A Comprehensive Study on Automation using Robot Framework

Neha S Batni[1], Jyoti Shetty[2]

[1]Student, RV College of Engineering, Bangalore, Karnataka, India

[2]Assistant Professor, RV College of Engineering, Bangalore, Karnataka, India

Abstract: *Recently, many organizations have shown improvement in the automation of applications when compared to manual testing as manual testing consumes a lot of human work, it is time consuming and may also affect the accuracy of testing. Automation has shown significant improvement in terms of availability, efficiency and quality of software products. Automation using Robot Framework has advantages such as it is plain, powerful and easy extended framework that uses the keyword driven development. It has table form of creating syntax which makes it easy and test cases are in a uniform way. The framework also has the powerto make re-use the keywords from existing keyword and makes sure it provides easy extensibility. It has simple library API for creating custom test libraries in Python or Java, it also has instruction interface and XML based output files that ease integration into existing build infrastructure, for example continuous integration systems. All these features make sure that Robot Framework can be used to quickly automate test cases. This paper is a study on Robot framework and its usage in automation testing.*

Keywords: software testing; integration testing; regression testing; test automation; robot framework

1. Introduction

Manual testing is the traditional method adopted by organizations.However, with the increasing number of web-based applications and quality tools within the market, this is slowly vanishing. In recent times, test automation has become a crucial part of the software development life cycle. The test automation process verifies the functionality, regression errors and executes all the features or use cases of the software product.It provides an efficient way to administer multiple features of the product simultaneously.

Test Complete automation tool was used by the organization to automate the product but the problem was, it did not support image recognition and it is a licensed tool. There are a number of tools for test automation available commercialized and open source, but few are suitable for black box testing (for a black-box testing, see [2]). Many of available tools are most suitable for the unit tests performed by the quality engineers or developers. The challenge is to choose the framework that is either paid or has the potential to work by reaching every UI part of the appliance.It should also provide, different libraries to communicate with the back end of the appliance that is under testing such as desktop applications or mobile applications.

2. Literature Survey

In software testing during an integration with software and hardware units to check for faults in the test, integration testing is carried out[3]. It is a continuous process and carried out throughout the software testing process. It is the best practice to carry out integration testing after the smoke testing for a short release.

The regression testing has a lot more of focus in the integration of functional tests [4]. This includes the testing of many protocols on the tool such as SOAP, CAN, Anybus, CODESYS. For every release that happens in the duration of three months, to run the test cases manually takes a lot of effort and time and may also be error prone. As the results of the tests are already known, automating of the test cases would save time and would make it more efficient.

Robot framework is used for automated regression testing [5] of the test cases that takes less time and gives a great success thus saving costs and enhances the quality of software. Robot framework supports all the platforms. It is built in Python. It is an open source framework and it is maintained regularly.

Here, it is used for VoIP communications [6]. As VoIP platform can be operated automatically it takes less time and it is less error prone. In [6], the researchers have made use of Robot Framework to extend the libraries to include some features to meet the necessary needs of server testing and to execute the automated installation of drives and firmware.

The researches[7] have used robot framework for the configuration of mobile networks since the framework is simple to use and has built-in keywords for automation. Network element configuration that is done by automation is more easier when compared to manual testing. There are many network elements and to configure so many network elements and parameters takes a lot of time with manual testing. So using robot framework is simple and it overcomes the difficulties.

A new tool called the Remote controller[8] has been designed and implemented for Robot framework. The tool solves the problem existed in robot framework such as when the report is generated in HTML format, it is hard to

manage all such logs and reports because the test development can take place in the same or different place at the same or different time.

The keyword driven testing feature of Robot framework[9] makes the framework more advantageous by making the automation faster. The Robot Framework IDE(RIDE) provides the keyword completion feature for the keywords that are already developed and for those keywords are that are to be developed newly. In some case, the keywords need to repeated for some of the test cases then those keywords are grouped into one sequence and then executed in RIDE.

To run the test multiple times in robot framework[10] from the command line execution would be an inefficient way of testing as selenium needs to open a new web page each time the application is launched.This can be well defined in robot framework if the task is to open different web pages using the same test variable or test case or the same keyword that is developed.

3. Why Robot Framework?

Robot framework [1] is an open source framework which understands text, HTML formatted test suite scripts, and features a keyword-driven methodology. It is a wrapper that's written over many frameworks with help of test libraries. These test libraries are selected consistent to the framework and to automate differing types of applications are elected. To drive the applications using an Appium framework, the robot framework's Appium library is preferred. Here most of the capabilities of Appium are framed in form of keywords, which are easily understood by a tester who would want to read the scripts and understand what is exactly being performed in the particular test. The figure 1 depicts the architecture of Robot framework [1].
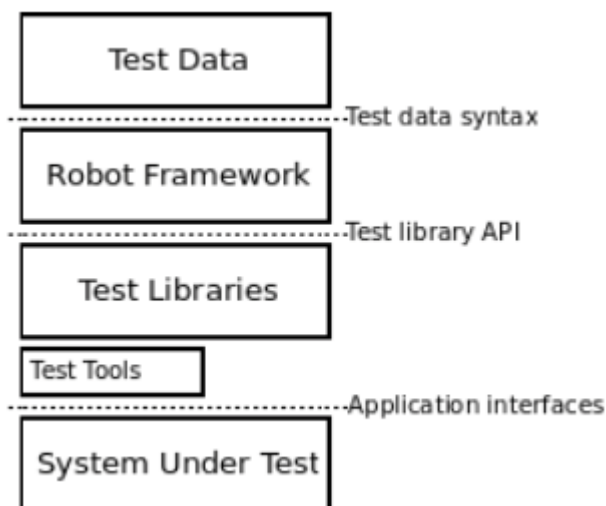
Figure 1: Architecture of Robot Framework

Test Data: Test data contains the test case data in TSV or HTML format, stored in .robot extension. Robot framework is responsible for handling the data, processing it, controlling the test execution and reporting of the status of testing. The test data cannot directly communicate the system under test so it uses robot framework that acts as a bridge between the input of the test data and test libraries.

Robot framework: Robot framework is a framework that is responsible to take the test data and process it to into a format that is appropriate for the attached libraries. To invoke the test libraries with suitable arguments about the test and test libraries and report it. It is implemented on Python or Java and has to run with it.

Test Libraries: Test libraries interact with the system under test and results in the formation of generating reports

Test tools: Optionally test libraries use some test tools, they interact with the system under test. This is not known to Robot framework and happens internally.

System under Test: An application under test must be used.

A. Robot Framework Components
1) **Test suite**: Launch_email_app.txt This is a test script file which has 3 test cases under "Test case" tag. Here the labels under this tag are considered as test methods by Robot framework. They are executed in sequential manner with top to bottom approach.
2) **Pybot** This is a robot framework module used to trigger the test scripts written in Robot framework format. Here the pybot reads the various framework files and executes the tests by performing different actions on the UI of apps. Robot Framework Code Base The folder structure which comprises of all scripts that use Appium libraries to speak and operate on the UI of the appliance. It comprises the different external libraries according to the need.
3) **Reports/Logs** The good part of using robot framework is its superior logging and reporting of features in the test case executions. The reports are generated in standard manner or they will be customized too. Test report is a report.html page which is automatically created after test runs. This page summarizes different test suites and their runs with pybot command that was triggered. These reports are interactive and may be drilled right down to have detailed analysis of how the steps in test scripts went.
4) **Test Logs** The following image shows the logs of the test case runs. Here is the detailed view and runtime values flowing in the variables. This approach saves time, avoiding the need to debug the code and then fix it.Test Logs image shows the logs of the test case runs. Here is the detailed view and run time values flowing in the variables. This approach saves time, avoiding the need to debug the code and then fix it.
5) **Jenkins**: Jenkins features a plug-in for robot framework, which shows interactive reports. We can drill down the reports to see details to analyze the health of the test cases running in consecutive job runs. The trend of each test suite in every job run also can be analyzed with this plug-in.

B. File Structure of Test Data

Robot Framework test data is in tabular form, supports HTML,TSV, text, or efficient Text format. Once start the Robot Framework, it selects a test data file translator according to the file extension name and executes the test file under test and the extension is not case-sensitive. generally HTML file documents are used. Test data file is collected of the following four types of form structure, as shown in Table 1.

Table 1: Different Test Data

Test Name	Used for
Settings	1) Importing test libraries, resource files and variable files 2) Defining meta data for test suites and test cases
Variables	Defining variable that can be used everywhere in the test data
Test cases	Generating test cases from available keywords
Keywords	Creating user keywords from existing lower-level keywords

C. Robot Framework features

There are many features the robot framework provides because of which it is possible to automate test cases quickly and efficiently. They are listed below:
- Simple Tabular Syntax
- High-Level Architecture
- Data-driven Test Cases
- Generic Test Libraries
- Remote test libraries and plugins for Jenkins/Hudson, Maven
- Separate Test Data Editor
- Clear Reports
- Detailed logs

D. Robot Framework IDE

Robot framework has a lot of standard test libraries and in addition it has many external test libraries available. They are used by the community and is maintained and served for different purposes. One advantage of all these libraries is that it can be developed together and used together. The keywords can be customized according to the requirement of the test case. It can be done combing the standard keywords with high-level keywords.

In addition to the libraries and functionality of the framework, robot framework provides graphical user interface called an integrated development environment(IDE) to maintain and develop the test cases. The figure 2 shows the Robot framework IDE with the libraries structure. PyCharm IDE is used for developing test cases to communicate with the products. PyCharm is a development environment when Python is used as a programming language.
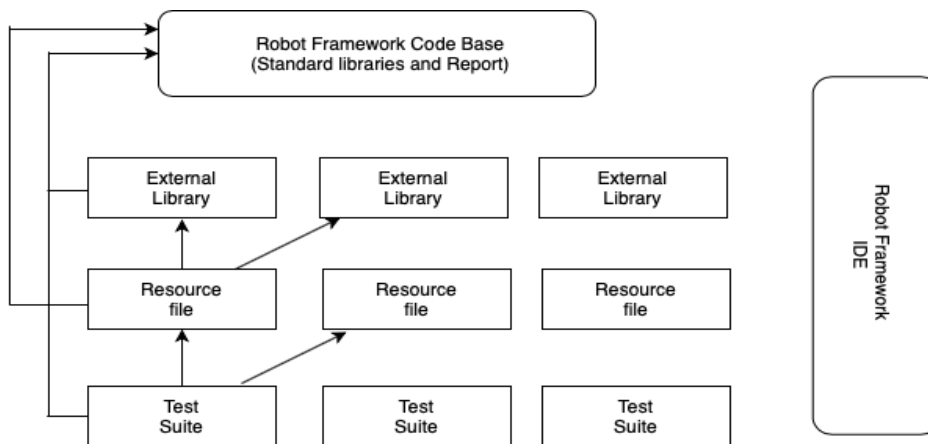
Figure 2: Robot framework IDE

Results Comparison

The comparison of manual testing with automation testing using robot framework is carried out. The advantages of using automation testing is also discussed.

Table 2: Comparison of manual and automated testing

Serial No.	Manual Testing	Automation Testing using RF
1	1 year- 3 releases 1 release- 24 hours Total time for testing- 72 hours	1 year- 3 release 1 release- 40 minutes 3 releases- 120 minutes
2	Document information- Release version, date, document number: Takes time	Built-in keyword is available in RF "Verify Text in File"
3	Manual tests can be error prone during the information verification	The screenshots are captured during the test execution and information is stored in the form of reports and logs in HTML document

The table 2 compares the manual testing with automation using RF.
1) The product release in any organizations may be 3 times a year or even more. The testing of the product before the release has to be carried out. In manual testing, during the release, testing may take upto 24 hours whereas in automation using RF it may take 40-45 minutes for smoke testing that is carried out during the

release. So, 3 releases during an entire year may take upto 72 hours in manual testing which is more when compared to automation and hence automated testing is chosen over manual.

2) The document verification of some files such as user manuals may contain information such as release version, date of the document and document number. There may be three documents or more than that, every time opening the document and checking for all these information and closing the document takes time. Instead, we can use "Verify Text in File" built-in keyword provided by RF.

3) The information verification can be error prone when done by a tester but with Robot framework the information can be stored in the form of screenshots or screen recording feature in the report and logs that the robot framework generates.

4) These are the advantages of choosing automation using Robot framework instead of manual testing.

4. Conclusion

Robot framework provides data driven, keyword driven and behavior driven style of writing the test cases. It is very easy to install and helps in creating and executing test cases. Robot framework provides natural flow of test cases with pre-conditions, actions, verification and also does clean up of the environment. Easy to build keywords with the lot of existing libraries and the developer can easily write his own test cases and keywords can be built with the help of existing keywords and also build high level keywords in Python or Java. Because of automation, manual tasks can be reduced and gives error free results. The main advantage of robot framework is that it also generates the reports that shows the status of the test case whether it has passed or failed and in the particular step with the information about the failure that has occurred. The cost of automation is reduced by preventing the use of manual testing and this makes robot framework a biggest strength.

References

[1] Nokia Solutions and Networks. Robot Framework User Guide. [online].available: http://robotframework.org/robotframework/latest/Robot FrameworkUser-Guide.html.

[2] R. Patton, Software Testing, Sams Publishing, 2005

[3] Castro L.M., Francisco M.A., Gulías V.M., A Practical Methodology for Integration Testing. In: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. (eds) Computer Aided Systems Theory - EUROCAST 2009. EUROCAST 2009, vol 5717. Springer, Berlin, Heidelberg

[4] Monica., Sanket N Shettar., Survey on Robot framework, International Journal of Engineering & Research & Technology, 2017

[5] Stanislav Stresnjak, Zeljko Hocenski, "Usage of Robot Framework in Automation of Functional Test Regression" in The Sixth International Conference on Software Engineering Advances, 2011

[6] Godasu Swetha, T. Ramaswamy, S.P.V. Subba Rao, "Automation of Telecom Networks using Robot Framework", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-5S3, February 2019

[7] Bhuvaneshwari, Hemanth Kumar A.R, "Automation for Configuration of Mbobile Networks using Robot framework", International Journal of Scientific Development and Research (IJSDR) ISSN: 2455-2631, Volume-2, Issue-6, June 2017

[8] Zhang Z., Guo H., Wang H. (2013) Remote Controller for Regression Test in the Robot Framework. In: Stephanidis C. (eds) HCI International 2013 - Posters' Extended Abstracts. HCI 2013. Communications in Computer and Information Science, vol 374. Springer, Berlin, Heidelberg

[9] Stanislav Stresnjak, Zeljko Hocenski, "Usage of Robot Framework in Automation of Functional Test Regression" in The Sixth International Conference on Software Engineering Advances(ICSEA), 2011

[10] Lisa Crispin 2010, "An Introduction to Test Automation Design", from The Testing Planet, No: 2 www.softwaretestingclub.com, July 2010