Framework for Testing Web Applications using Selenium Testing tool with respect to Integration Testing

Prasanth Yalla, Dr. L S S Reddy, M.Srinivas, T.Subha Mastan Rao
1Dept. of Freshman Engineering K.L.University, Guntur dist, AP India.
2LRBR College of engineering, Mylavaram, Krishna District. AP, India.
3Dept.of Freshman engineering, K L University, Guntur Dist., AP, India.
4Dept. of IST, K L University, Guntur Dist., AP, India.

Abstract
Testing has become most important parameter in the case of software development lifecycle (SDLC). The web application testing is more complicated as it involves different testing strategies and tools, because the tool has to support different characteristic properties and also different frameworks. Integration testing focuses the verification effort on the smallest unit of software design, the software component or module using top-down or bottom-up approach.

The main objective of this paper is to propose a framework that integrates or supports the architectural elements that have been already compared with respect to integration testing and finally doing the same by using the testing tools like QTP, Win runner and Load Runner. In this paper, we have compared the architectures of two tools and have come-up with different possibilities that are leading towards proposing a common framework. Also, in order to test any 2-tier web application with respect to Integration Testing test cases are required. Here, a framework is proposed that satisfies the test requirements.

In our proposed framework, we have taken a tool Selenium under integration testing and tested several web applications

Key words
Framework, Web application, Selenium, test case, testing tool, client, server.

I. Introduction
This paper deals with the testing of Web Applications using a testing tool named Selenium under Integration Testing. The main intension of the paper is to test several Web Application and also to verify the errors if they are present in it. Several test cases and requirements traceability logs were also considered for this.

Terminologies:

Framework
A framework is a defined support structure in which another software project can be organized and developed. A framework may include support programs, code libraries, a scripting language or other software to develop different components of a software project. A web application framework is a software framework that is designed to support the development of dynamic websites, web applications and web services. The framework aims to alleviate the overhead associated with common activities performed in Web development. For example, many frameworks provide libraries for database access, templating frameworks and session management etc and they often promote code reuse.

Web Application
A web application is an application that is accessed over a network such as the internet or an intranet. The term may also mean a computer software application that is hosted in a browser-controlled environment (e.g. a Java applet) or coded in a browser-supported language (such as JavaScript, combined with a browser-rendered mark-up language like HTML) and reliant on a common web browser to render the application executable. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility. Common web applications include webmail, online retail sales, wikis etc.

Two (2) – Tier Web Application
Two-tier architecture is client/server architecture, where a request to do some task is sent to the server and the server responds by performing the task. A Two-tier architecture is a client/server application. Client acts as Front End and Server acts as Backend like databases.

Fig.1.2.3: Two-Tier Architecture
The two-tier model is the one that most computer users are familiar with; it’s the basic foundation of the Internet. Much of the Internet, from web browsing to e-mail, is based in some fashion on this architecture. The web browser (client) sends a request to a web server, which determines what the user is looking for and sends the information back to the client. Two tier client-server design derives its name from how it distributes work between client and server. Clients access databases through the user system interface. Database management, on the server side, distributes processing between both client and server. Both tiers, the client and the server, are responsible for some of the processing management. Simply put, the client uses the user interface to make requests through database management on the server side.

Client Tier
At the client level we have either a Java client application or a web browser as the client. By using a web browser as the client allows the entire application to be located on the server. The client always gets the latest version when they start the application. Also the client can use any web browser from any location provided that they can connect to the application server. The client tier can also consist of an application running on the clients’ desktop (usually presenting a rich user interface) and connecting directly or through HTTP to the application server. This requires that the client have the application installed on their desktop.
Clients Side Elements
Html, xml, JavaScript, VB Script
Server Tier : The server is the central computer who gives instructions to the client computer, to make sure everyone can logon on any computer (client) can share the resources of sever like information, peripherals etc The server allow/restrict the client computers to perform any task
Server Side Elements: JSP, ASP, Data Mgmt Components, Security Components and other Components like Communication Components

Integration Testing
Integration testing (sometimes called Integration and Testing, abbreviated “I&T”) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. Integration testing is of two types
i) Top Down Integration
Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module (main program). Modules subordinate (and ultimately subordinate) to the main control module are incorporated into the structure in either a depth-first or breadth-first manner.
ii) Bottom Up Integration
Bottom-up integration testing, as its name implies, begins construction and testing with atomic modules (i.e., components at the lowest levels in the program structure). Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubs is eliminated.

Objectives
The main objective of this paper is to test any kind of web application using an automated integration testing tool. We have chosen an Integration testing tool named selenium. Generated the code in different languages that includes JAVA, C# etc.
1.4 Scope: The scope of the testing the Web Application using Selenium under Integration Testing is as follows:
• The user can use any operating system, but should have Firefox browser.
• The user should have Selenium tool to test the Web Applications.
• The user can get the coding in many languages like C#, JAVA, Ruby, Python, etc
1.5. Purpose: The purpose of testing web applications using Selenium because of following features:
• Even non-developers can develop tests in the easy-to-use, graphical Selenium IDE and export them as JUnit tests.
• You have a choice of target browsers to run the tests. Selenium supports browsers like Internet Explorer, Firefox and Safari. This is in contrast to tools like HTMLUnit and Canoo, which have their own implementations of a browser, or Watir/Watij, which always use Internet Explorer. Selenium supports Ajax calls.
• When a test fails, you can send an email with a Screenshot of the browser, choosing selenium.
• You can view Selenium test scripts even in browsers that don’t have the Selenium IDE/plug-in installed.
• Selenium Grid enables you to run tests in parallel on multiple servers reducing the time efforts

II Requirements

Testing Requirements
• Select an integration group
• Define test conditions
• Determine lowest level module for each test condition
• Reassign condition according to testability
• Select test conditions for a module’s initial test
• All test conditions assigned to a module Grouped into test cycles
• Construct integration test cycle dependency diagram
• Estimate effort for testing a test cycle
• Revise estimates on integration group dependency diagram

Functional Requirements:
In software engineering, a functional requirement defines a function of a Software system or its component. A function is described as a set of inputs, the behaviour, and outputs (see also software). In general, functional requirements define what a system is supposed to do. Functional requirements are usually in the form of “system shall <do requirement>”. In our paper, the Functional Requirements are Selenium tool, Firefox web Browser, Web Application

Non – Functional Requirements:
In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non functional requirements can be divided into two main categories 1. Execution qualities, such as security and usability and Evolution qualities, such as testability, maintainability etc.

Core Requirements:
• Browse urls
• Render web pages
• Mimic User Actions
• Validate results
• Ajax/CSS support

Pragmatic requirements:
• Language choices
• Platforms (Browser/OS)
• JS toolkit support
• Distributed tests
• Scheduled execution
• Reporting
• Configurability
• Build Integration
• Test Creation Environment

System Requirements Specifications:
Software Specifications:
• Operating System: Windows XP, MAC
• Programming Language : C, CPP
• Graphical User Interface : Java
• Documentation Tool : Ms Access
Tools / Platforms:
• Java compiler (JDK 1.6.0_10)
• Selenium Tool
• Firefox Browser
• Rational Rose

III. System Analysis:
System analysis is the dealing with analysis of sets of interacting entities, often prior to their automation as computer systems and the interactions with in those systems.

System Study:
The following are different Modules:
• Recording-to record the web application
• Testing- used to test the Web Application
• Code Generation - to generate the code
• Results-used to report the Runs and Failures occurred while testing

Existing System:
The existing system is manual or usage of the automated testing tools like win runner and load runner etc. Manual Testing takes more time efforts for a window-based or a web-based application. Even the automated testing tools like win runner have some limitations. All the test cases that are related to the software must be tested thoroughly in order to avoid the upcoming errors during the execution of the software.

Proposed System:
In the proposed System, the process of testing can be implemented in a different way with the help of integration testing tool Selenium. Failures can be easily detected. Whenever the test case is executed, the code is converted into script in java. Our paper proposes a framework with respect to some parameters that support the integration testing and we also proposed a web application that can be going to be tested by using a testing tool i.e., Selenium with respect to integration testing that helps to test the application and also helps to find the errors easily.

Feasibility Study:
Feasibility is a measure of whether something can be done. Feasibility also means that it can likely be completed successfully. Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

IV. DESIGN

Architecture:

The Selenium architecture mainly consists of two components Selenium IDE – The Selenium IDE is a Firefox plug-in that can be used to record tests as they run in the browser. It plays back the tests to simulate a user’s interactions with the browser.
Selenium Remote Control (RC) – Selenium RC consists of two components:
1. Selenium RC Server
2. Selenium RC Client
Selenium RC server – The server listens for connections from Selenium clients. It has the ability to launch a browser and control the browser to play back tests.
Selenium client(s) – A client is a set of libraries (in Java, Ruby, PHP, or .NET) that you can use to communicate with the Selenium server and send messages to the Selenium server to control the browser. The Selenium client runs the test script that you develop.

Fig. 3.3 Proposed Architecture

Working of the system: Selenium IDE
Selenium IDE is the only flavour of Selenium which allows you to record user action on browser window. It can also record user actions in most of the popular languages like Java, C#, Perl, Ruby etc. This eliminates the need of learning new vendor scripting language. For executing scripts created in these languages, you will need to use Selenium Remote Control. If you do not want to use Remote Control than you will need to create your test scripts in HTML format. Selenium can be accessed from tool --> Selenium IDE in your browser toolbar if the installation is completed successfully. As compared to most of the test automation tools it is very simple and lightweight. The small red button on the right hand side gives you an indication on whether Selenium is in recording mode or not. Also, Selenium IDE will not record any operation that you do on your computer apart from the events on Firefox browser window. So go ahead read your mail, open a word doc or do anything else, Selenium will record only your actions on browser. Other options present on the Selenium IDE toolbar are related to test execution. Run will execute the tests with the maximum possible speed, Walk will execute them with relatively slow speed and in step mode you will need to tell Selenium to take small steps. Final button present on the Selenium IDE toolbar is the Selenium Test Runner. Test Runner gives you nice browser interface to execute your tests and also gives summary of how many tests were executed, how many passed and failed. It also gives similar information on commands which were passed or failed. Test Runner is also available to tests developed in HTML only. If you open the option window by going to Option, you will see there are some self explanatory options available. For example, encoding of test files, timeout etc.
Selenium RC
Selenium RC allows the test automation expert to use a
programming language for maximum flexibility and extensibility in developing test logic. For example, if the application under test returns a result set and the automated test program needs to run tests on each element in the result set, the iteration / loop support of programming language’s can be used to iterate through the result set, calling Selenium commands to run tests on each item.

Selenium RC provides an API and library for each of its supported languages. This ability to use Selenium RC with a high level programming language to develop test cases also allows the automated testing to be integrated with the project’s automated build environment.

Automated Integration Testing with Selenium applications:
Automated integration tests can be useful particularly for the following types:
* Existing applications that haven’t run any unit tests (e.g., legacy applications)
* CRUD applications that have a very simple middle tier and therefore don’t have/require unit tests
* Applications that have business logic tightly coupled to the environment in which they run (e.g., business logic embedded in DAOs or servlets)

Options for implementing automated integration tests:
* Use a different framework for each tier
* Use tools, such as Watir or Watij

To create and run integration tests with Selenium, you must complete the following steps:
1. Use the Selenium IDE to record and play tests.
2. Export tests created with the Selenium IDE as JUnit tests.
3. Add the JUnit tests to your Java project in your IDE and run the tests.

Optionally, you may choose to run the tests as part of the build process using Ant or Maven as well.

1. Use Selenium IDE to record and play tests.
2. Export tests created with the Selenium IDE as JUnit tests.
3. Add the JUnit tests to your Java project in your IDE and run the tests.

Here are the steps for recording and playing test with Selenium:
1. Install Selenium IDE in Firefox.
2. Start browsing the web application that you wish to test.
3. Click on the red button in the top-right corner of the Selenium IDE. Selenium will record your actions as you click on links, fill in forms, and submit forms.
4. Click on the red button in the top-right corner to stop the recording. Selenium will have recorded the tests.

The case above does the following:
1. Types ‘test’ into a text field with name ‘text’
2. Selects an option with the label ‘three’ from a dropdown box named ‘choice’
3. Selects a radio button with value ‘button2’ from a radiobutton group with name ‘radiogroup’
4. Clicks a submit button
5. Waits for the page to load (timeout after 30 sec)
6. Verifies the text ‘test’ is present

This test case can be exported as a JUnit test. To export the test case in the Selenium IDE, select Options->Format->Java(JUnit)-Selenium RC from the menu.

Here is the JUnit version of above test case selenium.open("/selenium-1.0/test"); selenium.type("text","test"); selenium.select("choice", "label=Three");elenium.click("//input[@name='radiogroup' and @value='button2']"); selenium.waitForPageToLoad("30000"); verifyTrue(selenium.isTextPresent("Text entered: test"));

Add the JUnit Tests to your Java project in your IDE and run the tests.

Import selenium-java-client-driver-0.9.2.jar and selenium-server-coreless-1.0-0081010.060147.jar into your IDE (Eclipse, NetBeans, IntelliJ, etc.) project. Create a new JUnit4 test case and add the code snippet above to your test method.

```java
@RunWith(Suite.class)
@Suite.SuiteClasses ({ SimpleTest.class })
public class SeleniumTestSuite extends TestSuite {
    private static SeleniumServer seleniumServer;

    public static Test suite() {return new JUnit4TestAdapter (SeleniumTestSuite.class);};
    @BeforeClass
    public static void init() throws Exception{
        //Start selenium server
        System.out.println("***Starting selenium ***");
        RemoteControlConfiguration seleniumConfig=new
```
RemoteControlConfiguration();
SeleniumConfig.setPort(SeleniumConfig.getPort()); seleniumServer = new SeleniumServer(seleniumConfig);
seleniumServer.start();
SeleniumTestHelper.getInstance().init(); System.out.println("*** Started selenium ***");

public static void destroy(){
   //Stop selenium server
   System.out.println("****Stopping Selenium****");
   SeleniumTestHelper.getInstance().getSelenium().stop();
   seleniumServer.stop();
   System.out.println("***Stopped selenium ****");}

The init method in the JUnit4 test suite above will start the Selenium server. The destroy method will stop the Selenium server. Start the application server with the application that you are trying to test and then run the JUnit4 tests. The JUnit4 init and destroy methods will be called once only for each run and they will start the Selenium server. Lastly, the JUnit4 test cases will be run.

Run a Test Suite
Click the Run All button to run all the test cases in the currently loaded test suite.

Stop and Start
The Pause button can be used to stop the test case while it is running. The icon of this button then changes to indicate the Resume button. To continue click Resume.

Stop in the Middle
You can set a breakpoint in the test case to cause it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click, and from the context menu select Toggle Breakpoint.

Start from the Middle
You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a start point, select a command, right-click, and from the context menu select Set/Clear Start Point.

Run Any Single Command
Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly

1. Test case – K L University

Run Any Single Command
Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly

Screen 1: Opening the KLUniversity Website

Screen 2: Testing the Authentication of E-Learning Screen

V. Implementation
The implementation of the system is described in the following steps
- Take a URL
- Record the Web Application
- Test the Navigations and hyperlinks
- Obtain the Results

VI. Test Cases
A good test case is one that has high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. Test Cases in Selenium are nothing but recording the Web Application and testing that again using the Selenium tool. The IDE allows many options for running your test case. You can run a test case all at once, stop and start it, run it one line at a time, run a single command you are currently developing, and you can do a batch run of an entire test suite. Execution of test case is very flexible in the IDE.

To Run a Test Case
- Click the Run button to run the currently displayed test case.
Screen 3: Testing the Functionality of attendance

VII. Conclusion:
In this paper, we used an integration testing tool called Selenium that tests the automated web applications. In our proposed framework, Selenium under integration testing tool tests several web applications and generated the code in different languages that includes JAVA, C# etc. Several test cases and requirements traceability logs were prepared that support our paper. We can even extend the work by adding some new properties to the testing tool and new languages that will be proposed later.

VIII. Bibliography

BOOKS:
2) Practical software testing: a process-oriented approach By Ilene Burnstein.

WEBSITES:
1) http://seleniumhq.org/about/platforms.htm
2) www.apttest.com/resources.html
3) http://kluproject16.blogspot.com
4) http://www.testinggeek.com/index.php/testing-tools
5) http://www.slideshare.net/metapix/selenium-ide-tutorial-presentation
6) http://www.openresourcetesting.org/functional.php
7) http://portal.acm.org/citation.cfm?id=1321211.1321239
8) http://swik.net/Eclipse/del.icio.us%2Ftag%2Feclipse/Integration+testing+framework
10) http://www.softwaretesting.net/

Prasanth Yalla received his B.Tech. Degree from Acharya Nagarjuna University, Guntur (Dist), India in 2001, the M.Tech. degree in Computer Science and Engineering from Acharya Nagarjuna University in 2004, and presently pursuing the Ph.D. in Web services and SOA from Acharya Nagarjuna University, Guntur (Dist), India. He was an associate professor, with Department of Information Science and Technology in KL University, from 2004 to 2010. Presently he is working as associate professor, with the department of Freshman Engineering from 2011 in KL University. His research interests include Testing tools, Software Engineering, Web services and SOA. He taught several subjects like Multimedia technologies, Distributed Systems, Advanced Software Engineering, Object Oriented Analysis and design, C programming, Operating Systems, Database management systems, UML etc. He is the Life member of CSI.

Dr L.S.S.Reddy received his Ph.D from BITS, PILANI and presently Working as DIRECTOR, Lakireddy Bali Reddy College of Engineering, Mylavarm Krishna district. He is having more than 30 years of experience in teaching. Previously he Worked in various prestigious institutions like CBIT, Birla Institute of Technology & Science(BITS), PILANI, MVSR(Matrusri) Engineering College, Hyderabad, V.R.Siddhartha Engineering College, Vijayawada, as a principal, K.L. College of Engineering. He got many AWARDS/ HONORS like Chairman Academic Council KLCE (Autonomous), Recognized guide for guiding scholars of Ph.D in Computer Science of Mysore University, Acharya Nagarjuna University &JNTU, Hyderabad, Chairman, board of studies Computer Science and Engineering, and Information Technology, Acharya Nagarjuna University, Chairman,CSI Chapter-Vijayawada. Senior Member, IEEE Computer Society. Member, Expert Panel of DOEACC Society and many more.

M.srinivas received his MCA from IGNOU ,New Delhi and M.Tech.degree in Computer Science and Engineering from Acharya Nagarjuna University and presently pursuing the Ph.D. in K L University, Guntur (Dist),AP, India. He is working presently as an Associate professor in the department of Freshman Engineering in KL University .He is having 16 years Experience in Teaching in reputed institutions. His research interests include Software Engineering, Datamining. He is life member of professional bodies like CSI, ISTE. He taught various subjects in computer science and engineering like software engineering, computer organization, Database Management systems, Cryptography and network security, Digital Logic design, UML ,OOPS etc.

T SubhaMastan Rao received his B-Tech degree in Information Science and Technology from Acharya Nagarjuna University, Guntur (Dist), India in 2001, the M.Tech. degree in Computer Science and Technology from Bharat University in 2008, and presently pursuing the Ph.D. degree in Web Usage mining from Acharya Nagarjuna University Guntur (Dist), India. He is working as assistant professor, with Department of Information Science and Technology. His research interests include Testing tools and web usage mining. At present, He is engaged in Data mining concepts and techniques.