

One-Source Online Toolkit - OOT

For Software Test Design and Test Data Generation

Niveditha S.
4th Sem, M.Tech, Dept. of Soft.Eng
M.S. Ramaiah Inst of Tech,
Bangalore, India
nivedithas800@gmail.co.in

Dr. Aswatha Kumar. M
HOD of Information Science Dept.
M.S.Ramaiah Inst of Tech,
Bangalore, India

Abstract— This paper presents a toolkit for the testers which provides a good framework for software test design and test data generation especially for test diagnosis and validation. Basically, OOT is One Source Online Toolkit is a web based test case generator which provides a variety of tools and utilities to help improve test engineering efforts including software test documentation, software test case reduction, software test data preparation and a many other uncategorized software test functions. Test data preparations involves generating large volumes of custom data in a variety of formats for use in testing software, populating databases, and can be used as an input for some of the web application etc., Developing a software test case design generator provides an efficient utility for obtaining a minimal test case set out of the bulk. Over all this tool includes all these utility in a single framework within a single click the test case can be generated and more user friendly. Thus, OOToolkit is building innovative tool that do smarter test engineering and reduce testing cost and time.

Keywords— combinatorial testing, pair wise testing, subset test case generator, test case permutation generator, test coverage, test data generator, test design generator.

I. INTRODUCTION

Developing software testing tool with high quality is one of the important objectives of software engineering. In the past, numerous testing and maintenance techniques have been developed to facilitate software testing but those techniques were not so effective since testers has to spend lot of time on creating testcases manually for testing specific application. Nowadays people consider more and more about the completeness and effectiveness of such techniques in order to increase the developers' confidence in software quality and its efficiency.

To some extent, all these techniques have been developed based on the information collected about software requirements, design, implementation, testing and maintenance. If the information can be organized into a complete and tight-coupled

framework, then that system could then be used to support more systematic, complete and efficient software testing and maintainence.

In this paper, we will discuss on software testing tool framework called OOT- One Source Online Toolkit for generating test data and test design generator.

Testing the performance of any of the software application is obvious not an easy task. Manual testing consumes a lot of time as it requires a tester to play the role of an end user, and use most of all features of the system to ensure correct behavior. The challenge is not only to solve the testing problem by having an effective test case coverage but also in monitoring all the such type of utility which could really help the tester in his work by embedding all such utility in a single frame thus, saves the time and cost of testing. With this motivation, an application is developed in such a way, by generating the test cases through some of the test design generator like pair wise test case generator, subset test case generator, test case permutation generator, etc.,and also generating some random test cases through test data generators like size based file generator which creates a batch file of certain specified size, and the type of file to be generated like numeric or alphanumeric or alphabetic etc based on addressing the regular need to create several fake files with different types of known data and required size. Similarly the password generator allows creating multiple random password of a given complexity based on the constraints specified by the user. And same is the case of email address and random date (timestamp) generator where there may be usually many cases where a particular test may require a lot of dummy test email IDs in different sizes and type. Random date generator is also required to be generated for the use in loading the random dates into the database or for some performance testing. Testing may be on embedded software, drivers for peripherals, web applications or mobile applications, OOT service improves the test efficiency by-

- Reducing the test life cycle time and saving the cost.

- Helping in test case design and test data generation.

Modules in One Source Online tool kit:

1. Test case design algorithm
2. Test data generation

A. Randomized Automatic Test case Generation

Randomized Automatic Test case Generator in software testing which generates test case automatically having a good coverage of test case to be tested. As OOT provides some of the technical features like the tool is completely online, requests can be made through screens as well as file uploads and more over it has user friendly outputs and downloadables.

Toolkit include following testing technique for test design generation:

- It provides a web based test case generation tool using pair wise or combinatorial testing technique
- Also include subset test case generation technique and test case permutation technique.

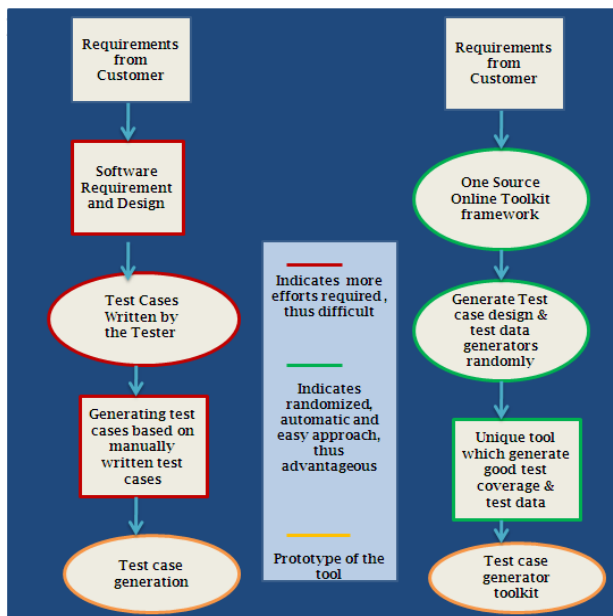


Figure 1. Difference between the Manual test case generator and Randomized automatic test case generator approach

The Figure 1 shows the difference in the traditional manual test case generation approach and the randomized automatic test case generation approach, where the traditional approach i.e., a manual test case generator is built using the given requirements and the test cases are usually been

written manually by the programmer and thus generates the test case based on the written test case.

While in Randomized test case generation approach (OOT), it involves generating a randomized test cases which is unique and since all the utility are put into a single package this forms an efficient framework with good test coverage and test data generation for all the software testing application.

II. GOAL

Testing the performance of any of the software application is obvious not an easy task. Manual testing consumes a lot of time as it requires a tester to play the role of an end user, and use most of all features of the system to ensure correct behavior. The challenge is not only to solve the testing problem by having an effective test case coverage but also in monitoring all the such type of utility which could really help the tester in his work by embedding all such utility in a single frame thus, ensures optimal test set which will effectively test the software system is desired, which results in reducing the effort of testing and obtaining the best tool which is well automated, also saves the time and cost of testing. With this motivation, an application is developed that tests the performance of some software by generating the test cases automatically as it is web based application, it does not require any software to be downloaded.

A. What is the need for generating randomized test cases?

There are several reasons why to randomize the test case:

- Speed up testing to accelerate releases
- Allow testing to happen more frequently
- Reduce costs of testing by reducing manual labor
- Improve test coverage
- Ensure consistency
- Optimal test set which will effectively test the software is desired
- Improve the reliability of testing
- Allow testing to be done by staff with less skill
- Define the testing process and reduce dependence on the few who know it
- Make testing more interesting
- Develop programming skills

One may find it's best to randomize the execution and leave the verification to be done manually. Or may choose to automate the verification and leave the execution to be done manually. Some

people say that it's not real automation unless it does everything. If one is simply looking for challenge, then he can try to do it all. But if he is looking for success, focus on where one can quickly get automation that can be used again and again.

III. PROPOSED ARCHITECTURAL DESIGN

This architecture in Figure2 mainly has 3 components the Application layer, Test case generator Engine (test design and test data generators) and the Web server. Basically web server used here is an Apache local web server to display the contents on the web page and making it more interactive for user with good UI design, simple and elegant input/output formats to be more quick and accurate in its function. Technology used to generate these test cases are C# implementation of the test case design generators and Php, JavaScripts for test data generators.

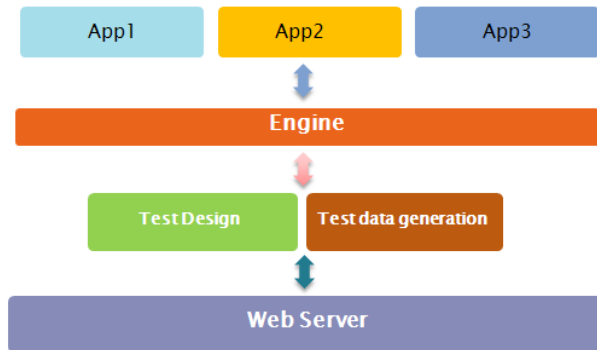


Figure 2. Architecture of OOT tool

A. User Application

Web application is an extensible platform for building a web page for the elegant use of user interaction. It provides a core of services for controlling and managing all the web page and making it more user-friendly. Here the tool is integrated with the apache web server for the web page display (front end application). This helps to create a platform where all the tools are wrapped together in a single framework toolkit..

B. Test Engine

The test engine involved in this tool is test design and test data generator. As they generate test cases in the backend in a randomized fashion. As the user calls for certain generator to generate the testcases, based on the same it generates. Traditional manual test case generation doesn't involve this kind

of randomized engine. Thus the tool is different from other approach.

C. Web Server

A web server can be referred to hardware, the computer or the software, the computer application that helps to deliver content that can be accessed through internet with the help of web page. The most common use of web servers is to host websites. In OOT apache web server plays a vital role to integrate the application (test engines) with other utilities and displays through web page.

D. Test Design Generator Engine

Test design part consists of designing or developing test cases based on algorithmic approach as follows:

i) Pair wise test case generator:

This is also called combinatorial test case generation technique. It is a part of test case design algorithm module of the tool using which a test engineer can generate reduced number of test cases. It's a test design tool that can generate reduced number of effective test cases out of the large number of possible test combinations with maximum test coverage data. This is a popular approach for combinatorial testing problems, whenever multiple variable interact. Several testers have judiciously used this tool to reduce millions of test combinations into just a vital few and successfully found defects in the short time.

➤ Why yet another pair wise test case generator?

Firstly, there are wonderful tool available in the market, but user cannot customize the tool based on their requirement as the source code is not available.

User Requirements –

- Directly read its input from an XML file or a SQL database
- Directly emit results in a custom output format.
- Introducing constraints (test input sets that are not permitted), by introducing required test sets, or changing how the tool generates its test set collection.
- Although pair wise test set generation tools are available on the Internet, some of these

tools are quite inefficient for providing the required data

For example, consider a situation with 20 parameters, each of which has 10 values. For this scenario there are-
 $10 * 10 * 10 * \dots * 10$ (20 times) = 10^{20} = 100,000,000,000,000,000,000 possible test-case inputs. This is a lot of test cases. Few tools provide more than 600 Test cases few tools provide more than 1000 Test cases but OOT can manage to give 200+ Test cases in above scenario

ii) *Subset Test case Generator:*

Often times, while choosing test inputs the problem of selecting a subset out of a given list arises, this tool shows all possible subsets of a given size from a list of test items. The number of such subsets is deduced by the conventional nCr formula from mathematical combinations and the generated list is lexicographically ordered. Generating all the subset combinations (k items taken out from n items) of a test parameter has proven to be a powerful way to drive test case input. Based on the subset number (k) the number of combinations of such subset combinations can be generated.

iii) *Test case Permutation Generator:*

Also called change orders test case combinations. Change in order of action can lead to change in behavior. It is least argued fact in the way how software applications work. Testers need to change the order of key test actions to see if they cohesively work well.

Thus the test cases designs are implemented and designed for *generating*) test cases with good test coverage

E. *Test Data Generator Engine*

This is a simple, powerful and fully customizable utility that generates data, tables, views, procedures etc., for database testing purposes: performance testing, QAtesting, load tests or usability testing. Currently database developers and administrators spending hours of dull work to create test data sets by writing it manually which is the tedious process, before the performance is examined. But this utility over comes manual process, instead it generate the test data more conveniently. Few of which as follows:

a) *Password Generator:*

Generates a set of valid password test input values to test password fields against test requirements

ie., those passwords which are randomly generated can be used as test inputs in testing the requirements of the password field. This generator create highly secure and extremely difficult to crack or guess due to an optional combinations of lower and upper case letters, numbers and punctuation symbols.

b) *Size based File Generator:*

It generates a windows batch file that can be successively creates files of the specified size (KB or MB or GB) and type. By addressing the regular need to create several files with different types of known data and required size. File type may be numerical, alphabetical, alphanumeric etc. This utility may be used in many of the applications like if CD Rom burning feature of publishing application is being tested and if it needs 700MB of alphanumeric data of some junk file, then this may be used.

c) *Email Address Generator:*

There usually are many cases where a particular test may need a millions of dummy test e-mail IDs in different sizes and types. They may be used as an email list builder to create email lists for marketing or advertising campaigns. Can create unlimited email addresses to send emails about our product.

Ex: An address book functionality being tested and 100's of ID needed to be loaded into test display, sorting and paging features.

d) *Date/Time (Timestamp) Generator:*

It can generate 1000's of random date and time at a time which may be used for loading dates into the database as a part of application input or for some performance test. Especially in financial related application this generator may come into use.

e) *Random number Generator:*

There exists number of applications requiring random numbers increases continuously. They are used for example in cryptographic applications, in scientific calculations or to generate passwords.

CONCLUSIONS

This paper provides an approach to the automatic generation of the test cases through some test data generators and also generates minimal test case for testing. This makes the testing process very easy and efficient by speeding up the testing process

and improving test coverage and has no manual labor, thus reducing the cost. This also frees up the testers from more testing tasks, which is again a competitive advantage. One can say this tool has the Super Saver Technique Tooling.

REFERENCES

- [1] Brian Marick, "When a Test Should Be Automated?". StickyMinds.com, 2010.
- [2] Rick Kuhn and Raghu Kacker, "Combinatorial Software testing National Institute of Standard and Technology", IEEE, Aug-2009.
- [3] G. J. Myers, T. Badgett, T. M. Thomas, and C Sandler, "The Art of Software Testing", Wiley, USA, 2004.
- [4] Jacek Czerwonka, " *Pairwise Testing in the Real World: Practical Extensions to Test-Case Scenarios*", Microsoft Corporation, February 2008
- [5] DeMilli R.A, Offutt A.J, " Constraint-based automatic test data generation", IEEE, Aug-2002.
- [6] J Edvardsson, "A survey on automatic test data generation", Proceedings of the 2nd conference in computer applications, Citeseer-1999.