

A STUDY AND ANALYSIS FOR TEST CASE GENERATION AND PRIORITIZATION TECHNIQUES FOR UML MODELS

Summary

Submitted to

Babasaheb Bhimrao Ambedkar University

(A Central University)

Lucknow

**BABASAHEB
BHIMRAO
AMBEDKAR
UNIVERSITY**



LUCKNOW
प्रज्ञा शील करुणा
ESTABLISHED 1996

For the degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE

By

WASIUR RHMANN

Under the Supervision of

Prof. Vipin Saxena

Dean

SCHOOL FOR INFORMATION SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY

'A' Grade Accredited by NAAC

(A Central University)

Vidya Vihar, Raebareli Road

Lucknow (U.P.), 226025, INDIA

2017

SUMMARY

Software testing is an important activity and integral part of the software development process. It consumes around 50% of the total software development cost. Software testing mainly involves three phases namely: test cases generation, test cases execution and test evaluation. In test cases generation phase, test cases are generated from different artifacts which are used in software development process. Test cases can be generated from specification of the software, code of the program, etc. A test case is a set of input test data given to the system under test to determine whether system is performing according to its specification. Software testing is performed to show presence of faults in the system. Effectiveness of testing depends on the quality of test cases. Good test cases have higher chance of fault detection. The main objective of testing is to detect the large number of errors with the minimum number of test cases. In context of model based on testing, test cases are test paths. These test paths are generated from design models. Input to the system under test is given to traverse these test paths and expected output is determined from model of the system. Generation of test cases is a creative part of the software testing. In the test cases execution phase, generated test cases are executed with the software under test. In test evaluation phase, test cases are checked which are passed and failed.

The present work deals with the generation of test cases. Unified Modeling Language (UML) is a standard modeling language which is used in analysis and design phase of the software development process. Most of the test cases generation techniques are based on code of the program. Test case generation from design models facilitate the availability of test cases earlier in the development process. In this work, test cases are generated from different UML diagrams. There are large numbers of test cases for even small size software. So execution of all possible test cases is not possible due to time and resource constraints. Test cases prioritization techniques schedule the test cases in an order which improves the effectiveness

of test cases. Prioritized test cases cover the critical components earlier and can expose the fault in the system at early stage of software development. Detection of faults and testing of critical components earlier in testing process can save time and resources. In this work, UML is used for generation and prioritization of test cases. For generation of test cases, information available in UML diagrams is used. Different types of UML diagrams which represent different views of the system are used for generation of test cases.

Chapter wise summary is given below:

CHAPTER I INTRODUCTION

This chapter contains the introduction and basic concepts related to software testing and history of UML. UML a standard modeling language has been evolving since 1990 and it was adapted by Object Management Group (OMG) in 1997. Major revision of UML, UML 2.0 came in 2005. In same year, it was adapted by ISO and became very popular among software engineers after adaption of UML by OMG. Different views of the system can be modeled with different types of UML diagrams. UML diagrams are mainly categorized in two categories namely: Structural diagrams and Behavioural diagrams. Some UML diagrams which are used for generation of test cases are presented with an example of cash withdrawal from aadhaar number based ATM system and methods which are used for generation of test cases are also described.

CHAPTER II REVIEW OF LITERATURE

This chapter deals with the review of the related work. Test cases are generated from different design models. Various researchers used UML activity, sequence, state, class for test cases generation. Markov chain models and Finite State Machine (FSM) are also used by researchers for generation of test cases. Single diagram is usually not sufficient to express System Under Test (SUT). So some researchers combined two or more diagrams for generation of test cases. Regression test cases are selected from UML diagrams. Different soft computing techniques like Genetic algorithm, Ant colony optimization, Firefly algorithm,

Particle swarm optimization, and Cuckoo search, etc. are used for optimized test cases generation. These optimization techniques are also used for prioritization of test cases and all these are well defined in the chapter.

CHAPTER III TEST CASES GENERATION USING GENETIC APPROACH AND EXTENICS THEORY

This chapter deals with the test cases generation based on Genetic algorithm and Extenics theory. Test cases are generated from UML activity diagram and genetic algorithm is used for test cases prioritization. A case study of book issue from library is presented to validate the proposed approach. In second part of the chapter, UML sequence diagram is used for test cases generation. Extenics theory is used for generation of minimized test cases. Extenics theory is applied for solving contradictory problems. Contradictory problems are those in which goals and given conditions are contradictory. In presented work, generation of minimized test cases with maximum faults is a contradictory problem. A case study of aircraft departure activity is taken for generation of test cases, .mdl file of UML sequence diagram is taken as input and converted it into XML form and generated XML is used to draw flow graph. Then flow graph is converted into its dual graph. Graphviz which is an open source tool and helps in graph visualization, is used for graph creation and obtained dual graph is used for generation of test cases.

CHAPTER IV REQUIREMENT AND RISK BASED TEST CASE PRIORITIZATION

This chapter deals with the generation and prioritization of test cases based on risk associated with the projects. The first part of the chapter deals with test cases prioritization and selection using 0-1 integer programming which selects or rejects test cases. In this method, 1 represents the selection of test case and 0 represents rejection of test case. Test cases are selected based on time constraints. Objective function is taken as maximization of statement coverage, requirements priority, risk exposure and faults exposed by test cases. Constraint is total time

allocated to the testing. Second part of the chapter deals with the risk based test cases prioritization from UML state machine diagram. A case study of ATM is presented for prioritization of generated test cases. Risk information is attached with different states of UML state diagram. UML stereotypes are used for risk information representation. Some states are more affected with risk than other state as risk is categorized in four categories namely annoying, hindering, catastrophic and damaging. Risk exposure values are used for prioritization of test cases and generated test cases have better risk coverage. All these aspects are well defined in the chapter.

CHAPTER V TEST CASES GENERATION AND PRIORITIZATION USING FIREFLY AND SLICING TECHNIQUES

This chapter deals with test path and test case generation using Firefly algorithm and Stationary Iterative Method (SIM). In the first part of the chapter, test paths are generated and prioritized from UML activity diagram using Firefly algorithm. Firefly algorithm is a nature inspired optimization algorithm. Firefly algorithm is inspired from flashing behaviour of fireflies. Fireflies attract to other fireflies based on their brightness. Brightness of fireflies depends on objective function and firefly of low brightness moves to higher brightness firefly. A case study of flight check-in process is used to validate the proposed approach. Second part of the chapter deals with test cases generation from UML activity diagram. Slicing techniques are used for easily handling the large and complex software. In context of programming, Slicing is used for extraction of statements relevant to particular computation. In this work, activity diagram is sliced and test cases are generated using stationary iterative method of numerical analysis. SIM is used for finding the error in the result to the approximate solution. A case study of ATM is used for generation of test cases. Decision functions are derived for each decision node of the activity diagram. System of equations is derived from these decision functions. Initial solution is guessed and increments are computed by solving the system of

equations and further generated test cases are minimized. All these are well described through examples.

CHAPTER VI TEST CASES GENERATION FROM UML SEQUENCE DIAGRAM FOR AADHAAR CARD NUMBER BASED ATM SYSTEM

This chapter deals with test cases generation from UML sequence diagram and Object Constraint Language (OCL). Sequence diagram is converted into Sequence Flow Graph (SFG). Pre and post conditions from OCL are used for test cases generation. A case study of Aadhaar Card Number (ACN) based ATM system is considered. Aadhaar card is an identity provided to the citizen of India from government of India and it is based on biometric and demographic data of the citizen. Test cases generated using the proposed approach satisfied the path coverage criteria and cyclomatic complexity is used to validate the number of generated test cases.

CHAPTER VII FUZZY EXPERT SYSTEM BASED TEST CASES PRIORITIZATION FROM UML STATE MACHINE DIAGRAM USING RISK INFORMATION

This chapter deals with prioritization of regression test cases using Fuzzy expert system. UML state diagram is used for test cases generation. Fuzzy expert system used triangular membership function for conversion of crisp input values to fuzzy values. Risk information and change in system due to the modification is used as input to the fuzzy expert system and outputs are categorization of test paths. Test paths are categorized as reusable, retestable and obsolete test paths. Obsolete test paths are discarded as they cover no risk and change. Retestable paths are of higher priority than reusable paths as they cover more risk and change. A UML diagram for ATM is used and after modification, its functionality is enhanced by adding Aadhaar number enable ATM. Security is enhanced with the fingerprint identification.

CHAPTER VIII CONCLUSIONS AND FUTURE SCOPE OF WORK

This chapter deals with concluding remarks of entire research work. In brief, it is concluded that test cases generated from UML diagrams are helpful for validation of the functionality of the system. Generated test cases are available earlier and can also be used by tester without programming language knowledge. Testing system with the UML may help in uncovering errors in behaviour of the system during early stage of software development. Genetic algorithm helps in prioritization of critical test paths. UML is helpful in identification of important test paths. Generation of test cases with risk information will helpful for safety critical systems and financial systems. Test cases are prioritized with risk information which have better risk coverage. UML sequence diagram is helpful for cluster level testing. UML is also helpful for regression test cases prioritization based on risk information and change in the system. Firefly algorithm based test path generation discarded the redundant transitions. Removal of redundant transitions can save time and resources for testing.