Automation of NetAct Integration Process using Robot Framework

Poornachandra Tejasvi T M,
Department of Mechanical Engineering, The National Institute of Engineering
Mysuru, Karnataka, India

K R Prakash
Department of Mechanical Engineering, The National Institute of Engineering
Mysuru, Karnataka, India

***Abstract*— This Process automation refers to the use of digital technology to perform a process or processes in order to accomplish a workflow or function. Here the process of NetAct integration is being automated to reduce the human efforts. NetAct is an EMS (Element Management System) which is used to supervise Network Elements by checking Dynamic adaptation, alarms (Fault Management - FM), configuration (Configuration Management - CM) and the performance (Performance Management - PM) of a Bare-metal and VNF. The Process of NetAct integration consists of seven phases which need to be carried out one after the other successfully with validation.**

***Keywords*— NetAct, Robot Framework, HLR, TIAMS, IMSNFM, Process Automation.**

I. Introduction

The Process of NetAct integration consists of seven phases node creation, dynamic adaptation, alarm upload, cm upload, cm editor, fault management, performance management, where each phase has different type of interactions to performed like Web application, java application, Linux application, windows application. Node creation, Dynamic Adaptation are web application where data need to be entered precisely, some data are obtained with the help of port listening of Linux system and with automation it as reduced human error. Alarm Upload (monitor app), CM upload (CM operation manager), CM editor are java-based application, when alarms are triggered, DN & DU objects will be created. For handling the security warning which is a windows-based warning windows application automation will be used. Performance management and fault management are generated log files which need to be verified according to time and date stamp of its creation, here Linux based automation place a major role.

A. NetAct

Regular network growth, a aggregate of various mobile broadband technologies and the need for simplicity calls for community management to evolve. Nokia netact meets those challenges due to the fact it is effortlessly scalable, no longer simplest can it handle evolving technology, but additionally community increase and modifications in carrier supplying.

NetAct gives a single, consolidated view of any cell network, even the most complex multi-domain, multi-era gadget. Its miles virtualized for minimal downtime and resilience. NetAct archive cloud presents the agile, revolutionary and cloud-prepared returned-up and repair system for the telco cloud market. The machine provides non-stop, real-time tracking abilities, giving service vendors greater visibility into network situations at any factor in time and answers the most crucial demanding situations of the again-up marketplace in cloud, records increase, enterprise adjustments and data evaluation and transport.

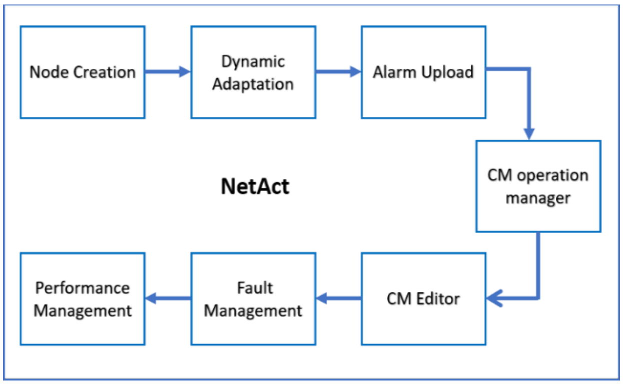


Fig. 1. NetAct integration Architecture

B. Robot Framework

Robot framework is a general automation framework for acceptance testing and reputation test-driven development (atdd). It has smooth-to-use tabular check information syntax and it utilizes the keyword-pushed testing technique. It's automating capabilities can be extended through libraries implemented both with python or java and users can create new higher-level keywords from existing ones using the same syntax that is used for creating check instances.

Robotic framework scripts are hosted on github where in documentation, source code, and problem tracker can be carried ahead. Downloads are hosted at pypi. The framework has a rich ecosystem, with numerous standards of libraries and tools that are evolved as separate initiatives. Robotic framework is operating device and alertness impartial.

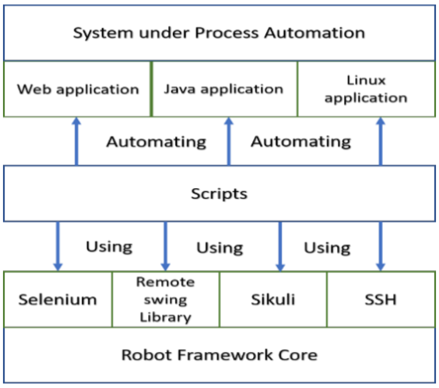


Fig. 2. Robot Framework

The center framework is implemented using python and runs additionally on jython (jvm) and ironpython (.Internet). Robot framework itself is open supply software program released under apache license, and most of the libraries and gear within the surroundings are also open source. The framework was initially developed at nokia networks and it is now a days backed by robotic framework foundation.

II. BACKGROUND

In this block we are going to see the prerequisites for the NetAct integration and provide an appropriate in-house solution.

A. Requirement 1

The home location registry in (hlr) is the primary database of permanent subscriber data for a cellular community. The hlr is combination of gsm tdma and cdma networks. Which is maintained by the subscriber's home carrier, the hlr contains pertinent personal information, consisting of user preference, account status, address and alternatives. The hlr interacts with the mobile switching center (msc), that is a switch used for control and processing of calls. The msc additionally serves as a access point to public switched telephone network (pstn - the constant network). The third indispensable detail is the visiting location registry (vlr), which keeps transient consumer statistics (which includes current area) to manage and control requests from subscribers who're out of the location covered by thier home system.

B. Requirement 2

Tiams (tsp installation, admin, and management server), a telecommunications service provider (tsp) is a sort of communications provider company that has traditionally offered telephone and similar support and services. This class includes incumbent local exchange carriers, competitive local exchange vendors or carriers, and mobile communication companies offering wireless services.

Even as some humans use the terms "telecom service provider" and "communications service provider company" interchangeably, the term tsp commonly excludes net service providers (isps), controlled service provider companies, satellite tv, and cable tv corporations. Tsps provide access to telephone and associated communications offerings. Within the beyond, maximum tsps were authorities owned and operated in maximum nations, due to the nature of capital expenditure involved in it. However nowadays there are numerous private company players in most areas of the world, even almost of the government owned businesses have been privatized.

III. OBJECTIVE

For a successful process automation of the NetAct integration at least the below objectives need to be satisfied.

A. Automation of all the seven steps process with validation.
B. No human intervention and errors done by human.
C. One click operation.
D. Automated Mail with Summary

IV. PROPOSED METHEDOLOGY

The NetAct integration of seven phases need to be automatically integrated irrespective to change in node name and its configuration, Even the respective seven phases consisting of different integration methodology and applications, process automation application or scripts should take care of all the dynamicity of the application. For the dynamicity of the application we use selenium, remote swing library, sikuli, ssh library and extended mail notifier plugins in robot framework to achieve the goal.

The seven phases consist of node creation, dynamic adaptation are of web application type, which can be only handled by the web drivers which selenium takes care of it.

A. Selenium

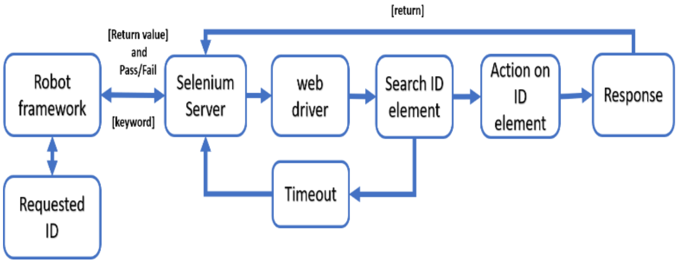


Fig. 3. Selenium Architecture

The Robot framework will first establish a connection with Selenium server. Selenium Server will do following
1. It will create a session for that particular request
2. It will launch the desired browser
3. Loads the Selenium cores JavaScript file into the specified browser (So as selenium server will have handle with the webpage for performing Selenium action) Now Robot framework will send the program that we have written in RIDE as selenium (by making conversion) and send it to Selenium server.

Selenium server is intelligent enough to understand the selenium command and triggers the corresponding JavaScript execution in the web browser. Here Selenium Server act as a “Proxy Server” between the AUP (Application under process). Being a proxy gives Selenium Server the capability of “lying” about the AUP’s real URL. Selenium server requests the actual webserver for the page open request and then it receives the page and sends it to the browser. Now any operation or request which the browser makes will eventually passes through Selenium RC server to actual webserver and vice versa.

The requested ID is searched in the webpage and if the ID is present in the entire page, corresponding return value will be returned to be robot framework from the selenium server. If in case, there is no element with that ID then there will be a time wait of 30 seconds to user defined time interval and a response with error code will be return by the selenium server to robot framework.

B. Remote Swing Library

In NetAct alarm upload, cm upload, cm editor are Java based application, which can't be handled by selenium, so we go with Remote Swing library which helps in automating java application.

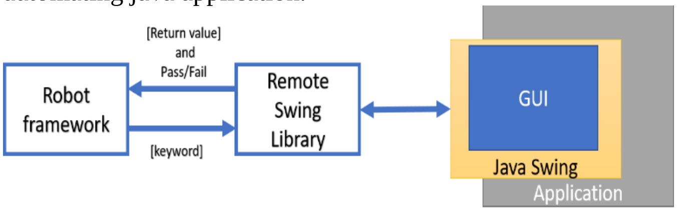


Fig. 4. Remote Swing Architecture

The library has keywords for application launching, keyboard events and handling various GUI components, like text fields, buttons and checkboxes. The keywords are documented in RIDE. Many of the keywords need an identifier parameter to specify the targeted GUI elements. A GUI element can be identified by name or by index.

Most of the keywords that operate on a visible component take an argument identifier of name type, which helps in locating the element. The primally matched element will be operated, depending on below conditions:

1. Zero-based index for the component type in the current context is used when identifier is a number type.

2. If the identifier matches to internal name of a component that component is chosen.

3. For components that have visible text, identifier is also matched against that.

Keyword List Components in Context lists all components and their names and indices in a given context. The robot framework just requested with keywords remote swing library will take care of handling of the java application by first searched outer frame of the java application and slowing access all the internal components of the application. The internals components traversal needs to be properly given by the user or else there will be no element visibility because of the security java application provides. If in case, there is no component found then there will be a time wait of 30 seconds to user defined time interval and a response with error code will be return by the remote swing library.

C. Sikuli Library

Before opening the java-based application for user security, the windows system prompts some windows asking trust on the application which is getting opened, as there are combination of windows based application and java based application, which only remote swing library can't handle them, as a supporting hand for remote swing library, sikuli library is used for the automation of windows application.

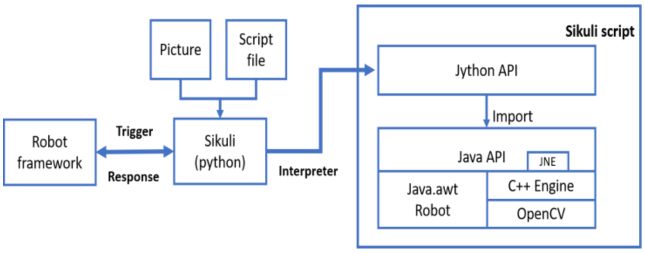


Fig. 5. Sikuli Architecture

Sikuli is a tool to automate graphical user interfaces (GUI) using Image Matching method. In Sikuli, all the web elements should be taken as an image and stored in the sikuli project. it will trigger GUI interactions based on the image matched. A Sikuli script (.sikuli) is a directory that consists of a Python file and the images in png format used by the script file. All images used in a Sikuli script are simply a path to the .png file in the sikuli bundle. When a script is passed to Sikuli IDE as an argument in command line, it will be recognized as its type by its filename extension. Sikuli can also be used in automation of web application which provides a supporting hand for selenium.

D. SSH Library

The last past of the NetAct integration is to monitor Fault and Performance of the system. To measure those parameter, linux server login need to be done. Which can't be handled by previously used libraries.

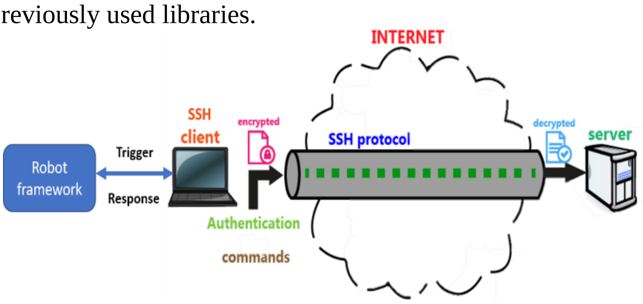


Fig. 6. Ssh Architecture

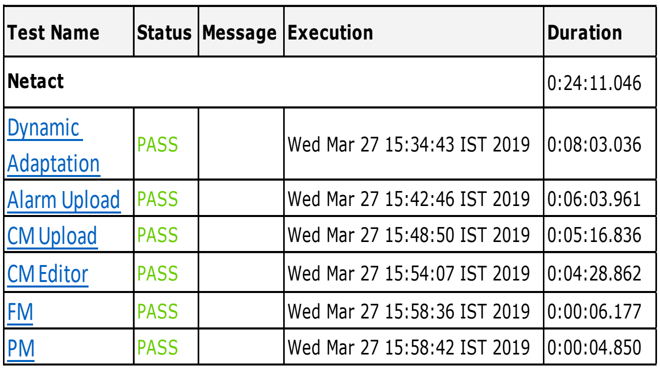
SSH, or Secure Shell, is a remote administration protocol that allows users to control and modify their remote servers over the Internet. Secure Shell (SSH) provides an open protocol for securing network communications that is less complex and expensive than hardware-based VPN solutions.

When user is connecting through SSH, user will be connected into a shell session, which will be of command-based text interface where user can interact with your server. In the duration of users SSH session, any commands that user types into the terminal are sent over an encrypted SSH tunnel and executed on linux server. The SSH connection is implemented using a client-server model. To establish a SSH connection, SSH daemon need to be running in the remote machine. Specific network port will be listened by SSH daemon software, which helps in authenticating connection requested by the user and appropriate environment will be spawns to the user on providing the proper credentials.

E. Extended Mail Notifier

After the process of all sever steps, easy analyser is needed for the better understanding of the process along with the time consumption of easy steps. Extended mail notifier provides a over view of the automated process and send them via mail.

TABLE. I. EMAIL NOTIFICATION



Test Name	Status	Message	Execution	Duration
Netact				0:24:11.046
Dynamic Adaptation	PASS		Wed Mar 27 15:34:43 IST 2019	0:08:03.036
Alarm Upload	PASS		Wed Mar 27 15:42:46 IST 2019	0:06:03.961
CM Upload	PASS		Wed Mar 27 15:48:50 IST 2019	0:05:16.836
CM Editor	PASS		Wed Mar 27 15:54:07 IST 2019	0:04:28.862
FM	PASS		Wed Mar 27 15:58:36 IST 2019	0:00:06.177
PM	PASS		Wed Mar 27 15:58:42 IST 2019	0:00:04.850

V. RESULT

TABLE. II. COMPARISON TABLE

cases	Time used (min)	
	Manual	Automated
Node Creation	5	4
Dynamic Adaptation	9	4
Alarm Upload	12	7
CM Upload	12	6
CM Editor	8	5
FM	4	0.5
PM	4	0.5
Total	54	27

The benefit of automating reduces the human intervention along with that the human created errors can be avoided easily, by the reference of table 2, time consumption is reduced to 50% and repeated runs can be achieved with out any issues.

Robot Framework also provide a feedback to the user where things went wrong which helps in diagnosing the issue caused while the automated process was in execution. With the help of email notification and automated triggering jobs the tests can we scheduled at any time and runner without the human attention and intervention. In industry time saving is given more prominence and automation helps in the conservation of time.

VI. CONCLUTION AND FUTURE WORKS

NetAct introduction and its benefits are exposed in this paper along with the integration process automation carried out with the help of Robot Framework and its plugins. This paper denotes the aspects of automation in industry with different types of application which can be automated without the limitation, here in this paper we were able to automate all the seven phases which are of different type of application and reduced the human intervention and human errors completely except once at the starting of the process. Which in turn helped in the reduction of time consumption for the process of NetAct integration by 50%, which helped in processing of other application and reduced human efforts. Automated mail summarizer helped in diagnoses of the issue encountered while execution of the automated process, which can be resolved by the department. Selenium, remote swing library, sikuli, ssh library utilization can be seen in maximum in this paper for the automation process.

The time consumption can be reduced by introducing parallel processing, which is again a tedious job, which can only be done once the decencies are removed from the NetAct system, which can help in achieving 20% more-time efficiency. Even by introducing Artificial intelligence, the errors occurred while execution of the automated process can be made to 0% by eliminating the human intervention at the starting of the trigger input process.

REFERENCES

[1] Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting, Vidroha Debroy, Lance Brimble, 0-7695-6388-0/18/ 2018 IEEE.

[2] Design and Application Of A Sikuli Based CaptureReplay Tool, Jin-lei Sun, Shi-wen Zhang, 978-1-5386-7839-8/18/ 2018 IEEE.

[3] Netact Based Test-Automation Framework Development For Ims Cmrepo, Jiya Johnson, Aarathi Elizabeth Jai, 978-1-5386-2745-7/17/ 2017 IEEE.

[4] Extension And Application Based on Robot Testing Framework, Qiu Na, Du Huaichang, 978-1-4799-8353-7/15 2015 IEEE.

[5] Nokia CUDO Documents.

[6] Repairing Selenium Test Cases: An Industrial Case Study about Web Page Element Localizatio, Maurizio Leotta, Diego Clerissi, Filippo Ricca, Cristiano Spadaro, 978-0-7695-4968-2/13 2013 IEEE.

[7] SURVEY ON SELENIUM TOOL IN SOFTWARE TESTING, Insha Altaf, 978-1-4673-7910-6/15 IEEE.

[8] Testing using Selenium Web Driver, Paruchuri Ramya, Vemuri Sindhura, 978-1-5090-3239-6/17 2017 IEEE.

[9] Security for the Digital Library - Protecting Documents Rather Than Channels, Ulrich Kohl, Jeffrey Lotspiech and Stefan Nusser.

[10] Abstracting Perception and Manipulation in End-User Robot Programming using Sikuli, Michael Kasper, 978-1-4799-4605-1/14/ 2014 IEEE.