN 2249-4510

A Novel Approach to Test Suite Reduction Using Data Mining Approach

Arjun Singh

B.Tech, IV Year, Northern India, Engineering College, Delhi, India

Abstract – The paper analyzes knowledge mining of the test case System. Widespread use of test case systems and explosive growth of databases require traditional manual data analysis to be coupled with methods for efficient computer-assisted analysis It is very important for us to utilize this kind of information effectively. Today, the design of test case with enhanced reliability is a real challenge as it needs expert designers with perfect knowledge about the whole system and also the traditional test case generation approach faces a challenge in test results analysis, because of the difference between the generated test cases and the expected results. So, it will be better if we could establish a method for automatic test case mining based on both program structures and the functional requirements in specifications. Mining approach can be used to have a perfect knowledge about the whole system. In order to improve the accuracy and efficiency of knowledge acquisition, it establishes a knowledge-mining model.

Keywords - Data Mining, Test Case, Test Suite, Clustering

----------X----------

1. INTRODUCTION

Software testing is the process of executing a program in order to find faults, thus helping developers to improve the quality of the product when the discovered faults are solved and reducing the cost produced by these faults. A software test consists of a set of test cases, each of which is made up of the input of the program, called test data, and the output that must be obtained. As the target of software testing is to find faults, a test is successful if an error is found. Testing is a very important, though expensive phase in software development and maintenance; a challenging part of this phase entails the generation of test cases. This generation is crucial to the success of the test, because it is impossible to achieve a fully tested program given that the number of test cases needed for fully testing a software program is infinite, and a suitable design of test cases will be able to detect a great number of faults. (Lilly Raamesh, 2009) If the Test suites tend to grow in size as software evolves, then testing becomes too cost to execute entire test suites. The test suite reduction techniques significantly reduce the size of the test suites. In this study, the application of data mining techniques with software testing is used for reducing the size of the test suit (SE Code Optimization using Data Mining Approach, 2012). The less, the number of test cases, the time taken for executing the program should also be less. This consequently improves the effectiveness of the test process (An Efficient Algorithm for Reducing the Test Cases, 2013), (Kartheek et.al).

Human experts usually plan software-testing activities, while test automation tools are limited to execution of pre-planned tests only. Evaluation of test outcomes is also associated with a considerable effort by software testers who may have imperfect knowledge of the requirements specification. Not surprisingly, this manual approach to software testing results in heavy losses to the world's economy. As demonstrated in this study, Data Mining algorithms can be efficiently used for automated modeling of tested systems. Induced Data Mining models can be utilized for recovering system requirements, identifying equivalence classes in system inputs, designing a minimal set of regression tests, and evaluating the correctness of software outputs. A test suite is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviors. Writing effective test cases for a application is a skill that can be achieved by some experience and in-depth study of the application on which test cases are being written (Lilly, et.al, 2010).

2. REVIEW OF LITERATURE

Data Mining is also popularly known as Knowledge Discovery in Databases (KDD). Data mining, an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data

mining process is to extract information from a data set and transform it into an understandable structure for further use (Reliable Mining of Automatically Generated Test Cases, 2010), (Prioritizing Test Suites Using Clustering Approach in Software Testing, 2012), (Shin Yoo and Mark Harman, 2007). Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly. Apart from regression testing, Automation testing is also used to test the application from load, performance and stress point of view. It increases the test coverage; improve accuracy, saves time and money in comparison to manual testing. A test suite is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviors. A test suite often contains detailed instructions or goals for each collection of test cases and information on the system configuration to be used during testing. A group of test cases may also contain prerequisite states or steps, and descriptions of the tests.

It is not possible to automate everything in the Software; however the areas at which user can make transactions such as login form or registration forms etc, any area where large amount of users. Can access the Software simultaneously should be automated. Furthermore all GUI items, connections with databases, field validations etc can be efficiently tested by automating the manual process. Test Automation should be uses by considering the following for the Software:

- Large and critical projects.
- Projects that require testing the same areas frequently.
- Requirements not changing frequently.
- Accessing the application for load and performance with many virtual users.
- Stable Software with respect to manual testing.
- Availability of time.

Automated software testing tools:

Following are the tools, which can be used for Automation testing:

- HP Quick Test Professional
- Selenium
- IBM Rational Functional Tester
- Silk Test
- Test Complete
- Testing Anywhere
- Win Runner
- Laod Runner
- Visual Studio Test Professional
- WATIR

3. AUTOMATED TESTING

High-volume automated testing involves massive numbers of tests, comparing the results.

Another approach runs an arbitrarily long random sequence of regression tests. Tests that the program has shown it can pass one by one. Memory leaks, stack corruption, wild pointers or other garbage that cumulates over time finally causes failures in these long sequences. (Wes Masri, Andy Podgurski, 2007)

Yet another approach attacks the program with long sequences of activity and uses probes (tests built into the program that log warning or failure messages in response to unexpected conditions) to expose problems. (Zheng Li, et.al,. 2007)

High-volume testing is a diverse grouping. The essence of it is that the structure of this type of testing is designed by a person, but the individual test cases are developed, executed, and interpreted by the computer, which flags suspected failures for human review. The almost complete automation is what makes it possible to run so many tests. (Dennis Jeffrey and Neelam Gupta, 2007)

The individual tests are often weak. They make up for low power with massive numbers.

Because the tests are not handcrafted, some tests that expose failures may not be particularly credible or a skilled tester often works with a failure to imagine a broader or more significant range of circumstances under which the failure might arise, and then craft a test to prove it. Some high-volume test approaches yield failures that are very hard to troubleshoot. It is easy to see that the failure occurred in a given test, but one of the necessary conditions that led to the failure might have been set up thousands of tests before the one that actually failed. Building troubleshooting support into these tests is a design challenge that some test groups has tackled more effectively than others. Testers continually learn about the software they're testing, the market for the product, the various ways in which the product could fail, the weaknesses of the product (including where problems have been found in the application historically and which developers tend to

make which kinds of errors), and the best ways to test the software. At the same time that they're doing all this learning, exploratory testers also test the software, report the problems they find, advocate for the problems they found to be fixed, and develop new tests based on the information they've obtained so far in their learning." (Mao ye, et.al,. 2006)

An exploratory tester might use any type of test--domain, specification-based, stress, risk-based, any of them. The underlying issue is not what style of testing is best but what is most likely to reveal the information the tester is looking for at the moment.

Exploratory testing is not purely spontaneous. The tester might do extensive research, such as studying competitive products, failure histories of this and analogous products, interviewing programmers and users, reading specifications, and working with the product. (Xu, Z. et.al., 2005)

What distinguishes skilled exploratory testing from other approaches and from unskilled exploration is that in the moments of doing the testing, the person who is doing exploratory testing well is fully engaged in the work, learning and planning as well as running the tests. Test cases are good to the extent that they advance the tester's knowledge in the direction of his information-seeking goal. Exploratory testing is highly goal-driven, but the goal may change quickly as the tester gains new knowledge.

Here, we prefer exploratory testing. For that we choose mining of test cases method, since mining of test cases provides the complete knowledge about the total system.

4. AUTOMATIC GENERATION OF TEST CASES

The techniques for the automatic generation of test cases try to efficiently find a small set of cases that allow a given adequacy criterion to be fulfilled, thus contributing to a reduction in the cost of software testing. Model checking techniques can be successfully employed as a test case generation technique to generate tests from formal models.

Knowledge representation and explain mechanism: It is to evaluate the function of Test Case mining from the user's point of view. More easily the knowledge to be understood by users, it plays a greater role. So a very important point is user can understand a new knowledge, this requires the knowledge must be explained in a simple way in the system, such as graphics, natural language and visualization technologies. When data mining discovered a new knowledge, it is able to explain it by the forms of relation, rule and concept. But user will not know the basic principles of the find or to distinguish the value of the knowledge until the system provides a better explanation mechanism. (T.Y. Chen, et.al., 2003)

CONCLUSION

In this paper we present a mining approach to have better knowledge about the test cases and about the full system. So those better tests cases can be generated selected and are used for testing. Because the main challenge in testing is to select & execute test cases. The knowledge mining of test case system has better way of mining the test suite and provides a better set of test cases to test the system performance.

REFERENCES

Lilly Raamesh, G.V. Uma (2009). "Knowledge Mining of Test Case System" International Journal on Computer Science and Engineering Vol.2 (1), pp. 69-73

SE Code Optimization using Data Mining Approach, (2012). International Journal of Computer & Organization Trends –Volume2 Issue 3.

An Efficient Algorithm for Reducing the Test Cases which is Used for Performing Regression Testing (2013). 2nd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2013) March 17-18,

Kartheek Muthyala et al., A Novel Approach To Test Suite Reduction Using Data Mining, Indian Journal of Computer Science and Engineering (IJCSE) .

Kartheek Muthyala et al., A Novel Approach To Test Suite Reduction Using Data Mining, Indian Journal of Computer Science and Engineering (IJCSE).

Lilly Raamesh et. al. (2010). An Efficient Reduction Method for Test Cases, International Journal of Engineering Science and Technology, Vol. 2(11), pp. 6611-6616.

Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification, (2010). International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3.

Prioritizing Test Suites Using Clustering Approach in Software Testing (2012). International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-4.

Shin Yoo and Mark Harman ”Pareto Efficient Multi-Objective Test Case selection”, Proc. ISSTA ’07, July 9-12,London, U.K. 2007 ACM.

Wes Masri, Andy Podgurski (2007). “An empirical study of test case filtering techniques based on exercising information flows”, IEEE transactions on software Engineering, vol.33, No.7.

Zheng Li, Mark Harman, robert M. Hierons. (2007). ”Search algorithms for regression test case prioritization”, IEEE transactions on software Engineering, vol.33, No.4.

Dennis Jeffrey and Neelam Gupta. (2007). “Improving fault detection capability by selectively retaining test cases during test suite reduction” IEEE transactions on software Engineering, vol.33, No.2.

Mao ye, boqinFeng, yao Lin 7Li Zhu. (2006). “Neural Networks Based Test Case Selection” Proc of IEEEtransactions.

Xu, Z.; Gao, K.; Khoshgoftaar, T.M. (2005). “Application of fuzzy expert system in test case selection for system regression test”, Information Reuse and Integration, Conf, 2005. IRI -2005 IEEE International Conference on volume, Issue, 15-17, Page(s): pp. 120 – 125

T.Y. Chen, Pak-lok poon, T.H. Tse (2003). ”A choice Relation framework for supporting Category-partition Test Case generation” IEEE transactions on software Engineering, vol.29, No.7.