ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 2, Ver. 2 (April - June 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Fuzzy Theory in Black Box Testing

Vishal Chandra
Affiliated to SGVU, Jaipur, Rajasthan, India

Abstract
In software development industry the most crucial stage or phase is software testing. After coding phase or stage testing phase starts. Testing uncover the faults in coding phase. There are broadly two types of testing techniques under which all testing concern they are white box and black box testing. In white box tester has full knowledge of codes and their functionality while in black box testing testers have only knowledge of input and corresponding output. Testers carefully design test cases so that they can analyze all functionality. Black box testing analyze functional requirement in the software. It is also known as blind testing. This research paper uses fuzzy theory in black box testing. Fuzzy is branch of mathematics in which there is no rigid boundaries. In traditional crisp set of simple set theory elements are either inside the set of not in the set but in fuzzy theory every element has degree of existence. They have their own membership function in other words degree of belonging. By using fuzzy there will be flexibility in testing methods. Before it testing is done by rigid values either acceptable or not acceptable. After completion of this paper one can determine the degree of test result in other words tester can determine how much the test result is acceptable or not acceptable their degree of existence. This paper used four criteria of testing they are graph based testing, boundary value analysis, equivalence class partition and orthogonal array testing. This paper applies fuzzy theory in all mentioned testing phases. If someone asks when testing is over the reply will never each time software is execute the testing is done.

Keywords
Black box testing, fuzzy theory, fuzzy logic, equivalenceclass partition, boundary value analysis, graph based testing, orthogonal array testing.

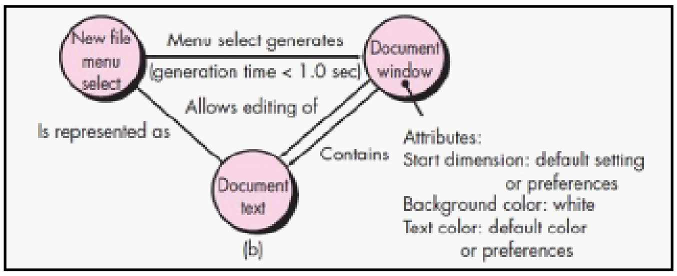
I. Introduction
Black box testing is also known as behavioral testing. In this type of testing we do not have idea of internal coding and also the algorithms. It is also known as blind testing. Testers only have to observe the behavior of software after feeding in put to the software. Software testing is very expensive and very laborious and also time consuming. So testing team has to decide test cases in such a way that it will be more economic and time saving. Black-box testing tries to find out errors in the following categories:(1) behavior or performance(2) missing or incorrect functions, (3) interface errors, (4) errors in datastructures (5) access of external databases, errors, and (6) initialization and termination errors

The major four approaches for black box testing are
1. Graph based testing
2. Boundary value analysis
3. Equivalence class partition
4. Orthogonal array testing

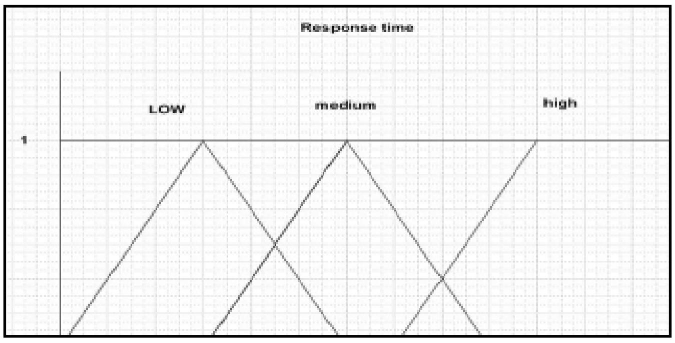
Test cases are determined to answer the following questions:
1. How software‘sfunctional validity tested?
2. How are system performance and behavior tested?
3. Whichclasses of input make good test cases?
4. Is the system is sensitive to certain input values?
5. How boundaries are of a data class isolated?
6. How much data rates and data volume the systemcan tolerate?

II. Goal based testing
Goal based testing is consider objects and relationship between objects. In graph based approach there are circle they refer to objects and their relationship is denoted by arrows. Arrows may be unidirectional, bidirectional, parallel. Each type of arrow has own importance and meaning. Arrows has also weight. They also have to use response time. This paper focuses response time. Because this is most important part of any software how fast the software executes

.

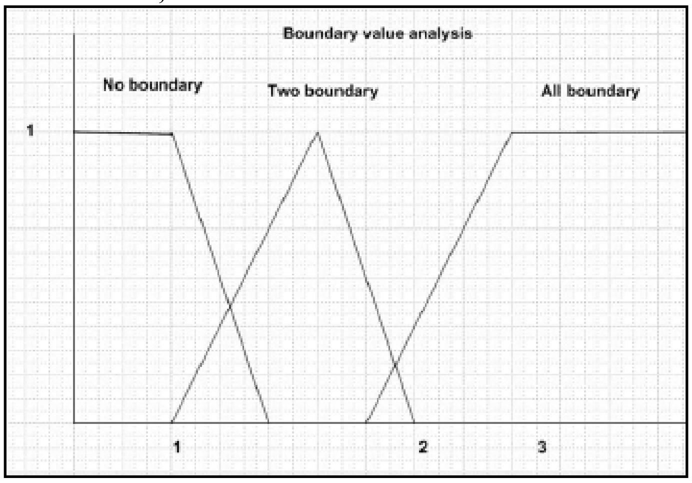

New file menu select
Menu select generates (generation time < 1.0 sec)
Document window
Allows editing of
Is represented as
Contains
Document text
Attributes:
Start dimension: default setting or preferences
Background color: white
Text color: default color or preferences
(b)

This paper uses response time of any software during testing. in this paper there are three linguistic variables linguistic variable {low, medium, high}
And their corresponding membership graph will be



If software has 0 to .6 sec response time then its low response time
If software has .6 to .8 sec response time then its response time is medium
If software has response time greater than 1 then its response time is so high
Most of errors occur at the boundaries of the input values rather thanin the “middle or center.” It is for this reason that boundary value analysis has been developed. The Boundary value analysis tries to a selection of testcases such that attempting bounding values.The Boundary value analysis is the test-case design

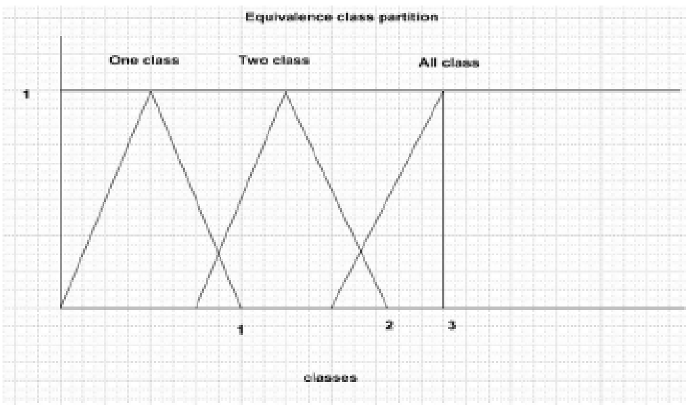
technique that is equivalence classpartitioning. Selecting any element of an equivalence classes, BVAleads to the selection of those test cases at the "edges" of the class., BVA derives test cases from the output values as well[Mye79].
Fuzzyfication of boundary value analysis
There are three linguistic variables {no boundary, two boundaries, all boundaries}



If during testing no boundary is verified then software is rejected
If during testing two boundary are verified then it is acceptable (minor)
If all boundaries are verified then software is completely acceptable

III. Equivalence class partition

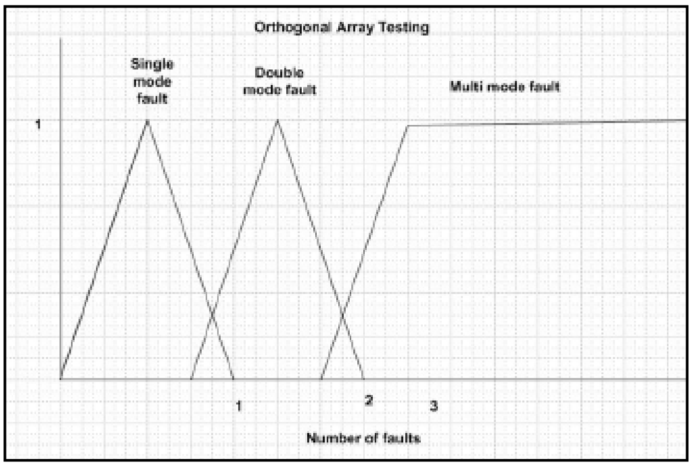
Equivalence class partitioning is a black-box software testing technique that divides the input values of a program into classes of data from which test cases can be derived. In an ideal test case single-handedly uncovers a class of errors else that might require many test cases to be executed before the most general error is observed. Test-case design for equivalence partitioning is based on an evaluation of equivalence classes partitioningfor an input condition. If a set of objects can be linked by relationships that are transitive, symmetric, and reflexive, an equivalence class is present [Bei95].
This partitioning is done such a way that the behavior of theprogram is much more similar for each and every input data belonging to the same equivalence class.The main idea behind definition of the equivalence classes is that testing of the written codewith any one value belonging to an equivalence class is full fill the testing thesoftware with any other value belonging to that equivalence class. Equivalenceclass partitioning for software can be designed by examining the output data and input.



Fuzzification of equivalence class partition
Linguistic variables {one class, two classes, all classes}
If classes between 0 and 1 then it is one class
If classes between 1 to 2 the it means it satisfy two classes
If more than two classes are verified then it is all classes

IV. Orthogonal array testing

There are many cases in software applications in which the input values are relatively limited. The number of input parameters is small and the values that each of the parametermay take clearly bounded. When these numbers are very small (e.g., four inputparameters taking on four discrete values each), it is notpossible to consider everyinput permutation and exhaustively test the input domain. As the numberof input values grow and the number of discrete values for each data item increases,exhaustive testing becomes impractical or impossible.Orthogonal array testing may be applied to those problems in which the input values isrelatively small but too large to accommodate exhaustive testing. The orthogonalarray testing method is particularly useful in finding region faults—an error categoryassociated with faulty logic within a software component.
The difference between orthogonal array testing and more conventional"one input item at a time" approaches, consider a system that has four inputitems, W,X, Y, and Z. Each of these input items has three discrete values associated withit. There are 4*3*2*1=256 possible test cases it is considerably big.
Fuzzy orthogonal array testing
If one fa ult is detected then it is single mode faul if two faults are deteced then it is dual mode
If faults are more than two then it is multi mode
Linguistic variable {single mode, double mode, multi mode}



IV. Conclusion

On the basis of this paper it is conclude that it can be used fuzzy theory in software engineering specially in testing. It shows the flexible testing methods. It shows the degree of acceptability in testing of software. All black box techniques are same as it is, but by this paper I added the feature of fuzzy. This paper makes testing more intuitive and more automated. It makes testing realistic like human thinking between beyond two rigid boundary of yes and no or 1 and 0

V. Future scopes

I have written this paper all major number of black box testing techniques. It might be possible that we can add some more techniques. There are various factors which affect black box testing. Black box testing is done by human not by computer. Using this paper we can make it automated. In future this type of testing is use to get more flexible result. With the help of this testing mechanism will be easy and flexible.

References

[1]. Arya, S.P. and Hazarika, D., Functions with closed fuzzy graph, J. Fuzzy Math. 2:593-600, 1994.

[2]. Bezdek, J.C. and Harris, J.D., Fuzzy partitions and relations an axiomatic basis for clustering, Fuzzy Sets and Systems 1:111-127, 1978.

[3]. Bhattacharya, P., Some remarks on fuzzy graphs, Pattern Recognition Letters 6:297-302, 1987. 62 2. FUZZY GRAPHS

[4]. Bhutani, K.R., Onautomorphisms of fuzzy graphs, Pattern Recognition. Letters 9:159-162, 1989.

[5]. Cerruti, U., Graphs and fuzzy graphs, Fuzzy Information and Decision Processes 123-131, North-Holland, Amsterdam-New York, 1982.

[6]. Chen, Q. J., Matrix representations of fuzzy graphs (Chinese), Math. Practice Theory 1:41-46, 1990.

[7]. Delgado, M. and Verdegay, J.L., and Vila, M.A., On fuzzy tree definition, European J. Operational Res. 22:243-249, 1985.

[8]. Delgado,lbl. andVerdegay, J.L., On valuation and optimization problems. In fuzzy graphs: A general approach and some particular cases, ORSA J. on Computing 2:74-83, 1990.

[9]. Ding, B., A clustering dynamic state method for maximal trees in fuzzy graph theory, J. Numer. Methods Comput. Appl. 13:157-160, 1992.

[10]. Dodson, C.T.J., A new generalization of graph theory, Fuzzy Sets and. Systems 6:293-308, 1981

[11]. Software engineering ,rajeeb mall

[12].Software engineering a practitioner's approach,R. pressman..

Profile

Name: - Vishal ChandraAcademic qualification: -B.Tech (CSE), M.Tech (artificial intelligence)(Pursuing)