

# An Efficient Method to Generate Automation Scripts using Selenium Tool

Brinda Sood<sup>1</sup>, Dhavleesh Rattan<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Punjabi University, Patiala, India, Mb. +91 81960 00950

<sup>2</sup>Department of Computer Engineering, Punjabi University, Patiala, India, Mb. +9198148 37334

Email:<sup>1</sup>brinda1230@gmail.com, <sup>2</sup>dhavleesh@rediffmail.com

**Abstract-** In the recent past, there has been a large increase in the number of web applications. As the internet has evolved beyond all limits, the web applications also called websites have become a huge concern with millions of website around the world. There are many techniques which are used to perform software testing. An automation testing increase the efficiency of system and it also saves time so that it must be completed with minimal time and effort. The proposed work generates record script automatically by taking three inputs i.e. application model, state diagram and application code. By taking UML diagram as input first, test cases are generated and after that automation test scripts are generated. In the proposed work, performance of the application is increased using optimization techniques. This optimization technique extract better property state by using state diagram and generate test case steps again to generate automation scripts. It increases the performance of the application.

**Keywords:** UML, software testing, QA, GUI

## 1. Introduction

Presently applications of computer have diffused into each sphere of life to control numerous sophisticated applications. Lots of these applications are of safety critical, substantial and complex nature. Subsequently, there must be reliable software. In other words, software of good quality and high reliability is crucial. Apart from numerous techniques for enhanced reliability, software testing is very common and important methodology. Thus, testing remains the most vital part of quality assurance in the practice of the development of software. Even though there are so many techniques available for quality assurance such as design reviews, formal specifications, inspection and checking of model exists. Since the software now has an imperative part in our lives both socially and economically. There is pressure for software professionals to concentrate on quality issues. The software with poor quality that can bring loss of property or life is no more adequate to society [1].

Software testing is very expensive and labor intensive; it represents pretty nearly 50% of the cost of development of software system. If the process of testing could be automated, the cost of software developing ought to be decreased fundamentally. Of the problems included in software testing, one is of specific importance here; the issue of creating test data. A test data generator is a device which helps a programmer in generation of test cases for a program. There are three types of generators of test data: pathwise test generators, data specification generators, and random test generators [2].

Software testing is an imperative activity in lifecycle of software development. Software organizations spend significant part of their budget in activities which are related to testing. Each time the execution of program is done, the program is tested by the user. So we need to execute the program with the particular intent of removing and fixing errors [3].

The UML sequence diagrams are utilized for modeling distinct behavior of an object through sequence graph. Such transitions and states are discriminating to choose the invocations of particular operation that would be made taking into account the conditions emerging during the execution of a scenario. For testing at unit level, we can get tests from state machine diagrams of UML, which embody the behavioral depiction of every component [4].

## 2. Automated Testing

Automated software testing is a procedure in which software tools execute pre-scripted tests on a software application before it is discharged into production. The aim of automated testing is to simplify as much of the effort of testing as conceivable with minimum set of scripts. Tests completed out with these tools can be repeatedly run at any time of day.

In most organizations, automation is implemented only when it has been resolved when it has been depicted that the program of manual testing is not fulfilling expectations and it is unrealistic to acquire most human testers.

The steps of automation process are:

1. Test tool selection
2. Define scope of automation
3. Planning, design and development
4. Test execution
5. Maintenance

### 3. Uml Diagram

Unified Modeling Language (UML) is a modeling standard used generally in the field of software engineering. The Unified Modeling Language incorporates a set of techniques of graphic notation to produce visual models of object-oriented software-intensive systems. The Unified Modeling Language was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1990s. It was adopted by the Object Management Group (OMG) in 1997, and is overseen by this association ever since. In 2000, the Unified Modeling Language was acknowledged by the International Standard Organization for Standardization (ISO) as standard for Industry to model software intensive systems. The Unified Modeling Language is becoming an essential skill for anyone virtually incorporated in software projects. There are mainly two reasons for the requirement of modeling language. Firstly, the model offers a blueprint for developers so they exactly know what they need to develop and for project managers so that they calculate precisely the cost of given project. Secondly and more significantly, UML is the bridge between non technical users and technical developers [5].

### 4. Related Work

A lot of research work has been done in the area of automated testing. Many efficient methods have been proposed for automated testing in the software development. Some important work in the area of automated testing is as follows:

TABLE I  
Comparative Study of Literature Survey

Source	Sample/Study Description	Purpose
Sunguk Lee [6]	Unified Modeling Language (UML) for Database Systems and Computer Applications	This presents the idea of systems of database and in addition the utilization of (UML) as a standard notation of real world objects in developing design methodology of object oriented computer applications.
A. V. K. Shanthi, G. Mohan Kumar [7]	Automated Test Cases Generation from UML Sequence Diagram	This paper proposed a novel approach for software testing at initial stage itself so that it will be simple for software testers for testing in later stages.
Isabella, Emi Ratna [8]	Study Paper on Test Case Generation for GUI Based Testing	This paper explains about the generation of test case for GUI based testing. Intent of this paper is to study few techniques utilized for generation of test cases and the procedure for many software application which are based on GUI.
ParamjitKaur, Rupinder Kaur [9]	Approaches for Generating Test Cases Automatically to Test the Software	In this paper, various techniques are studied to generate test cases from UML diagrams for automatic software testing. A new methodology is also proposed which will help in software testing automatically by test cases generation.
Aysh Alhroob, Keshav Dahal, Alamgir Hossain [10]	Automatic Test Cases Generation from Software Specifications	A new technique is proposed to expand the Integrated Classification Tree Methodology (ICTM). The approach presented in this paper separates just the legitimate test cases and those incomputable with the specifications of software.

## 5. Motivation

The scope of this research work is broader as it helps to develop abilities to execute record scripts generation automatically. This proposed work is an enhancement of automation testing. In the proposed work, three inputs are taken: application model, application code and state transition diagram all comprises to form UML diagram. By using this input, test cases are generated and with the help of test cases automation scripts are generated. Optimization technique is used for enhancing the performance of the GUI application.

The research work is based on following objectives:

1. Generate test case by taking application model, state transition diagram and application model i.e. UML diagram.
2. Generate automation scripts by using test case.
3. By using state diagram apply optimization technique on test case generation for enhancing the performance of GUI application.

## 6. Proposed Scheme

In our word, we generate record scripts automatically. So in our research methodology we describe this solution step by step. These steps are given below:

- In first step we will generate test cases. The generation of test case will be done by referring UML diagram.
- After generating test cases we will generate test case scripts by using these test cases.
- In third step we will use a state diagram. By using this state diagram we will extract those states which has better property and after that we will be performing again test case step on it for generating test scripts.
- At last check the enhanced performance of GUI application.

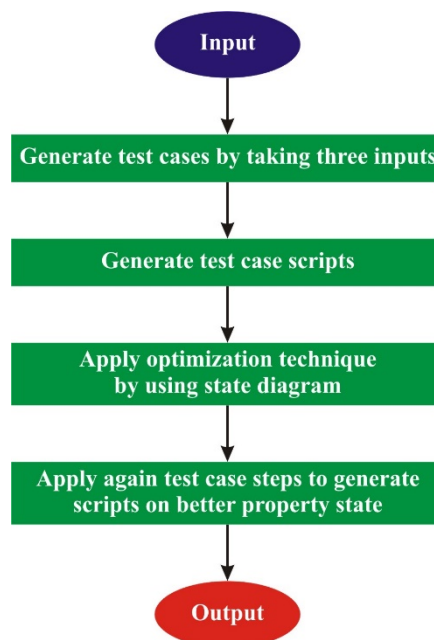


Fig. 1 Flowchart of proposed method

## 7. Results And Discussions

This section presents the simulated results of automated testing for enhancement of GUI application. Also this will help to understand the working of all over project, its features. It will provide the clear view of the inputs required and output.

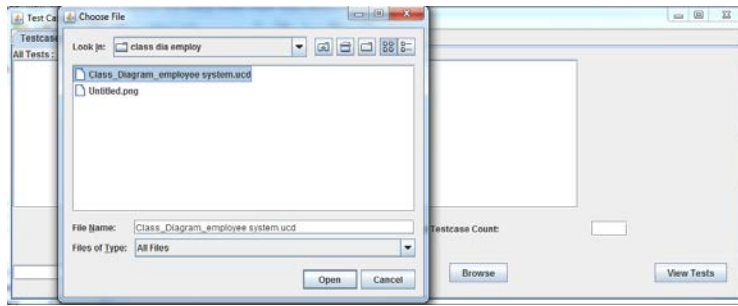


Fig. 2 Browse Class diagram

This window is used to browse the UML diagram from the system, which will be provided as an input. When Browse button is clicked, new window opens and one can select the destination folder of the UML diagram residing in the system. Only the files in .ucd format will be accepted else it will display an error message. After selecting the desired UML file, View Test Button is clicked so as to view the results. All the processing will takes place at the back end and further output is displayed. The purpose of class diagram is to find the classes within a model. In an object oriented application, classes have attributes, operations and relationships with other classes.

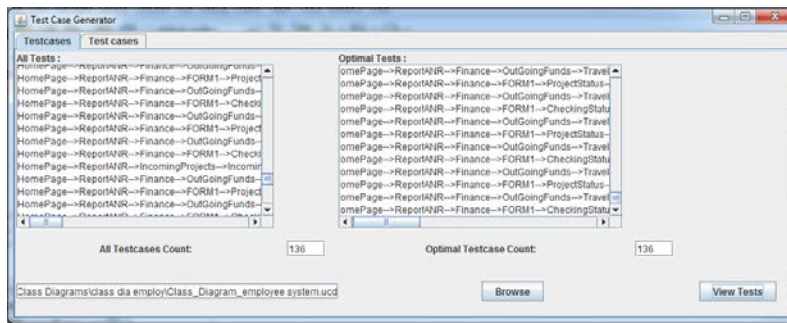


Fig. 3 Optimized test cases paths

This window will display all generated test cases and also the optimized test cases using the proposed method. Time taken for the generation of scripts and optimized test cases is very less. All the required steps are performed automatically at back end and the result is shown. A properly generated test suite may not only locate the errors in a software system, but also help in reducing the high cost associated with software testing. It is often desired that test data in the form of test sequences within a test suite can be automatically generated to achieve required test coverage.

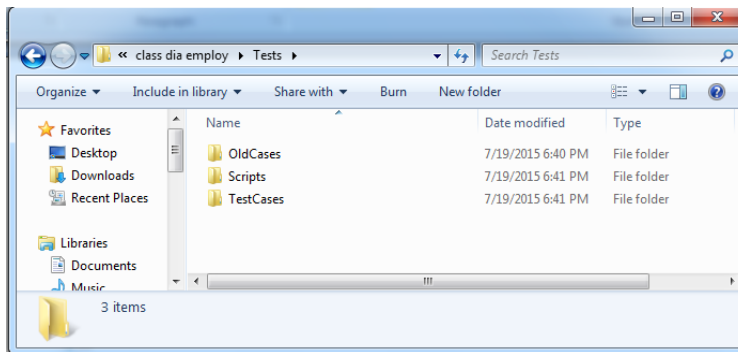


Fig. 4 Saved test cases

Here one folder is generated named as Tests as the destination folder of the UML diagram which was used as an input. Tests folder further contains 3 sub folders i.e. OldCases, Scripts, TestCases. OldCases folder contains the all test paths generated using the proposed method. Scripts folder contains the scripts generated for each optimal path using selenium



tool. TestCases folder contains all optimized test cases. Optimizing or even improving the quality of the test cases can be the aim of some researchers.

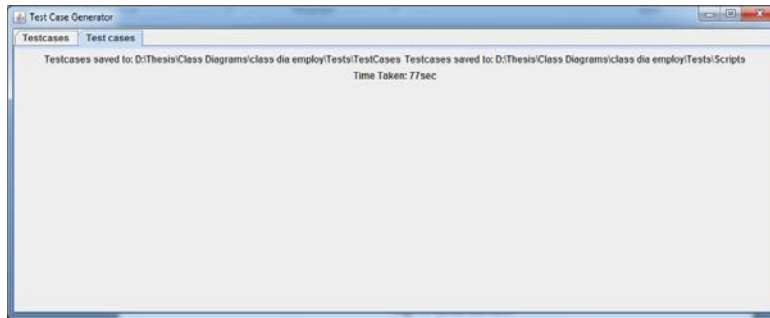


Fig. 5 Saving the test cases and time taken to generate the scripts

This window display the time taken to generate scripts and optimal test cases. The time taken for the execution of proposed work entirely depends upon the complexity of the software/web application/window application which is to be tested and system configuration which is used for testing. Performance of proposed system is very much efficient as compared to the previously developed systems. This system help to save the time as it provides output in very less time. Also it reduces the system development cost because human efforts required for testing are not required. Proposed system provides an efficient way for generating automated scripts and test cases using Java platform and Selenium tool.

```

package com.example.tests;
import com.thoughtworks.selenium.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
import java.util.regex.Pattern;

{
    private Selenium selenium;
    @Before
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
        "http://www.domain.com");
        selenium.start();
    }
    public void testHomePage{
        selenium.type(txtName, InputValue);
        selenium.type(txtPassword, InputValue);
        selenium.click();

    @After

        public void tearDown() throws Exception
        {
            selenium.stop();
        }
    }
package com.example.tests;
import com.thoughtworks.selenium.*;

```

Fig. 6 Selenium testing running in the back end of the test cases syntax.

The Selenium Integrated Development Environment (Selenium-IDE) is the tool used to develop Selenium test cases. It's an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases.

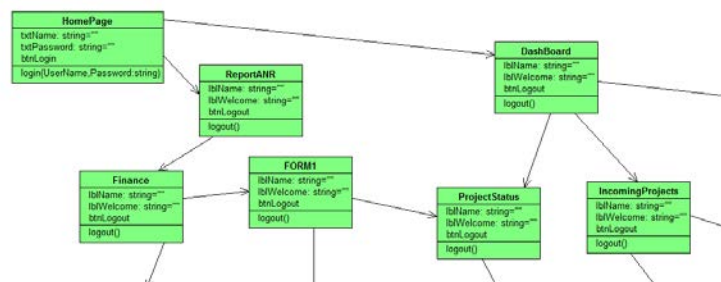


Fig. 7 Class diagram of Employee management system



This shows some part of UML diagram of employment system used as an input to the system. This diagram is generated in the software UML pad.

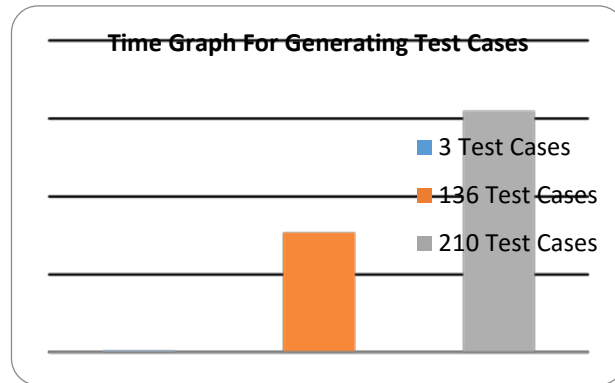


Fig. 8 Time Graph for Generating Test Cases

Three cases are considered here first with class diagram having 3 test cases which took 01 sec to generate the test cases and test scripts; second with class diagram having 136 test cases which took 90 sec to generate the test cases and test scripts; third with class diagram having 210 test cases which took 148 sec to generate the test cases and test scripts. The proposed technique takes less time to generate the test cases and test scripts. Time taken for the execution entirely depends upon the number of test cases of the system which is to be tested.

## 8. Conclusion

The main aim of this research is to enhance the automated testing. The proposed system helped to develop capabilities to perform generations of record scripts automatically. Previously, record scripts were generated by developing a coded UI tool and an automation framework. If the record script is generated automatically, it will be very helpful in many fields like industrial and technical filed and GUI applications. In this work, generation of automation scripts will be able to save time as scripts are generated by taking three inputs directly. In this research work, we have used the test cases for employee management system by making class diagram, which is shown in above section. Automation testing is very much useful for checking the functionality and behavior of GUI application by using various types of developed tool.

## 9. Acknowledgement

The paper has been written with the kind assistance, guidance and active support of my department who have helped me in this work. I would like to thank all the individuals whose encouragement and support has made the completion of this work possible I would like to thank Almighty God for providing me this opportunity and granting me the capability to proceed successfully.

## References

- [1] A. Verma and M. Dutta, "Automated Test case generation using UML diagrams based on behavior," *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 4, issue 1, June 2014.
- [2] B. Korel, "Automated Software Test Generation," *IEEE Transactions on Software Engineering*, vol. 16, pp. 8, August 1990.
- [3] M. Prasanna and K. R. Chandran, "Automatic Test Case Generation for UML Object diagrams using Genetic Algorithm," *Int. J. Advance. Soft Comput. Appl.*, vol. 1, pp. 1, July 2009.
- [4] V. Panthi and D. P. Mohapatra, "Automatic Test Case Generation using Sequence Diagram," *International Journal of Applied Information Systems (IAIS)*, vol. 2, pp. 4, May 2012.
- [5] M. Ibrar, "UML Diagrams: an aid to Database Design Specification: A Review," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, issue 3, March 2013.



- [6] S. Lee, "Unified Modeling Language (UML) for Database Systems and Computer Applications," *International Journal of Database Theory and Application*, vol. 5, pp. 1, March 2012.
- [7] A. V. K. Shanthi and G. M. Kumar, "Automated Test Cases Generation from UML Sequence Diagram," *International Conference on Software and Computer Applications (ICSCA)*, 2012.
- [8] Isabella and E. Retna, "Study Paper on Test Case Generation for GUI based Testing," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 3, pp. 1, January 2012.
- [9] P. Kaur and R. Kaur, "Approaches for Generating Test Cases Automatically to Test the Software," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 2, issue 3, February 2013.
- [10] A. Alhroob, K. Dahal and A. Hossain, "Automatic Test Cases Generation from Software Specifications," *e-Informatica Software Engineering Journal*, vol. 4, issue 1, 2010.
- [11] K. P. Jayant, R. Garg, V. Kumar and A. Rana, "An Approach of Software Design Testing Based on UML Diagrams," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.4, issue 2, February 2014.
- [12] Y. Zheng, J. Zhou and P. Krause, "An Automatic Test Case Generation Framework for Web Services," *Journal of Software*, vol. 2, pp. 3, September 2007.
- [13] A. Nayak and D. Samanta, "Automatic Test Data Synthesis using UML Sequence Diagrams," *Journal of Object Technology*, vol. 9, pp. 2, March-April 2010.
- [14] P. N. Boghdady, N. L. Badr, M. Hashem and M. F. Tolba, "A Proposed Test Case Generation Technique Based on Activity Diagrams," *International Journal of Engineering & Technology*, vol. 11, pp. 03, June 2011.