Survey on Robot Framework

Monica
PG Student, Department of Electronics and Communication Engineering
GSSS Institute of Engineering and Technology for Women, Mysuru

Sanket N Shettar
Assistant Professor, Department of Electronics and Communication Engineering
GSSS Institute of Engineering and Technology for Women, Mysuru

Abstract—**Recentlyin many applications of robot frame for IT automation have shown significant improvement in processes in terms of reducing tester participation, enhancing flexibility, efficiency and quality of the products. However, most applications are limited to point-to-point and noninteractive operations in which the availability of a highly-structured setup is a prerequisite. This prompts the vast emergency of researches on intelligent robotics that are aimed to improve the adaptability, flexibility and dexterity to enhance the intelligence of industrial robots.**

Robot Framework is a generic acceptance level test automation framework. Robot Framework is simple, powerful and easily extensible tool which utilizes the keyword driven testing approach. Easy to use tabular syntax enables creating test cases in a uniform way. Ability to create reusable high-level keywords from existing keyword ensures easy extensibility and reusability. Simple library API(Application Program Interface), for creating customized test libraries in Python or Java, is available, while command line interface and XML (Extensible Mark-up Language) based output files ease integration into existing build infrastructure, for example continuous integration systems. All these features ensure that Robot Framework can be used to automatetestcases.

Keywords—Shoulder surfing; Attack; Information security; graphical passwords;

I. INTRODUCTION

Robot Framework is a keyword-driven test automation which empowers testers to automate and manage complex workflow scripts efficiently. Before we discuss the architecture of Robot Framework in details we have to cover its core notion, keyword-driven or table-driven testing. Robotic assembly is regarded as a complicated task. Since the introduction of industrial automation, industrial manipulators have been widely applied to assembly operations. Most robotic operations involve pick-and-place operations that point-to-point motions are merely required. Sophisticated tasks such as peg-in-hole and debug operations are barriers to high-speed automation. At present, there is no industrial robot that can adapt rapidly to its changing environment and can manipulate objects without a precise definition effectively. Conventionally, control of robots is focused on model-based control such as position and force control. However, the major characteristic that differentiate human and machine is that human are capable of understanding, exploring, reasoning and learning under uncertain environment. In contrast, machine outperforms human under a pre-defined environment. Therefore, some researches have been focused on investigating systems that are capable of transferring and mapping human knowledge to robots [1].

Test cases are re-executed to check whether previous functionality of application is working fine and new changes have not introduced any new bugs. The method to verify is to perform this test on a new build when there is change in original functionalityor when there is a bug. Verifying that the bugs are fixed and the newly added features have not created in problem in previous working version of software.

Testers perform functional testing when new build is available for verification. The intend of this test is to verify the changes made in the existing functionality and newly added functionality. When this test is done, tester should verify if the existing functionality is working as expected and new changes have not introduced any defect in functionality that was working before this change. Regression test should be the part of release cycle and must be considered in test estimation. This test is very important. When there is continuous change/improvements added in the application. The new functionality should not negatively affect existing tested code [2][3].

This paper demonstrates general guidelines how to create good test cases using it. Both good practices and anti-patterns are presented. Most important goalsare Easy to understand, Easy to maintain, fast to execute for this frame work.Test suites often benefit from documentation explaining background etc. Well named tests created using well named keywords should not need extra documentation Reusable keywords must be documented – Good keyword and argument names help and are often adequate with higher-level keywords – Library keywords need detailed documentation

A. Test Automation in General

Test automation is carried out with a use of special software to control the execution of predefined test cases. The testing results (outcomes) are then compared to the expected outcomes. Automation helps in testing repetitive tasks, which would be difficult, or would require too much time and resources (manpower) to be run manually [2].

B. Benefits of Test Automation

Although implementing sufficient test automation is time consuming, it has a few benefits, when compared to human testers. Performing repetitive tests might be boring to a human tester, which might affect the accuracy of testing. Test automation is much faster than human testing. After test cases are implemented, the automation can accomplish hundreds or even thousands of tests in a matter of minutes. The same number of tests might take days from a human tester[6,7]

Obviously one of the main benefits of test automation is the amount of manual labor during the testing phase. The amount of manual work hours can be extremely small, compared to the time taken by manual testing.

C. Robot Framework

Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data syntax and it utilizes the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either with Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases. Robot Framework is operating system and application independent. The core framework is implemented using Python and runs also on Jython (JVM) and Iron Python (.NET). It has a highly modular architecture as illustrated in the figure3 [4].

D. Robot Framework Features

All the features, mentioned below ensure that Robot Framework can be used to automate test cases in a quick and proficient fashion.

- High-Level Architecture
- Simple Tabular Syntax
- Data-driven Test Cases
- Separate Test Data Editor
- Clear Reports
- Detailed logs
- Generic test libraries
- Webtesting, Swing, SWT, Windows GUIs, databases, SSH, Telnet
- Remote test libraries and other plugins for Jenkins/Hudson, Maven

E. Robot framework IDE

In addition to the Standard Test Libraries there are a lot of additional external Test Libraries available. Those are often contributed by the Robot Framework community and serve different purposes. The very good thing on all this libraries is that you can mix all the keywords from all the different libraries together in defining your own keywords or when writing a specific test case. Thus you can for example easily write tests for a web application using keywords from the Selenium Library to remote control the web frontend and use at the same time the Database Library to check results of certain operations from the database. In an ideal case all this can be done without a need to do any “real coding”, but by combining existing keywords from existing libraries to high-level keywords.

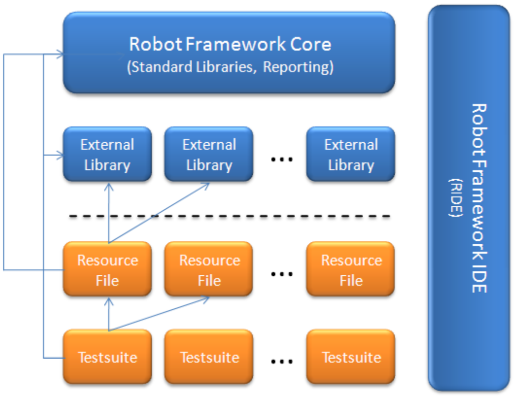

Figure 1. Robot framework IDE

In addition to the core functionality and the Test Libraries the Robot Framework provides a graphical user interface called RIDE (Robot Integrated Development Environment) that can help a lot in writing and managing test cases and keywords written in Resource Files. Please note that this has nothing to do with writing technical keywords in your own Test Library. This is entirely done in the development environment of your choice, e.g. Eclipse when using Java as a programming language

II. LITERATURE SURVEY

What We Do in Regression Test: Re-running the previously conducted tests and Comparing current results with previously executed test results. This is a continuous process performed at various stages throughout the software testing life cycle. A best practice is to conduct regression test after the sanity or smoke testing and at the end of functional testing for a short release [3].Automated Regression Testing is the testing area where we can automate most of the testing efforts. We run all the previously executed test cases on new build. This means we have test case set available and running these test cases manually is time consuming. We know the expected results so automating these test cases is time saving and efficient regression test method. Robot Framework is used for automation of existing regression test cases within short time and with great success and thus saving costs and enhancing the quality of the software project [4].Why Robot Framework: Robot Framework was found to satisfy all needed requirements. It is created in Python which can be implemented on all major platforms. Therefore, multiplatform requirement was completely fulfilled. Among other open source tools, Robot Framework seems to be one of the very few tools, which supports multi-platform environment and it is maintained regularly [5]. The tool is sponsored by Nokia Siemens Networks and released under Apache 2.0 license.

A. keyword-driven testing

Keyword-driven testing or some call it table-driven testing are the notions widely applied to an application-independent automation. The tester needs to develop data tables with keywords, independent of the test automation framework or any other tool used to run them. Then it is required to code the test script that will, in its turn “drive" the tested application and the data. Tables in a keyword-driven test will contain the information on the functionality of the tested application and step-by-step instructions for each test. Overall, we can speak of:

- Higher-level keywords: Those are testing a concrete aspect of the business logic of the system under test.
- Lower-level keywords: To keep the implementation of the higher-level keywords at a decent size one is often breaking down the required functionality to several lower-level keywords.
- Technical keywords: Those provide the technical implementation to access and thus test the system.

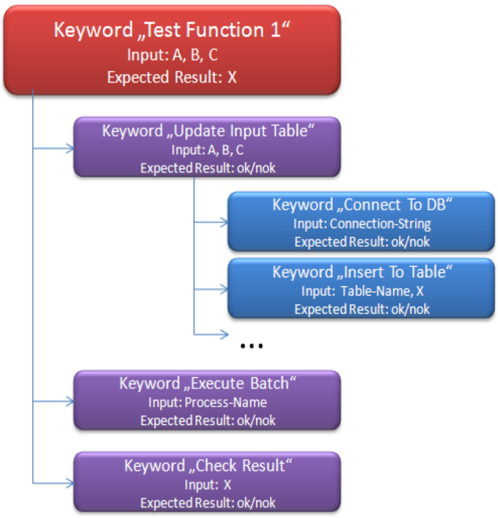

Figure 2. Nested keyword definatioin

III. WHY ROBOT FRAMEWORK

After careful analysis Robot Framework was found to satisfy all needed requirements. It is created in Python which can be implemented on all major platforms. Therefore, multiplatform requirement was completely fulfilled. Among other open source tools, Robot Framework seems to be one of the very few tools, which supports multi-platform environment and it is maintained regularly, as it is listed on [15]. The tool is sponsored by Nokia Siemens Networks and released under Apache 2.0 license, meaning it can allowed to be used for free (quite important topic, not only these days). Robot Framework is a generic, application and technology independent framework. It has a highly modular architecture illustrated in the Figure 1.

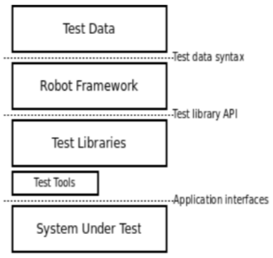

Figure 3 High Level Architecture

The test data is in simple, easy-to-edit tabular format. When Robot Framework is started, it processes the test data, executes test cases and generates logs and reports. The core framework does not know anything about the target under test, and the interaction with it is handled by test libraries. Libraries can either use application interfaces directly or use lower level test tools as drivers.

Test libraries provide the actual testing capabilities to Robot Framework by providing keywords. There are several standard libraries that are bundled in with the framework, and galore of separately developed external libraries that can be installed based on needs. Test libraries has standard libraries, external libraries and other libraries. For example, Swing Library is used from external libraries for testing Java applications with Swing GUI,Operating System library is used from standard libraries which enables various operating system related tasks to be performed in the system where Robot Framework is running.

Test Tools ease everything surrounding tests: editing, running, building and so on. Most of these tools are developed as separate projects, but some are built into the framework itself. There are tools like Built-in, Editors, Build and others. For example, Testdoc is a Built-in tool which generates high level HTML documentation based on Robot Framework test cases, Ride is an Editor tool for standalone Robot Framework test data editor [5].

IV. CONCLUSION

Benefit of working with Robot Framework is that writing test cases follows natural work flow with test case preconditions, action, and verification and finally clean-up. Real language is used for keyword description, so it’s easy to follow test case – even for non-technical person, which, together with its simple usage and easy library extension, make it great tool for test case automation. Everything is checked automatically and all reports are automatically generated and published on the web pages. This also saved lot of time when decision to introduce continuous integration was made. The cost of automating a test is best measured by the number of manual tests prevented from running and the bugs it will therefore cause to miss and this is probably the biggest strength of the Robot Framework.

REFERENCES

[1] Kiguchi, K. and Fukuda, T., ‘Intelligent positiodforce controller for industrial robot manipulators - applications of fuzzy neural networks’. IEEE Transactions on Industrial Electronics, Vol. 44 No. 6 (December, 1997), pp. 753 - 761.

[2] R. M. Sharma, “Quantitative Analysis of Automation and Manual Testing”, International Journal of Engineering and Innovative Technology (IJEIT), ISSN: 2277-3754, Volume 4, Issue 1, July 2014.

[2] Priyanka, Harish Kumar, Naresh Chauhan “A Novel Approach for Selecting an Effective Regression Testing Technique”, IEEE 2016.

[3] Jaspreet Singh Rajal, Shivani Sharma “A Review on Various Techniques for Regression Testing and Test Case Prioritization”, International Journal of Computer Applications, Volume 116 – No. 16, April 2015.

[4] Stanislav Stresnjak, Zeljko Hocensk-"usage of Robot Framework in Automation of functional test regression", ICSEA 2011: The Sixth International Conference on Software Engineering Advances.

[5] Overview on robot framework [online] available at http://robotframework.org/ [Accessed on 28-10-2016]

[6] Laukkanen P. Data-Driven and Keyword-Driven Test Automation Frameworks. Master's Thesis. Helsinki University of Technology - Aalto University; 2006.

[7] Introduction [online]. Robot Framework. URL: http://robotframework.org/#introduction. Accessed 5 February 2015.