Enhancement and Elimination of Roadblocks in Automation Testing Tool Selenium RC

Nidhika Uppal
AP, IT DEPARTMENT
GIMET,Amritsar

Vinay Chopra
AP,CSE DEPARTMENT
DAVIET,Jalandhar

ABSTRACT

Selenium is a suite of tools used in testing web applications. In this paper Firstly we described the selenium RC does not support model dialog and its effect when modal is opened, the accessibility to parent window is blocked and similarly window switching. Secondly how we enhance and to overcome this limitations we had created some methods. After updated in selenium server the test execution continue even after opening of the model dialog and window switching. When a popup opens another popup, it becomes necessary that each window should be identified uniquely in order for selenium to identify them correctly. It needs to move its handle across windows to perform its operation.

Keywords

Selenium RC(remote control)

1. INTRODUCTION

Selenium suite of tools is used for web application automation testing across all platforms. Selenium tests can be written in many programming languages and testing frameworks which can run on many browsers and operating systems and automatically validate functionality on a single click.Selenium IDE is used to create test cases quickly in Firefox and Selenium Remote Control runs test cases in multiple browsers and platforms and Selenium Grid is used to run tests on different servers. Selenium supports many commonly used programming languages like Java, Ruby, Groovy, Perl, Python and PHP and can be deployed on Windows, Linux and Mac platforms[1].Selenium is a framework of tools that enables development of test automation for web-based applications, compatible across various browsers. Selenium has almost all the features required to automate tests for web based applications and more from usability perspective. Open source Test Automation tool for web applications, javascript based.Selenium tests run directly in a browser, just as real users do.One of Selenium's key features is support the tests on multiple browser and operating systems[2].Selenium is a robust set of tools that supports rapid development of test automation for web-based applications. Selenium provides a rich set of testing functions specifically geared to the needs of testing of a web application. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behaviour .

1.1Selenium-RC:

Selenium Remote Control offers a more flexible and more complex approach to create run browser tests.With selenium RC it is possible to run tests inside every javascript compatible browser using wide range of programming languages. Your tests issue commands which the client library sends to the server. The server then 'runs' your actions for you in the browser and reports the results back to your client. Using Selenium-RC allows you to write automated tests in any supported programming language. Tests written in this way allow you to use standard programming practices to make them easy to maintain, robust and easy to collaborate on as a team. Selenium RC is a server, written in Java, that accepts commands for the browser via HTTP. RC makes it possible to write automated tests for a web application in any programming language, which allows for better integration of Selenium in existing unit test frameworks[3]. To make writing tests easier, Selenium project currently provides client drivers for PHP, Python, Ruby, .net, PERL and Java. The Java driver can also be used with JavaScript. A new instance of selenium RC server is needed to launch html test case – which means that the port should be different for each parallel run. However for Java/PHP test case only one Selenium RC instance needs to be running continuously[4].

Selenium RC components are:

1.1.1 Selenium Server

Selenium Server receives Selenium commands from your test program, interprets them, and reports back to your 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 program, 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[4].

1.1.2 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 Selenium commands from a program of 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. Within each interface, there is a programming function that supports each Selenese command[4].

2. PROBLEM FORMULATION WITH SELENIUM RC

Selenium works on Javascript. It needs to move its handle across windows to perform its operation. When showModalDialog is called the javascript gets suspended for the parent. Selenium whose handle is still pointing the parent window also gets suspended. As a result all the successive commands in Selenium script ultimately get suspended and automation gets blocked. To retain the normal flow of the selenium the only solution left is to bypass showModalDialog call with normal 'window.open' function call. To pass the argument from parent to modal we need to save the arguments in a variable on the parent window. Then we will inject a code on the modal window that will read this value from the parent and save it in 'window.dialogArguments' variable.If we open the window as non-modal, the biggest challenge will be to pass the return value from the modal back to the parent window. This is because when we open a window using 'window.open' commands the javascript do not wait for the return value and soon comes out of the function without performing its intended operation after retrieving the return value from the modal.[2] To overcome this we will call the instruction (probably a button click) that calls the showModalDialog function twice. First time when we do a click it will open the non-modal dialog and comes out of the function. On the second click it will use the return value that was saved by the non-modal dialog to perform post-showModalDialog operations. All the operations to be performed on the popup window will go in between the first and the second clicks. Internet Explorer has provided additional function, showModalDialog, to deal with Modal type windows. When we open a window using showModalDialog the java-script execution gets suspended till the window gets closed.[3] With this feature in place parent window can set itself under wait state expecting return value from the popup window. The popup window before closing itself needs to set its returnValue property, which will be used by the parent window.The parent window opens up a modal window and waits for the modal to return some value. The user enters some text in textbox on the modal window and clicks the link.

3. IMPLEMENTATION IN BLOCKING OF SELENIUM RC

A) To handle all modal dialog windows , Currently selenium RC does not support modal dialogs, add support for modal dialogs.

For Modal Popup problem: Opening a modal pop up in Firefox by using window.showModalDialog suspends execution of test as java script waits for popup to be closed by user and test being executed by the user stops. To overcome this problem we need to overrid showModalDialog function to open an separate window with URL supplied to Modal dialog page. This is achieved by creating a function to convert call to showModalDialog into open new window.

1. Updated selenium firefox launchers to do following:
    a. Updated Mozilla browser bot to open a new window every time modal popup is opened.
    b. Page to be opened in modal window will now be opened in new window.
    c. As page is opened in new window the test execution continues even after opening of the popup.

B) To enhance browser window switching ,Selenium RC has limitation in switching windows add functionality to switch windows properly.

Window switching API has been updated to implement new functions to selectwindows by giving sequence number in which window was opened, or by using selectNewWindow to open new open window. Window switching API will provide more functions than just one function provided by selenium, this will save time and maintenance of scripts with changing window identifiers like window title, window name or window id that are used by the selenium in selectWindow function to identify windows for redirecting test steps. With the new function need of updating scripts every time new window properties are changed will not be required and maintenance efforts will be reduced.

Changes done to achieve switching window:

1. Implemented a new method in selenium that changes the way of window switching.
2. Created method to check window and select a window with defined sequence.
3. Created a method for switching to new open window.
4. Added more flexibility and reduced risk of typing mistakes while typing long window names.

Implementation details:

1. Updated selenium server and added functions in selenium server files to do following:
    a. Select a window by sequence in which it was opened.
    b. Switch to new window by using command "SelectNewWindow".
    c. Updated selenium server files to handle switching with error handling.
2. Multiple modals : When a popup opens another popup, it becomes necessary that each window should be identified uniquely in order for selenium to identify them correctly.

4. CONCLUSION

In this paper describe the flaws in Selenium RC tools, find workaround to those and implement them in selenium. Selenium RC does not support modal dialogs, add support for modal dialogs after creating some methods the child window gain focus with parent window. So both windows can activated simultaneously. In selenium RC also not support window switching to add that functionality we had created some methods to remove this limitations so ,that we can run both windows parallel.

5. REFERENCES

[1] Selenium Documentation release 1.0, 06 Aug, 2010.

[2] Selenium http://en.wikipedia.org/wiki/Selenium

[3] http:// open2test.org

[4] P.Nirmaladevi, "Effective Automating testing with web application Using Selenium",International Journal of Communications and Engineering , 2012

[5] prasanth yalla ," framework for testing web application using selenium testing tool with respect to integration testing" ",International Journal of Computer science and technology , 2011

[6] 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

[7] Ali Mesbah, Arie Van Deursen," Invariant-Based Automatic Testing of Ajax User Interfaces ", Proceeding of the 31st ACM/IEEE International Conference on Software Engineering , 2009

[8] Mark Grechanik, Qing Xie, and Chen Fu, " Maintaining and Evolving GUI-Directed Test Scipts " 31st International Conference on Software Engineering(ICSE'09),2009

[9] Zeng Wandan, Jiang Ningkng, Zhou Xubo " Design and Implementation of Web Application Automation Testing Framework " Ninth International Conference on Hybrid Intelligent Systems,2009

[10] Leckraj Nagowah and Purmanand Roopnah " AsT-ASimple Automated Syetm Testing Tool " 3rd International Conference on Computer Science and Information,2010