2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia

Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm

[ab]Meiliana[*], [a]Irwandhi Septian, [a]Ricky Setiawan Alianto, [a]Daniel, [b]Ford Lumban Gaol

[a]Computer Science Department, School of Computer Science - Bina Nusantara University, Jl. K. H. Syahdan No. 9, DKI Jakarta, 11480, Indonesia
[b]Doctor of Computer Science – Bina Nusantara University, Jl. Kebon Jeruk Raya No.27, DKI Jakarta, 11530, Indonesia

Abstract

Software testing is an important and critical activity in software development that deals with software quality. However, the testing process is consuming activities that need to be automated to save a lot of resources. Towards automated testing, automating test cases generation as the first testing process is being highlighted. This research aims to generate test case automatically from UML diagram since model based testing that conducted on early phase of software development process show higher efficiency. UML diagrams used in this research are activity diagram, sequence diagram and SYTG as the combination graph. These three diagrams have been proved as the most compatible diagram to generate test case from previous research. Method proposed in this paper is Depth First Search algorithm that is modified to generate expected test cases. This paper proves that modified DFS algorithm applied to generate test case is provide accurate result, every node presented on the test case, include any condition (alt and opt). Comparison result from three different test cases generated shows that test cases from combined UML may not necessarily result in better test cases, due to the possibility of redundant test cases for some test cases. This paper also presenting an experiment result that proving sequence diagrams can produce better test cases.

Keywords: test cases, depth first search algorithm, UML diagram, software testing, test cases generator

* Corresponding author. Tel.: +6221-534-5830 ext. 2188.
E-mail address: meiliana@binus.edu

1. Introduction

Software testing is an important and critical phase that deals with software quality. However, software testing that consists of three phases (test case generation, test execution and test evaluation) [1] is time consuming activity that requires a lot of resource. Therefore, automated testing is strived to save resource spent in the terms of time, cost and effort and to give more accurate result than manual testing that vulnerable to human error. Towards automated testing, automating test cases generation as the first testing process is being highlighted.

Test cases can be generated automatically from source code or visual software model such as Unified Modeling Language (UML), Data Flow Diagram (DFD), or Entity Relationship Diagram (ERD). Research example about code based testing conducted by Srivastaval, et al. used genetic algorithm to optimize test case generation by applying conditional coverage on source code [2]. Another research from Alazzam et al. used information retrieval techniques for the automatic extraction of source code concepts for the purpose of test case reduction [3]. Compare to code based testing, model based testing where test cases are generated from model of the software showed higher efficiency of time and effort. Furthermore, generating test cases in the early phase of software development life cycle provide control management on construction and testing phase. Thus, this research will focus on generating test cases from several UML diagrams that are widely used on software modeling process.

Some previous researches have conducted test cases generation from UML diagrams. However, various methods used and different case provided by previous researchers lead to unclear comparison and evaluation about this field. This research provides one scenario case in different UML diagrams to be used in test cases generation process. As a preliminary work, two UML behavior diagrams which are activity diagram and sequence diagram will be used. An activity diagram can figure the sequential flow of activities of a use-case or business process from the start to the end activity and it can also be used to model logic with system. On the other hand, a sequence diagram can show more detail process about how processes interact with one another and the order of the interaction and indicate the life spans of objects relative to those messages. One additional graph as combination from activity and sequence diagram is formed and used as well for test cases generation in this research.

Modified DFS algorithm is proposed in this research as an enhancement of research from Tripathy et al.[4]. In our experiment, current DFS algorithm that applied for test case generation process generated some redundant node. Thus, a modification is needed to get optimal test cases result. Comparison test cases result for both aforementioned algorithms is provided in the fifth section. The second section will discuss about state of the art of this field. Subsequent section describes our proposed approach in generating test case automatically. Conclusions are given in the last section. This paper aims to provide better evaluation and comparison of test case generated from different UML diagrams with new proposed method; which is modified Deep First Search (DFS) algorithm.

2. State of The Art

Test cases are defined as a set of condition or variables which determine the level of correctness and quality of the product. Simple way to present test case is by providing test path to be followed when conducting a testing. The studied literature shows there are various methods described by numerous researchers for generating test cases and comparing test case from different UML diagrams. We have classified the literature according to different aspects of testing from UML design using different UML diagrams.

A.V.K. Shanthi and G. MohanKumar [1] presented an approach to automated generate test path using TABU search algorithm. In this paper, the activity diagram generated from software design, and then all possible information extracting using write parser in java. Based on the extracted information, an Activity Dependency Table (ADT) is generated. Test case is generated with the help of ADT by applying TABU search algorithm. This experiment show that this method has better performance. All possible test cases are generated and validated by prioritization. This approach can reveal all paths for software to be developed and also obtained test cases valid once. Similarly, some approaches [5–9] also used single diagram to generate test cases.

[Ketik di sini]