Test Case Design using Black Box Testing Techniques for Data Mart

Payal Pahwa
Bhagwan Parshuram Institute of Technology
I.P University, Delhi

Renu Miglani
GVM Institute of Technology and Management
DCRUST Murthal, Sonepat

ABSTRACT
Data Warehouse is a logical stockroom which accumulates and maintains enormous volume of data. It is very important for the enterprise that needs to analyze data obtained from heterogeneous sources to take tactical decisions. Any enterprise that wants to expand, survive and beat out the competition must have control over data. Any error or inconsistency in the data may lead to improper decision which may cause immense losses to enterprise. Data Warehouse testing is carried out to eliminate the errors and inconsistencies that arise due to data being collected from desperate sources in different formats. Data Warehouse testing is too expensive and time consuming practice as exhaustive testing is not possible. Therefore the concept of Data Mart comes into existence. Data Mart is a specialized subset of Data Warehouse which fulfills the data requirement of a specific group. Testing of a Data Mart is much easier and manageable process as compared to testing of a Data Warehouse. In order to test the data mart, there are a number of strategies that allow us to select a set of test cases and test data which are very effective in detecting errors. In our paper we have discussed black box and white box testing techniques concerning Data Mart in brief. We have explored black box techniques to select test cases as they have systematic approach to uncover a great number of errors. We have also proposed to design the test cases using Boundary Value Analysis, Equivalence Class Partitioning and the combination of both techniques. The test cases designed using these techniques are very successful to detect previously undetected errors or faults in a Data Mart. These are also proficient for testing the worst case and robustness of Data Mart.

Keywords
Data Warehouse, Data Mart, Data Mart testing, Black box testing technique, White box testing technique, Equivalence class partitioning, Boundary value analysis

1. INTRODUCTION
A data warehouse is a subject-oriented, time-variant, integrated and non-volatile collection of data in support of management's decision-making process[1][2]. It is a composite and collaborated data model that captures the entire data of an organization[3].Whereas data mart is a decision support system incorporating a subset of the enterprise's data focused on specific functions or activities of the enterprise[4]. This data may contain any inconsistency or error. The use of this erroneous data may affect the critical decision, strategic objectives or even harm the organization. Therefore testing the quality of the resulting information will support the trust worthiness of the system[5].Testing is a set of activities planned in advance having a series of steps containing specific test case design techniques and methods to test Data Mart in a systematic way. It is expected to test the Data Mart's response for every possible valid and invalid inputs. This process is extreme lengthy, costly, needs enormous time and even not possible. Thus to achieve our objective we settle something equivalents to complete testing. We test those critical and sensitive areas where probability of occurring a fault is maximum. Organizations develop strategies and policies to design test cases for detecting faults especially from these areas to make testing effective. These test cases are very much valuable and useful thus they need to be developed, reviewed, used, managed and saved[6]. Tests should have highest likelihood of finding the maximum errors using least efforts, cost and time. A rich set of techniques are available having systematic approach to design test cases. Two most common techniques are White-box and Black-box testing techniques. There are a number of strategies related to these techniques that can be used to design test cases which are very effective in discovering errors. We have discussed the Equivalence Class Partitioning, Boundary Value Analysis and the combination of both in our paper. These strategies provide a mechanism to ensure the completeness of tests and offer the highest probability for uncovering the errors in Data Mart. The test cases designed using combinations of strategies are very effective to uncover previously concealed errors or faults in a Data Mart and they are also proficient for testing the worst case and robustness of Data Mart.

The remainder of the paper is organized as follows: Section 2 describes the Data Mart testing techniques in brief stated as White-box and Black-box testing techniques. Section 3 explores the test case design strategies and proposes to design test cases using equivalence class partitioning, boundary value analysis and the combination of both as it becomes a valuable technique that produces a comparatively effective set of test data encompassing great likelihood of recognizing maximum faults. Section 4 illustrates these strategies to manipulate the data maintained as subset of an online banking application. Finally the summarized work has been concluded in section 5.

2. DATA MART TESTING TECHNIQUES
There are primarily two techniques which can be used to design test cases for testing Data Mart. This section describes these techniques in brief.

2.1 White-Box Testing Techniques
White box testing is a test case design method that uses the control structure of the procedural design to derive test cases[7]. This approach examines the inner organization of Data Mart and derives test data related with internal logic. Test cases are developed on the basis of procedural detail aimed to exercise the internal data structure exhaustively to ensure their validity. Tests are executed to confirm that all in-house components of Data Mart have been adequately exercised and operations are accomplished according to specification. This approach is known as white-box testing

technique. It encapsulates testing of Data Mart functions, triggers, logical views, and queries etc. which support refactoring. It checks rules of referential integrity and validates data models, database tables, schema and consistency etc. The common strategies used in this technique are basis path testing, structural testing, logic-based testing, fault-based testing, cyclomatic complexity etc.

2.2 Black-Box Testing Techniques

Black box testing is a test case design strategy that is used to ensure that each specified function of Data Mart is fully operational. They also search errors in each function. Tests are conducted at user interface to demonstrate that input is properly accepted and the resulting information is correctly produced as well as the integrity of Data Mart is sustained. The Data Mart functionality is tested with regards to its specifications and context. The accurate input/output relationship is verified thoroughly without concerning the interior organization of Data Mart. Black box testing encapsulates integration and interface testing of Data Mart which are accomplished using activities like data mapping, verification of incoming and outgoing data etc. Various strategies including in this category are equivalence partitioning, boundary-value analysis, cause-effect graph, comparison testing etc.

All the strategies mentioned above are very effective in designing test cases and detecting errors. We have discussed the Equivalence Class Partitioning, Boundary Value Analysis and the combination of .both in detail in the next section.

3. TEST CASE DESIGN STRATEGIES

The art of testing is to design a small, manageable set of test cases which are capable to discover maximum errors while minimizing the redundancy amongst of the cases. A lot of strategies related with above mentioned techniques can be used to design test cases which are capable of finding anomalies in the Data Mart effectively. Some of them are described in the remainder of this section.

3.1 Equivalence Class Partitioning

This black box testing strategy partition the input domain of a Data Mart into finite number of equivalence classes of data for which test cases can be derived. The input domain can be very big. The main goal of domain testing methods is to achieve a test suite, the size of which is considerably smaller than the count of all inputs of the programs, and which effectively reveals failures of the program as much as possible[8]. This method predicts that the test of a representative value of each class is as same as to the test of any other value of the same class. Therefore it anticipates that if a test case in a class discovers an error would be same as other test cases are finding. Conversely if a test case is unsuccessful in finding any error in a class, it predicts that no other test case would be able to find the error in the same class. It attempts to define a test case that discovers classes of errors that might otherwise require several cases to be executed, thus reducing the total number of test cases, efforts and time. Test cases are designed on the basis of evaluation of equivalence classes for an input condition.

3.1.1 Test case designing using Equivalence Class Partitioning

Equivalence classes are designing by evaluating each input condition and partitioning accordingly it into valid and invalid classes. Now these classes are used to generate test cases covering all the valid and invalid classes.

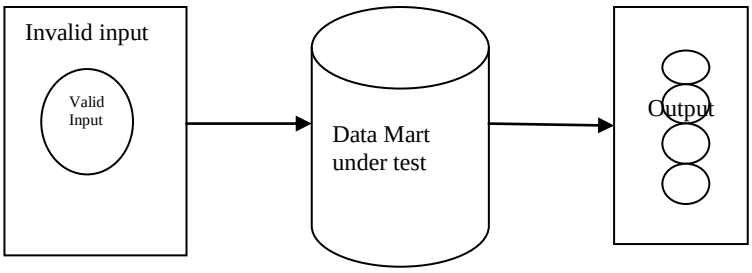

Fig 1: Equivalence Partitioning

Equivalence classes are defined on the basis of input conditions and represent a set of valid and invalid values. Input condition may be a range of values, a specific numeric value, a set of related values or a Boolean condition. Equivalence classes regarding testing of Data Mart for all these condition will be defined as mentioned below.

3.1.2 Procedure to design Equivalence classes

Suppose m is the representative member of any specific equivalence class then value of m will be specified as mentioned below

If an input condition indicates a range of values from x to y, then one valid, two invalid equivalence classes and three test cases will be identified.

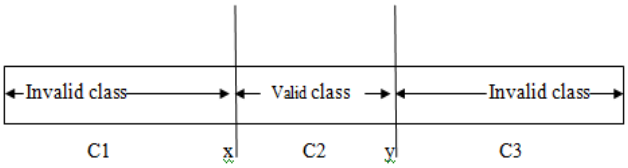

Fig 2: Equivalence class boundaries

Table1. Equivalence Class based on range of values

Test case	Equivalence class	Value of m	Class status
T1	C1	m<x	invalid class
T2	C2	x<m<y	valid class
T3	C3	m>y	invalid class

If an input condition indicates a specific numeric value x then one valid, two invalid classes and three test cases will be defined

Table 2. Equivalence class based on specific values

Test case	Equivalence class	Value of m	Class status
T1	C1	m<x	invalid class
T2	C2	m=x	valid class
T3	C3	m>x	invalid class

If an input condition indicates a member of a set S, one valid and one invalid class will be identified

Table 3. Equivalence class based on set of values

Test case	Equivalence class	Value of m	Class status
T1	C1	$m \in S$	valid class
T2	C2	$m \notin S$	invalid class

If an input condition is Boolean, one valid, one invalid class and two test cases will be identified

Table 4. Equivalence class based on Boolean values

Test case	Equivalence class	Value of m	Class status
T1	C1	True	valid class
T2	C2	False	invalid class

The idea of designing test case is to choose at least one member from each equivalent class. The number of test cases can be derived from the above mentioned input conditions. It is also possible to design equivalence classes for output domains. So we can derive test cases considering both input and output domains using equivalence class partitioning strategy.

3.2 Boundary Value Analysis

Boundary value analysis devotes special attention to boundaries of equivalence classes, because praxis shows that boundary values often reveal faults. Since the boundary between two partitions is the place where the behavior of the application changes. Suppose, programmers may improperly use < instead of <= or conversely <= instead of<.[5] For that reason the test case designer selects values at the extremes of the classes instead of choosing random values to design the test cases for testing Data Mart. Boundary value analysis is a test case design methods that compliments to equivalence class partitioning and leads to the selection of test cases at the edges of the class which exercise the bounding values. This technique also enforces to design test cases for the extreme conditions in output.

3.2.1 Test case designing using Boundary Value Analysis with single fault assumption

Single fault assumption predicts that failures are seldom the result of the simultaneous occurrence of two or more faults[6]. Thus it considers extreme value for only one input variable. Using single fault assumption theory, 4n+1 test cases will be designed, where n are number of input variables.

3.2.2 Boundary value analysis for range of values

Suppose we have range of values defined as $x_1, x_2, x_3, \ldots\ldots, x_n$, where boundary values are x and y such that $x=x_1$ and $y=x_n$. If an input condition with one input variable between the range of values from x to y then test cases will be defined as under

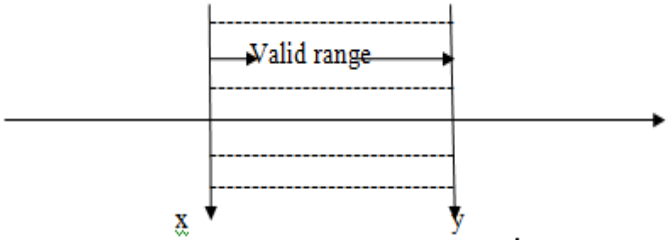

Fig 3: One dimension boundary values

Table 5. Boundary value analysis based on range of values with one input variable

Test case	Record	Record status
T1	x	Boundary value
T2	x+1	Adjacent to boundary value
T3	x_i	A nominal value
T4	y-1	Adjacent to boundary value
T5	y	Boundary value

Suppose we have an input condition with two input variables x and y such that a<=x<=b and c<=y<=d, then test cases will be defined as under.

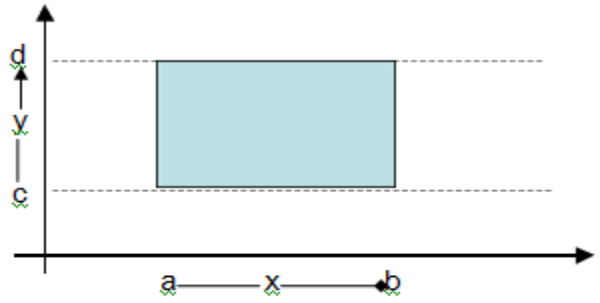

Fig 4. Two dimension boundary values

Boundary value analysis test cases can be accomplished by holding the values of all but one variable at their nominal values, and letting the other variable assumes its extreme values.

Table 6. Boundary value analysis based on range of values with two input variable

Test case	Record	Record status
T1	xnom,ymin	Boundary value
T2	xnom,ymin+	Adjacent to boundary value
T3	xnom,ynom	A nominal value
T4	xnom,ymax-	Adjacent to boundary value
T5	xnom,ymax	Boundary value

T6	xmin,ynom	Boundary value
T7	xmin+,ynom	Adjacent to boundary calue
T8	xmax-,ynom	Adjacent to boundary value
T9	xmax,ynom	Boundary value

Test cases should also be designed to exercise the internal data structure at its boundary as well as for output conditions.

4. BOUNDARY VALUE ANALYSIS OF EQUIVALENCE CLASSES

Equivalence class along with boundary value analysis is a valuable combination strategy that generates a comparatively effective set of test data having high probability of identifying maximum faults. Combination strategies are test-case selection methods where test cases are identified by combining values of the different test object input parameters based on some combinatorial strategy[9]. When these two strategies are combined, they discover additionally some more errors that are different from previously detected errors. Thus this testing strategy is complimentary. If a test case exercises an equivalence class on or just one side of its boundary, the probability of detecting an error increases.

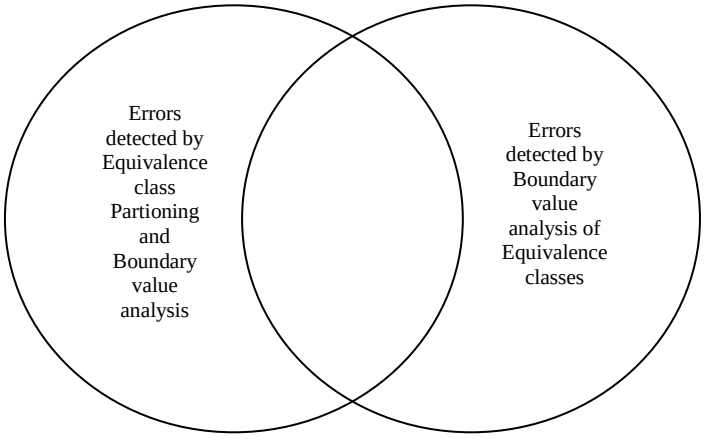

Fig 5. Complimentary strategies

When we extend the boundary value analysis by exceeding the extreme values slightly greater than the maximum and a value slightly less than minimum, it will test the robustness of the Data Mart as the input data goes outside the legitimate boundary of input domain. Thus total test cases in robustness testing are 6n+1 according to n input variables. Therefore, while testing the data mart using Boundary value analysis of equivalence classes having one input variable, the following test cases can be selected.

Table 7. Boundary value analysis of equivalence classes with one input variable

Test case	**Record**	**Record Status**
Test case 1	x-1 record	Adjacent to boundary value and member of equivalence class 1
Test case 2	x record	Boundary value and member of equivalence class 2
Test case 3	x+1 record	Adjacent to boundary value and member of equivalence class 2
Test case 4	xi record	Any random member of equivalence class 2
Test case 5	y-1 record	Adjacent to boundary value and member of equivalence class 2
Test case 6	y record	Boundary value and member of equivalence class 2
Test case 7	y+1 record	Adjacent to boundary value and member of equivalence class 3

Boundary value analysis of equivalence classes with two input variable using single fault assumption, the following test cases can be selected.

Table 8. Boundary value analysis of equivalence classes with two input variables

Test case	**Record**	**Record status**
T1	xnom, ymin-	Adjacent to boundary value
T2	xnom, ymin	Boundary value
T3	xnom, ymin+	Adjacent to boundary value
T4	xnom, ynom	A nominal value
T5	xnom, ymax-	Adjacent to boundary value
T6	xnom, ymax	Boundary value
T7	xnom, ymax+	Adjacent to boundary value
T8	xmin-, ynom	Adjacent to boundary value
T9	xmin, ynom	Boundary value
T10	xmin+, ynom	Adjacent to boundary value
T11	xmax-, ynom	Adjacent to boundary value
T12	xmax, ynom	Boundary value
T13	xmax+, ynom	Adjacent to boundary value

The above stated guidelines should be applied for input and output domain both to develop effective and efficient test cases.

When we reject the single fault assumption theory by inputting extreme values to more than one variables, it becomes Worst-case testing. For Worst-case testing 5^n test cases can be designed for n input variables.

5. CASE STUDY

We are considering the example of an online banking application whose data can be accessed through a card. This card can be used on a bank machine as well as on an Automatic Teller Machine. We can withdraw cash, check account balances, deposit money and receive a copy of our statement electronically by using the card and the password related with our account, which is called Personal Identification Number(PIN). PIN is a security code, which has been designed to protect the card account as a safeguard against unauthorized use. The PIN offers additional security for the account and must be used while making transactions through card. It is considered as our electronic signature. The most important and secured part of the card is Magnetic stripe. It stores the card number, card holder's name, expiry date, along with other bank-specific information. The user can manipulate the data through machine by swiping the card in front of its sensor, provides a four digit password and follows through a sequence of typed commands that activates various banking functions. During the log-on activities, the online application software may accept data in the form:

MII(Measure industry identifier) code – first digit of the card number

IIN (Issuer Identification Number) code – first six digit of the card number including MII

Account number –six digit number not beginning with zero

Check digit – embedded in the card number

Password – four digit number or alphanumeric string

Commands – deposit, check, bill pay etc.

The input condition related with each data element for the online banking application system may be specified as mentioned below

Table 9. Various data inputting format

Data element	Input condition	Values
MII	Range	Between 0 and 9
IIN	Range	Between 0 and 999999
Account number	range	Between 100000 and 999999
Check digit	Boolean Specific value	1 Specific single digit number
Password	Boolean Specific value	1 Specific four character string
Commands	Set	Commands related with system

Derivation of equivalence classes using proper guidelines helps to develop test cases for each input domain data item. Test cases are selected to exercise the maximum number of attributes of an equivalence class at once.

6. CONCLUSION

In this paper we empirically evaluate equivalence class partitioning, boundary value analysis and the combination of both the techniques from the test case generation point of view. The test cases designed using these techniques are very successful to detect previously undetected errors or faults in a Data Mart. The test cases designed using combinations of strategies are very effective to uncover previously concealed errors or faults in a Data Mart and they are also proficient for testing the worst case and robustness of Data Mart. These strategies provide a mechanism to ensure the completeness of tests and offer the highest probability for uncovering the errors in Data Mart. In future various data warehouse testing techniques can be analyzed for their applicability in data mart testing.

7. REFERENCES

[1] Eiad Basher Alhyasat, Al-Salt, Jordan and Mahmoud Al-Dalahmeh, "Data Warehouse Success and Strategic Oriented Business Intelligence: A Theoretical Framework", Journal of Management Research, Vol. 5, No. 3, 2013.

[2] W. H. Inmon, "Building the Data Warehouse", John Wiley and Sons, Second Edition, March 27, 1996.

[3] Manoj Philip Mathen, " Data Warehouse Testing", Developer IQ Magizine, Infosys Technologies Limited, Mar 2010,

[4] Payal Pahwa and Renu Miglani, "Domain Level Analysis of Data Mart Quality Factors along with Best-fit Tests to minimize defects", International Journal of Engineering Sciences Paradigms and Researches, Vol. 02, Issue 01, December 2012.

[5] Neveen ElGamal, Ali, El BastaWissy and GalalGalal-Edeen,"Towards a Data Warehouse Testing Framework" , Ninth International Conference on ICT and Knowledge Engineering, 2011.

[6] Rajib Mall, Fundamentals of Software Engineering, PHI, 2nd Edition, 2008.

[7] Mohd. Ehmer Khan , Farmeena Khan "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques ", International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.

[8] Vineta Arnicane, " Complexity of Equivalence Class and Boundary Value Testing Methods", Computer Science and Information Technologies" ,Vol 751, 2009.

[9] Mats Grindal, Jeff Offutt, Sten F. Andler, "Combination Testing Strategies: A Survey", GMU Technical Report ISE-TR-04-05, July 2004.