

## SYSTEMATIC STUDY OF A WEB TESTING TOOL: SELENIUM

**Chandraprabha<sup>1</sup>, Ajeet Kumar<sup>2</sup>, Sajal Saxena<sup>3</sup>**

<sup>1</sup>*Research Scholar, SRMSCET, Bareilly (India)*

<sup>2</sup>*Associate Professor, SRMSCET, Bareilly (India)*

<sup>3</sup>*Technology Analyst, Infosys, Pune (India)*

### ABSTRACT

*In software testing, test automation means to use special software to control the execution of tests and the comparison of actual outputs to predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually. Selenium is an open source automated testing suite for web applications across different browsers and platforms which supports multiple programming language .Selenium is a functional web testing tool. Selenium is not just a single tool even it is a suite which has four components: Selenium IDE, Selenium RC, Selenium Web Driver and Selenium Grid. Selenium IDE is a Firefox plug-in which is used to develop test cases, Selenium RC run tests inside every JavaScript compatible browser using a wide range of programming language, Web Driver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded and Selenium Grid allows you to run your tests on different machines against different browsers in. In this research paper firstly, we will discuss the different components of Selenium suite, architecture of Selenium components and then we will discuss the limitations of these components are and then way to enhance them.*

**Keywords:** *Selenium Core, Selenium IDE, Selenium RC, Web Driver, Selenium Grid, Selenium2.*

### I INTRODUCTION

Automated software testing is an essential element component of successful development projects. Automation testing draws maximum benefits with minimum efforts by increasing the quality and reliability of the products. There are various tools which are used in automation testing. These tools are able to playback pre-recorded and predefined actions compare the results to the expected behavior and report the success or failure of these manual tests to a test engineer. Once automated tests are created they can easily be repeated and they can be extended to perform task impossible with manual testing. There are various advantages of test automation like it improves accuracy, saves time and money, test coverage, it can do which manual testing cannot, and it improves team morale. There are number of commercial and open source tools are available for assisting with the development of test automation. In this paper we will discuss selenium web testing tool. The advantage of software testing process is to identify all the defects existing in a software product. Software testing provides a practical way of reducing defects existing in a system and increasing the user's confidence in a developed system. The solution

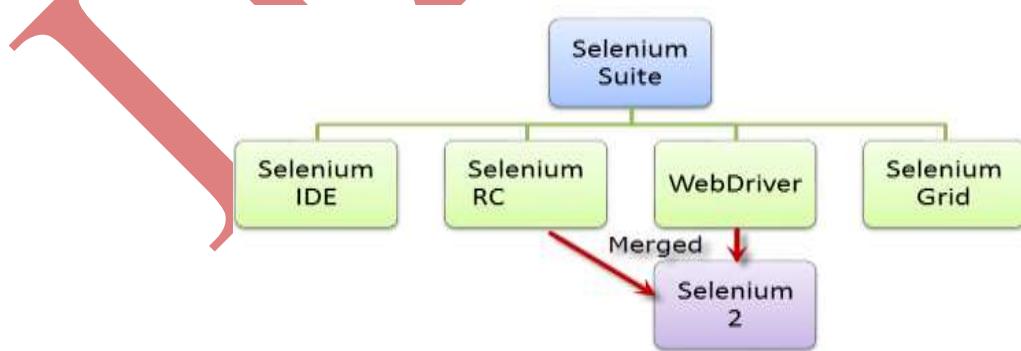
of this problem is automation testing. It is the process of automating the manual test case using software test tool for example Selenium.

### 1.1 Advantages

- 1) Automated Software Testing saves Time and Money- The code for the same object can be used across different applications. Duplication of work is minimized at every level which saves time. A time savings that translates directly into cost savings.
- 2) Improves Accuracy- In manual testing tester make many mistakes but in automated testing, tests perform the same steps precisely every time they are executed and never forget to record detailed results.
- 3) No scripting skills are required by the end user- No coding skills are required to automate and review the scripts. The scripts are user friendly with good readability.
- 4) Reliability- As the test cases are stored and well maintained, if any error comes, we can easily check against that error. So, reliability is here.

## II SELENIUM SUITE

Selenium is a browser automation tool, commonly used for writing end-to-end tests of web applications. Selenium automates the control of a web browser so that repetitive tasks can be automated. Selenium is an open source tool and writing tests in various languages like java, C#, Ruby and Python. Selenium is a suite of four components. First of these is Selenium IDE, which is an extension for Firefox that allows users to record and playback tests. Second component is Selenium RC which is a server written in java, that accepts commands for the browser via HTTP. Third component in the suite is Selenium Webdriver which provides APIs in variety of languages to allow for more control and the application of standard software development practices. The final component is Selenium Grid, which makes it possible to use the Selenium APIs to control browser instances distributed over a grid of machines, allowing more tests to run in parallel.



### 2.1 Selenium Core

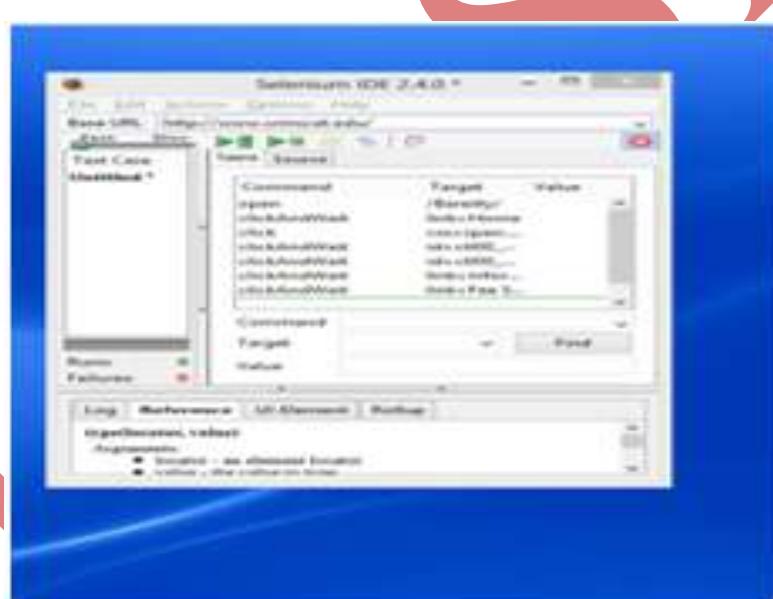
Selenium was created by Jason Huggins working in Thought Works in 2004. He was working on a web application that required frequent testing. He realized that manual testing repetition was becoming more and

more inefficient, he created a JavaScript program that would automatically control the browser's action. He named this program "JavaScriptTestRunner". Later he made this JavaScriptRunner open source which was later re-named as Selenium Core.

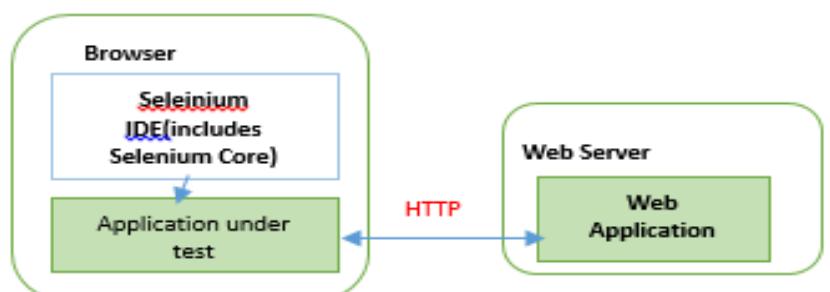
## 2.2 Selenium IDE

Selenium IDE (Integrated Development Environment) is a tool to develop Selenium testcases. Selenium IDE was originally created by Shinya Kasatani and donated to Selenium project in 2006. Selenium Ide is implemented as a Firefox extension. It allows recording, editing and debugging tests. It previously known as Selenium Recorder. On start-up of the Firefox plugin, the record feature is automatically turned on, allowing the user to record any action done inside the web page.

In Selenium IDE scripts may be automatically recorded and edited manually providing auto-completion support and ability to move commands around quickly. In Selenium IDE scripts are recorded in Selenese, a special test scripting language which is a set of Selenium commands. Selenese is used to test web application. Selenium commands are basically classified into three categories-actions, accessors and assertions.



### 2.2.1 Architecture of Selenium IDE



### 2.2.2 Features of Selenium IDE

- Simple and easy record and playback.
- Supports intellectual field selection options like ID's, XPath and Names.
- Supports for Selenium user- extensions.js file.
- Clever option for adding different asserts options in scripts.
- Save test scripts in several formats like Selenese, Ruby etc.
- Supports Walkthrough tests.
- Allow to set breakpoints and debug the scripts.
- Supports auto complete commands.
- Allow to customization through plug-ins.

### 2.2.3 Limitations to Selenium IDE

- Selenium IDE tool can only be used in Mozilla Firefox as it is an Add-on for only that browser and hence it cannot be used with other browsers.
- It is a record and playback tool and the script format can be written in Selenese.

## 2.3 Selenium RC

Selenium IDE has the limitation of “same origin policy”. To overcome this problem, ThoughtWork’s engineer Paul Hammant decided to create a server that will act as HTTP proxy to “trick” the browser into believing that Selenium Core and the web application being tested come from the same domain. This system known as Selenium Remote Control or Selenium1. With Selenium RC it is possible to run tests inside every JavaScript compatible browser using a wide range of programming language. Selenium RC has two components-

### 2.3.1 Selenium Server

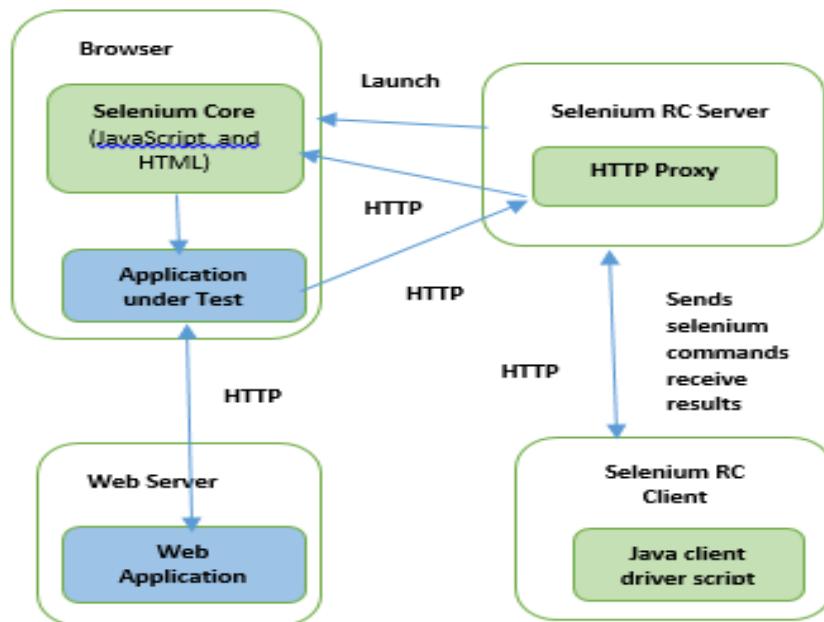
Selenium Server receives Selenium commands from test program, interprets them, and reports back to program the results of running those tests. Selenium Core is a JavaScript program, actually a set of JavaScript functions which interprets and executes Selenese commands using the browser’s built-in JavaScript interpreter. Selenium Server receives Selenium commands from test programs, interprets them, and reports back to the program the results of running those tests. The RC Server bundles Selenium Core and automatically injects it into the browser. This occurs when the test program opens the browser (using a client library API function). Selenium Core is a JavaScript program, actually a set of JavaScript functions which interprets and executes Selenese commands using the browser’s built-in JavaScript interpreter. The Server receives the Selenese commands from the test program using simple HTTP GET/POST requests. This means any programming language can be used, that can send HTTP requests to automate Selenium tests on the browser.

### 2.3.2 Client Libraries

Client libraries which provide the interface between each programming language and the Selenium RC Server. The client libraries provide the programming support that allows you to run your own design. There is a different client library for each supported language. A Selenium client library provides a programming interface

(API), i.e., a set of functions, which run Selenium commands from your own program. There is a programming function that supports each Selenese command.

### 2.3.3 Selenium RC Architecture



### 2.3.4 Limitations to Selenium RC

- Selenium RC is slow.
- Selenium RC struggles when running concurrent tests.
- Selenium RC does not allow simultaneously tests across different OS and browsers.

## 2.4 Selenium Web driver

Simon Stewart created WebDriver circa2006 when browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core .Web Driver was the first cross-platform testing framework that could control the browser from the OS level. Selenium Web driver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API. Selenium Web Driver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded. Web Driver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems. Web Driver is the name of the key interface against which tests should be written in Java, the implementing classes one should use are Android Driver, Chrome Driver, Event Firing Web Driver, Firefox Driver, HtmlUnitDriver, Internet Explorer Driver, iPhone Driver, Phantom JS Driver, Safari Driver.

Selenium Web Driver is the successor to Selenium RC. Selenium Web Driver accepts commands (sent in Selenese or via client API) and sends them to a browser. This is implemented through a browser-specific which sends commands to a browser, and retrieves results. Most browser drivers actually launch and access a browser

application; there is also an HtmlUnit browser driver, which simulates a browser using HtmlUnit. Unlike in Selenium1, where the Selenium RC server was necessary to run tests, Selenium WebDriver does not need a special server to execute tests. Web Driver directly starts a browser instance and control it. Selenium Grid can be used with Web Driver to execute tests on remote systems.

#### 2.4.1 Features of Webdriver

- WebDriver is designed in a simpler and more concise programming interface along with addressing some limitation in Selenium RC API.
- It drives the browser much more effectively and over comes the limitations of Selenium 1 which affected our functional test coverage, like file upload or download, pop-ups and dialog barriers.
- WebDriver overcomes the limitation of Selenium RC's Single Host Origin Policy.
- WebDriver can support the headless HtmlUnit browser.
- WebDriver interacts with page elements in a more realistic way.

#### 2.4.2 Limitations to Webdriver

- WebDriver cannot readily support new browsers, but Selenium RC can.
- It does not have a built-in command for automatic generation of test results.

### 2.5 Selenium Grid

Selenium Grid allows you to run your tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems. Selenium Grid support distributed test execution. Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the Hub. Tests contact the hub to obtain access to browser instances. The hub has a list of servers that provide access to browser instances (WebDriver nodes), and lets tests use these instances. Selenium Grid allows running tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally (instead of in each individual test).

- Test client sends commands to the Selenium Hub.
- Selenium Hub redistributes tests on the Grid environment
  - (i) Each server can have a different environment.
  - (ii) You target your tests to a specific environment.
- Selenium RC servers run on servers and drives browser instances.
- Tests run in parallel against the application server.

#### 2.5.1 Selenium-Grid 1.0

Version 1 was the first general release of Selenium Grid. Selenium Grid1.0 has its own remote control that is different from the Selenium RC server. They are two different programs. This version can only support Selenium RC command/scripts. Selenium Grid1 can only automate one browser per remote control.

#### 2.5.2 Selenium-Grid 2.0

Selenium-Grid 2.0 is the latest release as of the writing of this document (5/26/2012). It is quite different from version1 of Selenium Grid. In 2.0 Selenium-Grid was merged with the Selenium- RC server. Now we only need to download a single .jar file to get the remote Selenium- RC Server and Selenium Grid all in one package.

### 2.5.3 Difference between Selenium Grid 1.O And Selenium Grid 2.O

Selenium-Grid 1.0	Selenium-Grid 2.0
Selenium-Grid1.0 has its own remote control that is different from the Selenium RC server. They are two different programs.	Selenium Grid 2.0 is bundled with the Selenium Server jar file.
Selenium Grid1.0 can be used only after installation and configuration of Apache Ant.	Installation of Apache Ant is not required in Selenium Grid2.0
Selenium Grid1.0 can only support Selenium RC command/scripts	Selenium Grid2.0 can support both Selenium RC and WebDriver scripts.

### 2.5.4 Features of Selenium Grid

- Selenium Grid transparently distributes our tests across multiple physical or virtual machines so that we can run them in parallel
- Selenium Grid cutting down the time required for running tests.
- Selenium grid dramatically speeds up testing, giving us quick response and accurate feedback.

### 2.6 Selenium 2.0

Selenium is a well-known web application testing framework used for functional testing. Selenium2 merges the best features of Selenium1 and WebDriver. Selenium 2 overcomes the following drawbacks of Selenium RC-

- The driver will directly drive browser using WebDriver.
- Parallelization is easier: the test code takes care of it. Selenium 2 comes as a browser extension directly plugging into the browser automation feature. It avoids the JavaScript security model that restricts Selenium RC security. In Selenium2 test cases can be written in Ruby, Python, Java, C#.

## III CONCLUSION

Selenium is a very good framework for testing the user interface of a web application. It is an extremely powerful tool for running end-to-end functional tests. We can write tests in several programming languages and Selenium executes them into one or multiple browsers. In this research paper we did a systematic study of all components of Selenium suite. What are the limitations and how we can overcome those problems?

## REFERENCES

1. Olawa, Adam; Huizinga, Dorota (2007). Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press. p. 74. ISBN 0-470-04212-5.

2. Selenium Documentation release 1.0, 06 Aug, 2010.
3. Antawan Holmes and Marc Kellogg., "Automating Functional Tests Using Selenium", Proceedings of AGILE 2006 Conference, IEEE, ISBN 0-7695-2562-8/2006.
4. ] Eun Ha Kim, Jong Chae Na, and Seok Moon Ryoo, "Implementing an Effective Test Automation Framework", 33rd Annual IEEE Computer Software and Applications Conference, 2009.
5. Nidhika Uppal, Vinay Chopra "Design and Implementation in Selenium IDE with Web Driver" International Journal of Computer Applications (0975 – 8887) Volume 46– No.12, May 2012

IJARSE