## HEADLINE

Combinational test technique, as the name suggests, is a technique of combining the data / entities as input parameters for testing, to increase the scope. This technique is beneficial when we have to test with huge number data having many permutations and combinations. The beauty of this technique is that, it maximizes the coverage by comparatively lesser number of test cases.

There are many techniques of CTD, where OATS (Orthogonal array testing technique) is widely used.

## ORTHOGONAL ARRAY TESTING CHARACTERISTICS:

- OAT, IS A SYSTEMATIC AND STATISTICAL APPROACH TO PAIRWISE INTERACTIONS.
- EXECUTING A WELL-DEFINED AND A PRECISE TEST IS LIKELY TO UN-COVER MOST OF THE DEFECTS.
- 100% ORTHOGONAL ARRAY TESTING IMPLIES 100% PAIRWISE TESTING.

## USING OATS:

Orthogonal Array Testing Strategy (OATS) is a proven technique, especially for integration testing of software components. OATS can be used to reduce the number of combinations and provide maximum coverage with a minimum number of test cases. Endlessly executing tests take too much effort to find defects and does not increase the confidence in the system. Executing a concise, well-defined set of tests that uncovers most of the defects is a wise approach and a cost saving technique. Oats techniques create an efficient and concise test sets with fewer test cases than testing all combinations of all variables. You can create a test set that has an even distribution of all pair-wise combinations.

Implementing OATS technique involves the below steps:

1. Identify the independent variables. These will be referred to as “Factors”.
2. Identify the values which each variable will take. These will be referred as “Levels”
3. Search for an orthogonal array that has all the factors from step 1 and all the levels from step 2
4. Map the factors and levels with your requirement
5. Translate them into the suitable test cases

6. Look out for the left over or special test cases (if any)

**EXAMPLE:** Let us consider you have to identify the test cases for a Web Page that has 4 sections: Headlines, Details, References and Comments, that can be displayed or not displayed or show Error message. You are required to design the test condition to test the interaction between different sections.

In this case:
1. Number of independent variables (factors) are = 4
2. Value that each variable can take = 3values (displayed, not displayed and error message)
3. Orthogonal array would be 34.
4. Google and find an appropriate array for 4 factors and 3 levels. For this example, I am referencing the below table

| Experiment no. | Factor A | Factor B | Factor C | Factor D |
|---|---|---|---|---|
| 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 |

5. Now, map this array with our requirements as below:

- 1 will represent “Is Displayed” value
- 2 will represent “not displayed” value
- 3 will represent “error message value”
- Factor A will represent “Headlines” section
- Factor B will represent “Details” section
- Factor C will represent “references” section
- Factor D will represent “Comment” section.

Experiment no will represent “Test Cases #”

6. After mapping, the table will look like:

| Test Case | Headlines | Details | References | Comments |
|---|---|---|---|---|
| TC01 | is displyed | is displyed | is displyed | is displyed |
| TC02 | is displyed | not displyed | not displyed | not displyed |
| TC03 | is displyed | error | error | error |
| TC04 | not displayed | is displyed | not displayed | error |
| TC05 | not displayed | not displayed | error | is displyed |
| TC06 | not displayed | error | is displyed | not displyed |
| TC07 | error | is displyed | error | not displyed |
| TC08 | error | not displayed | is displyed | error |
| TC09 | error | error | not displayed | is displyed |

7. Based on the table above, design your test cases. Also look out for the special test cases / left over test cases.

**BENEFITS OF 'OATS':**

An orthogonal array is a type of experiment where the columns for the independent variables are independent to one another.

- Large savings in test effort.
- Analysis is easy [ready made tools and tables are available to describe the combination].
- Test design using OATS tools help in those test cases where we choose to make all the difference between:

1. Endlessly executing tests take too much effort to find bugs and do not increase confidence in the system.
2. Executing a concise, well-defined set of tests that uncover most of the defects will remove the inefficiencies and increase confidence in the system.

**CONCLUSION:**

None of the testing technique provides a guarantee of 100% coverage. Each technique has its own way of selecting the test conditions. In the similar lines, there are some limitations of using this technique:
Testing will fail if we fail to identify the good pairs.

- Probability of not identifying the most important combination which can result in losing a defect.
- This technique will fail if we do not know the interactions between the pairs.
- Applying only this technique will not ensure the complete coverage.

It can find only those defects which arise due to pairs, as input parameters.