Jeong-Jin Kang
Edward J. Rothwell
Nguyen Nang Dinh
Nguyen Thanh Thuy
Gyoo Seok Choi
Nguyen Ha Nam

Advanced and Applied Convergence Letters AACL 07

Advanced and Applied Convergence

2nd International Joint Conference, IJCC 2016
Hanoi, Vietnam, January 18-22 2016
Revised Selected Papers

IIBC

IPACT

UML Based Visualization based on Test Coverage Metrics

Dongho Kim, R. Youngchul Kim

SE Lab, Dept. of CIC(Computer Information Communication), Hongik University
2639, Sejong Campus, 339-701, Korea
ray@selab.hongik.ac.kr, bob@hongik.ac.kr

Abstract

Software Testing is an important step in Software Development Process. And It requires a lot of time and money. However, because of the limited time and money, it is impossible to excute all the test cases. And test coverage has been developed based on mostly white box. So there are many difficulties in applying the automatic traceability and refactoring techniques with models and code. In this paper, to solve this problem, we suggest visualization with diverse abstraction level of test case and test coverage techniques. This can guarantee the coverage of test cases possible even with a minimum of excuting the test cases.

Keywords: *SW Testing; SW Test Coverage; SW Visualization; SW Quality;*

1. Introduction

In recent software, the size and complexity of software rapidly are increased. There requires many requirement spec, rapid development, and high quality. As the software is complicated, software testing cost is limited although requiring high quality [1].

To solve this problem, it is required to do the software development and testing simultaneously. This method assists in improving the quality of software. In this paper, we solve this problem by visualizing the test case coverage in the model stage.

The paper is organized as follows: Chapter 2 describes Software Visualization and Test Coverage as related work. Chapter3 mentions the visualization method of Software Test Case and Test Coverage with Test Case Metrics. Chapter 4 describes the case study for cruise control system of the vehicle, Chapter 5 refers to conclusion and future research.

2. Related Works

Software Visualization Techniques automatically show software architecture and quality indicator through a tool-chain. This visualization focuses on visualizing software characteristics such as couplings and cohesions. The method of NIPA's software visualization is to visualize source code and its process for high quality of software [2]. Test coverage is measured on the white-box techniques (White Box) based on procedural language. There are 1) statement, 2) Branch, 3) Condition, 4) MC / DC coverage, etc. techniques [3]. Tester's goal is to get the Maximal Test Coverage with the Minimal Test Case. However, it is not suitable to use case-based, object-oriented test case coverage measurement, and test case generation techniques [4][5].

J.J. Kang et al. (Eds.): IJCC 2016, AACL 07, pp. 164~167, 2016

3. Visualizing with Test Coverage Metrics

Use case test cases coverage	Message sequence			State test case coverage	Object test case coverage	Method test case coverage
	Dialogue Test case coverage	MLU test case coverage	Reusable pattern test case coverage			
use case 1 (Product)	D1	MLU 1	RP1	s0	Object 1	m1
					Object 2	m2
				s1	Object 3	m4
						m5
		MLU 2	RP2	s1	Object 3	m3
				s2	Object 5	m4
		MLU 3	RP3	s5	Object 1	m5
					Object 3	m10
				s0	Object 5	m11
					Object 6	m12
use case2 (Coins)	D2	MLU 4	RP4	s2	Object 3	m6
			RP5	s3		m8
						m9
			RP6	s4	Object 4	m7
						m8
use case3 (Verify Deposit)	D3	MLU 5	RP7	s1	Object 3	m3
						m4
				s2	Object 5	m3
						m5
	D4	MLU 6	RP8	s2	Object 3	m8
			RP9	s3		m9
			RP10	s4	Object 5	m7
						m8

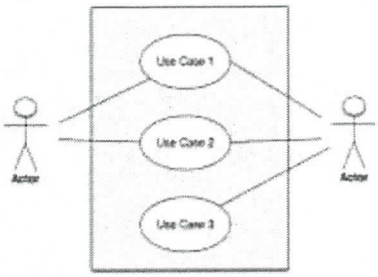

Figure 1. Use Case Test Coverage

Figure 1 shows Test Coverage of 'Use Case'. This coverage is defined, each use case of the actor's input and response of input, pre-condition, post-conditions, expected values. Table 1 shows test case coverage. Through this to extract the high level test case (level 7).

Table 1. Use Case' Test Case Coverage

Test Case	Use Case Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	UC1	UC	Pre-C1	In1	Post-C1	Out1
tc2	UC2	UC	Pre-C2	In2	Post-C2	Out2
tc3	UC3	UC	Pre-C3	In3	Post-C3	Out3

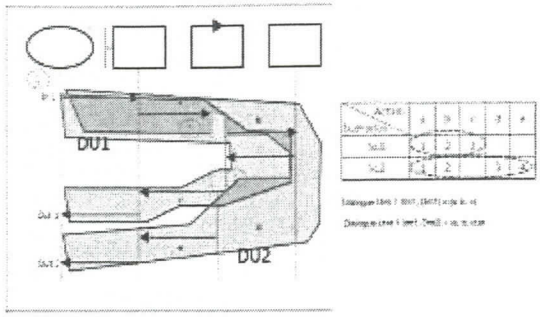

Figure 2. Dialogue Test Coverage

Figure 2 shows Test Coverage of 'Dialogue' There are two types of 'Dialogue'. DU 1 shows an unusual response to the user's input to the 'Dialogue' resulting from unusual cases. DU2 is usually generated 'Dialogue' which shows the usual response to user input.

Table 2. 'Dialogue' Test Case Metrics

Test Case	Dialogue Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	DU1	Dialogue	S0	In1	S0	Om1
tc3	DU2	Dialogue	S0	In3	S0	Om2

Table 2 is 'Dialogue' Test Case Metrics. TC1 of Figure 2 has as input In1. When a message is delivered, the initial state denotes the S0. DU 1 then delivers a message to the interface state because of the object and back to the initial state via the interface object passed back to the control object S0. As a result of expected DU1 generates OM1. Finally, the test case TC is shown in Table 1. TC2 can also be extracted in this way. Through this to extract the high level test case (level 6).

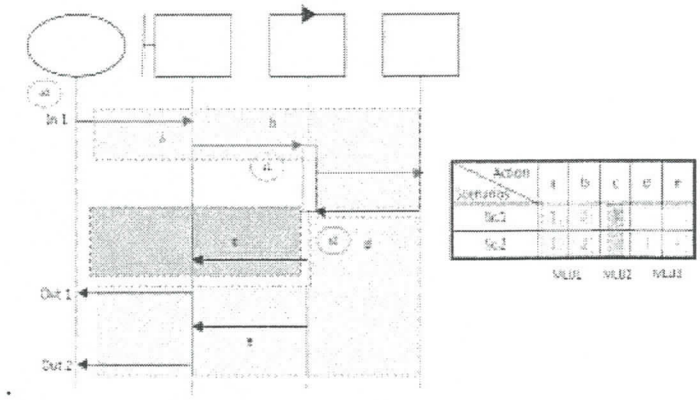

Figure 3. MLU (Maximum Linear Unit) Test Coverage

Figure3 shows Test Coverage of 'MLU'. In this figure, each message denotes a branch until it appeared by selecting a M1 is generated sequentially Im1 by the Method. 'MLU.' M2 generates a selection of Om2 by input Im1. M3 represents the expected value Om2 that may occur Because of selection of another input Im1

Table 3. MLU' Test Case Metrics

Test Case	MLU Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	M1	MLU	S0	In1	S1	M2||M3
tc2	M2	MLU	S1	None	S0	Om1
tc3	M3	MLU	S2	None	S0	Om2

4. Conclusion

The size and complexity of software are rapidly increased, but budget is limited. Moreover, the requirements for the quality is increasing. As a result, a lot of research have been in progress for the test method of high efficiency.

In this paper, we suggest this visualization way through the test coverage and test case metrics. We define each step of test case metrics and level of abstraction test case. Although this is accomplished by first critical test cases of higher level, it is guarantee the Maximal Test Coverage with the Minimal Test Case.

Future study will automatically extract level of test case, and apply to various cases.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2011601) and the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2015H1C1A1035548)

References

[1] Ilene Burnstein, (2003), Practical Software Testing, Springer-Verlag.

[2] NIPA, "SW Quality Management Manual (SW Visualization)," 2013.

[3] ISO/IEC/IEEE 29119 Software Testing, 2013

[4] R. Young Chul Kim, Carlson, C.R. (1999) "Scenario based integration testing for object-oriented software development", Test Symposium, 1999. (ATS '99) Proceedings. Eighth Asian pp. 283-288GV team., GraphViz User's Guide, http://www.graphviz.org/Documentation.php

[5] Sung Bin Ahn, Bo Kyung Park, Dong Ho Kim, Chae Yun Seo (2011) "Requirement Based Testing Using Test Unit, Proceeding of 2011 Korea Conference on Software Engineering, Vol.13, No1 pp. 291-298

ISSN 2288-6060

Advanced and Applied Convergence Letters

The AACL series is committed to the publication of proceedings of Advanced and Applied Convergence. Its objective is to publish original researches in various areas of Smart Convergence. This will provide good chances for academia and industry professionals as well as practitioners to share their ideas, problems and solutions relating to the multifaceted aspects.

Research papers were strictly peer-reviewed by program committees to make sure that the papers accepted were high quality and relevant to the current and future issues and trends in Advanced and Applied Smart Convergence.

The scope of AACL includes the entire area of advanced and applied convergence from the current and future trends. The language of publication is English.