

Developing a Model for Test Case Generation

Dr. Abdul Rauf

College of Computer and Information Sciences
Al-Imam Muhammad ibn Saud Islamic University
P.O. 5701| Riyadh 11432 | Saudi Arabia

Abstract

Generation of test cases is the conflicting issue in software industry. There are different methods, techniques and models are proposed to generate the test cases and minimize the test cases and cost of testing and improving the effectiveness and efficiency of the test cases. Test case generation is a method to identify the defects from the software. MBST test case generation is involved for generation of test case by using models. These models can help to generate the automated test cases for SUT. Main goal of this literature survey is to identify the models, techniques and model which are used for generation of automated test cases.

Keywords:

Testing, Test cases, MBST, Test Case Generation.

1. Introduction

MBST requires a subset of the requirements be modeled using a formal notation like FSM, UML etc. MBST describe the Software testing in abstract terms. MBST is a sub part of Grammar based testing which includes the string mutation using FSM, Model checking, Valid Strings and Traces are tests. Section 2 provides a literature survey of MBST methods and models. Section 3 Provides conclusion.

2. Literature Review

Y. Yin, B. Liu, et al [1] highlights the relationship of testing of real time embedded system with UML. They also proposed the test case generation mechanism for testing of embedded system by using extended UML. The proposed automatic test case generation method for testing of real time embedded system is effective, efficient and maintainable, so it also reduces the testing cost.

Researchers extend the Sequence Diagram, UML Class Diagram and Activity Diagram, in order to fulfill the test-case model requirement and suggested the model-based testing of real time embedded systems. They verify this proposed model by implementing it on avionic system.

In real time embedded systems, class diagram shows the static behavior of test case generation. By analyzing the

system researchers use extended class diagram to model the SUT by using OO methodology. In order to describe, complicated functions of real time-systems, layer activity diagram was used in which upper layer represents the top level flow of diagram whereas lower layer represents expanded activity nodes. In test case generation process, search activity diagram is used for all independent paths extraction of specific functions from starting node to ending node. Sequence diagram describe the dynamic behavior during generation of test case.

The activity diagram is micro level explanation of test case generation process because it cannot explain the interaction between objects. However sequence diagram is microscopic level explanation of test case generation process because it can explain the interaction between the objects. The test case generation process includes dynamic modeling, static modeling, generation of formal description of the test case and searching interacting scenarios.

C. Lizhi, Z. Juan, L. Zhenyu [2] proposed the method for generation of test cases using CPN (Color Petri Net) model. Their proposed technique comprises of three coverage criteria of testing. MBST generates the test cases using high level specification of software in formal methods like, FSM, UML, CPN etc. But the researcher proposed approach focused on test data generation from CPN. As compare to FSM, UML, CPN provides automatic state space analysis and random simulations.

For test cases generation process, a state coverage criterion needs coverage of all states in the state space by the test cases. CPN combines the strengths of ordinary PetriNet with the high-level programming language and it is often used for the analysis and simulation. The proposed techniques of test case generation all the state space of the model generating automatic states by using CPN Tool. W. Bang, Z. Chunhua et al [3] propose the improved automated unit test framework i-N unit. Which resolve the redundancy problem in MDA based test method for generation of unit test cases by separating the test code and test data. Automatic test cases were generated using MDA based approach for system under test on .Net Platform. The idea of i-N unit is explained by the figure.

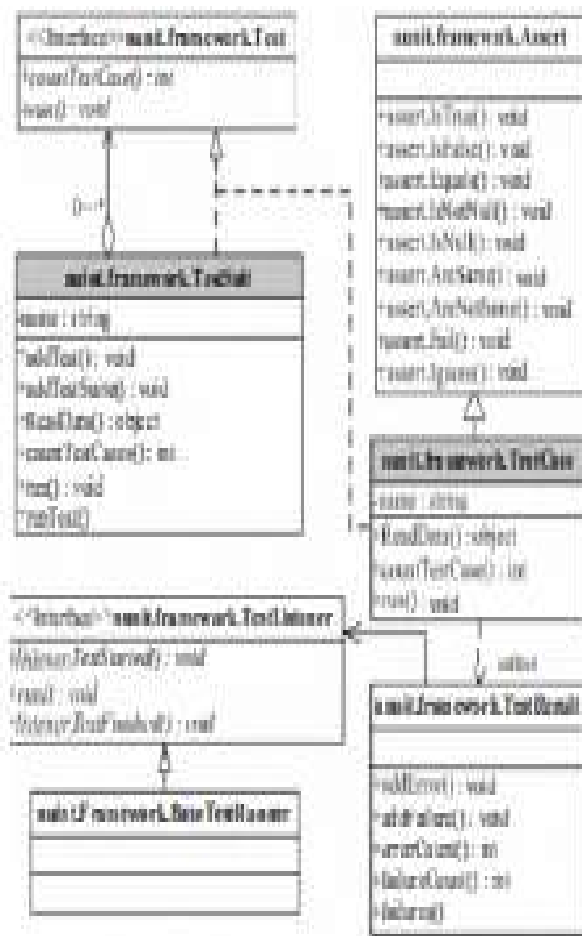


Figure1. Class Diagram of the i-NUnit.

The main steps of generating test cases for I-N unit are as follows:

- Form a model of SUT which is platform independent.
- Transformation of this model to the i-Nunit model named as horizontal transformation.
- Horizontal transformation is again transformed in to test called vertical transformation.

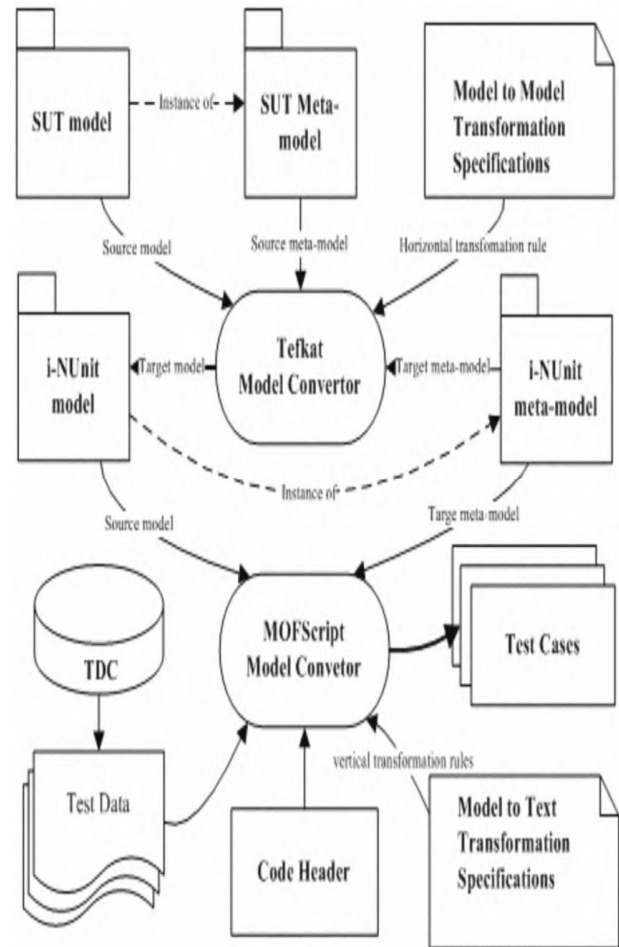


Figure 2. MDA –Based Model of automated Test Case

Test Data Container (TDC) was added in i-N unit frame work in order to separate test data from test code. i-N unit automatically generates test cases by using two-layer model transformation. P. Simon, J. Claude et al [4] proposed a tool and method for automatic test case synthesis from scenario and state-based model by using UML framework.

The method described by the researchers is prototype tool that support the automatic generation of UML test cases by using UML Test objectives and UML System Model. Researchers focus on following issues, which are concerning the conformance testing in a UML frame work.

- A complete formal basis process for test case generation from UML model.
- A formal operational semantic for UML model.
- A scenario based language in UML frame work for the expression of test cases and test objectives.

Proposed synthesis method has four main parts i.e.,

- Derivation of formal specification.
- Derivation of formal objective.
- Synthesis of test case based on formal method.
- UML Test case derivation.

Figure 3 describes the UML model semantics and test objective.

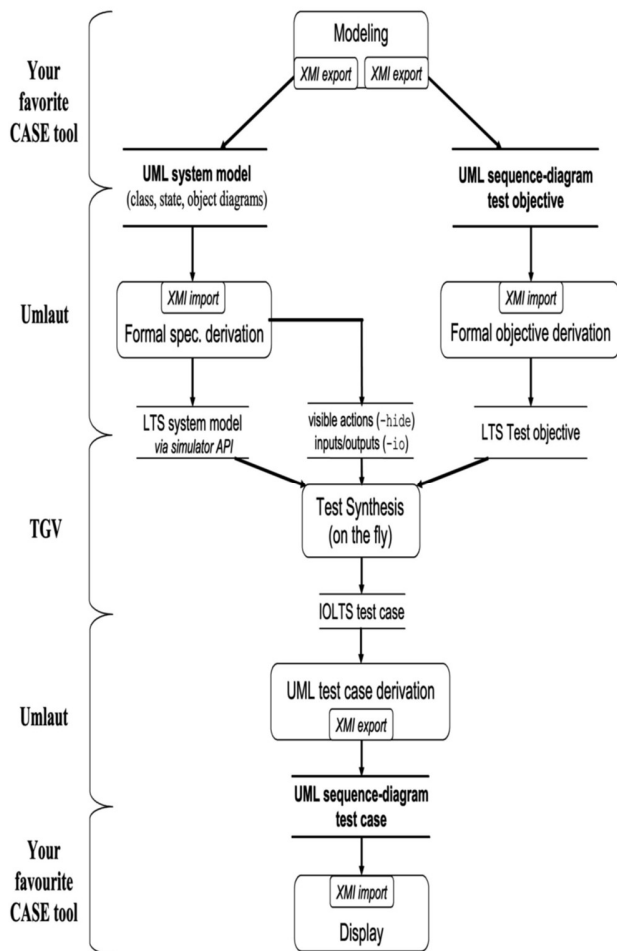


Figure 3. The synthesis Method

For tool purpose researchers use Umlaut frame work to automatically compile UML model.TGV Tool is used to derive the automated test case generation.

M. Zhong, J. Xingan et al [5] proposed PGA (Partheno-Genetic Algorithm) for problem of test case generation.This method used test generation project of DC16Vo1 Chip which used PGA for addressing the problem of instruction-set generation.

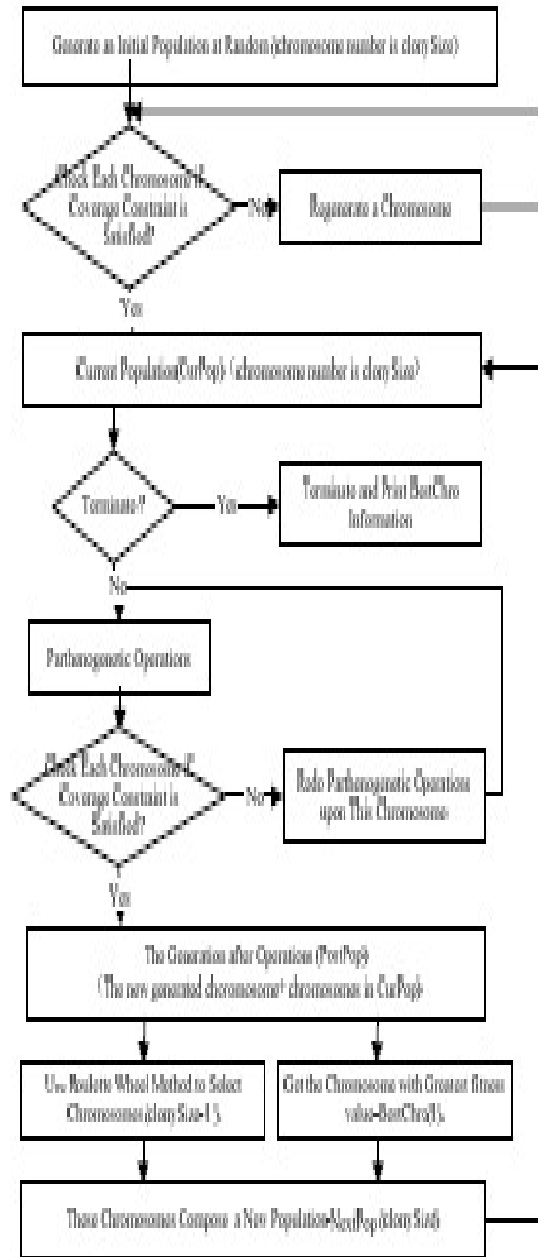


Figure 4. Flow Chart of the algorithms(PGA)

Y. Yuan, L. Zhongjie, S. Wei [6] proposed a search based approach for generation of BPEL test cases. For BPEL representation, this approach extends the CFG&BFG. The BPEL method of test generation comprises of four steps.

- Transform BPEL to BFG.
- Transverse the BFG to generate Test Path.

- Filter in feasible test case and test data generation for feasible test path.
- Generate abstract test case by combining path and test data.

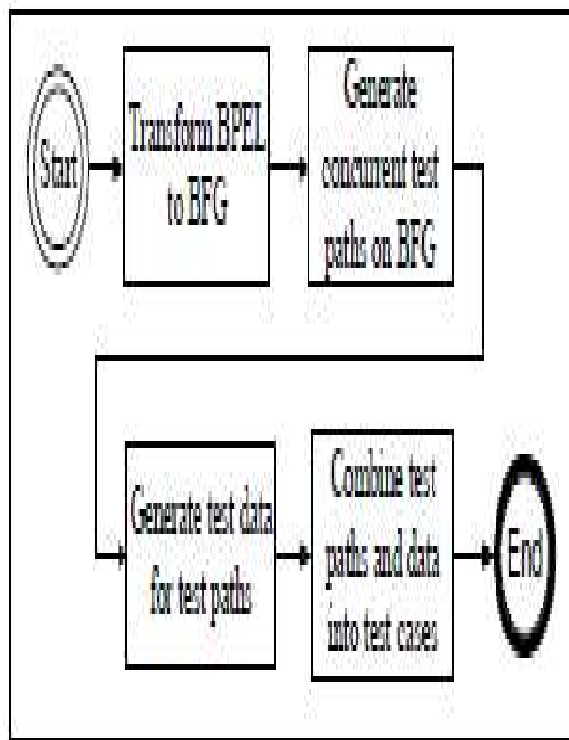


Figure 5. Top Level Work Flow of Proposed Method

As the test data and test path are prepared, they are automatically combined into test cases. R. Matthias, H. Michael [7] proposed a Traceability Driven Model for generation of test cases to reduce the gap between manual technique and automatic technique. The method of test case generation has four steps.

- Requirement Specification
- Activity diagram of the behavioral model
- Behavioral model consist of state charts
- Test cases are processed automatically

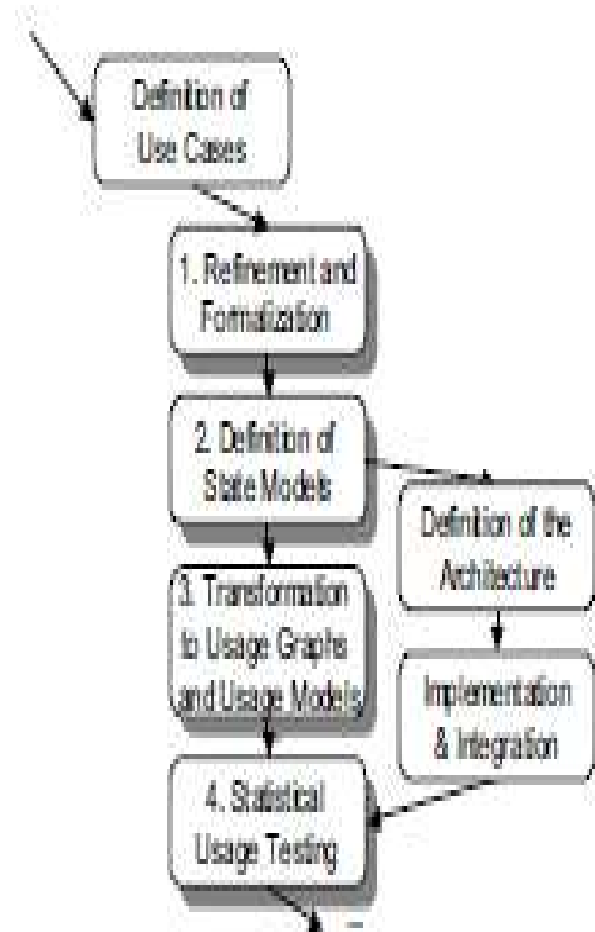


Figure 6. Overview the Test Case Generation Process

H. Ying, C. Rong, D. Zhenjun [8] proposed an automated method for GUI testing of software (J2ME). They adopt the automatic generation of test cases on the basis of FSM. Interpreted test cases are performed from script that initiates J2ME Program under test.

GUI testing of software (J2ME) is more challenging because of weak language characteristics. To overcome this difficulty FSM-Based automated testing approach is used. In this approach, specification file and SUT are given as input and FSM model is automatically structured.

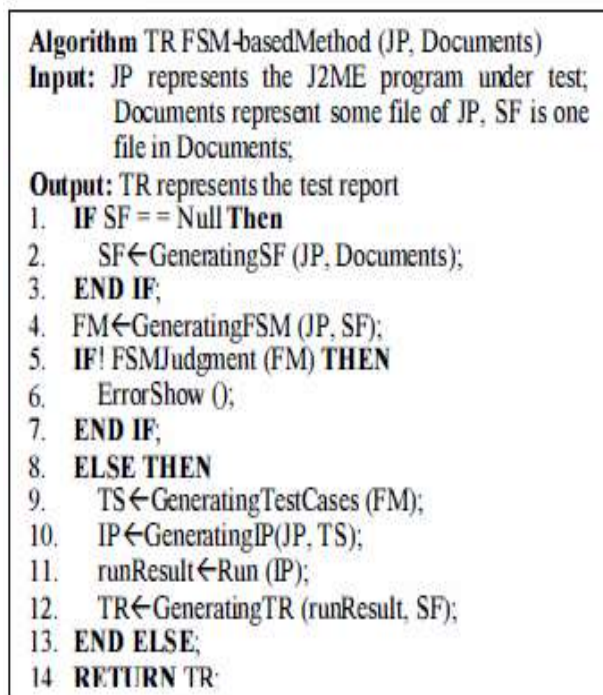


Figure 7. FSM Based Testing Method

In FSM-Based testing, test case generation method is based on WP-method. Advantage of test case generation is full fault-detection and general applicability capability.

K. Hyungchoul, K. Sungwon et al [9] proposed the method of test case generation which reduces number of test cases for a particular test case. Researchers design an explicit activity diagram Model from UML activity diagram which is transform into directed graph.

The directed graph is used for extraction of test cases and test scenarios. The IOAD explicitly shows external input and external output.

For test cases generation from IOAD, the model has twofold principles i.e.,

- Principles of black box testing
- Single stimulus principle

The test case generation procedure is:

- Drive a system of activity Diagram
- Drive IOAD model activity Diagram model
- On the basis of principles construct graph from IOAD Model

- Transverse nodes on the basis of all path coverage criterion
- Generation of test scenarios

A method avoids state explosion problem.

L.P. Beatriz et al [10] propose a model driven testing approach for automatic test case generation for SPL. The proposed approach is based on QVT language, UML 2.0, and UML Testing profile.

The proposal of MD Testing for SPL is explained in figure 8.

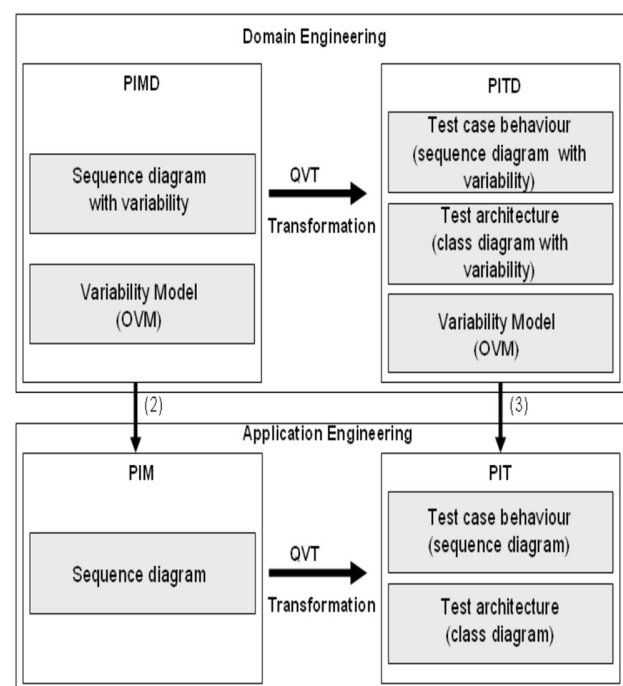


Figure 8. SPL Model driven testing

An approach is designed for PIM model in application engineering and domain using QVT language.

3. Conclusion

Automated generation of test case is main challenge in software industry for SUT. Different researchers focus on Model based test case generation and they propose different testing models for test case generation. Like., UML, extended UML, class diagram for i-Nunit based on MDA test method, PGA, BPEL test case generation method etc.

If these methods and models are implemented in different environment and produce efficient and effective results in their specific environment. These models and methods are reducing the test case and effort for generating test cases. There is need to develop a meta model for test case generation which is used to produce test cases for all type of environments and reduce the test case generation effort and model selection for the particular environment.

References

- [1] Y. Yin, B. Lin et al, "Test Cases Generation for Embedded Real-time Software Based on Extended UML", 2009, International Conference on Information Technology and Computer Science, IEEE.
- [2] C. Lizhi, Z. Juan et al, "Generating Test Case Using Colored Petri Net" 2010, IEEE.
- [3] W. Bing, Z. Chunhua et al, "MDA-Based Automated Generation Method of Test Cases and Supporting Framework", 2010, IEEE
- [4] P.Simon, J. Claude et al, "Test Synthesis from UML Models of Distributed Software", 2007, IEEE Transactions on Software Engineering
- [5] M.Zhong, et al "Partheno-Genetic Algorithm for test Instruction Generation", 2008, IEEE, The 9th International Conference for Young Computer Scientists
- [6] Y. Yuan et al, "A Graph-Search Based Approach to BPEL4WS Test Generation", 2006, International Conference on Software Engineering Advances
- [7] R. Matthias, H. Michael, "Traceability-Driven Model Refinement for Test Case Generation", 2005, IEEE, 12th International Conference and Workshops on the Engineering of Computer-Based Systems
- [8] H. Ying et al, "Automated GUI Testing for J2ME Software Based in FSM", 2009, IEEE
- [9] K. Hyungchoul et al, "Test Cases Generation from UML Activity Diagrams" ,2007, IEEE 8th ACIS International Conference on Software Engineering, AI, NW and Parallel/ Distributed Computing
- [10] L.P. Beatriz, "Model-Driven Testing in Software Product Lines", 2009, IEEE