Advanced and Applied Convergence Letters

Advanced and Applied Convergence

2nd International Joint Conference, IJCC 2016
Hanoi, Vietnam, January 18-22 2016
Revised Selected Papers
Vulnerability of USB Device Based on IoT
Byungho Park, R. Youngchul Kim, Dukyun Kim, Sungdeok Cha

UML Based Visualization based on Test Coverage Metrics
Dongho Kim, R. Youngchul Kim

State-Transition Mapping Tree based Representation Model
Soo-Kyung Choi, Je-Ho Park, Young B. Park

A Concept of Geofence to support Scenario-based Context-Awareness
Jinhyung Park, Byungkook Jeon, Sungkuk Cho, Hwanseok Kim, Byungchul Lim

MVDR and Feedforward ANC based Real-time Embedded De-noising Solution on DSP system
Chai-Jong Song, Sung-Ju Park, Chang-Mo Yang, Chil Kim

An Improved Method of Large Data Processing using the External Table
Heewan Kim, Yong Gyu Jung

The Study On Defects Prediction of Production Process Using Big Data
JingJing, Xu, Seung-Jung Shin

The Study On Supervision of Public Institution Using Big Data
KyoungSook Jeon, Seung-Jung Shin

Methodology for the Impact Assessment of DDoS Attacks and Defenses using Virtual Botnets Testbed
Jinyl Kim, Seung-Jung Shin

A Study on Application of Laser Display in Multimedia Environment
Sanghyun Lee, Sungjung Shin

A Study of Prevention System of Broadcasting Facility/Infrastructure utilizing with a mobile
Jaehun Oh, Sungjung Shin

Isolated VM Disk Image Analyzer for Digital Forensics
Han Seong Lee, Yeong Chang Jo, Hyung-Woo Lee

Iterative Channel Equalizer of FBMC/OQAM at an Oversampling Rate
Yong Ju Won, Jong Gyu Oh, JinSeop Lee, Joon Tae Kim

Influencing Factors of Hypertension on Body Activities and Quality of Life
Ungu Kang, Youngho Lee

A Noble Image Magnification Algorithm
Soo-Mok Jung

Implementation of Remote High Power LED Lighting System for Android Bluetooth Application
Inkyu Park, Gyooseok Choi

The Development of a Routing Solution for an Energy Saving System with WiFi Protocol over Android Platform
Joy long-Zong Chen, Yueh Chen
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 execute 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 executing 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. Chapter 3 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].

© The Institute of Internet, Broadcasting and Communication 2016
3. Visualizing with Test Coverage Metrics

<table>
<thead>
<tr>
<th>Use case test cases coverage</th>
<th>Dialogue Test case coverage</th>
<th>Message sequence</th>
<th>Message sequence</th>
<th>State test case coverage</th>
<th>Object test case coverage</th>
<th>Method test case coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>use case 1 (Product)</td>
<td>D1</td>
<td>MU1</td>
<td>RP1</td>
<td>d1</td>
<td>Object 1</td>
<td>m1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MU2</td>
<td>RP2</td>
<td>d2</td>
<td>Object 2</td>
<td>m2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MU3</td>
<td>RP1</td>
<td>d3</td>
<td>Object 3</td>
<td>m3</td>
</tr>
<tr>
<td>use case 2 (Coins)</td>
<td>D2</td>
<td>MU4</td>
<td>RP4</td>
<td>i2</td>
<td>Object 5</td>
<td>m5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RP1</td>
<td>x3</td>
<td>i3</td>
<td>Object 6</td>
<td>m6</td>
</tr>
<tr>
<td>use case 3 (Verify Deposit)</td>
<td>D3</td>
<td>MU6</td>
<td>RP9</td>
<td>i2</td>
<td>Object 5</td>
<td>m7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RP9</td>
<td>x3</td>
<td>i3</td>
<td>Object 6</td>
<td>m8</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Test Case</th>
<th>Use Case Name</th>
<th>Testing Type</th>
<th>Pre-Condition</th>
<th>Input Value</th>
<th>Post-Condition</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tc1</td>
<td>UC1</td>
<td>UC</td>
<td>Pre-C1</td>
<td>In1</td>
<td>Post-C1</td>
<td>Out1</td>
</tr>
<tr>
<td>tc2</td>
<td>UC2</td>
<td>UC</td>
<td>Pre-C2</td>
<td>In2</td>
<td>Post-C2</td>
<td>Out2</td>
</tr>
<tr>
<td>tc3</td>
<td>UC3</td>
<td>UC</td>
<td>Pre-C3</td>
<td>In3</td>
<td>Post-C3</td>
<td>Out3</td>
</tr>
</tbody>
</table>
Figure 2. Dialogue Test Coverage

Figure 2 shows Test Coverage of ‘Dialogue’. There are two types of ‘Dialogue’. DU1 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

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Dialogue Name</th>
<th>Testing Type</th>
<th>Pre-Condition</th>
<th>Input Value</th>
<th>Post-Condition</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tc1</td>
<td>DU1</td>
<td>Dialogue</td>
<td>S0</td>
<td>In1</td>
<td>S0</td>
<td>Om1</td>
</tr>
<tr>
<td>tc3</td>
<td>DU2</td>
<td>Dialogue</td>
<td>S0</td>
<td>In3</td>
<td>S0</td>
<td>Om2</td>
</tr>
</tbody>
</table>

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. DU1 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

Figure 3 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

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Name</th>
<th>Testing Type</th>
<th>Pre-Condition</th>
<th>Input Value</th>
<th>Post-Condition</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tc1</td>
<td>M1</td>
<td>MLU</td>
<td>S0</td>
<td>ln1</td>
<td>S1</td>
<td>M2</td>
</tr>
<tr>
<td>tc2</td>
<td>M2</td>
<td>MLU</td>
<td>S1</td>
<td>None</td>
<td>S0</td>
<td>Om1</td>
</tr>
<tr>
<td>tc3</td>
<td>M3</td>
<td>MLU</td>
<td>S2</td>
<td>None</td>
<td>S0</td>
<td>Om2</td>
</tr>
</tbody>
</table>

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

Advanced and Applied Convergence Letters

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