International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

Generation Of Automated Test Cases Using UML Modeling

Anjali Sharma
Department of Computer Science &Engineering
Lovely Professional University, Punjab

Maninder Singh
Department of Computer Science & Engineering
Lovely professional University, Punjab

Abstract

The generation of test cases is the most important aspect of the testing process. Mostly software fails the reason behind that the written test cases are not able to detect the errors. Commonly the testing is done on the requirements or the code, while the design is seldom concerned but in present work designed based test case generation is proposed.

Present work is to design such algorithm which is use to generate the test cases and also calculate the server hitting cost per test cases by the help of UML modeling using use case diagram, class diagram and sequence diagram. Use case diagram provides the whole entity information before the constructing or design the actual system. Class diagram which provides the static view information of the system. Sequence diagram gives the information of time ordering messages.

The UML models are naturally the sources of test case generation. This research proposes approach for generating test cases and calculate server hitting cost using UML modeling diagram. Firstly, draw a UML sequence diagram which then converted it into the control flow graph (CFG) of each test case and then CFG nodes augmented with different information necessary to compose test vectors. It recommends a technique to optimize generated test cases, error-handling and reduced the regression testing.

1. Introduction

Software testing is a type of procedure in which faults are identified, separated and ensure for modification and results that the product is fault free and make quality product in the end and hence user satisfaction. Unified Modeling language(UML) using object oriented modeling technique plays an important role in the development of business software.UML is the industry standard for software development, UML diagrams are effective enough to hold the performance of phase one of the most. Software testing is one of the most important elements. While using the concept of directly reuse the design model for test cases generation which avoids the cost of building test models. [1].

The Unified Modeling Language (UML) is "a standard language for building, visualizing, maintaining, and model the solutions of software systems" .And respectively used for non-IT systems as well as for business purpose .The UML represents a addition of best engineering practices that have successful proven in the constructing or modeling of large and critical systems. The UML is a very important aspect of developing software and the software development process and the UML uses mostly graphical notations to express the design of software projects. Using the UML, it helps project teams communicate, expand potential designs, and also validate the architectural design of the software.

Testing is the stepping stone work throughout the complete life-cycle of software systems. The different stages of software life-cycle have involve different kinds of programmers or stakeholders whose knowledge and understanding perception of software engineering is differs much. As a suitable level semantically talk about a software system, software architects employ the Unified Modeling Language(UML) which is the standard of software industry.

The Unified Modeling Language [7] is "a graphical language for visualizing, specifying, constructing, and documenting the artifacts of software intensive system". The UML gives you a standard way to write a system's blue prints, covering conceptual things, such as business processes and system functions, as well as concrete things, such as classes written in a specific programming language, database schemas, and reusable software components.

The collaborative work done by Booch, Jacobson, and Rumbaugh as begun of unification, establishes three goals of their work:

- To model systems, from concept to executable artefacts, using object-oriented techniques

- To address the issues of scale inherent in complex, mission-critical systems
- To create a modeling language usable by both humans and machines

2. Related Work

Linzhang et al. [1] proposed a method to automatically generate test cases from UML activity diagrams using a gray-box method. In their method they generate test cases directly from UML activity diagrams. Their proposed method exploits the advantage of black box testing to analyze the expected external behavior and white box testing to cover the internal structure of the activity diagram of the system under test to generate test cases.

In this paper [2] G.Myers mentions the strategies of finding possible errors in program testing .The knowledgeable reader will recognize that the most important component in the bag of tricks of program tester is the knowledge of how to write effective test cases.

In this paper [3] Rajib Mall et al. proposed the approach for the test cases generation automatically by transforming UML sequence diagram into a graph called sequence diagram graph (SDG) and augmenting the SDG nodes with different information necessary to compose test vectors. Retrieving the information those are required for the specification for input, output, pre- and post-conditions etc. of a test case from the extended use cases, data dictionary expressed in OCL 2.0, class diagrams (composed of application domain classes and their contracts) etc. and are stored in SDG. Approach for Providing a tool that straightway can be used to automate testing process.

B.N.Biswal et al. [4] proposed a approach that UML models provide or it is the important source of information for test case design.UML based activity diagram model describes the realization of the operation in the design phase and also supports the information or description of parallel activities and synchronization aspects involved in different activities perfectly. In this approach generating the test cases by analyzing respective sequence and class diagrams of each scenario, which efficiently achieves maximum path coverage criteria and in this approach also having an advantage of the cost of test model criterion is reduced as design is reused.

V.Panthi et al. [5] proposes a technique for Test Sequence Generation using UML Model Sequence Diagram.Uml model provides a lot of information that should not be ignored in testing. In this approach extracting the main features from Sequence Diagram after that it should be write that features in Java Source code according to ModelJUnit Library. ModelJUnit is a extended library of JUnit Library. By using that generated source code it generate the Test Case Automatic and test Coverage. In this approach it mainly for by using the Sequence diagram it describes the systematic test Case Generation technique performed on model based testing (MBT).

According to N.Kosindrdecha et al. [6] introduces an effective test sequence generation technique to minimize the time, cost and size of the tests while maximizing the test coverage. Their purpose to propose the technique which aims to derive and generate tests from state chart diagram. They introduces “3S” classification of test case generation techniques, which are as follow: specification-based technique, sketch diagram based technique and source code-base technique.

3. Proposed Work

By using Test Set Generation (TSG) and server hit count algorithm it will give the optimized result and it will appropriate and effective. This testing approach to be effective with less cost and time consuming. It will generate test cases for the large system, it handles large system which having number of events or operations.

Report Generator: Report Generator to create a good (sub optional) test case coverage, and error reduction rate of conversion of the situation and the resulting paper. Correspondence in the past for the future of the printed report in detail about the best test case of a Note, submitted to the user.

Use Case Template: Use cases are define for capture the USER and stakeholder needs as well as business decisions. It is not the place for design decisions or technical solutions. It provide the information for the system that will be designed and developed later.

Class Diagram: It provides an overview or idea about the targeted system by defining the objects and classes inside the system and also define the relationship between them. It provides a huge variety of usages, you can use class model in the other model diagram i.e. interaction diagram for modeling the detailed design of the dynamic behavior.

Test Case Generator: The test case generator produces new test cases by taking set of sequence of transitions from sequence diagram as Test set generation (TSG) as input. In general we have <input> are as defined in our test case, <transitions of the sequential order > and <the observed output>. The automated test cases are generated by using TSG (Test Set Generation) algorithm.

Test Cases and Coverage File: This module has been produced in all the test cases and their

respective coverage of the information is stored. For storing a stack test cases and their coverage of the data, stack is used in this module.

Data Dictionary: it defines the metadata repository and also defines the format rules, constraint and rules for data integrity. Without this modification to the application could become impossible.

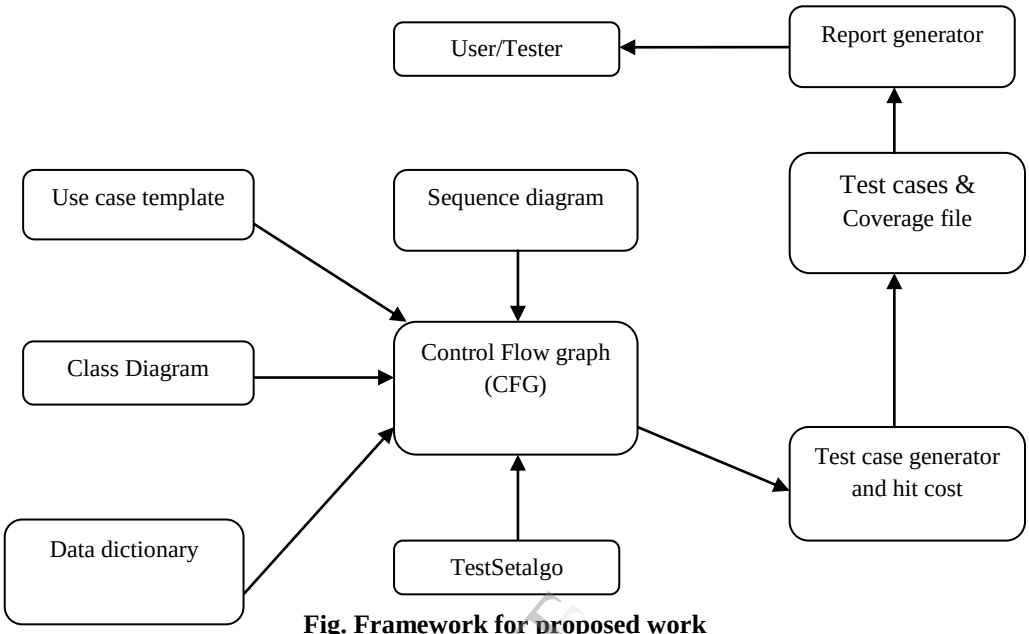

Fig. Framework for proposed work

4. Conclusion

We show the automatic generation of test cases from the sequence diagram and the methodology has been proposed to convert the sequence diagram into the control flow graph (CFG). Hence, our approach to design a algorithm that simply can be used to automate the process of testing. It also provides to calculate the server hitting cost per test case of automated system. It is uniqueness of this research that it easily recognize the testing cost by getting the server hit cost. This implies that our approach can handle large design i.e. scalable convenient and efficient.

5. References

[1] W. Linzhang, Y. Jiesong, Y. Xiaofeng, H. Jun, L. Xuandong, and Z. Guoliang, "Generating test cases from uml activity diagram based on gray-box method", 11th Asia-Pacific Software Engineering Conference, IEEE Computer Society, 2004, pp. 284 –291.

[2] G. Myers," The art of software testing", Hoboken, New Jersey: John Wiley & Son, 2nd ed., 2004.

[3] M.Sarma, D.Kundu and R.Mall," Automatic Test Case Generation from UML Sequence Diagram", IEEE, Computer Society, 2007.

[4] B.N.Biswal, P.Nanda, D.P.Mohapatra, "A Novel Approach for Scenario-Based Test Case Generation", International Journal on Information Technology, IEEE, 2008.

[5] V.Panthi, D.P.Mohapatra," automatic Test Case Generation using Sequence Diagram", International Journal of applied Information System, 2012.

[6] N.Kosindrdecha, J.Daengdej,"A test generation method based on state diagram", Journal of Theoretical and Applied Information Technology, 2010.

[7] G.Booch, J.Rumbaugh and I.Jacobson," the unified modeling language user guide", 2009.