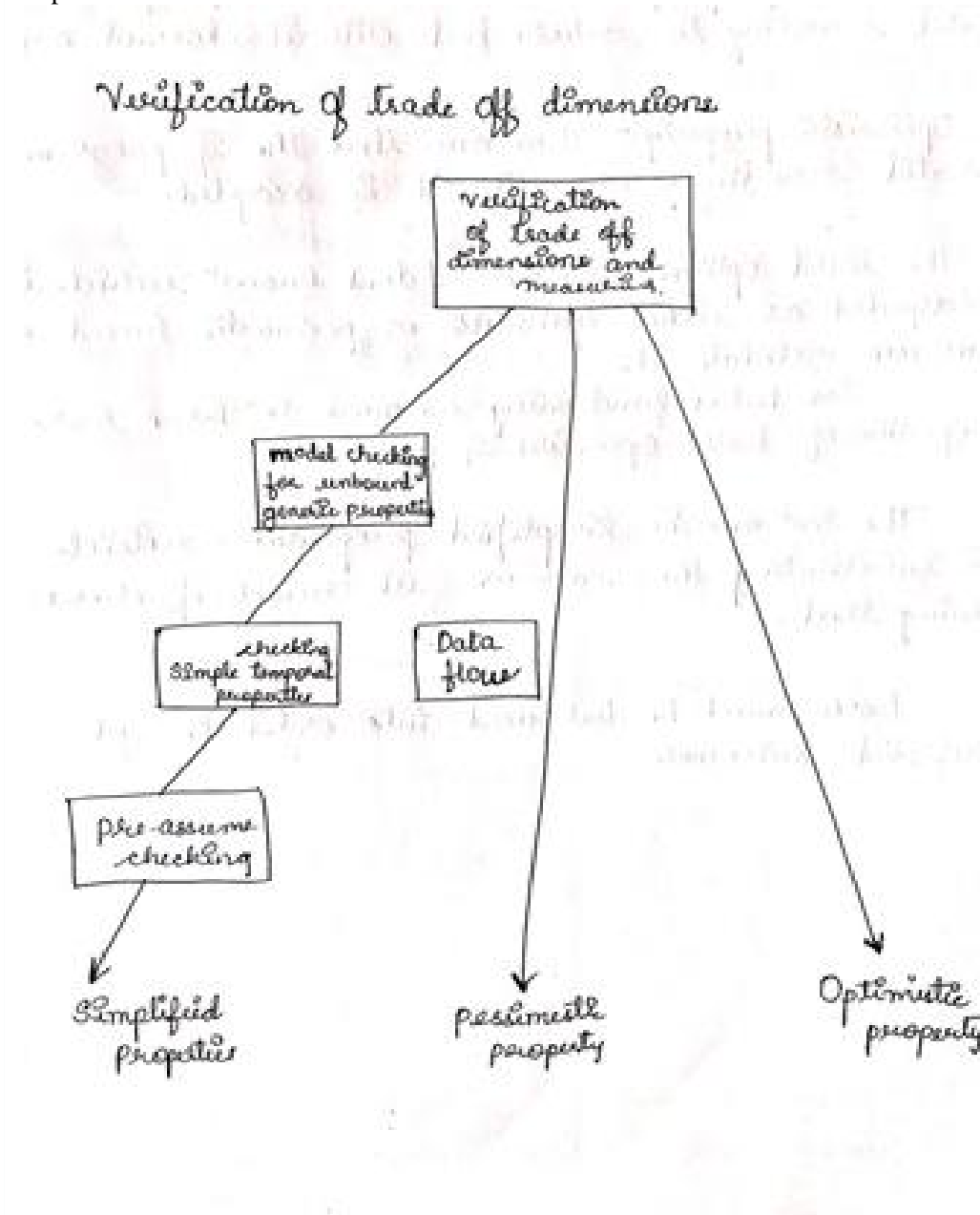


Scheme and Solutions - Improvement Test

Sub:	SOFTWARE TESTING						Code:	10IS65	
Date:	29 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ISE

Marks	OBE	
	CO	RBT
[05]	CO4	L4

1 (a) Explain verification trade off dimensions



Pessimistic property - It means that the program is valid according to criteria but still it doesnot accept it.

Optimistic property - It means that the if program is invalid accordingly but still it is accepted.

The third option can be "I dont know" which is interpreted as either optimistic or pessimistic based on how we validate it.

For better functioning we need to have balance proportion of both optimistic & pessimistic.

The 3rd one is Simplified properties - which is substituting the main or just bunch of classes doing that.

Both must be balanced in order to get proficient outcomes.

(b) Explain mutation analysis software fault based testing

[05] CO5 L4

Mutation Analysis :-

Mutation Analysis is the most common software fault based testing.

It is used to create hypothetical fault based program by changing the original program. The original program is changed based on the pattern. The pattern which is used to change the original program is called as mutant operation.

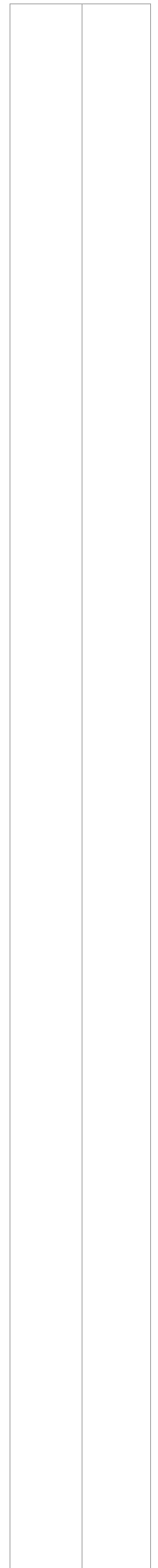
The variant which leads to change in original program leading to new program called mutant. (variant)

Suppose we have a function void transducer() which have several statements of which

```
void transducer()
{
    #define maxlen 100
    char den[maxlen]
    -----
    switch(a[char])
    {
        -----
    }
}
```

> Suppose we change switch statement to while
It is not valid mutant, it leads to compile time error.

- > Similarly if we change ~~0~~ 1000 to there is no error it is valid but it does not make any sense.
- > Therefore the mutants must not only be valid their behaviour must be different from the original program.
- > These programs are then tested if mutants differ from original program they are killed or mutants are kept alive.
- > This type of Analysis is called mutation Analysis for software fault based testing.



2 (a) What is Scaffolding? Describe generic and application specific scaffolding.

[05]

CO5

L1

Scaffolding :-

The code that is generated for testing is called as scaffolding. It is in analogy with the buildings around when there is a construction happening.

Scaffolding includes test drivers (substituting main function call), test harnesses (substituting deployment environment), and test stubs (substituting functionality call or software under test).

It helps in executing test cases and analysing the results of those test cases.

Generic and specific scaffolding

The simplest scaffolding is that the test driver. It includes the program to run for one test case or for specific test case.

For example suppose we want to make method calls in a particular sequence only then we write a code for it which involves method calls in that sequence. This is for specific test case.

When there is a thousands of code lines then the above technique becomes cumbersome, which leads to difficulty use of debugging, analysing etc.

(b) Describe the test oracles with a neat diagram.

[05]

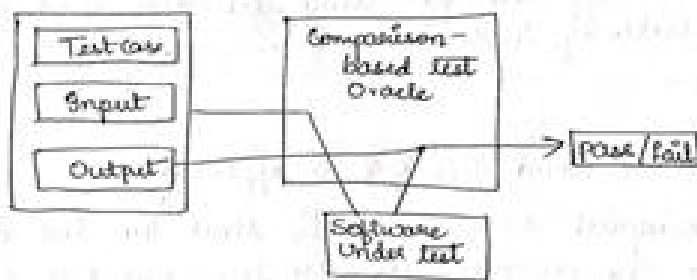
CO5

L1

Therefore it is better to find the common test driver code which can be filtered to make it common to all test cases.

This is the generic scaffolding where we use generic code which handles all types of test cases. It must be syntactically correct according any program language used.

Test Oracles :-



It is used to tell that success (pass) or failure of program execution is called Test Oracle named as oracle.

If the execution of a program is success then it gives success (pass) or fail. It is more handy when compared to human. It is more efficient.

3 (a) List the fault based adequacy criteria and explain.

[05] CO5 L1

Fault Based Adequacy criteria

Let Program P has test cases T then we have following steps to mutational Analysis

- (i) Select the mutants - Suppose we have specific faults then the mutants can be selected for that specifically.
- (ii) Generate mutants - We can generate mutants based on the program and specification.
- (iii) If the mutants are different from the original program then kill the mutants, the other mutants are said to be alive.

Eg:- Suppose $TS = \{10, 1U, 2D, 2U\}$

		10	1U	2D	2U
M_i	(buffer ≥ 0)	-	-	-	-
M_j	(buffer $\neq 0$)	-	x	x	x
M_k	(pos > 0)	-	-	-	-
M_l	(pos ≥ 0)	-	-	-	-
M_m	(atchar = 0)	-	-	-	-

Now M_j mutant can be killed as it is different from the original program.
Other are said to be alive.

(b) Briefly discuss dependability properties in process framework.

[05] CO4 L1

Dependability properties in process framework

(i) Correctness :- It tells whether the program is correct (success) or incorrect (fail). It is most important property. ~~eg~~

We cannot have somewhat correct or partly correct. It must be fully correct or wrong. It is relative to specification.

(ii) Reliability :-

Reliability is also relative to correctness. Similarly to ~~spec~~ specification. It is the measure of likelihood how correct the program is with respect to the specification.

(iii) Availability :-

It is the measure of the time between failures.
Eg:- network router failure we say router is down or unavailable.
When it's working we say "Up" or is available.

(iv) Safety :-

Safety is nothing but system safety property. Overcoming undesirable conditions called Hazards is safety.

It is not bringing new specification but just taking measures to overcome the hazard only.

(V) Robustness :-

Safety is kind of robustness. Safety measure taken to prevent hazard, but it mean taking measure to work in undesirable conditions.

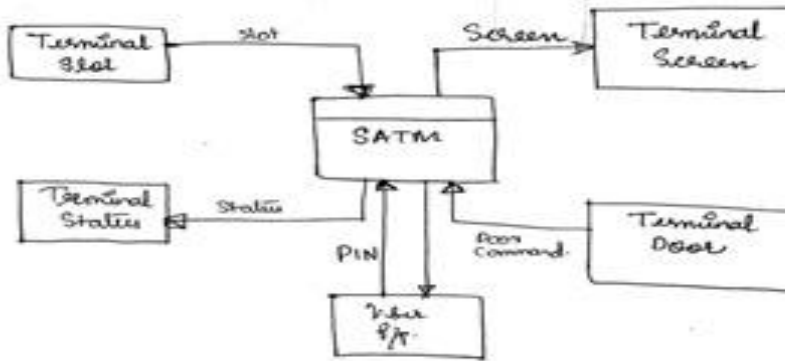
We can come up with augmented safety which helps to handle if undesirable conditions occur.

4 (a) Explain the simple ATM application with the help of context diagram and level 1 data flow diagram. [10] CO1 L4

SATM :-

The simple ATM consists of 15 screens in which we can do your bank transactions of deposit, withdrawal, ~~class~~ balance check.

Context Diagram :-

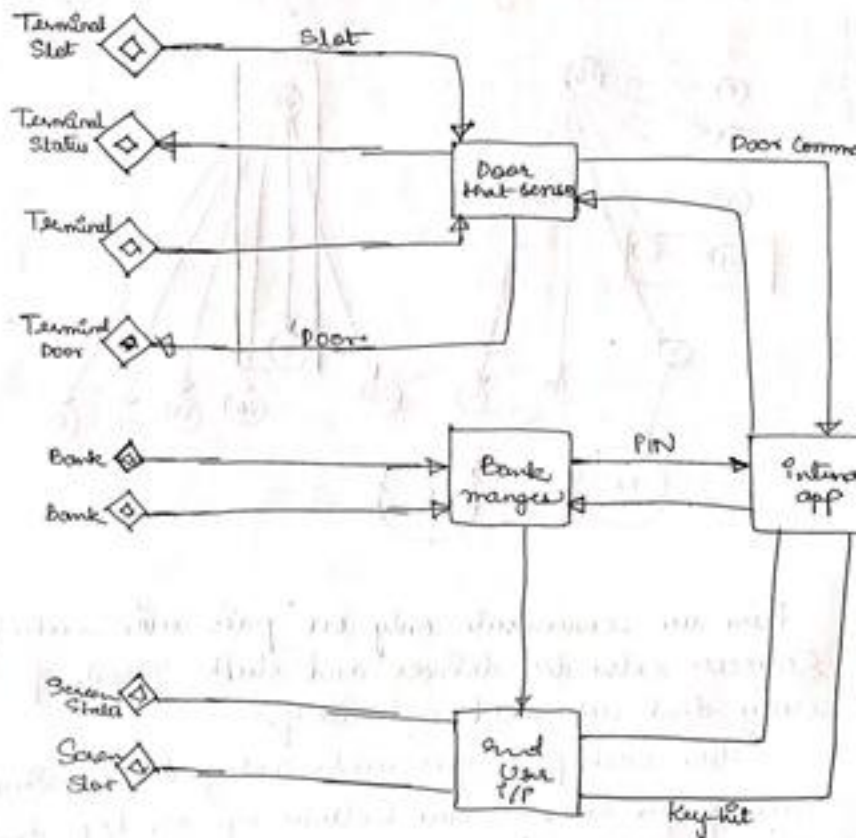


This uses a CASE tool for structural Analysis of the SATM. It is used to explain the slots, screen, door, status of the SATM.

The PIN is entered if correct only then processing is done if not message is shown.

There are transactions that can be done deposit, withdrawal, balance check.

Level-1 data flow diagram -



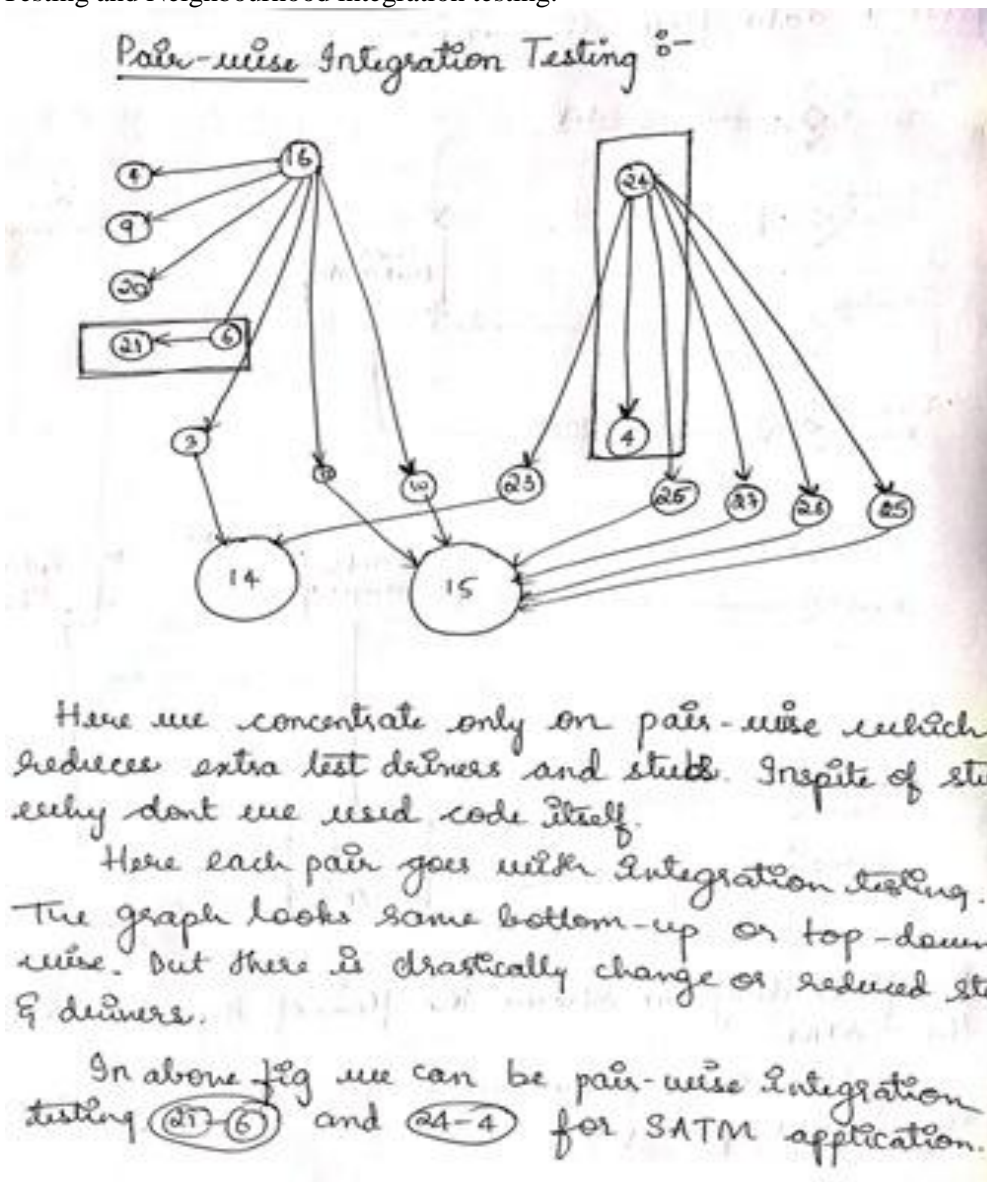
Dataflow diagram shows the flow of the data in the ATM.

It is one of the most efficient way to represent the flow of process.

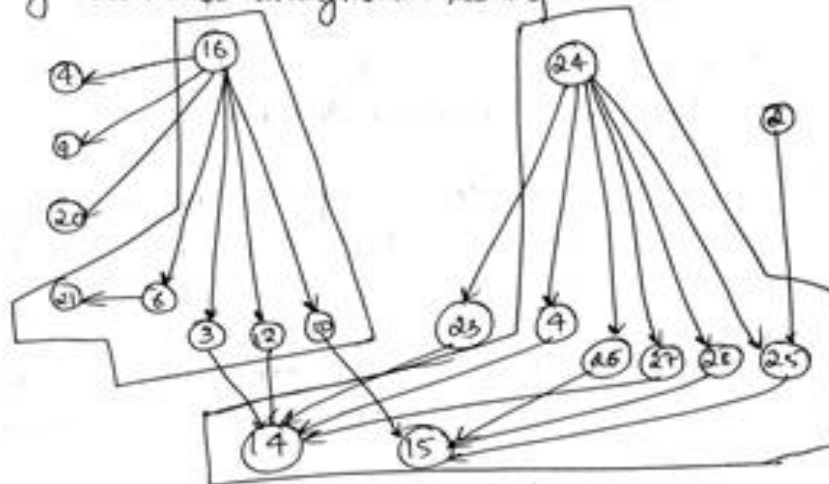
5 (a) What is call graph based integration testing? Briefly explain Pairwise Integration Testing and Neighbourhood integration testing.

[10]

CO1 L4



Neighbourhood Integration testing:



Neighbour of 1-radius means that for any node 1-node difference becomes neighbour.

So we take predecessor & successors of that particular node as its neighbour.

In above figure we can see 2 groups where we find neighbours, i.e. predecessor & successor.

We can calculate the neighbour

$$\text{Interior nodes} = \text{nodes} - (\text{sink} + \text{source})$$

$$\text{Neighbourhood} = \text{Interior nodes} + \text{sink}$$

- 6 (a) Why organizational factors are needed in process framework? Explain [06]
- (b) Write short notes on quality process. [04]

CO3	L1
CO6	L4