Lecture Notes in Electrical Engineering 424

Kuinam Kim Nikolai Joukov *Editors*

Information Science and Applications 2017



Kuinam Kim · Nikolai Joukov Editors

Information Science and Applications 2017

ICISA 2017



Contents

| Prioritized Process Test: More Efficiency in Testing of Business Processes and Workflows | 585 |
|---|-----|
| Static Testing Using Different Types of CRUD Matrices | 594 |
| Extracting Test Cases with Message-Sequence Diagram for Validating the Photovoltaic Energy Integrated Monitoring System Woo Sung Jang, Bo Kyung Park, Hyun Seung Son, Byung Kook Jeon, and R. Young Chul Kim | 603 |
| Automatic Test Case Generation with State Diagram for Validatingthe Solar Integrated SystemBo Kyung Park, Woo Sung Jang, Hyun Seung Son, Keunsang Yi,and R. Young Chul Kim | 609 |
| Comparison of Software Complexity Metrics in Measuring the Complexity of Event Sequences Johanna Ahmad and Salmi Baharom | 615 |
| Implementation of Ceph Storage with Big Datafor Performance ComparisonChao-Tung Yang, Cai-Jin Chen, and Tzu-Yang Chen | 625 |
| Web Technology | |
| Predicting Engaging Content for Increasing Organic Reach on Facebook | 637 |
| Learning Performance Evaluation in eLearning with the Web-Based Assessment Cheng-Ying Yang, Tsai-Yuan Chung, Min-Shiang Hwang, Cheng-Yi Li, and Jenq-Foung JF Yao | 645 |
| Improving Teaching and Learning in Southeast Asian SecondarySchools with the Use of Culturally Motivated Weband Mobile TechnologySithira Vadivel, Insu Song, and Abhishek Singh Bhati | 652 |
| Game-Based Learning to Teach Assertive Communication ClickTalk for Enhancing Team Play Bah Tee Eng | 660 |
| Cloud Storage Federation as a Service Reference Architecture Rene Ivan Heinsen, Cindy Pamela Lopez, Tri D.T. Nguyen, and Eui-Nam Huh | 668 |

Automatic Test Case Generation with State Diagram for Validating the Solar Integrated System

Bo Kyung Park, Woo Sung Jang, Hyun Seung Son, Keunsang Yi, and R. Young Chul Kim^(⊠)

SElab, Department of CIC (Computer and Information Communication), Hongik University, Sejong Campus, Seoul 339-701, Korea {park,jang,son,Keunsang,bob}@selab.hongik.ac.kr

Abstract. For safe software development on the solar integrated monitoring system, it is very important how to identify safe behaviors of the system behaviors. Therefore, it needs to test the system behaviors after the software development. To solve this problem, the existing studies have proposed the use case based test coverage analysis at all software development stages [1]. With this method, we identify the test cases based on priority of the system behaviors. In this paper, we proposes automatic test case extraction method based on state diagram among the use case-based test coverage extraction methods. That is, we can use state diagram for a system behaviors with which generates test cases to validate the system. We show an applicative case on the system behaviors of a solar integrated system with this approach.

Keywords: Automatic test case generation \cdot State diagram \cdot Use case based test coverage \cdot A renewable energy integrated monitoring system

1 Introduction

With the recent development of convergence software, the relative importance of software is increasing in the automobile, aviation, and railroad industries. Also, software is applied in various fields and thus requires complex functions. Therefore, it is the importance of quality in increasing software, for example, safety, reliability, etc. As suggested in 2010 Toyota recalls, software defects cause personnel and material loss. For this reason, safe software development is an important issue.

To development a safer software, software test is very important. Software defects are discovered after carrying out a test. If software test is executed earlier for that error can be discovered quickly, we can reduce period of development and cost. That is, the actual cost for correcting deficiency becomes cheaper. But the actual test of the system behaviors is executed after the software is implemented. The existing studies have focused on a use case-based testing combined with software development stage [1]. This method can discover and modify the problem that can occur in early development stage in advance as it conducts a testing from the requirement stage. The current method can test all possible input values. Also, the more the generated test cases are existed, the more the time and cost are consumed for testing. This method extracts all possible test

cases and focuses on effective testing. The finally extracted levelled test case identifies order of priority. The identified order or priority executes maximum coverage testing through minimum test case. This paper proposes an automatic test case establishment method through State Diagram. This method is included in Use case-based test coverage extraction method. The extraction process is generating State Diagram, State Table, and State Transition Tree and extracting test case. This method can detect error quickly and reduce period of development and cost because it can conduct a test in design stage. HIMEM, developed in this research laboratory as case study was applied to photovoltaic monitoring system [2].

This paper is as introduced below. Section 2 describes our test coverage on use case approach. Section 3 describes automatic test case generation with state machine based on the system behaviors. And Sect. 4 describes a case study on apply the solar energy total monitoring system with this approach, followed by conclusion and further studies.

2 Our Test Coverage on Use Case Approach

Figure 1 shows the use case oriented testing procedure [1]. For use case oriented testing, users should analyze domain and set business goals. In other words, users should understand the entire system clearly through domain analysis and set business goals. This is why the testing should also appreciate for business organization to be a great expense.



Fig. 1. A procedure of use case oriented testing

If such test is conducted despite unclear goal, it may lead to indiscriminate tests. This may result in a waste of time and cost, which has also the most critical impact on the entire cycle of development. Most entire software errors occur in requirement analysis

stage. If such requirement is analyzed through exact identification, it can reduce the time, cost, and personnel for software test. Requirement analysis identifies functional and non-functional requirements. The identified requirements can have a 1:1 or 1: n relationship through the existing requirement analysis. Use case is extracted from the functional requirements. Sequence and State Diagram are generated from Use case. Test scenario is extracted from non-functional requirements. Test scenario has a 1:1 or 1: n relationship. Finally, all possible test cases is extracted from use case, sequence, and state diagram. Test case has a 1:1 or 1: n relationship. Through this process, a test is conducted based on requirements. Table 1 shows the abstract levels of a test case coverage matrix.

| Use Case Test Case Coverage | Message Sequence | | | | | |
|-----------------------------------|-----------------------------------|------------------------------|--|--------------------------------|------------------------------|---------------------------------|
| | Dialogue Test Case Coverage | MLU Test Case Coverage | Reusable Pattern Test Case Coverage | State Test Case Coverage | Object Test Case Coverage | Method Test Case Coverage |
| UC 1 | D 1 | MLU 1 | P1 | s0, s1 | Object 1 | m1 |
| | | | | | Object 2 | m2 |
| | | | | | Object 3 | m4 |
| | | | | | | m5 |
| | | MLU 2 | P2 | s1, s2 | Object 3 | m3 |
| | | | | | Object 5 | m4 |
| | | MLU 4 | P6 | s5, s0 | Object 1 | m5 |
| | | | | | Object 3 | m10 |
| | | | | | Object 5 | m11 |
| | | | | | Object 6 | m12 |
| UC 2 | D 2 | MLU 3 | P3,P4,P5 | s2,s3,s4 | Object 3 | m6 |
| | | | | | | m8 |
| | | | | | | m9 |
| | | | | | Object 4 | m7 |
| | | | | | | m8 |
| UC 3 | D 3 | MLU 2 | P2 | s1,s2 - | Object 3 | m3 |
| | | | | | | m4 |
| | | | | | Object 5 | m3 |
| | | | | | | m5 |
| | D 4 | MLU 4 | P3,P4,P5 | s2,s3,s4 | Object 3 | m8 |
| | | | | | | m9 |
| | | | | | Object 5 | m7 |
| | | | | | | m9 |

Table 1. The abstract levels of a test case coverage metrics

3 Automatic Test Case Extraction Based on State Diagram

State transition testing is a model based technique. This technique generates test case based on the system behaviors of the solar integrated monitoring system. We implement a test case extraction tool based on state diagram.

Figure 2 shows a test case generation procedure [3]. First, we model the state diagram of a target system. The state model is converted to state table. The state table is composed of all events of states and events. The state can express each one of all situations. The event is a factor to cause the transition of state. The state table displays each state in the top side of table, and also represents the event in the left side of table. And the rest displays a movable state when a state meets an event. N/A represents a case that can't be movable. After then, we use the information of state table, and generate state transition tree. The state table. Therefore, to look for the transition route of state and generate test case, the state table is converted to the state transition tree. This transition tree generation method displays all states in the upper side of table. The next stage represents all accessible states. The generated test case is a scenario executed by each state in state transition tree.



Fig. 2. State transition testing method

4 A Case Study

This study is an automatic test case generation based on a state diagram, that is, the system behaviors of our integrated renewable energy monitoring system. For this, we use our tool, that is, Hongik MDA based Embedded Software Development Methodology (HIMEM v1.0), developed in our research laboratory of this study. HIMEM v1.0 is an automatic test case generation tool.

This System monitors such information of electric energy and temperature collected from the solar cell. Figure 3 shows the communication architecture of a photovoltaic monitoring system. This system stores the data transmitted from many power plants into our web-sever. And the data sent from each power plant are integrated through metamodel based standard interface. This meta-model is used because the inverter installed to each power plant uses different communication packet. Meta model converts different packets to the same types of packet.



Fig. 3. The communication architecture of a photovoltaic monitoring system



Integrated Monitoring System Development

Fig. 4. The whole system structure of integrated monitoring system

Figure 4 shows the whole system structure of Integrated monitoring system. In this current system, a central server does manage information data occurred from the existing monitoring system. An individual user can possibly monitor the generated energy on the web. Therefore, we can diagnose and resolve problems through monitoring at the central server. Figure 5 shows a test case extraction method based on state diagram for validating the system behaviors. We use our HIMEM v1.0 tool to draw the state diagram to represent the system behaviors of our monitoring system. In the state diagram of our integrated photovoltaic monitoring system in Fig. 5(A), the 'Idle' state is the initial state of the integrated monitoring system. The 'Ready' state represents the beginning state and resetting state of monitoring system. The 'Run' state represents the execution state of system, and processes the data, graph, and connection information of electric energy. 'Run' state is divided into two sub states like data collection (collectData) and data analysis (analyzeData). In a case of data collection, the data of electric energy and environmental sensor are collected through the inverter in a power plant. After then, the collected data are transmitted to monitoring system. If it does occur defects of the equipment, Alarm sends a message to administrator. In other case of data analysis, the collected data are analyzed. This data are printed out in graph after daily/monthly/yearly electric energy analysis. If disorderData occurs, the system is converted to watch out state. At this time, the system checks MPVMSClienct, MPVMSServer, or MPVMSMonitor and sends a warning message. Figure 5(B) shows a state table of integrated monitoring system. The state table represents a state and event when a particular event is generated in an applicant state from the state diagram. The upper side of table represents state, and the left side of table represents an event. The intersection represents an accessible state, that is, to the next stage. Figure 5(C) shows the state transition tree of integrated monitoring system. The State Transition Tree is generated based on state table. The state transition tree expresses all possible states in the form of tree. The test case is generated based on state transition tree. Figure 5(D) shows the finally extracted test case. The extracted test case shows an event and action occurred in the current state. Also, the next state can be identified in the current state. The total of 107 test cases were extracted.



Fig. 5. Test case extraction method with the state diagram

5 Conclusion

Software error is discovered after testing the system. How to validate the behaviors of the system? How to automatically generate test case to check the system behaviors? To solve such problems, we propose an automatic test case generation based on a state diagram for validating the behaviors of a system. In further studies, we will apply the use case-based test coverage extraction method. Also, Future studies need to look at the test case extraction method from Use case and sequence diagram.

Acknowledgments. This work was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2015H1C1A1035548).

References

- Ahn, S.B., Park, B.K., Kim, D.H., Seo, C.Y., Young Chul Kim, R.: Requirements based testing using test units. In: KCSE2011, vol. 13. no. 1, pp. 291–298 (2011)
- Son, H.S., Young Chul Kim, R.: Modeling a photovoltaic monitoring system based on maintenance perspective for new & renewable energy. In: The 2nd International Joint Conference on Convergence, AACL07, pp. 144–147 (2016)
- Kim, W.Y., Son, H.S., Young Chul Kim, R.: A study on test case generation based on state diagram in modeling and simulation environment. In: Kim, T., Adeli, H., Robles, R.J., Balitanas, M. (eds.) Advanced Communication and Networking. CCIS, vol. 199, pp. 299–305. Springer, Heidelberg (2011)