ISSN: 2321-7782 (Online)

Volume 2, Issue 2, February 2014

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Paper / Case Study

Available online at: www.ijarcsms.com

Automation Tool for a Layman to Test Remedy Applications

Jayshree Ghorpade[1]
Computer Department
MIT College of Engineering
Pune – India

Ankita Bidave[2]
Computer Department
MIT College of Engineering
Pune – India

Chaitanya Malse[3]
Computer Department
MIT College of Engineering
Pune – India

Sandip Kolawale[4]
Computer Department
MIT College of Engineering
Pune – India

Rutuja Sonawane[5]
Computer Department
MIT College of Engineering
Pune – India

Abstract: ***Since last two decades, researchers and practitioners came up with a variety of software tools to automate the testing process. The present paper discusses the proposed work/technique, which can be used to develop a tool to automate test case generation process. It proposes to build an Automation tool that records and replays the use cases, thus providing a simple application for a layman (Non-Technical Person) who is an internal/external consumer. The protocol used for recording is HTTPS, so the encryption is performed while recording the calls. Server/Mid-tier calls will be recorded in .java file which is compiled and executed using command line interface and a Graphical User Interface (GUI). When there is repetition of same scenario, the tool prevents redundancy of use cases. If there is a change in format of input or response, Auto parameterization of input is performed.***

Keywords: Secure Socket Layer (SSL), Extensible Markup Language (XML), Action Request System (AR System), Hyper Text Transfer Protocol (HTTP), Hyper Text Transfer Protocol Secure (HTTPS).

I. INTRODUCTION

The Software Development Life Cycle of a project has various phases. Software testing is considered as one of the important phases of software development as it assures the quality of the software process and product. There are two types of software testing: Manual testing and Automation testing. Manual testing is a traditional method. It has many limitations, to name a few, limited resources, low efficiency and limited environment coverage. As human beings have the tendency of cutting corners, there is also a fear of incomplete test execution. To overcome these limitations nowadays most of the testing is done using automation tools.

The dominating commercial products such as WinRunner and Robot4 are designed without much consideration about compatibility and portability. These traditional testing products care little about the operations on cross-platform, distributed control, and distributed execution capabilities [2].The existing Automation tools in demand are Scapa, Silk Performer etc. The limitations of these tools are as follows:

- The use cases are stored in .java file, so it is impossible to replay the use cases without learning JAVA technology.
- Existing systems do not support replaying use cases on change in input or response parameters i.e. there is no auto-parameterization.
- A Tester doesn’t have control on starting and stopping the recording of a use case.

- When browser window is already open the recording of browsing traffic does not work.
- They are version dependent.
- The response and input from HTTP call is stored in unstructured format, so it is hard for updating scripts.
- Record functionality is limited to multiple user system only.

Hence, to overcome the shortcomings of Scapa and Silk Performer, a new Automation tool is proposed. The rest of this paper is organized as follows. Section II explains the necessity of the application, Section III explains the architecture and Section IV concludes the paper.

II. Necessity of the Application

The other tools such as Scapa, SilkPerformer etc. are dependent on technical skills of tester in Java, XML, HTTP calls. So, a person who does not have these skills, but is an intended user for the business applications cannot understand how to test it. There are many risks associated with handling a tool [3]. They are:

- Expectations are generally unclear from the tool.
- People generally ignore the time, cost and effort require for initial introduction of tool.
- People generally fail to understand the effort required for maintaining the tool.

Every automated tool is claimed to be ***"easy to use"***, but the scripts resulting from this method contain hard-coded parameter which must change if anything changes in the application [1]. In order to keep testing simple and easy to understand a tool that can provide interactive user interface for recording and testing scripts has to be developed. Also, if there is change in input/output format of mid-tier then already recorded scripts will not work. In such case, the tester has to manually edit and update each and every script. In an advanced application, the tester has to do nothing with input/output format. He is only concerned with replay of scripts and results.

III. System Architecture

A. System Requirements

The proposed Automation Tool uses the following technologies:

- IntelliJ Idea: It is a Java IDE developed by JetBrains, for developing enterprise application.
- Apache Http Client: It provides different packages and libraries for developing client-side application that use HTTP/HTTPS protocol [4].
- XML: Extensible Markup Language (XML) is used for encoding documents and it is widely used for representation of data structure in web services

B. AR System Architecture

The architecture consists of 4 layers:

1. Client Tier:

The tools in Client tier are used to run, manage and build applications.

2. Mid-Tier:

The Mid-Tier lets you access the AR System Server form Web Browsers and wireless clients.

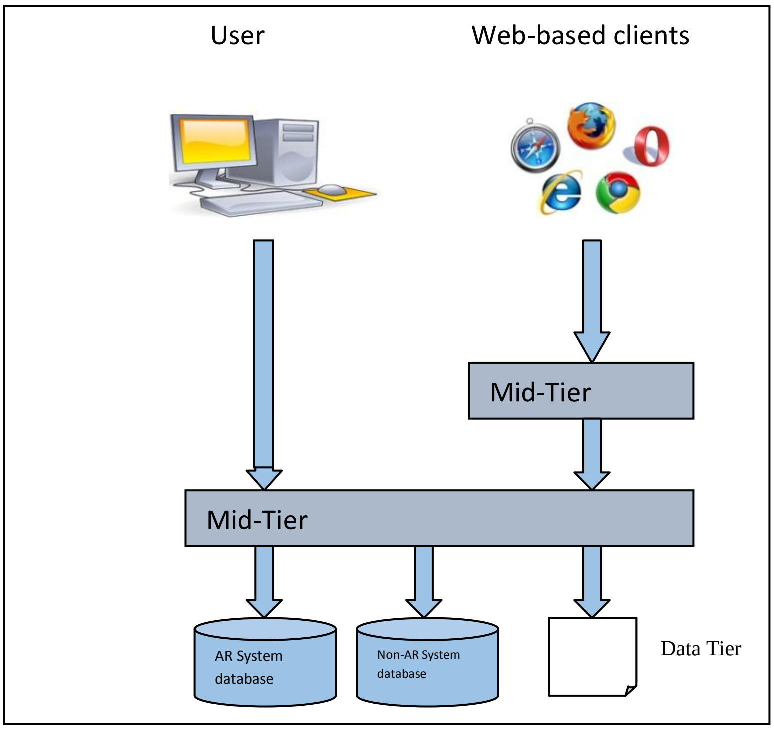

Fig1.AR System Architecture

3. Server Tier:

The Action Request System (ARS) performs various tasks. The heart of ARS is the AR Server and it performs the main task. In AR Server set of services like Email engine completes the server functionality of the system. AR System provides a Service Process Management platform which automates and manages Service Management business processes. The AR Server contains applications and software for creating new applications. The tasks performed by AR System are as follows:

- Replace outdated manual systems with process automation that speeds the handling of unique processes
- Integrate processes with systems across the enterprise
- Adapt and evolve your processes to continually align with the needs of the business
- Manage business process performance in real-time
- Capture and track critical business data

4. Data Tier:

The Database Server holds data that applications create and manipulate.

C. *Project Modules*

The proposed system has various project modules such as:

- Record Module:

In Record Module we will record the use cases and a script will be generated. The scripts will be saved in *.java* format.

The request and response parameters are saved in *XML* format.

- Parameterization Module:

A script recorded on a server can be replayed easily on same server. For replaying the Script on another server, this module is used to change parameters in accordance with the new server parameters.

- Playback Module:

While replaying, the configuration file of server is loaded. A scenario is created using workload file and configuration file. Then scripts are executed and the results are displayed to user.

- Validation Module:

When source code of the application is changed, this module identifies the changes and calls Script Update Module to modify the existing scripts related to the change.

- Script Update:

It modifies the Script according to the changes in application.

D. *Algorithms*

The following algorithms will be used by the Automation Tool:

- Pattern matching and Validation for input parameters passed and expected.
- Automation algorithm for replaying when there is change in input format.
- Management of login credentials that is required in further Http calls.
- Storing and modifying scripts automatically.

Fig.2 explains the testing process used by the tool. The figure describes the flow diagram of the proposed system.

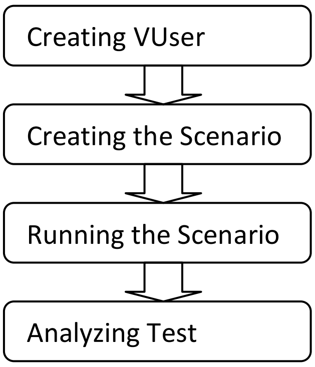

Fig2. Flow diagram of Proposed System

E. *Proposed Methodology*

The Automation tool divides the application into scenarios. A scenario runs a use case. In ideal condition, there are 100-200 use cases in a scenario. A scenario defines an event that occurs during each testing session. Thus, the scenario defines and controls the number of users to replicate the actions that they perform and the machines on which they run their replication. In the scenario, the tool replaces the human user with the virtual user (VUser). When we run a scenario, VUsers emulate the actions of human users submitting input to the server. Basically, a scenario can contain tens, hundreds, or thousands of VUsers. The actions that a VUser performs during a scenario are saved in a script. When we run a scenario each and every VUser executes the Script. While testing the application for the first time, a script is generated.

When there are any changes in the input parameters, during validation the change is detected and the scripts associated with the change are updated. Since numerous VUsers can run on a single computer, the tool helps us to reduce the hardware requirements. Thus, testing for every drop can be done in minimal time. The tool can record the scripts for multiple users simultaneously, thus, inducing parallelism and improving performance. The next major advantage of these tool is recording of calls is done using HTTPS. HTTPS stands for Hypertext Transfer Protocol Secure. It is a communication protocol used for secured communication over a network. The securities layer of SSL/TSL are added to standard HTTP communications, for reliable transfer of data. The use of SSL certificate provides security over an insecure network. As there are multiple clients working on the same AR Server the https prevent from eavesdropping.

IV. Conclusion and Future Enhancements

We have proposed an Automation tool for testing, which is one of the important phases of the SDLC of a project and can record and replay use cases. The tool is handy for layman and can be used without any technical knowledge. Also, the tool is version independent and supports auto-parameterization. Thus, the tool overcomes the problem of platform dependency. The tool uses HTTPs protocol for recording the calls so many organizations can use it to avoid corruption while recording calls.

Yet, there is scope for future enhancements. Some of them are: The testing can be done smartly by generating input data sets randomly; the application can be operated from cloud and Scripts can be scheduled and replayed.

Acknowledgement

We express our profound gratitude and deep regards to our guides Prof. Jayshree Ghorpade-Aher and Mr. Rahul Shah (BMC Software, Pune) for their exemplary guidance, constant supervision and encouragement throughout the course of this research work. Also, the work was supported by BMC Software.

References

1. Keith Zambelich, “Totally data-driven automated testing”, http://www.sqa-test.com/w-paper1.html, 2006
2. PEI Songwen, WU Baifeng, ZHU Kun, YU Qiang, “Novel Software Automated testing system based on J2EE” Tsinghua Science and Technology, July 2007, 12(S1): 51-56
3. “Why test Remedy Applications for performance”, http://www.vyomlabs.com/wp-content/uploads/2012/07/VyomLabs_Performance_Testing_WebinarV1.1.pdf
4. HTTP Client, http://hc.apache.org/httpclient-3.x/