Automation for Configuration of Mobile Networks using Robot Framework

Bhuvaneshwari[1], Hemanth Kumar A.R[2]

[1]**Student,** [2]**Professor**
Digital Electronics and Communication
Department of Electronics and Communication Engineering.
BIT, Bangalore

***Abstract*—Mobile networks are more complex and networks have become increasingly borderless. Configuration of the networks is becoming complex because the networks have been fortified with much functionality and should satisfy the several requirements for sophisticated multi-tenancy, high-level security and so on. The complexity of such a system dictates the use of automated network management tools. i.e Network Management System. Nokia NMS is a new generation network management system (NMS) for Multi-vendor and multi-technology networks. With NMS both the network and the services within the network are managed centrally so that the operator can view network element failures, service quality indicators, and traffic from one screen. Southbound interfaces are used for integrating the Network elements and lower level systems into NMS. NMS provides the management functions called FCAPS. In this paper we proposed automation for configuration of network Elements (NE's) i.e BSC of mobile network using Robot Framework. Robot Framework is a generic test automation framework based on keyword-driven and has been extensively used in the automated testing of NMS. Robot Framework is stretched by test libraries i.e Selenium2Library, Remote Swing Library and Screenshot etc.**

***Index terms*- Mobile Networks, NMS, Configuration of NE's, BSC and Robot Framework**

INTRODUCTION

Today's Mobile Networks are more complex than past, such as 2G , 3G and 4G technologies. If we consider 2G technology, there is a use of BTS, BSC and MSC Network elements for mobile communication. As 3G supports both voice as well as data (Internet) with the developed network elements such as SBTS, RNC and MGW etc. as the networking technology is growing, Network elements features, functionalities and number of NE's also increases. More complex networks supporting more applications and more users. As these networks grow in scale, two facts become painfully evident:

- Network and its related resources and applications become crucial to the organization.
- Many things can go wrong, restricting the network or a small portion of the network or corrupting performance to an unacceptable level

Network configuration is becoming complex because the networks have been fortified with many functionalities. and should satisfy the many requirements for sophisticated multi-tenancy, high-level security and so on.The complexity of such a system dictates the use of automated network management tools. i.e Network Management System(NMS).NMS serves as both NMS and Element Management System(EMS).NMS provides the following management functionalities:FaultManagement(FM), Configuration Management(CM),Performance Management(PM) and Security Management(SM).Configuration of Network Elements and managing the services of Network using the NMS satisfy the customer requirements.

There are two methods for configuring and managing the networks and network elements using NMS: 1. Manual and 2. Automation

In this paper we proposed method of Automation for configuring the networks and network elements using Robot Framework Robot Framework is the test automation framework based on keyword-driven and used in the automated testing of NMS. Robot Framework is stretched by test libraries i.e Selenium2Library, Remote Swing Library and Screenshot etc.

I. GENERALIZED NETWORK MANAGEMENT SYSTEM

Nokia NMS is a new generation of network management system for multi-vendor and multi-technology networks. it serves as both NMS as well as EMS.

A. Architecture of NMS

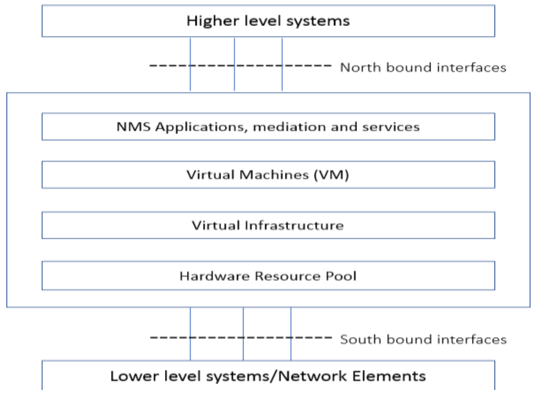


Figure 1: Architecture of NMS

Architecture of NMS is based on virtualized infrastructure (VI) technology that means physical servers are separated from software by a virtualization layer. Virtual machines host NMS applications and services. Services are distributed on multiple virtual machines. Each virtual machine provides the basic services such as a virtual operating system.
In NMS there is no dedicated hardware for any purpose, but the physical servers are combined into one hardware resource pool. Then, the hardware is dynamically allocated for different purposes because of the virtual infrastructure

B. Management Functionalities of NMS.

NMS provides the following functionalities:
1. Fault Management (FM):
Fault management is one of the main functions in network management. The FM enables to collect, process, store and display the alarms by analyzing the network topology. Network elements will rise alarms if there is any faults, depending on the received alarms NMS will correct the errors.

2. Configuration Management (CM):
The basic functionality of NMS Configuration is to define and manage parameter data in the network.
Configuration management (CM) applications in NMS can be divided into three main categories: Data editing, Data analysis and Data management. Data editing applications allow the planning center to display actual parameter values. In addition, data editing also enables the planner to create, modify and delete plans and then send these configuration plans for analysis.Data analysis is responsible for configurations plans inspection and error detection. Data analysis runs consistency checks on plans based on user-specific or NMS default consistency checks. Moreover, data analysis compares the created or modified plan with actual CM data. Data Management means implementing actions to configure a newly deployed network for the first time or changing the actual network configuration by the means of provisioning a configuration plan to the network. Data Management provides a convenient and fast way to execute commands to network elements. It also includes the Multipoint Configuration Assistant (MCA) application which allows efficient management of complex Multipoint configurations.

3. Performance Management (PM):
Performance management applications can be divided into two main categories: performance monitoring and performance reporting. Performance monitoring is online-oriented and provides near real-time information on critical network behavior. Performance reporting is offline-oriented and provides information on what happened in the network over a certain period of time. Performance reporting applications mainly rely on performance indicators and produce reports which can be used, for example, while troubleshooting, planning, or optimizing the network.

4. Security Management(SM):
The security solution in NMS consists of **user security, network security, data security and software security**. It is based on Nokia's security policy and guidelines that consist of customer requirements and international security standards.
User security: consists of authentication, authorization and user event logging. Authentication in NMS is managed by user management and authorization by permission management. **Network security**: includes traffic access control, encryption, integrity protection, separation and filtering. **Data security**: consists of file system and database access control, encryption and integrity protection, and secure key storage. **Software security:** includes verifying software upgrade and patching processes, file integrity checking and software signing.

Proposed methodology

Configuration of the networks and NE's using the NMS has two methods: Manual testing and Automation.
In this paper we proposed Automation method using Robot Framework as it has many advantages than manual testing.

C. Robot Framework

Robot Framework is a general test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data syntax and it uses the keyword-driven testing approach. Its testing skills can be stretched by test libraries implemented either with Python or Java, and users can also create new higher-level keywords from current ones using the same syntax that is used for creating test cases.
Robot Framework IDE(RIDE) is the editor used to write the test cases using python language. There are many libraries used to automate the Test cases. Few of them are:
Selenium2Library –Used for web testing applications
Remote swing Library-Used for java based applications
Screenshot-used to take screenshots

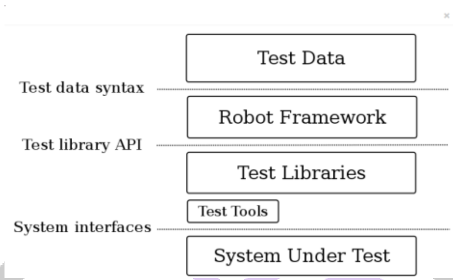


Figure2: Syntax of Robot Framework

Test Data: This is the data defined in files. Files has test cases and test suite is formed by test cases. It uses tabular editing format. Test data supports html and tsv format.
RIDE is the editor used to write the test case.

Robot Framework: It automates the test case by calling the services, test data and test libraries. Exchanges the data during test and generates output in HTML or XML format. It uses "pybot" command to start the framework and execute the test case .

Test Libraries: Robot Framework has in-Built libraries such as selenium library, remote swing library and screenshot etc. User can also create new libraries by using python script.

Test Tools: It is software/auxiliary equipment need to test the applications.

*System Under Test:*It is the system to be tested/ test product.

D. 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 I.

TABLE I. 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

II. CONFIGURATION OF NETWORK ELEMENTS

A. Integration of Network Elements to NMS.

To configure the Network elements, first we should integrate that NE into NMS using some interfaces. Such interfaces are called Southbound Interfaces.

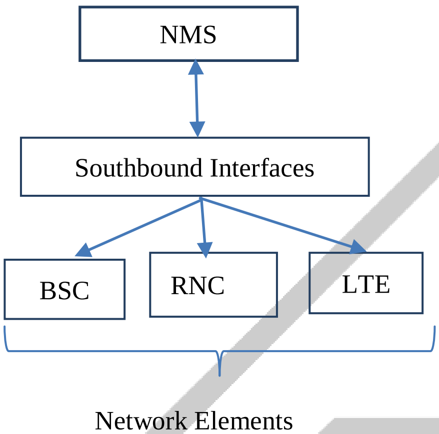


Figure 3: NE's interfaces

In this paper we used Base Controller Station (BSC) network element to integrate and configure the measurements plans to BSC.
Interfaces for BSC:FTP, MML, NE3SPROXY-1 AND OSI-1
FTP: This protocol used as interface between CM, PM of NMS and BSC. It transfers the data from BSC to NMS and vice versa with reliable and good efficient

NE3SPROXY: This stands for Nokia Enhanced SNMP Solution Suite PROXY. This interface behaves as a mediation between Fault management of NMS and BSC.

MML: This interface is used for Element management. BSC element details can be checked by using MML Interface.

OSI: This interface is the Q3 mediation between BSC and NMS. Because the data sends by BSC will not be understand by NMS. So OSI converts the BSC data into NMS understanding dataform.

Once the BSC is integrated in the Network Management System, the child objects such as BCF,BTS and TREand will automatically integrate to the NMS.

B. Administration of Measurements (AOM)

The Administration of Measurements (AOM) application is responsible for managing the collection of network elements(BSC)measurements. AOM configures the measurements to the BSC or Managed Objects(MO).This application is used to view, create, modify and delete measurement plans and templates.

Measurement plans can be used to activate or deactivate the measurements on the network elements. AOM also helps to upload measurement plans from the network elements to NMS

C. Managed object(MO)

Managed object presents the current status of the NE's. The reason for this is, customersrequire to observe the current status and configuration as well as all changes that made inside NE's, both hardware related and software related. All required parameters will be visible up to date.

All managed object is uniquely described by distinguished name that is called distname(DN). Dist name consists of MO class, instance identifier and MO distname of his parent.

Let's consider following MO distname(DN): MRC-MRC/PLMN-PLMN/BSC-375478

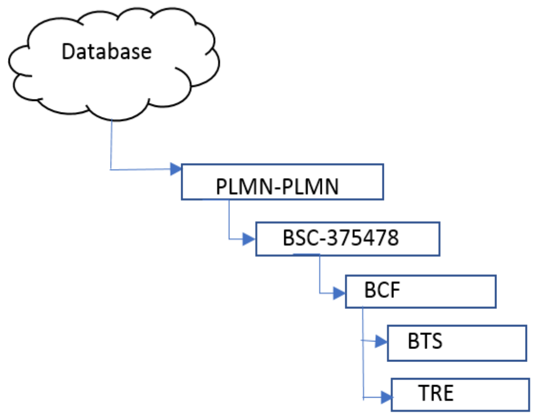


Figure 4:Managed object

III. DOCUMENTS AND EXPERIMENTAL RESULTS

The following four tables from Table II to Table V, are the structure of the Test Data file (test.html).This file has six Test cases for configuring measurements to the BSC NE. Robot framework is a automation tool using which we create measurement plans, enabling, disabling the measurements by activating and deactivating the plans respectively.

TABLE 2. SETTINGS

Setting	Value
Documents	Test suite performs AOM operations.
#Suite Setup	Test Suite setup
#Test Setup	Test setup Actions
#Test Teardown	Test Teardown Actions
#Suite Teardown	Webapp Teardown procedures
#Force Tags	GENERICAOM,BSC

TABLE 3. VARIABLES

Name	Value
${NE_Used}	BSC
${AdaptationName}	NOKBSC
${Test Plan Name}	Test plan name
${DN}	PLMN-PLMN/BSC-375478
${Measurements}	Traffic

TABLE 4. TEST CASE

Test case Name	Action
AOM_Plan_upload	This TC performs Upload Operation
AOM_plan_creat	Creates measurement plan to BSC
AOM_plan_Activation	Enables the measurements to BSC
AOM_plan_Deactivation	Disables the measurement plans to BSC
AOM_plan_Deletion	This Test case is used to delete the measurement plan from the BSC

TABLE 5. KEYWORDS

Keyword	Argument1	Argument2	Argument3
AOM upload	${Adaptation}	${NE_used}	
Create plan	${Test plan name}	${Adaaptation}	${Measurements}
Plan Activation	${Test plan name}		
Plan Deactivation	${Test plan name}		

When the test execution starts, robot framework parses the test data. Uses the keywords which are provided by the test libraries defined in test case and interacts with SUT. Directly libraries can interact with system or uses some test tools . test execution starts from "pybot" command. The generated output is in HTML or XML format.

Command:pybot-v/home/user/robot/sdv.pyv-/home/user/robot/elementdata.py---v—element file path: introduces the element file to set .

Test execution starts by the above command then robot framework executes the test cases. When the keyword AOM upload executed, the BSC sends its information to NMS.

After Create plan keyword execution, the measurement plans "Traffic" are configured into BSC. Similarly plan activation and deactivation test cases performed the operation of enabling and disabling the "Traffic" measurement on the BSC.

When the command line execution starts using the "pybot" command then test suite will start to run the test cases whisch is shown in the below figure 5.

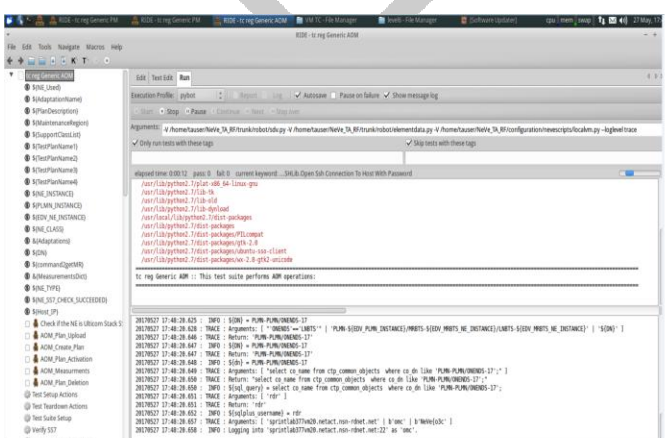
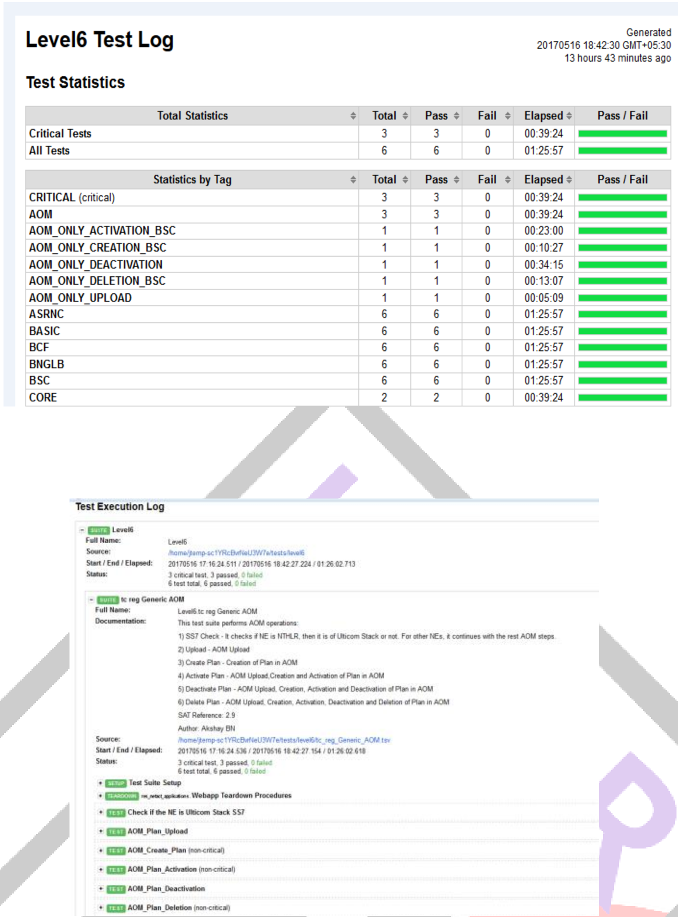


Figure5: Command line execution

The output file generated as follows:
Test execution is completed will generate three types of files, including log.html, report.html and output.xml.
When all the key tests are performed successfully (pass), the background colour will be green, otherwise it will be red. The two test results is as shown in Figure 6



Level6 Test Log

Generated
20170516 18:42:30 GMT+05:30
13 hours 43 minutes ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	3	3	0	00:39:24	
All Tests	6	6	0	01:25:57	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
CRITICAL (critical)	3	3	0	00:39:24	
AOM	3	3	0	00:39:24	
AOM_ONLY_ACTIVATION_BSC	1	1	0	00:23:00	
AOM_ONLY_CREATION_BSC	1	1	0	00:10:27	
AOM_ONLY_DEACTIVATION	1	1	0	00:34:15	
AOM_ONLY_DELETION_BSC	1	1	0	00:13:07	
AOM_ONLY_UPLOAD	1	1	0	00:05:09	
ASRNC	6	6	0	01:25:57	
BASIC	6	6	0	01:25:57	
BCF	6	6	0	01:25:57	
BNGLB	6	6	0	01:25:57	
BSC	6	6	0	01:25:57	
CORE	2	2	0	00:39:24	

Test Execution Log

SUITE Level6
Full Name: Level6
Source: /home/jtemp-sc1YRcBvfNeU3W7e/tests/level6
Start / End / Elapsed: 20170516 17:16:24.511 / 20170516 18:42:27.224 / 01:26:02.713
Status: 3 critical test, 3 passed, 0 failed
6 test total, 6 passed, 0 failed

SUITE tc reg Generic AOM
Full Name: Level6.tc reg Generic AOM
Documentation: This test suite performs AOM operations:
1) SS7 Check - It checks if NE is NTHLR, then it is of Ulticom Stack or not. For other NEs, it continues with the rest AOM steps.
2) Upload - AOM Upload
3) Create Plan - Creation of Plan in AOM
4) Activate Plan - AOM Upload,Creation and Activation of Plan in AOM
5) Deactivate Plan - AOM Upload, Creation, Activation and Deactivation of Plan in AOM
6) Delete Plan - AOM Upload, Creation, Activation, Deactivation and Deletion of Plan in AOM
SAT Reference: 2.9
Author: Akshay BN
Source: /home/jtemp-sc1YRcBvfNeU3W7e/tests/level6/tc_reg_Generic_AOM.tsv
Start / End / Elapsed: 20170516 17:16:24.536 / 20170516 18:42:27.154 / 01:26:02.618
Status: 3 critical test, 3 passed, 0 failed
6 test total, 6 passed, 0 failed

SETUP Test Suite Setup
TEARDOWN Webapp Teardown Procedures
TEST Check if the NE is Ulticom Stack SS7
TEST AOM_Plan_Upload
TEST AOM_Create_Plan (non-critical)
TEST AOM_Plan_Activation (non-critical)
TEST AOM_Plan_Deactivation
TEST AOM_Plan_Deletion (non-critical)

Figure 6: Test result logs

Once the measurements are configured to BSC we can check these measurements using element management function using the MML interfaces. This is shown in below figure 6.

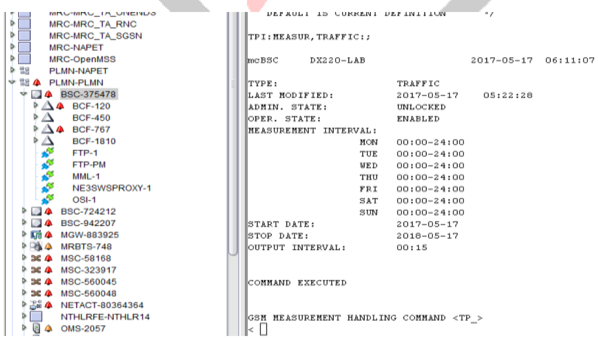


Figure7: BSC configurations

V. Conclusion

This paper analysis the configuration of mobile networks i.e configuration of measurement plans to BSC using automation based on robot framework. Robot framework utilizes the keywords to run the test cases which are written in python.
Network element configuration done by automation method that has many advantages than manual testing. There are many Network elements and configuring the network elements and its parameters is very difficult and takes lot of time with manual testing. So automation method using Robot framework overcomes all the difficulties and problem of manual testing. Robot Framework is very easy to learn and develop the test case. And it is very powerful.

REFERENCES

[1]J. Wu, C. K. Tse, F. C. M. Lau, and I. W. H. Ho, "Analysis of communication network performance from a complex network perspective," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 12, pp. 3303–3316, 2013.
[2] L. C. Freeman, "A set of measures of centrality based on betweenness," Sociometry, vol. 40, no. 1, pp. 35–41, 1977.
[3] J. Wu, C. K. Tse, and F. C. M. Lau, "Concept of node usage probability from complex networks and its applications to communication network design," IEEE Trans. Circuits Syst. I, Reg. Pap., vol. 62, no. 4, pp. 1195–1204, 2015.
[4]3GPP. (2012, June). Transmission and reception—Release 11 [Online].Available:http://www.3gpp.org/ftp/Specs/html info/36101.htm
[5]3GPP. (2012, June). E-UTRA and E-UTRAN overall description—Stage2Release11[Online].Available:http://www.3gpp.org/ftp/Specs/html-info/36300.htm
[6]J. Liu, R. Love, K. Stewart and M.E. Buckley, "Design and Analysis of LTE Physical Downlink Control Channel", IEEE VTC-Spring 2009, pp.1-5, Apr. 2009.
[7] 3GPP TR36.921, "Evolved Universal Terrestrial Radio Access (EUTRA);FDD Home eNode B (HeNB) Radio Frequency (RF) requirements analysis", V11.0.0.
[8] V. Garcia, Y. Zhou and J.L. Shi, "Coordinated Multipoint Transmission in Dense Cellular Networks with User-Centric Adaptive Clustering", accepted by IEEE Trans. Wireless Comm., Apr. 2014.
[9] 3GPP TR36.921, "Evolved Universal Terrestrial Radio Access (EUTRA);FDD Home eNode B (HeNB) Radio Frequency (RF) requirements analysis", V11.0.0.
[10]Ericsson. "5G radio access, " white paper, 2013.
[11] R. Kantola, J. Llorente Santos, and N. Beijar, Policy-based communications for 5G mobile with customer edge switching, Security andCommunication Networks, 2015.
[12] Wu JianJie, Chen ChuanBo, Xiao LaiYuan. Software testing technology base. Huazhong university of science and technology press. 2008. pp.10-15.
[13] Author: Louise Tamres, Translator: Bao XiaoLu, Wang XiaoJuan, Zhu GuoPing. Software Testing. Posts & Telecom press. 2004. pp.20- 26.
[14] Deng Bo, Huang LiJuan, Cao QingChun, etc. Software test automation. Mechanical industry press. 2003. pp.6-8.
[15] Cai JianPing. Software testing university tutorials. Tsinghua university press. 2009. pp.42-46
[16] Nokia Siemens Networks. Robot Framework User Guide. 2008-2009.pp.6-74.