

SIT Automation Tool: Failure Use Case Automation and Diagnosis

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Arian Shams Sanchez

In partial fulfillment

of the requirements for the degree in

Telecommunication systems ENGINEERING

Advisor: Jaume Comellas Colomé

Barcelona, October 2016

Abstract

The scope of this thesis is the SIT (System Integration Testing) process which is the testing procedure executed in customer test environment before the software goes on production environment. The main objective for this thesis is no other than improving the current process step by step taking into account the automation, efficiency, missing checks and much more.

This project is a kind of Industrial process to create a powerful testing tool which can allow the company to deliver quality adaptor products efficiently, do better in less time helping to reduce costs, as Adaptors are the most demanded product of MYCOM OSI portfolio. Take into account that business is not only generated when an Adaptor is delivered for first time but also when Vendors provide with new releases and new functionalities and operators needs to order an upgrade of the Adaptor to be able to monitor the new functionalities deployed on their network.

Resumen

El campo de aplicación en el que está centrado esta tesis es el SIT (System Integration Testing), proceso de testeo ejecutado en un servidor de testeo del cliente antes de desplegar el software en el medio de producción. El objetivo principal de esta tesis no es otro que mejorar el proceso actual paso a paso teniendo en cuenta la automatización, eficiencia, la falta de verificaciones, entre otros.

Este proyecto es una especie de proceso industrial para crear una aplicación potente de testeo que pueda permitir a la compañía entregar adaptadores de calidad con eficiencia, que hagan más en menos tiempo ayudando así a reducir costes. Los adaptadores son el producto más demandado del portfolio de MYCOM OSI. Hay que tener en cuenta que el negocio no se genera solamente cuando se entrega por primera vez el adaptador al cliente, sino que cuando los proveedores lanzan nuevas versiones con nuevas funcionalidades y los operadores necesitan encargar una mejora del adaptador para poder monitorizar las nuevas funcionalidades desplegadas en su red.

Resum

El camp d'aplicació en que es basa aquesta tesi és el SIT (System Integration Testing), procés de testeig executat en un servidor de testeig del client abans de desplegar el software al mitjà de producció. L'objectiu principal d'aquesta tesi no és un altre que millorar el procés actual pas a pas tenint en compte l'automatització, l'eficiència, la falta de verificacions, d'entre altres.

Aquest projecte és una mena de procés industrial per crear una aplicació potent de testeig que pugui permetre a la companyia lliurar adaptadors de qualitat amb eficiència, que facin més en menys temps ajudant així a reduir costos. Els adaptadors són el producte més demandat del porfolio de MYCOM OSI. Cal tenir en compte que el negoci no només es genera quan es lliura per primera vegada l'adaptador al client, sinó que quan els proveïdors llancen noves versions amb noves funcionalitats i els operadors necessiten encarregar una millora de l'adaptador per poder monitoritzar les noves funcionalitats desplegades a la seva xarxa.

Acknowledgements

After four months of hard work, my degree thesis is finally finished. With it, I conclude my Telecommunication Systems Engineering Degree and, therefore, a stage of my life. This time has been an amazing incredible journey of self-discovery. There were times were I faced several difficulties that completely overtook me but I always came up with a solution. I will always remember this sweet moment when your written code works as expected.

Doing this thesis in a Telecom business in a foreign country, far from family and friends it's being a tough challenge. Living in a city like London as stressful as it is and working in a small town like Slough (more than one hour of travel in every trip) it was also an uncomfortable situation to deal with.

But nevertheless, it's been a great opportunity and a constantly learning process which I would never forget.

I would like to thank my colleges from the ETSETB Alfons Soler Romeo, Joan Adrià Ruiz de Azua and Gerard Sánchez Gasulla for their help and interest. Of course, thanks to my tutor Jaume Comellas for his guidance and advices in every step of the thesis.

I would also like to thank all the MYCOM OSI UK team for all their support during my internship there, helping me every time when I could not find a way to continue. But in special to Patricia Delgado, the person in charge of me there, who never stop teaching and helping me every single day (even via Skype when she was not at the office).

Finally, I would like to thank all the amazing people I have met during this time at the university, that won't be only my old classmates but true friends.

Thank you all!

Revision history and approval record

Table 0.1 Revision history and approval record

Revision	Date	Purpose
0	27/06/2016	Document creation
1		Document revision
2		Final revision
3		Document approval

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Arian Shams Sanchez	arianss90@gmaill.com
Jaume Comellas Colomé	comellas@tsc.upc.edu

Written by:		Reviewed and approved by:	
Date	27/06/2016	Date	
Name	Arian Shams Sanchez	Name	Jaume Comellas Colome
Position	Project Author	Position	Project Supervisor

Table of contents

Abstract	2
Resumen.....	3
Resum.....	4
Acknowledgements.....	5
Revision history and approval record.....	6
Table of contents.....	7
List of Figures	8
List of Tables.....	8
Glossary.....	9
1. Introduction	10
1.1. Project requirements and specifications.....	11
1.2. Methods and procedures	11
1.3. Work Plan	11
2. State of the art of the technology used or applied in this thesis:	16
2.1. MYCOM OSI.....	16
2.2. PrOptima™	18
2.3. Adaptors.....	19
2.4. SIT.....	19
3. Project development.....	21
3.1. Current SIT process	21
3.2. SIT_AA_01 current process	23
4. Results	23
4.1. SIT Process solution proposal.....	23
4.2. SIT_AA_01 solution proposal	24
4.3. Next steps solution proposal.....	27
5. Budget	27
6. Conclusions	28

List of Figures

Figure 1: MYCOM OSI Software Distribution	16
Figure 2: Adaptor Information	19
Figure 3: Example of the content in a used file.....	25

List of Tables

Table 0.1 Revision history and approval record	6
Table 1.1 Milestones	13
Table 1.2 Gantt diagram.....	15
Table 3.1: Current Steps and description	22

Glossary

CAPEX: Capital Expenditures

CM: Configuration Management Data

CSP: Communication Service Provider

EMS: Element Management Systems

ETL: Extract, Transform and Load

FM: Fault Management

GIS: Geographical information System

GUI: Graphical user Interface

IK: Import Key

IoT: Internet of Things

ISP: Internet Service Provider

MIS: Mycom-osi Interface Specification

MOC: Managed Object Class (in PrOptima)

NE: Network Element

NOC/SOC: Network/Security Operational Center

OPEX: Operating Expense

PM: Performance Management Data

QoS: Quality of Service

ROI: Return on Investment

SI: System Integrator

SIT: System Integration Testing

TELNET: Application layer protocol used on networks to provide a virtual terminal connection

UAT: User Acceptance Testing

UFS: UNIX File System

UI: User Interfaces

VTI: Vendor Tech Interface (Name of the Adaptor)

VTP: Vendor Technology Plug-in

XML: Extensible Markup Language

1. Introduction

As networks become larger and virtualized, services become more complex to manage and customer experience expectations for service quality rise, Communications Service Providers (CSPs) require future-proof solutions to address the new challenges of the smart digital world. MYCOM OSI provides Performance management solutions for the digital era and the advent of hybrid – physical and virtualized – networks.

To do it MYCOM OSI has developed PrOptima™ a carrier grade multi-vendor and multi-technology performance management software solution for Communication Service Providers. It's a solution that manages the performance of both mobile and fixed networks and provides critical data for network and service operations centres, network engineering and optimization departments, as well as capacity planning and infrastructure rollout teams.

The software packages used on this mediation are called Adaptors. They parse Vendor specific input files into standard MYCOM OSI file format (MIS) so they can be manipulated by MYCOM OSI products. The parsing results are then loaded into the Data Warehouse. It is important to note that every adaptor is designed per vendor, technology and release to maximize the efficiency.

To check this parsing software, a test process called SIT (System Integration Testing) is performed on the customer test Server by MYCOM OSI engineers, which checks things like the extraction and loading of data, different releases behavior, updates of the software, modifications and much more.

The purpose of this project is to optimize the current SIT process, increasing the efficiency and the quality in order to achieve better results in less time. This should be done analysing the existing steps of the Testing process and identifying the gaps and unproductive commands, to develop an automation framework.

At the moment, the SIT process takes between 3 and 5 days to get passed but even then there is room for more improving by increasing efficiency and adding newest cases.

The project main goals are:

1. Improve my knowledge in Scripting with UNIX and the OS itself.
2. Analyse the whole testing process and be able to identify gaps and unproductive steps.
3. Develop a framework to optimize and automate the process with which the R&D department could do further improvement.
4. Improve the timing and the quality of Testing Reports.

1.1. Project requirements and specifications

Project requirements:

- ❖ Previous knowledge on UNIX scripting and UNIX OS in order to develop the new instances.
- ❖ Previous knowledge on C++ (or similar) to be able to learn quickly Perl coding language.
- ❖ Theoretical base of any mobile transmitting technologies (2G, GRPS, 3G, 4G...) and the Network Elements.
- ❖ A level of English that allows to write the required documentation.

Project specifications:

- ❖ Identify the gaps in the actual process.
- ❖ Improve the existing testing process to reduce the test time and give more accurate information.
- ❖ Develop an automation framework for one of the main steps

1.2. Methods and procedures

This project is based on a MYCOM OSI software application PrOptima™, tool to monitor any of their customer networks. Then, there are a couple of other complementary software like ICM (Import Control Manager) which is used to Import the data from the company servers (MIS data).

Putty, an SHH and telnet client to access the servers or Eclipse, an integrated development environment for computer programming are other software used during the project.

This project does not start from scratch since the current test tools are already developed by the company. But this is not a continuation of a previous project either since the pre-existing material has been developed by the R&D department and not by any other student.

1.3. Work Plan

Tasks

WP1- Introduction to the software: Get to know the product which the project is based.

-Internal task T1: Pass the main PrOptima trainings.

-Internal task T2: Practice all the theoretical knowledge achieved on the actual software.

WP2- Introduction to SIT Process: Get used to the testing process both in theory and practice.

-Internal task T3: 10h training with an expert engineer on the process.

-Internal task T4: Study of the SIT process.

-Internal task T5: Perform couple of complete SIT in the test server.

WP3- Business documentation: Compilation and writing of the company.

-Internal task T6: Find and write information about the company and the products involved on the project.

WP4- Perl: Introduction to the coding language.

-Internal task T7: Read and understand the documentation of the main structures and modules.

-Internal task T8: Pass an intermediate Perl course.

WP5- SIT process analysis and solution development: Analyze the process, design and develop a solution.

-Internal task T9: Recompilation of the Max Priority Problems during SIT process (12 last months).

-Internal task T10: Analysis and gap finding on the process.

-Internal task T11: Step analysis.

-Internal task T12: Solution design.

-Internal task T13: Solution development.

WP6- Tool testing and bug fixing: Test and fix any possible bug on the new tool.

-Internal task T14: Tool testing.

-Internal task T15: Bug fixing.

WP7- Final Project: Write and Review.

-Internal task T16: Write up the Thesis.

-Internal task T17: Thesis Review.

Milestones

Table 1.1 Milestones

WP#	Task#	Short title	Milestone/Deliverable	Date (week)
1	1	State of art	Do the main trainings of PrOptima	02/05
1	2	Tool practice	Familiarize with PrOptima	23/05
2	3	SIT Training	10h interactive training with an expert	30/05
2	4	SIT Process	Familiarize with SIT process	30/05
2	5	SIT Practice	Pass couple of SIT on test server	6/06
3	6	Company Documentation	Find and Write info from MYCOM OSI and the products	27/06
4	7	Perl Documentation	Learning Perl from Theperl.org	4/07
4	8	Perl Course	Pass an intermediate online course	11/07
5	9	PR1 Recompilation	Recompilation of the Max Priority Problems during SIT process (12 last months)	01/08
5	10	SIT Analysis	Analyze the process and Identify the gaps	08/08
5	11	Step analysis	Find a solution focusing on this specific step	15/08
5	12	Solution design	Design of the final solution	22/08
5	13	Solution development	Development of the code	29/08
6	14	Tool testing and bug finding	Test the new tool	19/09
6	15	Bug fixing	Fix the new errors	19/09

7	17	Final Report	Write the final report	26/09
7	18	Review	Review the final report	03/10

Deviations and incidences:

The original plan was to analyze the whole testing process and build up a solution to improve and automate every step on the SIT.

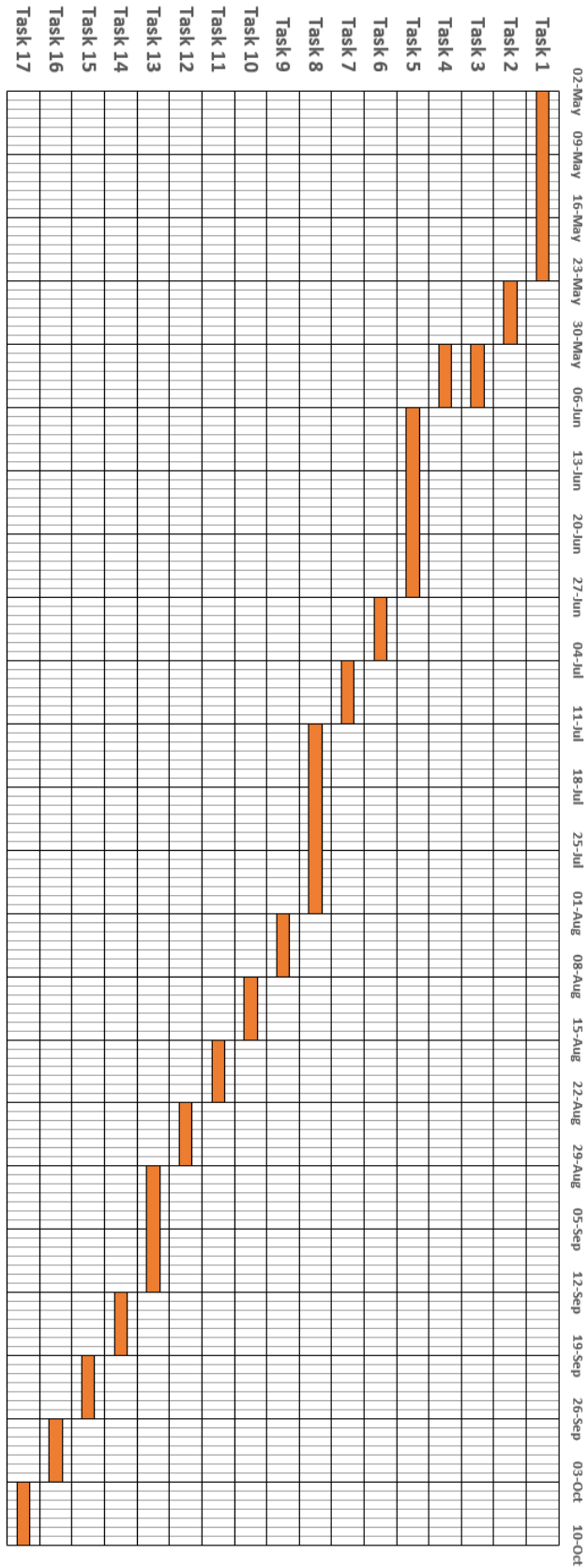
The first incidence came when I had to decide the coding language to be used. The original plan was to use C++ to develop any of the needed scripts, but after talking with the R&D department, they urged me to use Perl. The main reason they gave me is that, as an agreement, everybody has to develop the scripts in the same language so the whole team can work on the same project in case the solution it's not completely finished or something is need to be changed or added. So as I never used that coding language before I had to take an intermediate level course changing the original timing.

As a consequence of this first issue and after analyzing the current process, I've decided to focus on the first step of the test due to the limitation of time. This step checks that the data fetched from the Remote Server of the customer is properly stored on MYCOM OSI Servers. The files have to be the same in both, and also they have to accomplish some predefined parameters like the file name format, the time stamp or extension.

The main changes on the work plan are the 4 weeks Perl course and changing the step by step solution design and development for only one of the steps. Even then, the complete analysis of the current process and the recompilation of the most important problems occurred on the last 12 months is done and delivered to the R&D department for further improvements of the SIT.

Time Plan

Table 1.2 Gantt diagram

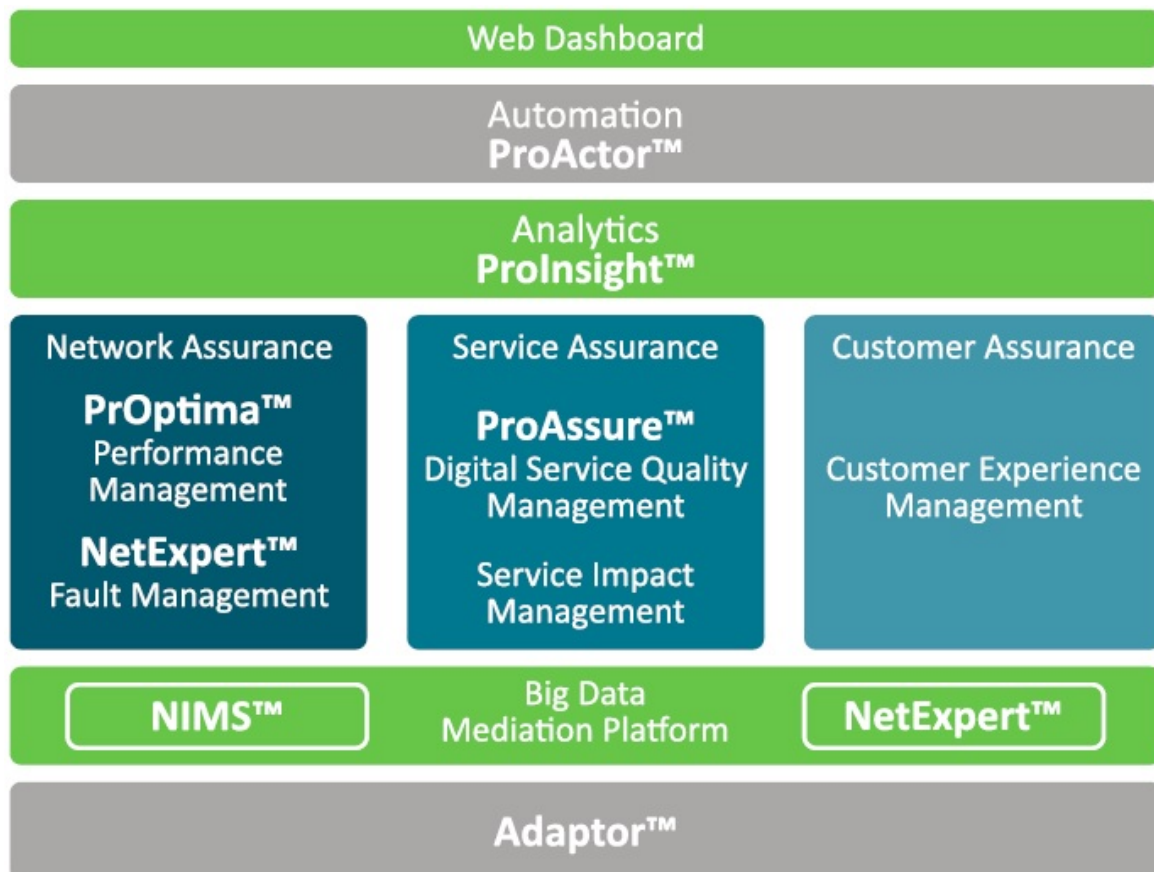


2. State of the art of the technology used or applied in this thesis:

2.1. MYCOM OSI

MYCOM OSI is an independent provider of Assurance, Automation and Analytics solutions to worldwide CSPs including AT&T, Deutsche Telekom, Maxis, Sprint, STC, Telefonica, Telenor, T-Mobile, Verizon and Vodafone.

Figure 1: MYCOM OSI Software Distribution



Experience Assurance and Analytics

MYCOM OSI’s Experience Assurance and Analytics (EAA) addresses three initiatives of CSPs: managing the Customer Experience, evolving to Network Virtualization (NFV) and exploiting the Digital/IoT environment.

By integrating quality products and solutions for Performance Management, Fault Management, Digital Service Quality Management, Customer Experience, Automation and Network Analytics into a big data-based platform that is tightly integrated by common resource, mediation and visualization layers, it enables CSPs to use consolidated information to address any of these aspects.

Experience Assurance and Analytics Benefits

- ❖ Capex efficiency through pre-integrated network, service, customer assurance and big data analytics
- ❖ Multi-team efficiency from collaborative platform for technical, marketing and care teams
- ❖ Customer insights and network monetization with predictive analytics
- ❖ Increased employee productivity by closed-loop automations
- ❖ Cloud-based, scalable, secure, network-independent platform

MYCOM OSI Benefits for CSPs

- ❖ Use of an end-to-end independent view of network, service and customer experience, to facilitate collaboration across teams
- ❖ Proactively detect network and service issues, to improve end-user experience
- ❖ Simplify operations via pre-integration and automation, to address complexity and to increase operational efficiency
- ❖ Make decisions based on network and service/device/customer related metrics, to optimize infrastructure investments and to maximize ROI

Network Analytics

The company provides Analytics solutions to enable the optimal use of network and service/device/customer data from different sources (including big data stores) at the same time and as part of familiar workflows/processes.

MYCOM OSI helps CSPs make intelligent use of the wealth of data at their disposal to reveal network utilization and subscriber behaviour insights, enabling new digital services and increase the return on network infrastructure investments.

Solutions

MYCOM OSI helps CSPs deploy a Next Generation NOC/SOC by building on the core platforms of fault management, performance management and service management, and adding use case-based solutions. These enable advanced operations in NOC/SOC that:

- ❖ Deliver customer centricity
- ❖ Enable proactive and preventative operations
- ❖ Increase productivity with automation
- ❖ Solve complex incidents with guided diagnostics

2.2. PrOptima™

PrOptima™ is a fixed/mobile network performance solution that scales to meet the needs of CSPs with support for multiple technologies, domains and equipment vendors across access, backhaul, core and service networks. It processes very large volumes of performance management data in near real-time, has advanced correlation, analysis, reporting and visualization modules, and is highly flexible via automation, workflow and configuration capabilities.

The software consumes highly valuable resource and service performance management data containing very detailed and statistically significant control and user plane events, element configuration and network element resources data. This may be complemented with other data, such as fault management, trouble ticketing and other data sources.

The data store can then be managed with automated and manual methods to provide 24x7 issue detection for resolution departments, network engineering planners, optimizers and management.

The basic modules available on the application:

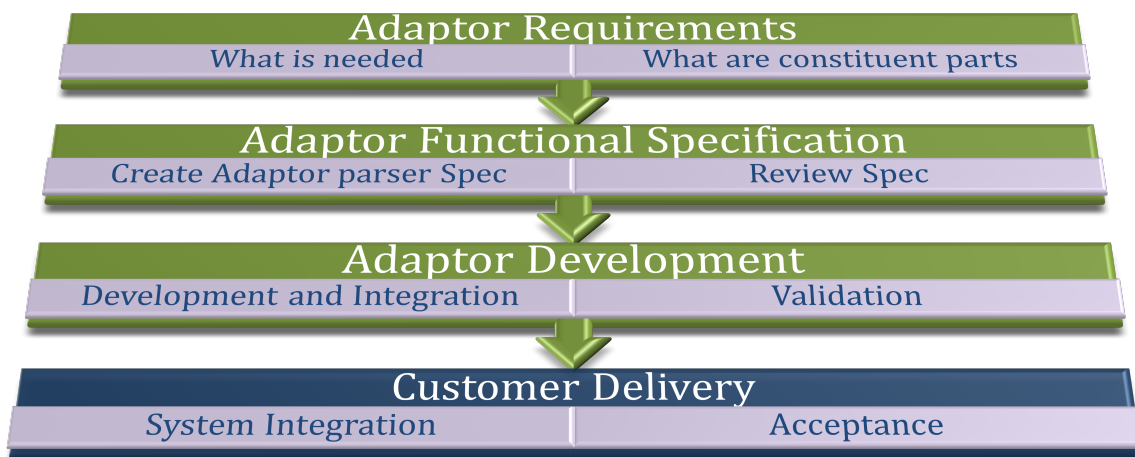
- ❖ **Workspace:** Allowing to build professional reports and dashboards combining tabular and graph views
- ❖ **Alarm Module:** Providing multiple deviation algorithms to generate performance alarms in various formats for consumption within or outside PrOptima™
- ❖ **Profiling Module:** Providing trending and envelop analysis and possibly feeding the alarm module for deviation identification
- ❖ **Geographical Information System (GIS) module:** Representing statistical information on geographical and vector maps provisioned locally or through Google Map integration
- ❖ **Decision Support System:** A workflow analysis module allowing to capture your operational and engineering business processes
- ❖ **Network Planning:** A forecasting module providing various extrapolations and projection mechanisms to assess the potential resource status in the future

2.3. Adaptors

Adaptors or Vendor Technology Interfaces are software in charge of parsing the information. They parse Vendor specific input files into standard MYCOM OSI file format (MIS) so they can be manipulated by MYCOM OSI products, such as ProOptima™ in this case. The parsing results (MIS files) are then loaded into the Data Warehouse with different lifetime depending on the granularity and the information type.

Adaptors Functional & Technical Specifications Documents define the required functionality of the parser and contains all the needed information to install and deploy the adaptor in each network, making them personalized for each vendor and technology.

Figure 2: Adaptor Information



2.4. SIT

Every Adaptor goes through a detailed testing process before is completely deployed.

First of all, the R&D department, which is in charge of the implementation of each adaptor, performs a QA (Quality Assurance) test to ensure that everything is working in in a controlled environment.

Then the software is delivered to one of the qualified engineers in the company to pass the SIT (System Integration Test) in a test server, as the QA it's performed in a lab environment with no real time data.

After that, the adaptor is delivered to the customer to pass the UAT (User Acceptance Test), a very similar test to the SIT but performed by the customer engineering team in their own environment (but still on the test Server).

Finally, the live installation is done and the data starts to be parsed.

Most of the issues related to the adaptor are found on the SIT, as this test is the most exhaustive one and that's why this thesis is based on it.

The current process has 20 steps and checks:

- ❖ Extracting process
- ❖ Multi-Release issues (every time an adaptor is updated a new SIT is performed)
- ❖ Network entities, counters and attributes properly parsed and stored
- ❖ Duplication of data
- ❖ Preview of data
- ❖ New data or structures (any modifications have to be checked)

3. Project development

3.1. Current SIT process

As said in the previews sections, the aim of the project is to develop a framework to improve and automate the current testing process of the parsing software.

An exhaustive study it's performed on every step to determinate the different changes to propose in each case.

Every case is named with the same structure, SIT_AA_XX, where XX is the number of step. Then, each of them is organized by areas and just as the name of the step the structure starts with SIT_ and added to that some specific information to sort them all.

To understand the testing process is important to understand first that there are 2 main platforms involved. In one hand, we have the Databases from both customer and MYCOM OSI, Servers from where the network data is fetched and stored (from the customer to the company database). And in the other hand, the Performance Manager tool, PrOptima™ which is in charge of manipulating this information and present it in many different ways accomplishing the user needs. Data is organized and shown into palettes from where the user can choose the different options, such as NE (Network Elements), AE (Attributes Elements) and CE (Counters Elements).

The process is shown step by step with a brief description of each one in the next table:

Table 3.1 Current Steps and description

Step name	Step Area	Description
SIT_AA_01	SIT_ADMIN_TEST	CM/PM file name check: The purpose of this check is to ensure that every single file is properly parsed and the filename structure of the files presented matches the file naming convention expected by the Adaptor, the time stamp (sending and arrival times) and extension.
SIT_AA_02	SIT_MULTI_RELEASE	Release Identification: - Identify the release tags available in data - Identify the ACCEPTED Releases configured. Ensure that only the PM/CM data for the applicable release is processed.
SIT_AA_03	SIT_MULTI_RELEASE	Check the ETL processing of Managed Object Instances and correct discrimination of releases
SIT_AA_04	SIT_MULTI_RELEASE	Check the ETL is correctly discriminating performance files by release.
SIT_AA_05	SIT_MULTI_RELEASE	Check that Import Keys have NOT changed between releases BEFORE enabling the Adaptor. This is a priority test case. It is essential that unplanned IMPORT KEY changes are NOT applied to Test or Production environments to protect against NE duplication and the expensive operation to revert the situation.
SIT_AA_06	SIT_DUPLICATES_CHECK	Check if duplicates have appeared in the system after the multi-release vti has been installed. This is a priority test case. It is essential that unplanned IMPORT KEY changes applied to Test or Production environments are detected.
SIT_AA_07	SIT_DUPLICATES_CHECK	Check whether the Client is showing duplicated NE NAMES normally as a result of IK Change.
SIT_AA_08	SIT_CALC_PALETTE_CHECK	Check to ensure that old and new palettes are merged.
SIT_AA_09	SIT_PREVIEW_NE	Check to ensure NE previews are correct for the interface.
SIT_AA_10	SIT_PREVIEW_NE	Check to ensure NE previews are correct for the interface (Negative). Historically, it has been observed that incorrectly defined DATETIME patterns return data by example specifying the DATE but not the time component.
SIT_AA_11	SIT_PREVIEW_CONF	Check to ensure AE previews are correct for the interface.
SIT_AA_12	SIT_PREVIEW_CONF	Check to ensure NE previews are correct for the interface (Negative). Historically, it has been observed that incorrectly defined DATETIME patterns return data by example specifying the DATE but not the time component.
SIT_AA_13	SIT_PREVIEW_COUNTER	Check to ensure CE previews are correct for the interface.
SIT_AA_14	SIT_PREVIEW_COUNTER	Check to ensure CE previews are correct for the interface (Negative). Historically, it has been observed that incorrectly defined DATETIME patterns return data by example specifying the DATE but not the time component.
SIT_AA_15	SIT_NE_PALETTE	Check to ensure the NE Palette Hierarchy matches that defined in the specification.
SIT_AA_16	SIT_CALC_PALETTE	Check to ensure the CLIENT palette presentation is correct.
SIT_AA_17	SIT_COUNTER	Check to ensure the raw COUNTER data is being parsed correctly.
SIT_AA_18	SIT_CALC_PALETTE	Check to ensure the raw ATTRIBUTE data is being parsed correctly.
SIT_AA_19	SIT_NIMS_NOM	Check to ensure the Parent-Child Relationships are in place.
SIT_AA_20	SIT_FLATFILE_SIZE	Check flat file size post MR upgrade is reasonable and that there is sufficient headroom on the server (i.e. Flat files should take no more than 75% of disk space).

3.2. SIT_AA_01 current process

After analyzing the whole process and due to the limitation of time, the scope of the project changes and the solution it's focused on the first step of the test solution.

As seen in Table 3.1, this first step is in charge of checking the ETL process from the customer Remote Server (where all the raw data is stored) to the MYCOM OSI Databases.

This case checks that the Extract Process is extracting raw file from the remote EMS correctly, and the data coming into PrOptima™ data base is consistent. It's also in charge of checking the file name format, the time stamp (sending and arrival time) and the extension of the files match the structure set on the Technical Specification document.

All this procedure is done manually, so the cost of time for the person in charge of the testing is really high, and also introduces the possible human errors.

4. Results

4.1. SIT Process solution proposal

As a result of the analytic study some notes of every step are being delivered to the R&D department. There is no specific solution for any of them (less for the first one, AA_01) but at least there is the way to follow.

There are two different types off checking during the SIT, the ones which concern to the files stored on the databases and the ETL process and others that test that once the data is parsed, PrOptima is correctly updated and this information or options are available for the user.

Regarding on this, 2 different frameworks should be developed to cover all the steps. In this project the framework developed corresponds to first type explained in the previews paragraph.

4.2. SIT_AA_01 solution proposal

As a result of the analysis done, an automation framework was developed trying to satisfy the needs of the company in this concrete area (SIT). It's true that this solution is focused in the first step of the process but from here, other steps can be included.

First of all, there was a customer Server accessing and occupying problem. Only the Project Manager and some of the qualified engineers involved on every project have permission to enter on the customer Database and like in most of the Telecom companies, this people are involved in other projects at the same time. So it is not easy to schedule an SIT, taking into account that the person in charge of the process has to be fully available for at least 3 days (normal execution is between 3 and 5 days, depending on the complexity of the network and the amount of data). In the other hand, being inside the Remote Server could affect to the correct operation of the system and other associated issues.

To solve this, instead of entering every time to the Remote Server to check the files, for this first step we only need some specific information of them. The solution proposed was to enter one time to the Server (all of them are in Unix OS), find the desired files, list them with the appropriated command (`ls -lhtr`) and save this list into a txt or csv file. By doing this, both problems are almost solved.

Anyone with access to the servers (MYCOM OSI or Customer people) can generate this files, and any of the company qualified engineers can pass the SIT, so the dependence of the PM or the others involved in the project is no further an issue. It also solves the second problem, the occupation of the Server, because the Server is only once consulted. After all the needed files for the test are saved, there is no more need to access again to the Database.

An example of file used for the code is shown in the next figure:

Figure 3: Example of the content in a used file

```
-rw-r--r--. 1 mycom mycom 508K Feb 12 2014 spm_erics_201308291200+0100_60.csv
-rw-r--r--. 1 mycom mycom 508K Feb 12 2014 spm_erics_201308291100+0100_60.csv
-rw-r--r--. 1 mycom mycom 519K Feb 12 2014 spm_erics_201308291000+0100_60.csv
-rw-r--r--. 1 mycom mycom 543K Feb 12 2014 spm_erics_201308290900+0100_60.csv
-rw-r--r--. 1 mycom mycom 573K Feb 12 2014 spm_erics_201308290800+0100_60.csv
-rw-r--r--. 1 mycom mycom 599K Feb 12 2014 spm_erics_201308290700+0100_60.csv
-rw-r--r--. 1 mycom mycom 613K Feb 12 2014 spm_erics_201308290600+0100_60.csv
-rw-r--r--. 1 mycom mycom 625K Feb 12 2014 spm_erics_201308290500+0100_60.csv
-rw-r--r--. 1 mycom mycom 630K Feb 12 2014 spm_erics_201308290300+0100_60.csv
-rw-r--r--. 1 mycom mycom 638K Feb 12 2014 spm_erics_201308290200+0100_60.csv
-rw-r--r--. 1 mycom mycom 639K Feb 12 2014 spm_erics_201308290100+0100_60.csv
-rw-r--r--. 1 mycom mycom 635K Feb 12 2014 spm_erics_201308290000+0100_60.csv
-rw-r--r--. 1 mycom mycom 631K Feb 12 2014 spm_erics_201308282300+0100_60.csv
-rw-r--r--. 1 mycom mycom 630K Feb 12 2014 spm_erics_201308282200+0100_60.csv
-rw-r--r--. 1 mycom mycom 628K Feb 12 2014 spm_erics_201308282100+0100_60.csv
-rw-r--r--. 1 mycom mycom 628K Feb 12 2014 spm_erics_201308282000+0100_60.csv
-rw-r--r--. 1 mycom mycom 625K Feb 12 2014 spm_erics_201308281900+0100_60.csv
-rw-r--r--. 1 mycom mycom 622K Feb 12 2014 spm_erics_201308281800+0100_60.csv
-rw-r--r--. 1 mycom mycom 622K Feb 12 2014 spm_erics_201308281700+0100_60.csv
-rw-r--r--. 1 mycom mycom 599K Feb 12 2014 spm_erics_201308281600+0100_60.csv
-rw-r--r--. 1 mycom mycom 578K Feb 12 2014 spm_erics_201308281400+0100_60.csv
-rw-r--r--. 1 mycom mycom 530K Feb 12 2014 spm_erics_201308281300+0100_60.csv
-rw-r--r--. 1 mycom mycom 501K Feb 12 2014 spm_erics_201308281200+0100_60.csv
-rw-r--r--. 1 mycom mycom 493K Feb 12 2014 spm_erics_201308281100+0100_60.csv
-rw-r--r--. 1 mycom mycom 505K Feb 12 2014 spm_erics_201308281000+0100_60.csv
```

Then, the other important aspect to be solved was the automation of the process. This was by far the most difficult part of the solution for many reason.

First, the organization of the files are really different for every customer and technology and as the networks become more complex the structure of the files increase in levels and quantity. So the first idea of developing a script that could fetch automatically the files was discarded. A unified methodology of storage was proposed to the R&D department for further improvements in this direction.

Then, a similar problem happened for the file name structure. Every customer chooses the naming of the files, so is really difficult to develop a script that could compare each of the files with the name convention agreed in the Technical Specification Document.

In order to solve this, a unique file name standard should be used. As the solution is developed in Perl and for the actual requirements (name structure, time stamp and extension) the 3GPP file naming convention fits perfectly.

<Type><Startdate>.<Starttime>-[<Enddate>.]<Endtime>[_-<jobId>][_<UniqueId>][_<RC>]

- ❖ The whole naming standard it's further explained on the annex, but here it is a brief explanation:
The first field contains the results for single or multiple NEs and the granularity periods.
- ❖ The Stardate field indicates de data of the beginning of the granularity or the first granularity period.
- ❖ The Starttime, indicates in the same way as the Startdate field, the time.
- ❖ Endtime and Enddate correspond to the finishing date and time periods of granularity

So the automation framework was developed in Perl code and it's based in 3 different modules (Preview, Comparison and Report).

Every time the user selects one of the options, the program asks for the path of the files locate on the same computer as the script is running.

The first option, Preview Mode, asks for one file and after entering the correct path gives different options to choose. Size, Month, Day, Time or Year, File Name, File Extension or the Complete Line are the 7 options to be previewed on the screen by the user. When an option is chosen, the script shows a numbered list by rows with the different data and a final counter of the total number of rows.

On the Comparison Mode, the user is asked for two file paths. After entering the correct ones, the script shows the same 7 options to the user, but this time the output is, as the name of the module says, a comparison between rows in both files. The output is shown on the screen and sorted by rows, giving information even if the data are the same or in case they are different, the differences between them. As in the Preview module, a final counter with the total of files is shown.

The last option, the Report Mode, is almost the same as the Comparison Mode, but in this case all the options are selected automatically and only when there are differences between files the row is printed. Another characteristic of this mode is that the output is shown on the screen but it also generates a csv or txt file to save the Report. After entering the file paths, the script asks the user for a path to save the output file.

With this framework and developing little modifications on the name option is easy to accomplish the target of this Step. For now, if the company do not agree to urge the customers to use one unique naming standard, the solution proposed is the most automated one and although this framework is not completely ready to use, it could be the way to lean on for developing a useful automated testing tool.

4.3. Next steps solution proposal

Once the naming standard is chosen, is easy to include more steps to the actual framework with no much programing effort.

AA_02 and AA_03/AA_04 can be automated by adding always in the same position the release tag to the file name, making easy to filter and compare with the valid releases of the adaptor.

AA_20 is in charge of checking the file sizes by technologies with that value below a certain threshold. With a small change on the size option of the framework, a sum counter could be easily introduced and this step would be passed.

5. Budget

The funds for this project have been many hours of work and the Perl programing course which has allowed me to write the code in the same language used for the projects in the company. MYCOM OSI provides the programs, trainings and access to the different servers. If you were to finance this project from scratch, this would be its costs:

- ❖ Perl Course 60€

- ❖ 720 hours x 10 euros/hour (estimated price of a junior engineer) = 7200 euros.

Total cost: 7260 euros.

6. Conclusions

The aim of the project was always to develop an automation framework for one of the testing processes of the main software on the company. After working really hard on this solution with no much time, doing everything from the scratch and with little support on the creative area, I think that the final results are quite consistent.

The framework developed is able to test the steps that involve the ETL process and the Database files, which are the ones that requires more time and effort.

This was my first time coding outside of the University and with a completely new coding language, with the difficulty that concerns.

I've been working with different departments from all over the globe, working alone or in a team, learning a lot from the company business and their methodology. All of this to try to reflex it latter on this project.

Now the R&D department, which is in charge of improving and adapting the software to their needs, has a framework to start working with and a big amount of notes and analysis of the current process.

To summarize, it's being a great project to start my carrier outside of the University, where I learned a lot of different things related with the Telecom industry and gave me the capabilities and the confidence to work in any other place and in almost any other project.