

The Selenium based GUI Automation Tool

Bhavna Arora¹ Tejal Rachh² Nida Parkar³ Sowmyashree⁴
^{1,2,3,4}Assistant Professor

^{1,2,3}Department of Computer Engineering ⁴Department of Information Technology
^{1,2,3,4}Atharva College of Engineering, Mumbai, India

Abstract— Automated Testing is the most preferable way to test the product with increased efficiency. Automated tests are generated and can be easily repeated to perform the tasks that are impossible with manual testing. Automated testing is considered critical for big software organizations and are too expensive for small companies to implement. Automated software testing is used to improve the accuracy and test coverage within short interval of time at low cost. In the competitive world, companies or testers have many options of picking the license or open source automation tools such as Quick Test Professional (QTP), selenium etc., based on their need of project. This paper proposes a GUI automation framework called SMEASY (Selenium Made Easy) built on the top of the Selenium, which overcomes the shortcomings of sauce builder and provides a better functionality. Eclipse is used as backend for the tool we are using. Selenium support many browsers, so in our tool we have given the facility to execute the script on different browsers. The tool significantly reduces the script creation time and makes it faster and increases the efficiency.

Key words: Selenium, Automation Tool

I. INTRODUCTION

Every software development group tests its products, as the delivered product contains defects. Test Engineers strive to catch them before release of the product with the best manual testing processes but they end up in vain. Automated Testing is the best way to test the product with increase in efficiency. Automated tests are created and can be easily repeated to perform the tasks impossible with manual testing.

Automated testing is considered critical for big software organizations or too expensive for small companies to implement. Automated software testing improves the accuracy, test coverage within short interval of time at low cost. Test automation has specific advantages for improving the long-term efficiency of a software team's testing processes at low cost and time.

A. Automation Testing Tools

In the competition world, companies/testers have multiple options to pick licensing or open source automation tools such as Quick Test Professional (QTP), selenium etc., based on their requirement.

Different automation testing tools such as QTP, QA Test, QA Load, QA Wizard Pro, Win Runner, Test complete, Silk Test, Silk Performer are available for performing Functional, Performance Testing, Regression and Rational Robot. QTP- a Functional Testing Tool has the attribute for storing screenshot of each and every page navigated throughout the execution. So it can be utilized as verification for completion of testing, and also if there is any need to refer the screenshots of previous executions, we can refer them easily. Test report can be automatically written to A customized report page can be automatically written that will ensure accuracy of the report. The look & feel of the report can also be improved.

Test automation is a way of knowledge sharing and increase output. It is easy even for a non-programmer to understand QTP and create adding test cases. Hold up for record and playback and ability to edit scripts after recording. Different footage modes are provided in QTP viz. Normal, Analog & Low level. Excellent Object Identification process. Also Hold up for different software's like Java, Oracle, SAP, .NET, Web Forms, Siebel, PeopleSoft, Web services, Main frame (Terminal Emulator) etc. .Capability to allow you increases the existing tests even without the AUT (Application under test) from active screen. It supports all famous Automation frameworks - Keyword driven testing approach, Data driven testing approach, Modular testing approach, Hybrid frameworks etc. QTP comes with an integral IDE, which is easy and trouble-free to use. QTP inculcates VBScript which is English-like and very simple to learn and gain proficiency. We have abundance of supply accessible to learn VBScripts. Microsoft Object model can be entrenched in QTP without difficulty (Example – Word document object, Excel Object, Outlook Object, ADO objects, File system objects, DOM etc. Simple to maintain test iterations and data pouring the tests through configurations. Test with all necessary details for analysis is provided.

B. Selenium Components

To create complete automation suite for your web applications following variants of Selenium that can be used in isolation or in combination.

C. Selenium IDE

It is a Mozilla Firefox add-on which is capable of recording the user proceedings and also play-back. Since selenium IDE is an integrated development environment, Tester can also edit and debug the tests in above mentioned languages. At a basic level it allows us to convert recorded scripts into programming languages such as Java, Ruby and more. It also allows adding verification

and synchronization steps to the draft throughout the recording process. IDE provides excellent support for writing automated test scripts in Selenium and gets better with every release.

II. LITERATURE SURVEY

Literature re-examine is the effectual assessment of selected documents on a research topic. A review may form an essential fraction of the research procedure or may constitute a research project in itself. It is a critical synthesis of previous research. The evaluation of the literature leads logically to the research question.

Most published research on test automation frameworks presents case studies or feasibility studies. Artzi et al [1], describes rudimentary test computerization framework to execute feedback-directed test generation for JavaScript web applications. The author offered the case study of a precise system (rather than general or all web applications) also it needed access to AUT's source code. James M. Slack [2] described a test automation framework using "AutoIt" tool and an Excel Sheet as a Data Container for test data. The framework proposed was tool specific and the disadvantages of using excel sheet as a data container were not enclosed, as excel sheet representation is not suitable for more complex and dynamic web applications. Manpreet Kauretal [3] "Xml Schema Based Approach for Testing of Software Components" mentions apt use of XML format for representing test data.

AshutoshJha [4] described aptly some basics for implementing data driven frameworks but again it does not provide a generic model or architecture for general design. W.T.Tsai[5] described an XML based framework named Coyote which was designed for testing Web services. Again, this paper was a case study and presented no conclusions. Merchant, et. Al [6] presented a browser agnostic UI test structure for Web applications and concluded that using the structure condensed the time required to create test case scenario by 50% compared to a physical approach.

Nowadays many software applications are written as web-based applications to be run in an Internet browser. Selenium is a set of influential software tools, each with a different approach to bear automation test for testing web-based applications. It works with many encoding languages, browsers, testing frameworks and operating systems. To test its power or to analyze general performance under different load types, JMeter is used to replicate an intense load on a server, network or object. JMeter operates at the procedure-level; on the other hand, the outcome show that the new software structure improves software products quality and develops efficiency at the user-level based on the Selenium and JMeter, Fei Wang &Wencai DU includes an automatic software testing structure for web applications. We skill fully recover the extensibility and reusability of automated test with the use of the software structure. [7]

NavarajJavvaji, AnandSathiyaseelan and Uma MaheswariSelvan[8] explained Selenium Remote Control (RC) as a test tool that simulates web browsers and supports many programming languages. "Selenium-java-client-driver-1.0.1" is used in this structure for driving the browsers using java. Selenium RC provides only basic functionalities using the client drivers. But there is an effective way of expanding the functionalities that creates a customized logger file and generation of customized test summary report by implementing the thought process to capture user actions automatically.

Jeff Offutt has developed the web application bypass testing technique. The common technique in web applications is to perform the information validation on client by utilizing HTML attributes and scripting language like JavaScript. The most regular activity of web applications is to validate the user's information. Bypass testing technique is used for the web applications. This technique skips the client side testing mechanism and automatically generates the input data to verify security, performance and other quality aspects of the web applications. Ricca and Tonella proposed an analysis model and associated test strategies for analysing web page. Kung et al. have developed a model to represent the websites using graphs and provides the definitions for developing tests based on these graphs.

III. SYSTEM ARCHITECTURE

Researchers have developed the numerous testing tools for automated GUI testing and web application testing. Now it's a contest to develop a single abstract tool for testing GUI and web applications. Numerous of the testers do not have complete java information, so increasing selenium structure or using selenium IDE, RC is extremely serious and time consuming work. Keys like sauce builders have flourished in construction of Selenium with no programming knowledge and faster error elimination. Still it fall a short to offer a solution to provide file upload facility and reducing script design time.

The projected SMEASY (Selenium Made Easy) - a GUI automation structure built on the top of the Selenium that overcomes the failings of sauce builder and gives aim proved functionality. The projected structure supports Selenium 1.0 and Selenium 2.0 (web drivers). The tool is Java based, so it can be arranged and used on various platforms. Eclipse is used as backend for the gizmo being. Selenium support numerous browsers, so in our tool we have set the facility to execute the code on various browsers. The tool knowingly reduces the script making time. The framework provides the facility to upload the files from local machines and also imports data from excel and database. It produces test case flow reports and HTML reports making it simple and extra user-friendly. The focus of the structure is making it data driven, so that you can pass 'n' number of data through its structures. The most significant benefit of the automated tests is the reusability. Moreover the test maintenance is easy. This needs the preparation of the data sheets which is completely independent of the test automation tool.

The projected system will have the tool that support selenium 1.0 and selenium 2.0 (web driver).Thetool is java based so we can deploy and use this tool on any of the platforms. Selenium support many browsers so in our tool we have given the facility to execute the script on different browsers. Script creation time can be reduce and tester can easily use this tool. Creating Project, package, Method and class using selenium tool. Creating test flows using selenium tool. Data driven is the key features of data through its features.

IV. PROPOSED SYSTEM OVERVIEW

The projected automated tool is vigorous, swift and platform independent, used to organize, design and execute test cases.

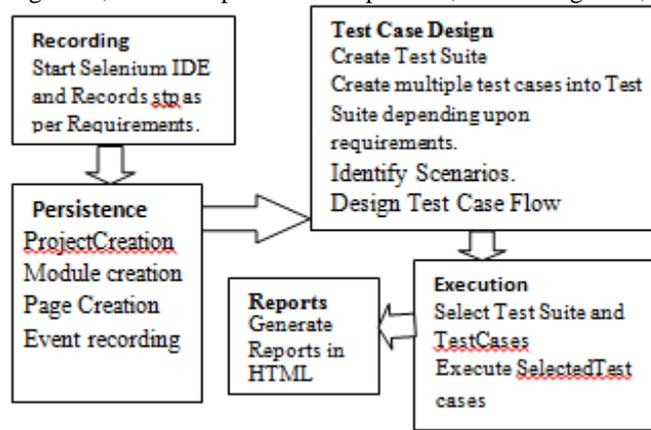


Fig. 1: Proposed System

The projected system shows two major external components.

- 1) Selenium IDE
- 2) SMEASY Automation Tool

Selenium IDE is used to read and record the web browser actions. Java source code for these events generated in Selenium IDE is copied and used in SMEASY event IDE for further processing.

SMEASY shows four main activities:

- 1) Persistence
- 2) Test Design
- 3) Test Execution
- 4) Reports

Persistence shows creation of new project, modules, pages and events. Test design creates test suites. It allows creation of multiple test cases under test suite depending upon requirements. Tester identifies the scenarios and designs the test case flow. After designing the test case flow corresponding test script is saved. For Test Execution corresponding test suite and test cases are selected and after execution simple HTML Report is generated for the user.

V. SYSTEM IMPLEMENTATION FLOW

This section will present a detailed internal functioning of the tool. The projected tool is implemented using Java technology. Software tools such as JRE 1.7, Selenium IDE, Eclipse Development are used and for testing Microsoft Windows 7, Tool with 2G/3G Internet Connection is required. Test Suite is collection of Test Cases (TC) where every TC corresponds to a test scenario. The tool works around TC designing, execution and reporting.

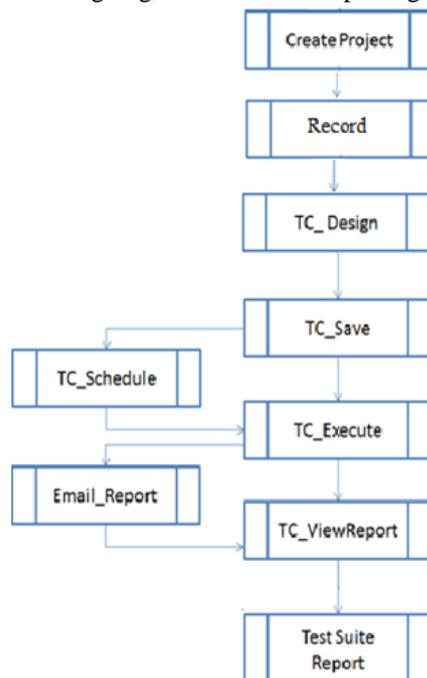


Fig. 2: System Implementation Flow

- Create Project: Tool provides a facility to create multiple Projects and multiple Test Cases under each Project. Test Case should be created in Module, which will be created under a Root.
- Record: Web Browser activity is recorded in Selenium IDE and automatically generated source code for these events is copied in SMEASY IDE.
- TC_Design: A test case is designed for a specific scenario. The fields associated to test case (TC) like project details, along with TC description are filled.
- TC_Save: TC is designed and saved. TC get saved for permanently, until explicitly deleted.
- TC_Scheduler: Once the TC is saved with its all details, it can be run at any time without manual intervention. TC scheduler makes it possible to schedule test run on particular data and time; so that one can schedule Test run when the PC is an ideal.
- TC_Execute: Finally, properly designed TC is executed. Tool shows a process status bar for user information. This tool offers a function to perform more than one test case simultaneously under one Test group.
- Email Report: Once the test run is finished, a generated report is automatically send through a mail to the person who set test schedule.
- TC_View Report: The outcome of test execution is three different types/levels of reports are generated.
- Test Suite Report: A summary report of test suite execution shows the no. of pass and fails test cases with percentage of pass/fail in one Test Suite. Report consists of fields like total execution time, total no of records, result status with graphical representation using pie chart.

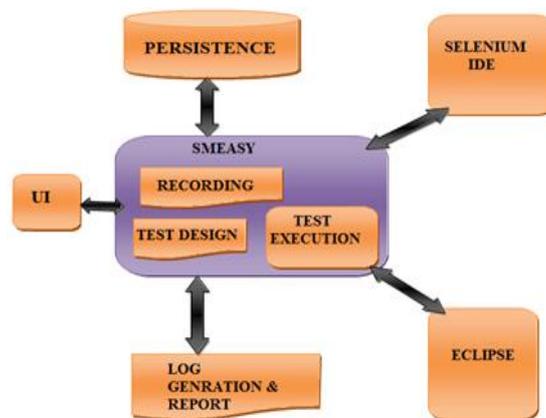


Fig 3: System Architecture Diagram

VI. CONCLUSION

SMEASY shares a significant role in making Selenium more efficient and user friendly. The tool significantly reduces the script creation time. It generates test case flow reports and HTML reports making it simple and more user-friendly. The highlight of the framework is making it data driven, so that you can pass 'n' number of data through its features. The greatest advantage of the automated tests is the reusability. Additionally the test maintenance is easy. This requires the preparation of the data sheets which is totally independent of the test automation tool. The facilities made available by SMEASY have really decreased the burden of software testers from the concern of the sound programming knowledge. Data driven functionality is the important feature that makes it extra powerful as compared to its existing solutions.

VII. FUTURE WORK

Though SMEASY has stepped ahead for improving the existing GUI automation framework systems, it shows possibilities for future enhancements. Forthcoming work on SMEASY may try to include as numerous browsers as possible into the structure. In its data driven mode, SMEASY gives a real time data set creation option. This can be improved by providing direct uploading of excel data sheets. This will save the valuable time of the tester from creating data set at real time. Also there are dependencies between the events created while performing the data driven testing. Removing these dependencies will greatly enhance the performance of the tool.

REFERENCES

- [1] Shay Artzi, Julian Dolby, Simon Holm Jensen, Anders Møller, and Frank Tip, "A framework for automated testing of JavaScript web applications", In Proceeding of the 33rd international conference on Software engineering, ICSE 11, pages 571–580, New York, NY, USA, 2011. ACM.
- [2] James M. Slack, "System Testing of Desktop and Web Applications", Information Systems Education Journal (ISEDJ). Volume 9, No. 3 August 2011.
- [3] Manpreet Kaureta, "XML Schema Based Approach for Testing of Software Components", International Journal of Computer Applications, Volume 6– No. 11, September 2010

- [4] AshutoshJha“Development Of Test AutomationFramework For Testing Avionics Systems”,Digital Avionics Systems Conference (DASC), 2010, IEEE.
- [5] W.T.Tsai, R. Paul, S. Weiweiand C. Zhibin, "Coyote: an XML-based Framework for Web Services Testing", Proceedings ofthe7thIEEE International Symposium on High Assurance Systems Engineering, 2002,pp. 173-174.
- [6] C.Merchant, M.TellezandJ.Venkatesan, "A Browser yu6Agnostic Web Application UI Test Framework
- [7] FeiWang,Wencai DU, “A Test Automation Framework Base on WEB”, IEEE/ACIS 11th International Conference on Computer and Information Science,2012
- [8] NavarajJavvaji, AnandSathiyaseelan, Uma MaheswariSelvan, “Data Driven Automation Testing of Web Application using Selenium”, STEP-AUTO,2011