Test Case Generation Technique for IoT Mobile Application

Aman Jaffari, Jihyun Lee, Cheol-Jung Yoo, and Jun Hyuk Jo
Dept. of Software Engineering, Chonbuk National University
{aman, jihyun30, cjyoo, bionicle73}@jbnu.ac.kr

Abstract

Currently, IoT mobile applications are growing fast in number and complexity. As a result, the applications quality issue became crucial, hence to ensure their quality a proper testing is highly required. Testing such mobile applications is always tedious, time-consuming and expensive. To cope with these issues, we propose a testing approach using activity diagram with data flow information. The main functionality of IoT applications is exposing the meaningful data obtained from the sensors to the users by doing a lot of analysis, comparison, and computation. Therefore, our focus is on identifying and selecting the most appropriate paths at which calculation is taking place and the paths at which predicate exists. In our case study, we have used a real-world IoT mobile application and identified a total of ten test paths with two predicate uses and two computation uses through an example. With applying only this four critical paths, we can adequately test the application's core functionalities while significantly reduce the testing effort and cost.

1. Introduction

Nowadays, the internet of things (IoT) applications is exploding in use (i.e. from home automation to healthcare, agriculture, etc.), creating new business opportunities, and having a great impact on the economy by optimizing the operations, improving the productivity, reducing the cost and failures. Particularly, in agriculture, due to the continuous growth in the world's population demand for food production is remarkably raised. Hence, precision agriculture became one of the fastest expanding fields in IoT. In farming, the IoT mobile applications are used to improve the quality by real-time, remote monitoring and allowing the farmers to control the factors, such as the soil pH, CO2, moisture, temperature, humidity, the intensity of radiation, and so on [1]. Recently, researchers have predicted that over 30 billion of IoT applications will ship by 2020 [2]. As a result, the IoT applications quality, as well as reliability, is getting more important which highly requires a proper testing. The way we are testing IoT applications is significantly different from testing traditional applications in many ways like dealing with a more dynamic environment (i.e. a large number of connected sensors providing data continuously), and intensive data (i.e. real-time monitoring, collecting, and visualizing a huge amount of data). Testing mobile applications are always challenging, costly and time-consuming [3, 4]. Definitely, in testing the complex IoT mobile applications identifying all the possible paths, and possible inputs within a limited amount of time and resources is even harder.

To address these problems and enhance testing IoT mobile applications, in this research, we propose a testing approach for generating test cases based on the activity diagram with data flow information. The motivation behind this research is to minimize the testing cost and effort while exploiting the efficiency and effectiveness by identifying and selecting the most critical paths. By critical, we mean identifying those paths at which calculation is taking place and the paths at which predicate exists (i.e. *c-use* and *p-use* paths). To show the effectiveness of our proposed approach, in our case study we have used a real world IoT mobile application developed by a software company working on IoT and precision agriculture. The result of the research indicates that the approach is feasible to minimize the testing effort, time and cost while maximizing the test efficiency and effectiveness.

2. Related Works

In this section, we indicate some of the related works on testing mobile applications, and IoT applications testing challenges. Whereas mobile applications are moving beyond low-priced recreational applications to more complex business-critical uses (i.e. IoT mobile applications), researchers are always looking for a better solution for testing them by introducing new techniques and tools. Researchers have also marked the importance of the IoT applications in precision agriculture, including some of the recent ongoing issues in the field. For instance, J. Shenoy [1] introduces IoT as a good solution to the major challenges in agriculture. The source of test challenges in IoT from different points of view has been introduced by [2]. C. Chouhan et al. [5] have applied activity diagram to generate test cases for mobile application. Currently, model-based test case generation is a promising approach and an active area of research. It generates test cases from the existing models representing the software application such as specification document. Usually, the specification-based testing is a technique used for testing systems functionality without any knowledge of the system's internal structure. Allowing early design of test cases is one of the advantages of specification-based testing. Also, in specification-based testing, the complete knowledge about system internal functionality is not required. However, in our approach, to get benefits of data flow information, we need a full understanding of the detailed system functionality. Data flow testing is common approach, used for exposing the improper use of data in a program by modeling the program as a control flow graph, though it is a white-box testing technique [6].

Up to now, there are in fact some studies, proposed for testing mobile applications, and a few of them have utilized model-based testing for their test case generation. However, to the best of our knowledge, first of all, there is no explicate research on testing IoT mobile applications. Most of the existing approaches are revealing the ongoing issues and limiting themselves to testing traditional applications. As stated above in the introduction section testing IoT applications are significantly different from traditional ones. In this research, we focus on testing IoT mobile applications using data flow information strategies for identifying the most critical paths. Doing so helps us to examine the core functionality of the application under test with minimum effort and maximum efficiency.

3. Test case generation approach

In this section, we present our main approach for testing IoT mobile application using activity diagram with data flow information. It can be a good question asking how testing IoT applications are different from other applications. Therefore, we think that before introducing the activities in our main approach, it would be better to point out to some of the characteristics of the IoT applications which make them different from other applications. Unlike the traditional applications, IoT applications are more data intensive, means dealing with huge amounts of data generated by a large number of sensors, so that we have to manage and visualize the collected data as well as responding to them in real-time. This fact of handling with massive amounts of data alone makes IoT testing very different from the traditional one. Also, the IoT applications are more complex and dynamic, means many devices and sensors are interconnected while providing data continuously. Hence, the testers of IoT applications need not take into account only providing test cases for simulating human user actions, but also sensors and other devices. For example, simulating sensor values coming from the information center is tricky comparing to values provided by a human user to the system.

Testing is always difficult and time consuming therefore it is almost impossible to exercise all the paths and every possible inputs. Therefore, the objective of this study is to identify the most critical paths like *c-use* (the paths used for computation) and *p-use* (the paths used for predicate) in order to reduce the overall testing effort and increase the efficiency. In other words, our focus is on identifying and selecting the most appropriate paths at which calculation is taking place and the paths at which predicate exist.

Figure 1 illustrates the basic concept of our approach comprising the four components: the class diagram and activity diagram used as inputs, the activity diagram with data flow information, the data flow graph which is a linear repetition of the activity diagram and finally the test paths as an output. The activity diagrams are used to model the system behaviors and the way in which these behaviors interact with each other by defining the sequence of actions among them. To attain our purpose, at this point we need to mark each activity with dada flow information to identify either the activity is: defined activity, used in a predicate activity or used in a computation activity. After obtaining the activity diagrams with data flow information, it is necessary to convert them into a data flow graph to remove the cycles in the activity diagram. Due to the cyclic nature of activity diagrams, the path traversing can be a difficult task. With converting them into data flow graph, we will be able to analysis and manipulate them easily for extracting the test paths. At last, the test table presents the outcomes containing the strategies and the testing paths. The next case study section further explains each step in more detail.

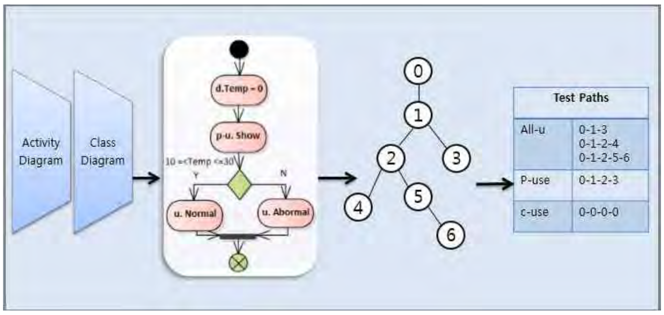

Figure 1. The main concept of our approach

4. Case Study

To show the effectiveness of our approach, we have selected a real-world IoT mobile application for our case study. The application is developed by a software company working on IoT and precision agriculture. Its functionalities are to facilitate remote controlling of the greenhouse, provide real-time monitoring, and to visualize data for the users. It also allows the users to query the stored data from the information center. The application can further help farmers in different ways to reduce the farming failures and increase the productivity by providing automation. The users can also benefit accessing to lots of available shared information about farming quickly.

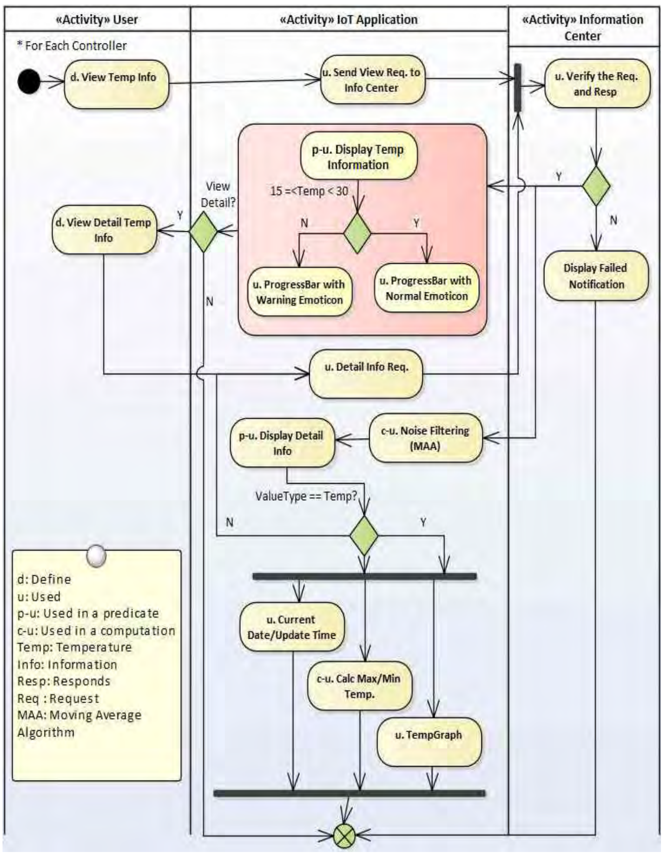

Figure 2. Activity diagram for temperature checking

Figure 2 above depicts the activity diagram used to show the sequence of action for checking the temperature. Each activity in the diagram is marked with either *d* for defined, *p-u* for used in a predicate or *c-u* used for calculation followed by the activity name. A define activity is an activity in which the variables are defined and initialized. An activity annotated with *p-u* is an activity in which the comparison is taking place whereas an activity with *c-u* annotation is an activity that involves computation. For instance in Figure 2 "Display Temp Information" is a *p-u* activity and "Noise Filtering" is a *c-u* activity. There are many *use* activities, but they are not of our interest. Thus, our focus is only on the two most critical ones though they comprise the most significant part of any IoT applications.

Figure 3 demonstrates the data flow graph which is a linear representation of all the paths in the activity diagram. It is used to facilitate traversing the paths by removing the cycles in the activity diagram. The activity flow graph can be used for traversing the paths automatically.

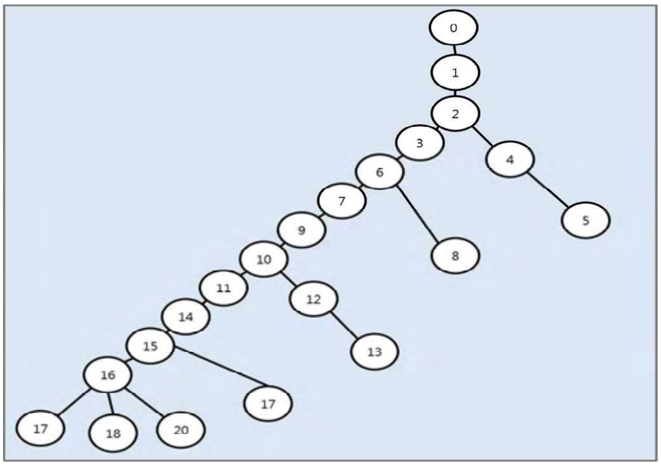

Figure 3. Data flow graph for the activity diagram in Figure 2

Table 1 presents the extracted testing paths that are divided into three categories: a) *All uses*, the paths from every 'define' to 'all use' activities; b) *All p-use*, the paths from 'define' to 'all p-use' activity; c) *All c-use*, the paths from 'define' to 'all c-use' activities. However, we have excluded the simple *use*, because considering all of them will lead to an exponential number of paths and exercising them is not practical. An example of a *p-use* path can be from 'View Temp Info' to 'Display Temp Information' (node 0-1-2-3-6). We have marked only ten *All use* paths in the table, but there can be much more of them in the case of '*all define*' to '*all use*'.

Table 1. Extracted testing paths

Strategies	***Testing Paths***
All uses (AU)	0-1-2-3-6 0-1-2-3-4-5 0-1-2-3-6-8 7-9-10-11-14 7-9-10-12-13 14-15 14-15-16-18 14-15-16-19 14-15-16-20 14-15-17
All p-uses (APU)	0-1-2-3-6 14-15
All c-uses (ACU)	7-9-10-11-14 14-15-16-19

5. Conclusion and Future Works

We have proposed an approach for testing IoT mobile applications using activity diagram including data flow annotation. IoT mobile applications are more complex, dynamic and data intensive. Therefore, calculation and comparison of a large amount of data for the purpose of obtaining and visualizing meaningful data to the users comprise their core functionalities. Testing such kind of application is very expensive, time-consuming and tedious. Hence, we have followed a strategy for identifying and selecting the most appropriate paths. A large enterprise IoT application can contain a vast number of paths to test, and writing test cases for everything is not feasible. Hence we need to put our focus on determining precisely the most critical erroneous paths or the paths that matter the most. To demonstrate it, in our example based on a real-world application, we have identified two *p-use* and two *c-use* paths among a large number of possible paths, so that with only exercising these four paths alone, we can make sure that our application's core functionalities are adequately tested. Moreover, the testing effort and cost are significantly reduced. The future work will focus on applying it to different case studies and focus on further automation.

6. Acknowledgment

This research was supported by the Korea Institute for the Advancement of Technology(KIAT) (N0002018, 2016).

References

[1] J. Shenoy, and Y. Pingle, "IoT in Agriculture", Computing for Sustainable Global Development (INDIACom), IEEE, 2016, pp. 1456 - 1458.

[2] E. J. Marinissen, C. Reyes, and ten others, "IoT: Source of Test Challenges", IEEE European Test Symposium (ETS), 2016, pp. 1 - 10.

[3] H. Muccini, A. Di Francesco, and P Esposito, "Software Testing of Mobile Applications: Challenges and Future Research Directions", 7th International Workshop on Automation of Software Testing (AST), IEEE, 2012, pp. 29-35.

[4] Anthony I. Wasserman, "Software Engineering Issues for Mobile Application Development", FoSER 10 Proceedings of the FSE/SDP workshop on Future of software engineering research, ACM, 2010, pp. 397-400.

[5] C. Chouhan, V. Shrivastava, and P. S. Sodhi, "Test Case Generation based on Activity Diagram for Mobile Application", International Journal of Computer Applications (IJCA), 2012, Vol. 57, No.23.

[6] J. Badlaney, R. Ghatol and R. Jadhwani, "An Introduction to Data-Flow Testing", NCSU CSC TR-2006.