Black Box testing

Outline

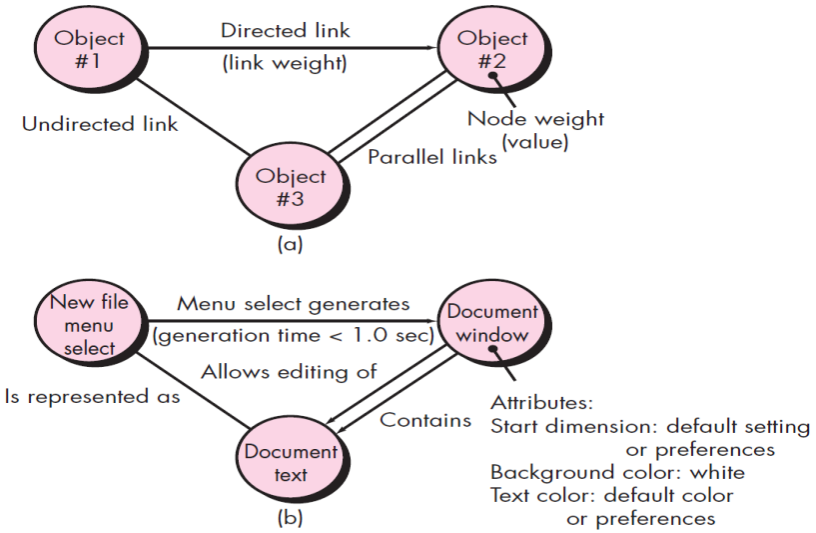
- What is black box testing ?
- Black-box techniques
  - Graph based testing method
  - Equivalence partitioning
  - Boundary value analysis
  - Orthogonal array testing.

Black box testing

- Black-box testing, also called behavioral testing, focuses on the functional requirements of the software.
- Black-box testing attempts to find errors in the following categories:
  - (1) incorrect or missing functions,
  - (2) interface errors,
  - (3) errors in data structures or external database access,
  - (4) behavior or performance errors, and
  - (5) initialization and termination errors.
- Applying black-box techniques, let to derive a set of test cases.
- Graph based testing method, equivalence partitioning, boundary value analysis, orthogonal array testing.

Graph based testing method

- Software testing begins by creating a graph of important objects and their relationships and then devising a series of tests that will cover the graph so that each object and relationship is exercised and errors are uncovered.
- The symbolic representation of a graph.
  - Nodes are represented as circles
  - Arrow represents
    - A *directed link in only* one direction.
    - A *bidirectional link, also called a symmetric link, implies that the relationship* applies in both directions.
    - *Parallel links are used when a number of different* relationships are established between graph nodes.
- Derive test cases by traversing the graph and covering each of the relationships. These test cases are designed in an attempt to find errors in any of the relationships.



Equivalence Partitioning

- Equivalence partitioning is a black-box testing method that divide the input data of software into different equivalence data classes. Test cases are designed for equivalence data class.
- A use of this method reduces the time necessary for testing software using less and effective test cases.
- Guidelines:
  - **1. If an input condition specifies a range, one valid and two invalid equivalence** classes are defined.
  - **2. If an input condition requires a specific value, one valid and two invalid** equivalence classes are defined.
  - **3. If an input condition specifies a member of a set, one valid and one invalid** equivalence class are defined.
  - **4. If an input condition is Boolean, one valid and one invalid class are defined.**
- ***Example :*** ***Test cases for input box accepting numbers between 1 and 1000 using Equivalence Partitioning:***
  - *One input data class with all valid inputs. Pick a single value from range 1 to 1000 as a* ***valid input data class test case.***
  - *Input data class with all values below lower limit. I.e. any value below 1, as a* ***invalid input data class test case.***
  - *Input data with any value greater than 1000 to represent third* ***invalid input data class test case****.*

Boundary value analysis

- Boundary value analysis (BVA) has been developed as a testing technique that leads to a selection of test cases that exercise bounding values.
- BVA derives test cases from the output domain.
- Guidelines for BVA :
  - 1. If an input condition specifies a range bounded by values *a and b, test cases* should be designed with values *a and b and just above and just below a and b.*
  - 2. If an input condition specifies a number of values, test cases should be developed that exercise the minimum and maximum numbers. Values just above and below minimum and maximum are also tested.
  - 3. Apply guidelines 1 and 2 to output conditions. For example, assume that a temperature versus pressure table is required as output from an engineering analysis program. Test cases should be designed to create an output report that produces the maximum (and minimum) allowable number of table entries.
  - 4. If internal program data structures have prescribed boundaries (e.g., a table has a defined limit of 100 entries), be certain to design a test case to exercise the data structure at its boundary.

Orthogonal array testing

- Orthogonal array testing can be applied to problems in which the input domain is relatively small but too large to accommodate exhaustive testing.
- The orthogonal array testing method is particularly useful in finding region faults—an error category associated with faulty logic within a software component.
- Consider, the *send function for a* fax application. Four parameters, P1, P2, P3, and P4, are passed to the *send function.* Each takes on three discrete values. For example, P1 takes on values:
    - P1 =1, send it now
    - P1 = 2, send it one hour later
    - P1 = 3, send it after midnight
  - P2, P3, and P4 would also take on values of 1, 2, and 3, signifying other send functions.
- Normally, the number of test cases required is $3^4 = 81$. In L9 orthogonal testing strategy, the following sequence of tests (P1, P2, P3, P4) would be specified: (1, 1, 1, 1), (2, 1, 1, 1), (3, 1, 1, 1), (1, 2, 1, 1), (1, 3, 1, 1), (1, 1, 2, 1), (1, 1, 3, 1), (1, 1, 1, 2), and (1, 1, 1, 3).

Test case	Test parameters			
	P1	P2	P3	P4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1