



A Functional Testing Modeling for Enhanced Software Testing

Jalil Khojasteh¹, Tawfik Saeed Zeki², Hamid Reza naji³, Hamid sanatnama⁴

¹Department of Software Engineering, Islamic Azad University, U.A.E. Branch

²Department of Computer Engineering, Islamic Azad University, U.A.E. Branch

³Department of Computer Engineering, Islamic Azad University, U.A.E. Branch

⁴Department of Computer Engineering, Islamic Azad University, U.A.E. Branch

ABSTRACT

During the recent decays, software is more complicated than before. This means that finding errors in this software is very critical issue. According this, finding errors in complex software is a problem for Software testing team.[7] There are many testing techniques such as: Black and White box testing techniques which are most popular in testing. In this paper, we focus on white box testing, functional testing and merging Path testing and Data flow testing (two techniques of white box testing) to each other. The main objective of this research is designing and developing a model for functional testing and introducing a comprehensive algorithm in order to improve functional testing. A system test is selected and divided to some functions then it testes each function with this technique and generates some test cases. The results are reported based on the following 1.In this model functional testing is done more easily in complex software 2.The number of codes are less than before by using this model 3.If a software is large it is very hard to test all of test cases thus with this model we can find the best test cases and improve finding errors during software testing more easily than before.

Keywords: *software testing, White box Testing, Data flow testing, Path testing*

1. INTRODUCTION

Software testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item [1, 2].Goals of testing Identifying the correctness, completeness, security and quality of developed computer software; determining the status of the product during and after the build; Testing process must balance the written requirements, real-world technical limitations and user expectations [3].The testing process and its results must be repeatable and independent of the tester, i.e., consistent and unbiased [4].Software can be tested at many different testing levels, for example at levels of units, components of integration and of the whole system [5]. There are many techniques for testing .The most important testing techniques are Black and White box testing techniques. The paradigm of Black box testing states that test suites must be derived from the *specification* or the *documentation* of the component under test. In this kind of testing it is not used any information concerning the internals of the tested component. White box testing deals with the internal logic and structure of the code. White box testing also needs the tester to look into the code and find out which function / statement / chunk of the code is mal functioning .The main purpose is introducing a comprehensive model for finding errors in complicated software.

2. RESEARCH METHODOLOGY

In this research the steps for testing are:

1. Selecting a system test
2. Defining functions
3. Generating test cases

4. Selecting the best test case

For functional testing a system test case should be selected. The ATM system is a suitable system test case because it has many functions.[9] After selecting the system test, some functions should be defined according user requirements. Generating some test cases and selecting the best test case using our new algorithm that proposed.

3. PROPOSED ALGORITHM

When all functions designed in software. Each function should be tested by an algorithm. We proposed an algorithm that white box testing techniques for test each functions. Our algorithms have following steps:

1. Write code for each function in .net environment.
2. Drawing control flow graph.
3. Compute cyclamates complexity.
4. Used DPT testing technique as new technique.
5. Generate test cases.
6. Finding errors.

3.1 Writing Code

It is tried to write a code for each functions and run them with input data .In this part we select the function (check pin code) of an ATM system to demonstrate our proposed algorithm. After interrering the card into the ATM system the system asks for a pin code. If the pin code is valid the system gives access to other functionalities, otherwise the system sends a message to the customer about invalid pin.

```
PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    Dim k AsString
    Dim Id, I, max AsInteger
```



```

Id = Val(TextBox2.Text)
i = 1
maxrows = ds.Tables("pinno").Rows.Count
While i <> max
k= ds.Tables("pinno").Rows(i)("pin")
If Id =k Then
Me.Close()
Text1.text = ds.Tables("pinno").Rows(i)("acc_no")
Form3.Show()
ExitWhile
i += 1
If i = max Then
MsgBox("INVALID PIN number!!!")
ExitWhile
EndIf
EndIf
EndWhile
Endsub
    
```

3.2 Drawing Control Flow Graph

The code is simulated to flow chart by sections below: Lines (or arrows) called *edges* represent flow of control Circles called *nodes* represent one or more actions Areas bounded by edges and nodes called *regions* a *predicate node* is a node containing a condition.

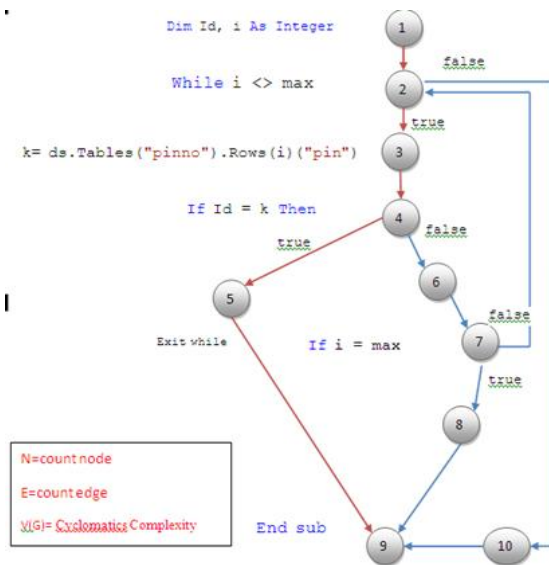


Figure 1: Control flow graph

P is the number of unconnected parts of the control flow graph as shown in figure1.

For flow graph above, cyclamates Complexity is:

$$V(G) = 12 - 10 + 2 * 1 = 4$$

That is upper bound for number of linearly independent paths.

3.4 Used DPT Testing (Data-path Testing) Technique as New Technique

Data flow-based testing method inters data and gives output .[10] This technique generate some test cases which can be used as input for path testing technique to produce paths. Selecting test paths of a program will be possible by the locations of the definitions and uses of different variables in a program. In this paper it is merged two techniques such as data flow tests and path testing. According to previous steps four paths should be tested.[8] In propose code we inter input data to determine output and generate some test cases.

The identified variables in this unit are:

- I: counter for count while statement;
- K: For each record in database we have a pin that should compare with id;
- Max: maximum record in database that we must search for find pin code if exist;
- Id: Data enter by customer as shown in table 1.

Table 1: Generating Test Case

| id | Data flow testing | | | Path testing | | | |
|----|-------------------|------|-----|--------------|-------------|-----------------|-----|
| | I | K | Max | id | output | Path | sum |
| 1 | 5 | 1111 | 5 | 2222 | Invalid pin | 1-2-10-9 | 4 |
| 2 | 3 | 1111 | 5 | 2222 | loop | 1-2-3-4-6-7-2 | 7 |
| 3 | 2 | 1111 | 5 | 1111 | 1111 | 1-2-3-4-5-9 | 6 |
| 4 | 4 | 1111 | 5 | 2222 | Not exist | 1-2-3-4-6-7-8-9 | 8 |

In this table there are four paths and the best test cases should be selected .In next step a relationship between nodes and edges is introduced which can find the best test cases.

3.5 Generate Test Cases

Using new technique by entering input and determine output we can generate some test cases by composing paths test and data.[6].

$$V(G) = \text{edges} - \text{Nodes} + 2p$$



3.6 Finding Errors

In this part the best test cases should be selected and finding errors for each functions .There are some variables such as:

N=nodes;

E=edges;

V (G) = cyclamates Complexity;

And Sum which is number of nodes.

Value of these variables in the path of figure1 are N=10; E=12 In step 4, 3 Cyclamates complexity is computed which is equals to 4.

Finally a relation between nodes and edges is introduced:

$$Sum = (N+E-V)/(v-1);$$

$$Sum = (10+12-4)/(4-1) =6;$$

For each test case the sum of the nodes in each path is computed and puts in the last column of table1.

To find the result of the relationship above \varnothing this number should be compared with all numbers in columns. The number shows the best test case for testing.

4. MODELING TESTING TECHNIQUE

In this model as shown in figure 2, after applying the steps in proposed algorithm, Cyclamates complexity will be computed in order to know how many test cases are exist. By this model tester can easily find the best cases and errors during software testing process.

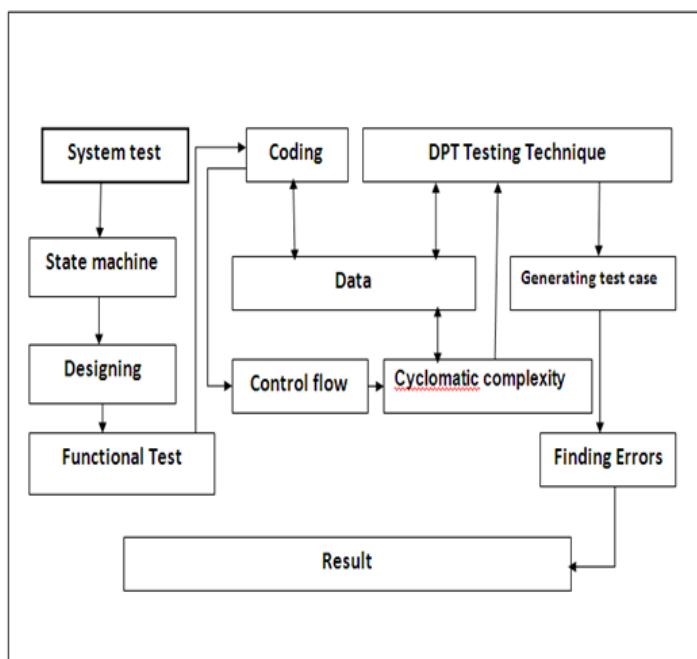


Figure2: Functional Testing Modeling

5. CONCLUSION AND FUTURE WORK

In this paper we presented a model for testing process based on white box techniques .The tester can generate some test cases according to data flow testing and path testing by using this approach. Selecting the best test cases is another objective in this research that we found it with a relationship between edges and nodes. The proposed model can improve writing and testing of programs. The structure of the program is systematic with this model and if the programmer wants to change the program and finds errors, he can do it easily.

Future work is about Internet paths and failures due to software, hardware, and configuration errors often disrupt packet delivery on the Internet.

Many approaches to avoiding these problems use multiple paths between two network locations. These approaches rely on a path independence assumption in order to work well.

REFERENCES

- [1] IEEE, "ANSI/IEEE Standard 1008-1987, IEEE Standard for Software Unit Testing," no., 1986.
- [2] IEEE, "IEEE Standards Collection: Glossary of Software Engineering Terminology," IEEE Standard 610.12-1990, 1990.
- [3] Quadri, S., and Sheikh Umar Farooq. "Software Testing–Goals, Principles, and Limitations." *International Journal of Computer Applications IJCA* 6.9 (2010): 34-41.
- [4] Edward L. Jones —Grading student programs – a software testingl, Proceedings of the fourteenth annual Consortium for Computing Sciences in Colleges, 2000.
- [5] Mark Utting and Bruno Legeard. Practical Model-Based Testing: A Tools Approach.Morgan Kaufmann,2007.
- [6] Ahmed s Ghiduk . A New Software Data- Flow Testing Approach via Ant Colony Algorithms. *Universal Journal of Computer Science and Engineering Technology* 1 (1), 64-72, Oct. 2010.
- [7] Roya Alav and Shahriar Lotfi. The New Approach for Software Testing Using a Genetic Algorithm Based on Clustering Initial Test Instances. *International Conference on Computer and Software Modeling* 2011.
- [8] Maha Alzabidi and Ajay Kumar . Automatic Software Structural Testing by Using Evolutionary Algorithms for Test Data Generations. *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.4, April 2009.



[9] Yingx wang and Yanan Zhang . The Formal Design Model of an Automatic Teller Machine (ATM).International Journal of Software Science and Computational Intelligence, 2(1), 102-131, January-March 2010

[10]Ruchika Malhotra and Mohit Garg . Empirical Validation of an Efficient Test Data Generation Algorithm Based on Adequacy based Testing Criteria. Software Engineering : An International Journal (SEIJ), Vol. 2, No. 1, March 2012,