Automation Development and Optimization for OR-SIM

Submitted By

Ms. Annu Manik

13MCEN31

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2015

Automation Development and Optimization for OR-SIM

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering(Networking Technologies)

Submitted By

Ms. Annu Manik

(13MCEN31)

Guided By

Prof. Prajakta Rathod

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2015

Certificate

This is to certify that the major project entitled **"Automation Development and Optimization for OR-SIM"** submitted by **Ms. Annu Manik (Roll No: 13MCEN31)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Prajakta Rathod
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Gaurang Rawal
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Ms. Annu Manik**, Roll. No. **13MCEN31**, give undertaking that the Major Project entitled "**Automation Development and Optimization for OR-SIM**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering(Networking Technologies)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date:May 23, 2015
Place:Ahmedabad

Endorsed by
Prof. Prajakta Rathod

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Prajakta Rathod**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

I am extremely thankful to my Manager **Mr. Aashish Nadkar** from Oracle for his constant support and guidance and for giving me valuable information regarding to this work.

- Ms Annu Manik
13MCEN31

Abstract

A new democratic relationship between consumer and retailer is emerging. The single biggest challenge for retailers is to understand, empower and represent the new digitally empowered consumer operating across all touch points including mobile. Oracle explores new retail democracy and establishes opportunities for retailers to get real-time access to store management and reporting functionality wherever it on the front need register, the back-office PC, or tablet devices. Oracle provides many retail solutions like Oracle Retail Merchandising System, Oracle Retail Store Inventory Management System, Oracle Retail Point Of Sale,etc to helps the retailers work smoothly.

Oracle Store inventory management system helps store to perform in-store operations including item look-up, receiving, adjustments, stock count, and inter-store transfers. SIM helps to maintain accurate inventory positions, track where the inventory is and can provide visibility to this information, saving the sale. OR-SIM manages transfer or return merchandise from warehouse or directly from vendors.

Before releasing a newer version of any product it needs to be thoroughly tested and all the possible scenarios need to be covered. Hence, in this project the current version 14.1 of the Oracle Store Inventory Management product will be tested by automating the various test scenarios. The main objective of this project is to execute the existing test scripts written using VBScripts for automating the regression testing of the OR-SIM product using the automation tool called HP-QTP. These testcases are to be migrated to Oracle's newer solution SYNERGY using JAVA language for getting an optimum automated testing suite for SIM. Also the methods of Data Generation will be written using PL SQL queries in Oracle Database 12c, for creating foundation data for regression testing and addition of new functionality to be done for making automation testing optimized.

Abbreviations

OR-SIM	Oracle Retail Store Inventory Management
HP-QTP	Hewlett Packard QuickTest Professional.
MO-KEY	Modular - Keyword driven.

–

Contents

List of Figures

Chapter 1

Introduction

1.1 Oracle Retail

One of the biggest challenges a retailer faces in todays customer centric environment is loss of lot many sales opportunities which may cause due out-of-stock positions or may be worse than that, because the store associate may be unable to locate an item. Oracle wanted to explore new retail democracy and to establish what opportunities are available for retailers to differentiate and grow their business in a crowded market dominated by consumer power, the global market place and brand convergence.

Retailers are able to have an open and integrated suite of business applications, server and storage solutions. These are engineered to work together to optimize every aspect of their business from Oracle suites. Top 20 retailers worldwide including fashion, grocery, hardlines and specialty retailers use Oracle solutions to drive performance, fuel growth across traditional, mobile and commerce channels and deliver critical insights [3].

1.2 Overview of OR-SIM

Oracle Retail Store Inventory Management(OR-SIM) provides a brand platform and enable true multichannel retailing and helps retailers to deliver an experience that meets and exceeds the ultimate needs and expectations of consumers. SIM can provide inventory's in-store activities visibility streamline, improve merchandise management and productivity in-store, supports remote-store processes, reduce labor costs, and manage the store-level profit and loss. The store managers and store personnel can now quickly and easily perform all kind of in-store operations including receiving merchandise from warehouse or directly from vendors, ordering merchandise transferring stocks, managing physical inventories, conducting stock counts, creating, editing, and dispatching returns from the store to a company-owned warehouse, a finisher, or directly to a vendor using a high level speed internet connection and portable handheld wireless devices [3].

Oracle Retail store inventory management is a desktop application organized by functional areas. SIM has three tiers in its technical architecture: a client tier, a server tier, and a data tier. The client tier contains a PC client (a Java desktop application) and handheld devices. The SIM server (deployed as a J2EE application inside the Web logic Application Server) comes under Servier tier, also called as middle tier. The data tier consists of an Oracle 11-g database and an LDAP directory.

1.3 Testing Automation

It is a challenging task for IT industry is to deliver software products that meet the business needs. It is essential to deliver such a software which is bug free. The bugs in software can cause heavy loss for IT organization in case they are not solved before delivering the software to customers. To deliver a good quality software, an efficient testing is to be conducted. According to statistics, half of the total cost occoured for software development is incurred due to software testing and it is even more if it is the case of critical software. Testing can be classified as unit testing, integration testing, system testing, regression testing, performance testing, alpha testing, beta testing,acceptance testing,stress testing,mutation testing, etc., on basis of time, scale and performing methods. The test data can be designed either manually or automatically.

Large emphasis are put by software engineering research on automating the software development process as it helps producing more complex codes with comparatively less effort. For testing these softwares, there is need of finding advanced innovative and easy to implement supporting procedures for automating the test process.

1.3.1 Testing Automation Tool: HP-QTP

Automation tools designed for testing some particular test environment automatically, which may include Windows and web automation tools, etc. Tools serve as a driving agent for an automation process. Oracle uses Hewlett Packard's QuickTest Professional 11.0 as automated functional testing tools for product test automation [4].

These tools can be used to create the test scripts that are easy to maintain and reuse and that allows the customers to verify the functionality of their implementation and avoid any regression in quality early in the testing cycle.

1.4 Project Motivation

Oracle- Retail is planning to change their testing automation system using HP-QTP. The plan is to use Oracle's newer testing automation software SYNERGY instead of HP-QTP due to several reasons listed below:

- HP-QTP is a licensed software, so need to pay quite a big amount of money to buy the license every year.
- Frustration with the performance and stability of QTP.
- Too much time being spent on automation infrastructure instead of product automation.
- Lack of good and stable support of emerging technologies (Adobe Flex, ADF).
- Need to move towards test case creation using a keyword-driven methodology.
- Need of a very flexible automation platform (support for languages such as Java, JavaScript, Python, Scala).

In order to over come the above stated issues, there is need to first undrstand how the HP-QTP is used for test automation by implementing all the existing 425 Test cases of OR-SIM, need to explore the advantages and functionality of SYNERGY and to write all the test cases with the help of SYNERGY.

Chapter 2

Literature Survey

2.1 Oracle Retail Store Inventory Management

Oracle Retail store inventory management provides a brand platform and enable true multichannel retailing. OR-SIM enables the retailers to deliver such an experience that helps meeting and exceeding customers' needs and expectations. SIM can provide inventory in-store activities' visibility streamline, improvement of merchandise management and productivity, support remote store processes, reduce labor costs and manage store level profit and loss [3]. SIM contains four major functionality under which there are multiple features as follows.

- Look-ups- At any time, user can look up detailed information about inventory items, suppliers, and containers.
- System and Store admin options: Through the Store Admin window, the administrator can set values for options that control a variety of SIM behaviors.
- Shipping and Receiving: SIM manages transfer or return merchandise from warehouse or directly from vendors. It includes three features as follows.
- DSD: Stands for Direct store delivery in which the supplier can directly drop the merchandise at the store.
- Transfers: Transfers handles the movement of goods from one store to another within a company.

- Returns: Through Returns user can create, edit, and dispatch returns from the store to a company-owned warehouse, a finisher, or directly to a vendor.
- Sequencing: Sequencing helps in knowing the relative location of an item in a store.
- Item Request: Item Request is use to request items to cover shortages and increased demand. Security: SIM roles and permissions are designed and assigned by a system user, probably a system administrator, whose role has the necessary permissions to create and modify SIM roles and assign.

2.1.1 OR-SIM Technical Architecture

OR-SIM high level technical architecture consist of three tiers : Client Tier, Middle(Server) tier and Database Tier. SIM Application is basically used in inventory and stores and the users may not be having pure technical skills. Keeping that point into consideration SIM has been made such that it can be easily deployed in variety of clients including a desktop computer, handheld wireless device , etc.

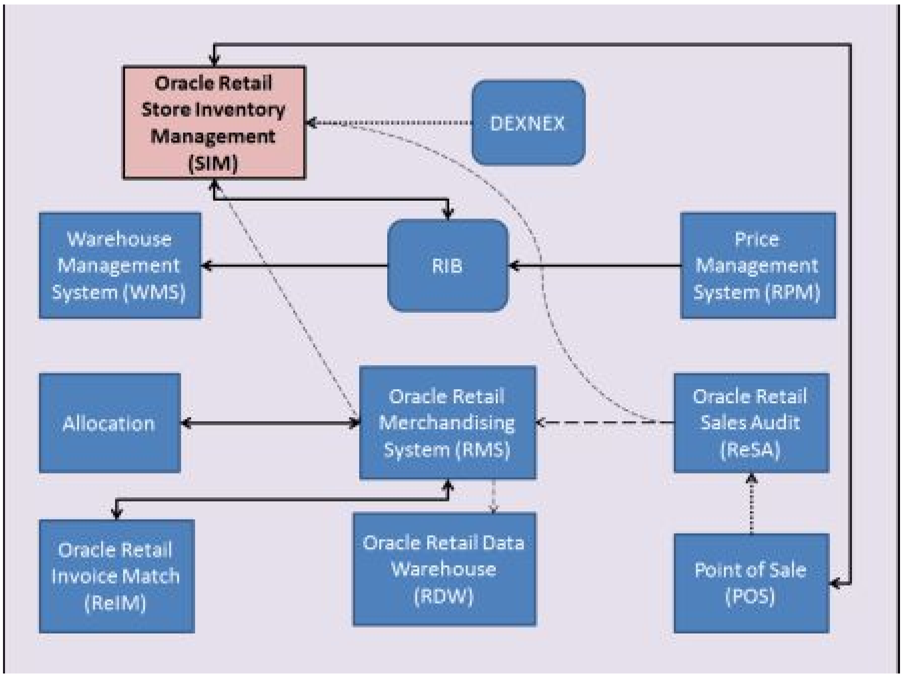
Through graphical user interface (GUI), data are presented to the user, as well as data is received directly from the user through the front end [6]. The PC side of client is developed using Swing, a toolkit which is used for creating rich GUIs in Java applications. the Oracle Retail Wireless Foundation Server helps the handheld devices to communicate with the SIM server.

The middle (Server) tier provides the link between the SIM client and the database. It handles all of the business logic processing that occurs within multi-tiered architecture of SIM, that too virtually. The middle tier handles business functionality services. Work can be distributed seamelessly among the servers as the SIM servers runs on multiple application servers in a stateless system.

The database tier is the storage platform for application which contains the physical data used throughout the application [6]. Databases are actually the data tables and views that are used in SIM servers and then passed to clients.

2.1.2 SIM Integration with other Oracle Products

The above diagram illustrates various Oracle Retail products and databases that SIM interfaces with and the data flow among all the products. The Solid line with double arrows represents files that are goin back forth in read and write fashion. being sent



Oracle Retail Store Inventory Management (SIM)
DEXNEX
Warehouse Management System (WMS)
RIB
Price Management System (RPM)
Allocation
Oracle Retail Merchandising System (RMS)
Oracle Retail Sales Audit (ReSA)
Oracle Retail Invoice Match (ReIM)
Oracle Retail Data Warehouse (RDW)
Point of Sale (POS)

Figure 2.1: SIM Integration

from Dotteedone application to other. SIM is fully integrated with RMS, i.e., Oracle Retail Merchandising System, with WMS, i.e., Warehouse Management System, Invoice Management System, ReSA (Oracle Retail Sales Audit) and POS(Point Of Sale) using RIB(Oracle Retail Integration Bus). Most of the transactions are done in near real time using real time integration.

2.2 Testing Automation

It is a challenging task for IT companies is to deliver software products that meet the business expectations and needs. It is a necessity to produce and deliver a software which is bug free. The software having bugs causes heavy loss for IT industries if at all they are not solved before delivering the software. To deliver a good quality software, an efficient testing is to be conducted.

Half of the total cost incurred for development of software is due to software testing, and if it is a critical software the cost is even high, according to the statistics [1]. Testing can be classified as unit testing, integration testing, system testing, regression testing, performance testing, alpha testing, beta testing,acceptance testing,stress testing,mutation testing, etc., on basis of time, scale and performing methods. The test data can be designed either manually or automatically.Large emphasis are put by software engineering research on automating the software development process as it helps producing more complex codes with comparatively less effort. For testing these softwares, there is need of finding advanced innovative and easy to implement supporting procedures for automating the test process.

2.2.1 Test Automation Principles

A systematic literature review was done on test automation for academic views while a survey was done for getting the practitioners' views to know how the data was collected from software professionals. The survey shows that benefits of test automation were related to test re-usability, repeatability, test coverage and effort saved in test executions. The limitations were high initial invests in automation set-up, tool selection and training. According to a study quater of the respondents said that available tools in the market offer a poor fit for their needs. Hence a customized test automation framework has to be developed according to their respective functionality.

The main motivation for automating the test cases is that the non-programmers should not loose the comfort level they would have with the degree of informality, flexibility and ambiguity which is inherent in natural-language expressions and which would be closer to their thought processes. Thus the ultimate goal of automation testing is to improve the efficiency of automating the manual tests by automating the testing task.

Automated software testing is automating the manual process of testing software. Automated Testing is gaining interest these days as it reduces the time and increases efficiency for testing the applications. The basic terms used in automation testing domain are:

- Test Automation: Developing software for automatically test a software product.
- Test data: Data used from database specially designed to be used in testing an application.
- Testcase: Test cases are the list instructions which has simply the call functions to call the testing keywords. These set of instructions are executed one after other in specified series.
- Test suite: A bunch of test cases which are executed in sequence in order to test an application.

Following are the steps of Automation Test Process.

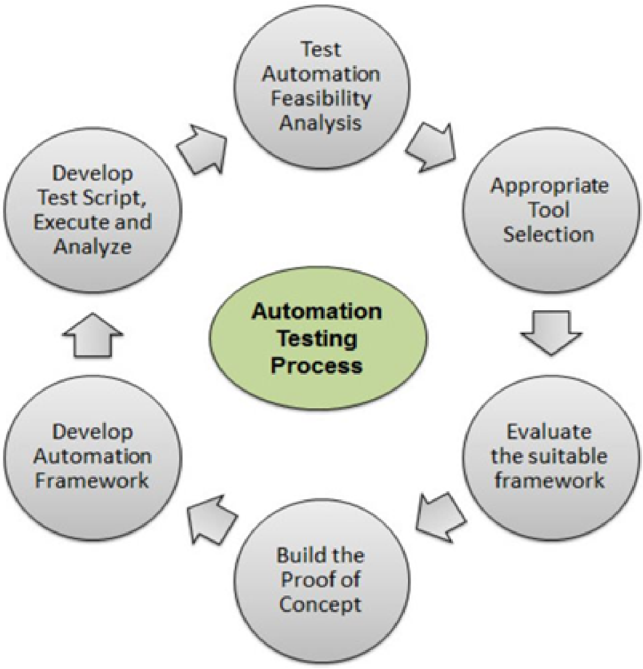

Test Automation Feasibility Analysis
Appropriate Tool Selection
Automation Testing Process
Evaluate the suitable framework
Build the Proof of Concept
Develop Automation Framework
Develop Test Script, Execute and Analyze

Figure 2.2: Automation Test Process [7]

1. **Test Automation Feasibility Analysis:** Checking the feasibility of the AUT that is to check if the application is eligible to get automated or not. All applications may not be able to get automated due to its limitations.

2. **Appropriate Tool Selection:** Selection of tool is dependent on the technology in which the application is built.

3. **Evaluate the suitable framework:** There are various kinds of frameworks that supports automation, each of which has its own significance.

4. **Build the Proof of Concept:** Proof of Concept(POC) is developed with an end to end scenario to evaluate if the tool can support the automation of the application. As it is performed with an end to end scenario which will ensure that the major functionalities can be automated.

5. **Develop Automation Framework:**After building the POC, framework development is carried out, which is a crucial step for the success of any test automation project. Framework should be build after diligent analysis of the technology used by the application and also its key features.

6. **Develop Test Script, Execute and Analyse:**On completion of Script development, the scripts are executed, their results are analysed and defects are logged, if any. The Test Scripts are usually version controlled.

2.3 Testing Automation Tool: HP-QTP

HP QuickTest Professional (QTP) is the HP Unified Functional Testing (UFT) software which is meant to provide the functional and regression test automation suite for software applications and environments. HP UFT is generally used for quality assurance of enterprises[1].

Keyword and scripting interfaces and features of a graphical user interface are supported bu HP-QTP. Visual Basic Scripting Edition (VBScript) scripting language is used in HP-QTP to specify a test procedure, for manipulating the objects and controling the application under teste [4].

HP QuickTest Professional 11.0 for E-Business Suite simplifies creating and maintaining automated tests of Oracle E-Business Suite applications. The QuickTest Professional E-Business Suite Add-in enables customers to create and run tests using special test objects and methods that have been customized for Oracle E-Business Suite.

QuickTest Professional provides customized test objects, methods, and properties that make tests and components simple to read, maintain, enhance, and parameterize, enabling both advanced and novice users to create sophisticated tests and components for applications developed using Oracle E-Business Suite [4].

Features provided by HP QTP for test-automation are:

- **Object Repository:** Collection of object and properties with which QTP identifies the objects. Various actions can be carried out on these collection of objects.

- **Actions:** Actions helps testers to divide scripts into groups of QTP statements called actions.

- **Data Tables:** A Data Table is similar to Microsoft Excel helps testers to create data driven test cases that can be used to run an Action multiple times.

- **Checkpoints:**Checkpoint refers to a validation point that compares the current value for specified properties or current state of an object with the expected value which can be inserted at any point of time in the script.

- **Synchronization:** Synchronization point is the time interface between Tool and Application under test. Synchronization point is a feature to specify delay time between one step and another of the test script.

- **Smart Identification:** QTP is unable to find any object that matches the recognized object description or it may find more than one object that fits the description, then QTP ignores the recognized description and uses the Smart Identification mechanism to recognize the object.

- **Debugging:** Debugging is in automation testing context, is a systematic process of spotting and fixing the coding issues in the automation scripts so that the script will be more robust and can spot the defects in the application.

- **Error Handling:** There are various ways on handling errors in QTP. There are three possible kinds of error type one would encounter while working with QTP.

- **Recovery Scenarios:** In order to recover the test from unexpected errors, and continue executing the rest of the script, Recovery Scenarios are used.

- **Environment Variables:** QTP environment variables are special types of variables that can be accessed by all actions, function libraries and recovery Scenarios.

- **Library Files:** In order to modularize the script, library Files are added to the QTP Script. It contains variable declaration, Functions, Classes etc. They enable re-usability that can be shared across test scripts. They are saved with an extension .vbs or .qfl

- **Test Results:** The Test Results Window gives us sufficient information to show the steps passed, failed etc.

- **GUI Objects:** There are various GUI objects with which QTP interacts during the script execution. Hence it is important to know the basic methods for the key GUI objects using which we will be able to work on it effectively.

- **Accessing DB:** using VBScript it is possible to connect and interact with databases using ADODB objects.

- **Descriptive Programming:** When objects in the application are very dynamic in nature, descriptions of the Objects are created using Descriptive programming to perform an operation on an object that is not present in the object repository.

2.4 MoKey Framework

In QTP there are different tyes of frameworks uch as Linear, Data-driven, Functional decoposition, Keyword driven, Hybrid and BPT framework. But Oracle products uses Keyword driven framework. **MoKey** stands for **Modular Keyword** Driven Framework. MoKey is a Hybrid Framework made up of Keyword Driven and Functional decomposition framework. Functional decompositon is also knoewn as Modular framework.

In **Modular framework**, one can write the entirecode in different functions and then these functions are called in the scripts whenever required. These functions can be usd as both re-uasble and non-reusable. In **keyword driven framework**, on can create different functions and can assosciate diferent actions with these keywords.When a particular framework is encountered by QTP, it executes the action associated with that framework. Oracle products uses a hybid of both of these frameworks.

Figure 2.3 shows the whole process handled in MoKey Framework.

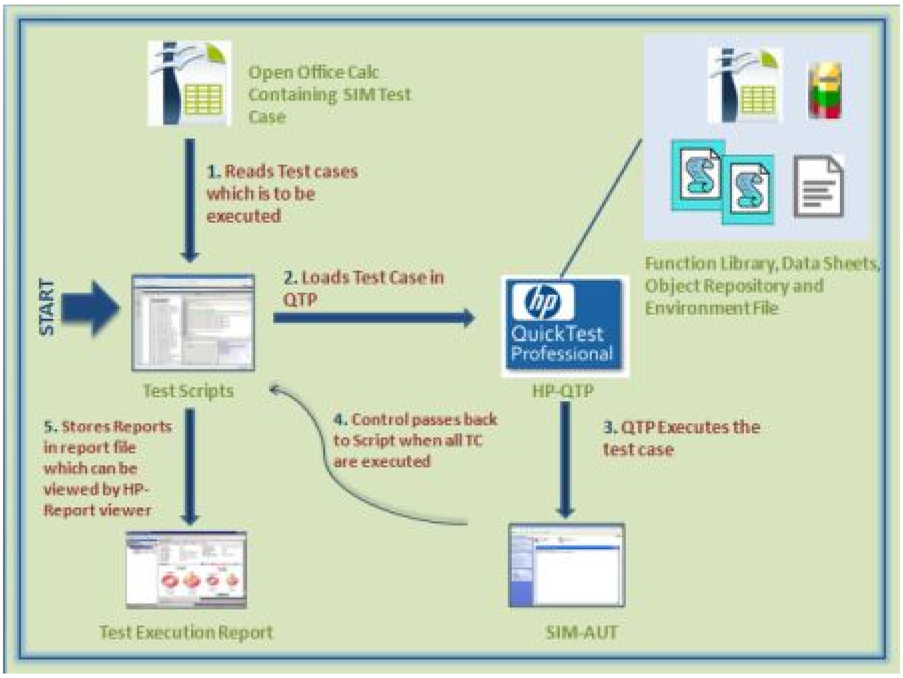

Open Office Calc Containing SIM Test Case
1. Reads Test cases which is to be executed
START
2. Loads Test Case in QTP
hp
QuickTest Professional
Function Library, Data Sheets, Object Repository and Environment File
Test Scripts
HP-QTP
5. Stores Reports in report file which can be viewed by HP-Report viewer
4. Control passes back to Script when all TC are executed
3. QTP Executes the test case
Test Execution Report
SIM-AUT

Figure 2.3: WorkFlow of MoKey Framework

MoKey framework basically consist of 7 major components.They are:

- **Test Scripts:** Test Scripts wors as he driver of the automation testing. Once the testcript is run in QTP it loads the TestCase and does the data parameterization. It loads the libraries and object repositories and the environment file in QTP. Steps are written in the Test Script for testing, those steps gets executed in QTP.
- **Environment File:** It contains nothing but the environment variables which can be used by any function. It is kind of global declaration of variables or objects.
- **Object Repository:** Object repository is the repository which has a heirarchy of objects based on Application. It is application specific.
- **Libraries:** Libraries are nothing bt the pool of Functions. These contains all the Application specific functions as well as generic keywords which can be used by all the testcases simultaneously.
- **Test Cases:** Test cases contains the actual steps for testing the application. It is written in the Excel sheets. Function calls are made in the test cases which are used from the function libraries.
- **Master Data table:** Master Data table is the table containing all the data related to application. These data gets used as well as updated by the SQL queries written in the testcases for validating the application.
- **Application Under Testing:** Application which is to be tested and is in process of testing is considered as AUT.

2.4.1 Need of QTP MODULAR Framework

Code re-usability and data re-usability are the major advantage of Modular frame work. The advantage of this approach is that the re-usable code would always stay at one place and thus it would be easy to maintain the code as it just needs to make the changes at a single place only. Calling the function wherever required makes sense of code re-usability. Here functional libraries and the object repositories are re-used by many testcases.

2.5 SYNERGY

2.5.1 Introduction and Need of SYNERGY

Oracle Retail uses HP-QTP for testing automation system. But there are some reasons for which Oracle is now opting for SYNERGY in place of HP-QTP. Several reasons for this replacement are listed as below:

- HP-QTP is a licenced software, so need to pay quite a big amount of money to buy the licence every year.
- Frustration with the performance and stability of QTP.
- Too much time being spent on automation infrastructure instead of product automation.
- Lack of good and stable support of emerging technologies (Adobe Flex, ADF).
- Need to move towards test case creation using a keyword-driven methodology.
- Need of a very flexible automation platform (support for languages such as Java, JavaScript, Python, Scala).

2.5.2 WorkFlow In SYNERGY

Synergy is Oracle's own Testing Automation tool which has a wide range of features that helps over come the issues faced with the HP-QTP tool. Oracle is planning to follow the same MoKey framework for automation as it does with the QTP with some changes according to the features provided by Synergy.

Basic components of Synergy Framework are explained as below:

- **Project:** Project is the Parent folder which contains all the components required for automation in Synergy. A project will gives the access to additional features such as easy version control system integration.
- **Test Suite:**A test suite is a collection of tests that validate whether a software program exhibits a specified set of behaviors and any supporting constructs/information necessary for the execution of contained tests. Synergy test suites may contain the

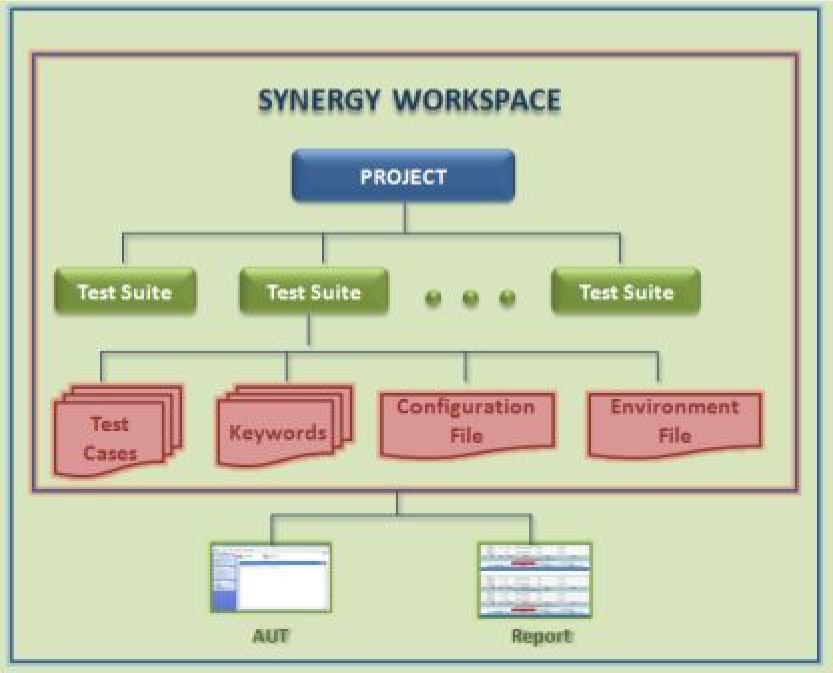


Figure 2.4: WorkFlow in SYNERGY

following components: Test Cases,Folders,User Keywords, Resource, links, Global Variables and Configuration settings.

- **Automation Libraries:** Automation libraries contain those lowest-level keywords, often called library keywords, which actually interact with the AUT. All test cases always use keywords from some library, often through higher-level user keywords.The recorder will automatically import any automation libraries needed during recording

- **Test Case:** Test Case are just the instructions given to the tool for automating the test scenarios. Any test cases in a folder are executed based on the position of the test case in the folder and the position of the folder in the test suite.

- **Configuration File:** This file contains the details of configuration need for specific application under testing. It mainly contains the configurations such as database driver details and username and passwords for database access.

Chapter 3

Exisitng Work

3.1 Automation of OR-SIM using HP-QTP

Automation of OR-SIM in Oracle is done using HP-QTP,an automation tool in which MO-KEY framework has been incorporated. The Test data and seed data are established using Foundation data scripts. These foundation data scripts are written using VB script as a scripting language in HP-QTP. Code for testing scenarios are written in test scripts. These test scripts is the parent script in which other resources are called as child script.

Depending on the test scenarios to be tested, the selection of resources is done. for navigating through application UI, object libraries are made. These object libraries stores the X-path for each object on the UI screen and its properties. These object libraries are further used in script for automatically navigating to the required object on UI. New objects can be added by record and play function provided by QTP. application data are stored in CSV files. These files are called as a resources in the test script.

Test script first reads the test cases which is to be executed. Test cases are written in open office calc. Once the loading of all the functional libraries, object repositories, environment file, which basically contains environment details and the configuration details, and test case, the execution of test begins. QTP plays the script and the AUT gets tested automatically according to the steps written in test case. Once the testing is done, a report is generated in QTP. which can be viewed in HP-Report viewer. HP-Report viewer gives a step wise result of the test case with a reason behind failure, if any, with screen shots of AUT.

3.2 Foundation Data Schema

For testing any application four types of data are required. These are 1.Seed data, 2.Application data 3.Foundation data and 4.Test data. Seed data is the base data required for setting up the base for establishing the application. For example data like organization unit details, merchandise details, etc required for testing a retail application. Application data is the data which is used to feed in the UI fields. For example data regarding types of actions available in a combo box. Foundation data is the test bed which is created in the initial step before testing is started. This data is used for testing various test scenarios. Test data actually comes under foundation data. Data added through application gets added to the database, such kind of dynamic data can be considered as test data. Test data are more specific to test cases.

Existing schema for foundation data depends on the test cases. Data is fetched from all the application databases and stored in a view called master data table. This master data table has an individual row specific for each test case scenario. A fresh item is created for each new test case scenario. As a result every time a new row is added to master data table for each test case. When the application is to be tested the data for that particular test case is being fetched from master data table.

This scheme has its own pros and some cons. It reduces the direct use of application data tables, resulting into less chances of getting application data tables altered. As it needs to fetch or add data in only one table for each new test case it becomes easy to maintain the data. On other side, there are in total 15 types of Items used for testing the AUT. For each test case a fresh item id is generated. One test case may have n number of item creation. For one item creation, 7 data tables gets reflected. For each DB query, data connection is done with Server database. Reason for using this scheme was, continuous uses of same item ID cause shortage of stock on hand for that particular item id. If other test case uses same item id, it will fail due to lack of stock on hand for that item id. For this reason the existing scheme of foundation data creation creates a new item for every test scenario.

Chapter 4

Project Motivation

The main objective of this project is to increase the performance of the automation testing of OR-SIM in terms of time. There are two approaches for taken for performance enhancement of automation testing. One is to select a better automation tool for automating the test scenarios and secondly to optimize the existing foundation data creation scheme and functional code which ultimately reduces time complexity.

4.1 SYNERGY Over HP-QTP

Though HP-QTP is a highly used automation tool in current market, it has some noteworthy cons which motivates to use Synergy tool instead. As HP-QTP is a licensed software,Oracle needs to pay quite a big amount of money to buy the license every year. To work on QTP one should have enough knowledge of VB scripting so it is more of a language specific tool which demands for a skillful human resources. Too much time being spent on automation infrastructure instead of product automation. Lack of good and stable support of emerging technologies (Adobe Flex, ADF). There is a need of a very flexible automation platform that can support for languages such as Java, JavaScript, Python, Scala.

Synergy provides a full range of consulting expertise in Oracle Solutions and Tools, including Oracle Database Administration, BI, Data Warehousing, Data Mining, OLAP, Supply Chain Management, Finance Management, and more. It provides easy and fast way of adding new automation plug-ins for new technologies and common automation integrated development environment across all automation plug-ins. It also provides HPQC integration for metric reports. Synergy is Oracle's own automation tool so no need to pay as in case of licensed software QTP. As Synergy is Oracle's own automation tool, the tool itself can be mold as per automation requirements.

Automation infrastructure in Synergy is quite simpler compared to one used in QTP. In QTP all the resources are called and loaded in the script at run time. While Synergy supports all the resources to be within the tool itself. It doesn't need any other third party resources like excel sheet,etc. though is flexible enough to support importing the external files as well if needed. Hence the transitions between accessing files is less compared to QTP.

4.2 Code Optimization

4.2.1 Foundation Data creation Scheme

In Automation testing of OR-SIM, for each test case scenario one or more than one item are created. There are around 1000 test scenarios to be automated for testing OR-SIM. For each test scenario one or more items are created. For one item creation seven application data tables are reflected. For each DB query execution data connection is done with Server data base. This takes noticeable time when the automation run is done for a batch of test cases.Reason behind creating fresh items for each test scenario is that the continuous use of same item results the drainage of total stock on hand. If other test case try to use same item then due to shortage of stock on hand, it fails.

Other reason for having individual item creation for each test scenarios is that if any of the test case in a batch of test cases fails than the status of item may be in-transit which causes improper stock calculation. The next test case will not be able to use the item with in-transit status and it will ultimately fail. Hence some failure recovery code is required so that failure of one test case doesn't affect the further lined up test cases.

4.2.2 Test suite Execution Failure

To test an AUT all the individual test cases are clubbed into one test suite. A test suite consist of a bunch of test cases which are to be executed in sequence in order to test an AUT. The test cases are executed one by one in a particular sequence to test the functionality and behavior of AUT after every step and validate the data change in database accordingly.During execution one or more test case may get failed due to several reasons like technical issue, erroneous code, application performance issue, validation failure, database connection failure, sudden unavailability of environment or database, etc. If one of the test case fails due to any of the mentioned issue, rest of the further test cases fails in the test suite. Reason of the failure can be any of the two, application or database.

Consider a scenario where one test case is testing one functionality on screen x and due to some reason it fails in between. Now the execution of next test case starts. The next test case tries to locate an object on say, screen y. But the currently open application has screen x. So due to unavailability of that particular object on particular screen, the test case fails. In this case, some recovery scenario should be incorporated to avoid failure of further test cases.

Chapter 5

Proposed Work

5.1 New Scheme for Foundation Data Creation

As mentioned earlier, the existing scheme for creating foundation data for testing AUT creates a new fresh item for every new test case and each test case may need one or more than one item creation. Hence each time the test case runs, a new item id is generated. This ultimately result to a bulk of data in database. The reason behind this scheme was just to prevent further test case failure due to usage of same item, as the stock on hand decreases and item becomes unavailable.

The new proposed scheme reduces the bulk data creation. There are in total 15 types of items used for testing OR-SIM application.So foundation data creation script will create only 15 items, each of different type. Test cases will pick item ids from this 15 items and proceed with the testing script. Once the testing code has run, in the end of the test case, it will run the SQL query to replenish the stock on hand of the used item.

Following are the Steps for the new Foundation Data Creation scheme:

1. **Create Foundation data:**

```
For each itemType
    $CreateItemId(itemType)$
```

2. **Automation Test Script Begins:**

   (a) Fetch the prerequisite data: How many items needed, respective item type, qty = required SOH

(b) Check if the required number of items for required item type are available in DB. If not, call *createItemId(itemType)*

(c) Check if ItemStockOnHand less than qty, call *replenishItemSoh(qty)*

3. **Continue with the functional and DB automation test script.**

4. **Call *replenishItemSoh(qty)***

5. **End Test script**

In each test case an existing item will be used and at the end the stock on hand of that item is added by the amount used. There is a possibility that a test cases uses more than one item of same item type. For this scenario a condition is checked before item creation itself. In prerequisite for a test case the information regarding number of items required ant their type are taken. Using this prerequisite information, it can be checked about the item requirement and if the number of item required is more than available item for particular item type, the item creation queries can be executed from the test case and the dynamically created item than can be used further.

5.2 Robust Automation Test Suite using Recovery Scenario handling Functionality

As mentioned earlier,in a test suite the test cases are executed one by one in a particular sequence to test the functionality and behavior of AUT after every step and validate the data change in database accordingly.During execution one or more test case may get failed due to several reasons like technical issue, erroneous code, application performance issue, validation failure, database connection failure, sudden unavailability of environment or database, etc. If one of the test case fails due to any of the mentioned issue, rest of the further test cases fails in the test suite. To avoid the failure of other further test cases, a failure recovery handling code is needed. The proposed piece of code handles this scenario.

Following are the steps of the Function for Recovery Scenario:

Follow the below steps until the Login Page of AUT is open.

1. **if** the screen = Logout screen

```
            call Login()
        else
2.          if the screen has BACK option
                until screen = Logout screen
                        click Back
3.          else if screen = alert dialog box
                call Handle Alert() function
4.          else
                until screen = Logout screen
                        click Cancel
```

This code is incorporated at the end of each test case. Whether test case fails or passes this code will ultimately get to the Login screen of the application. Consider a scenario where a test case is testing some functionality of AUT. Due to some reason it fails. So now the application has the current screen open where the test case got failed. Now in test suite, the next test case starts executing. It tries to find the login screen to login the application but the current screen is something different. This kind of scenarios are handled by this function. The function first checks if the current screen is Logout screen. If the current screen is a Logout screen, it clicks *Login* button and test case ends here. If not, it finds for the available button or dialog box. If it finds *Cancel*, *Ok* or *Back* it will click the button to navigate to the Logout screen. If it finds any Alert box or any dialog box, it calls Handle Alert() function. In Handle Alert function it checks for the alert/error message, prints the error message to the result log and then clicks *Close*. This helps at the time of analyzing the result as it prints the error message.

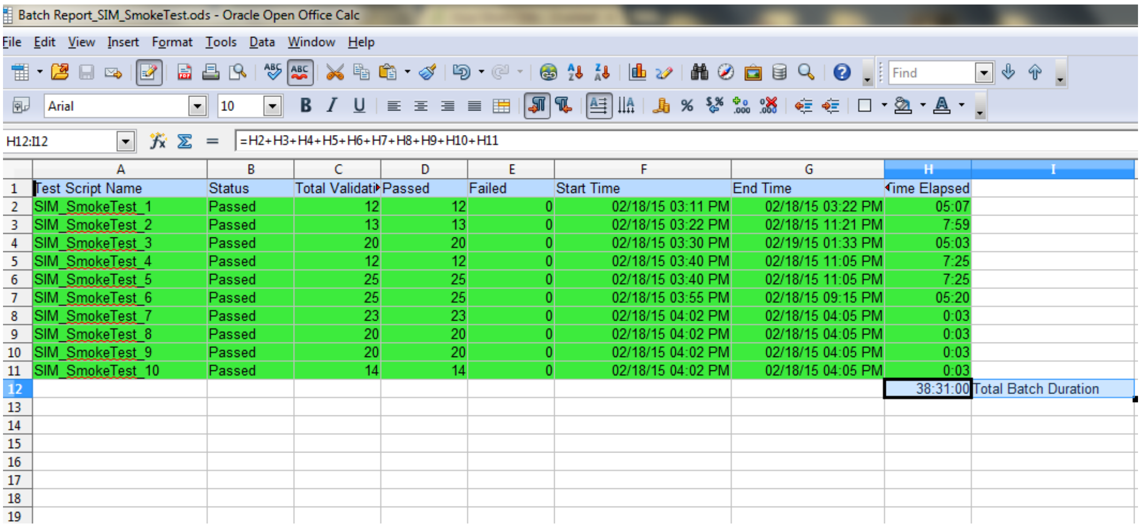
Chapter 6

Implementation and Results

As a part of phase-I the test scenarios were scripted and the AUT was tested using QTP with existing scheme.There after as a part of Phase-II, using Synergy tool, the same existing scheme was used for automation and testing and new foundation data scheme. As a part of Phase-III the AUT is tested with the proposed scheme for foundation data creation and new functionality.

A test suite of 1000 test cases was executed in QTP and were coded and executed in Synergy with existing functionality scheme and New functionality scheme with new foundation data creation scheme. A test suite of 200 test cases was taken as test bed for comparison of all three phases. Figure 6.1 shows the run result for phase-I in which a test suite of 10 test cases was executed in QTP. Total execution time taken was 38 minutes 31 seconds. Figure 6.2 shows run result for phase-II in which same set of test cases was executed and the total time taken was 19 minutes 10 seconds. Same set of test cases with additional functionality of Recovery scenario and new foundation data creation scheme was ran in Synergy, and it took 13 minutes 14 seconds for complete execution which is shown in figure 6.3.

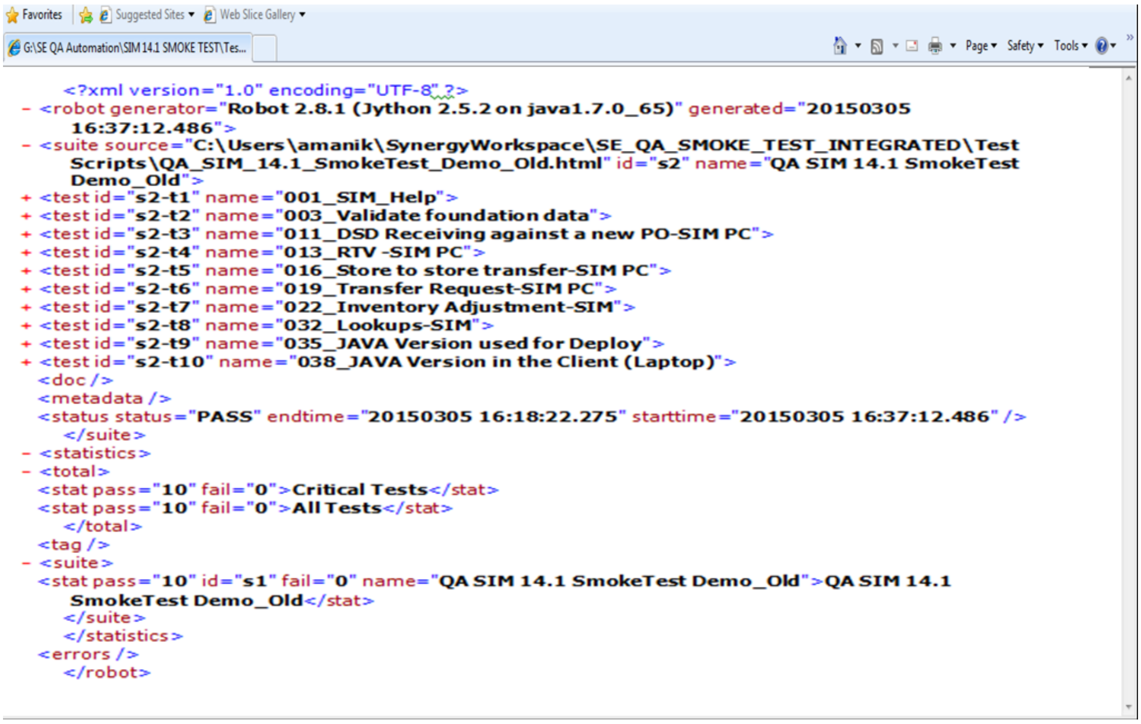
Figure 6.4 shows the comparison of run results for all three phases. The execution of existing functionality code using QTP takes maximum time compared to the time taken by execution of existing as well as new functionality added test suite ran using Synergy. Moreover, the test suite with new functionality scheme takes less time compared to the existing scheme.


Batch Report_SIM_SmokeTest.ods - Oracle Open Office Calc
File Edit View Insert Format Tools Data Window Help
Find
Arial
10
H12:I12

=H2+H3+H4+H5+H6+H7+H8+H9+H10+H11

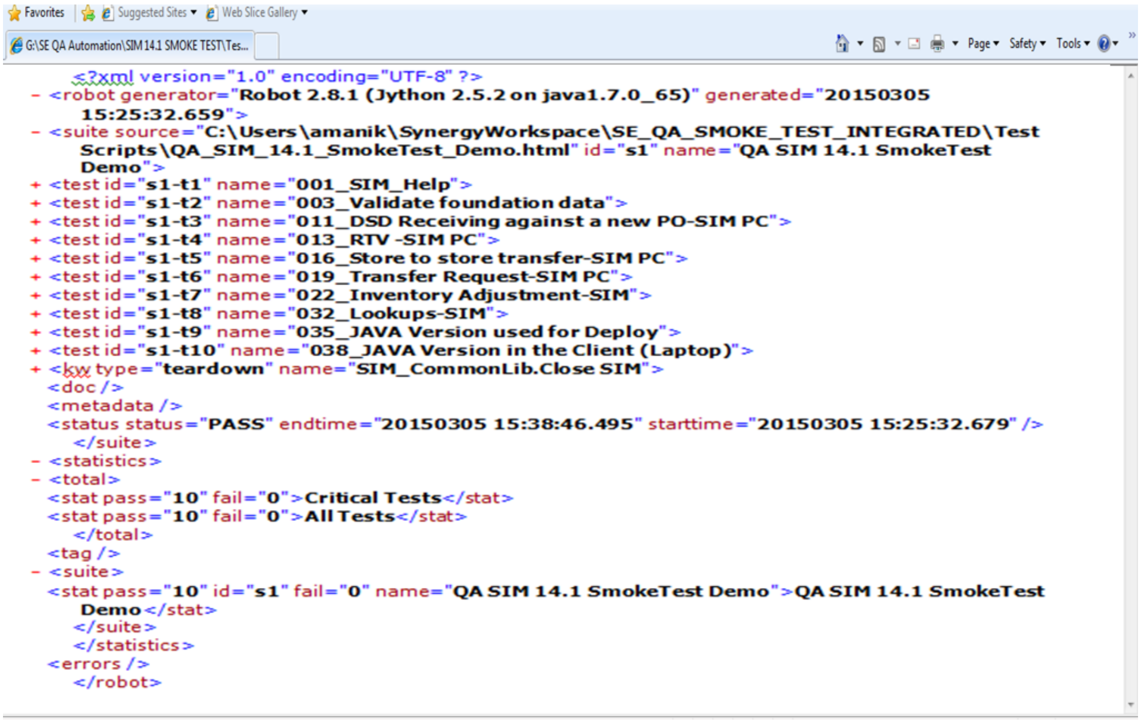
	A	B	C	D	E	F	G	H	I
1	Test Script Name	Status	Total Validati	Passed	Failed	Start Time	End Time	Time Elapsed	
2	SIM_SmokeTest_1	Passed	12	12	0	02/18/15 03:11 PM	02/18/15 03:22 PM	05:07	
3	SIM_SmokeTest_2	Passed	13	13	0	02/18/15 03:22 PM	02/18/15 11:21 PM	7:59	
4	SIM_SmokeTest_3	Passed	20	20	0	02/18/15 03:30 PM	02/19/15 01:33 PM	05:03	
5	SIM_SmokeTest_4	Passed	12	12	0	02/18/15 03:40 PM	02/18/15 11:05 PM	7:25	
6	SIM_SmokeTest_5	Passed	25	25	0	02/18/15 03:40 PM	02/18/15 11:05 PM	7:25	
7	SIM_SmokeTest_6	Passed	25	25	0	02/18/15 03:55 PM	02/18/15 09:15 PM	05:20	
8	SIM_SmokeTest_7	Passed	23	23	0	02/18/15 04:02 PM	02/18/15 04:05 PM	0:03	
9	SIM_SmokeTest_8	Passed	20	20	0	02/18/15 04:02 PM	02/18/15 04:05 PM	0:03	
10	SIM_SmokeTest_9	Passed	20	20	0	02/18/15 04:02 PM	02/18/15 04:05 PM	0:03	
11	SIM_SmokeTest_10	Passed	14	14	0	02/18/15 04:02 PM	02/18/15 04:05 PM	0:03	
12								38:31:00	Total Batch Duration
13									
14									
15									
16									
17									
18									
19									

Figure 6.1: QTP Result File for Existing Scheme



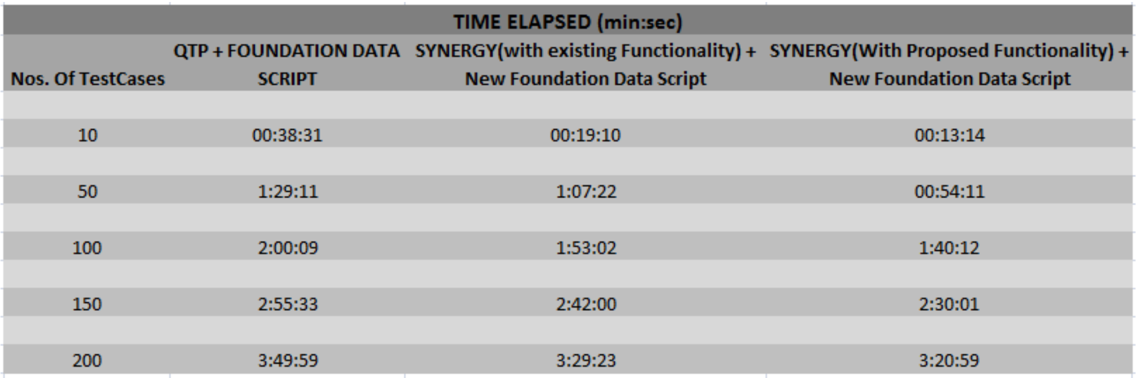
```
    <?xml version="1.0" encoding="UTF-8" ?>
- <robot generator="Robot 2.8.1 (Jython 2.5.2 on java1.7.0_65)" generated="20150305
    16:37:12.486">
- <suite source="C:\Users\amanik\SynergyWorkspace\SE_QA_SMOKE_TEST_INTEGRATED\Test
    Scripts\QA_SIM_14.1_SmokeTest_Demo_Old.html" id="s2" name="QA SIM 14.1 SmokeTest
    Demo_Old">
+ <test id="s2-t1" name="001_SIM_Help">
+ <test id="s2-t2" name="003_Validate foundation data">
+ <test id="s2-t3" name="011_DSD Receiving against a new PO-SIM PC">
+ <test id="s2-t4" name="013_RTV -SIM PC">
+ <test id="s2-t5" name="016_Store to store transfer-SIM PC">
+ <test id="s2-t6" name="019_Transfer Request-SIM PC">
+ <test id="s2-t7" name="022_Inventory Adjustment-SIM">
+ <test id="s2-t8" name="032_Lookups-SIM">
+ <test id="s2-t9" name="035_JAVA Version used for Deploy">
+ <test id="s2-t10" name="038_JAVA Version in the Client (Laptop)">
  <doc />
  <metadata />
  <status status="PASS" endtime="20150305 16:18:22.275" starttime="20150305 16:37:12.486" />
    </suite>
- <statistics>
- <total>
  <stat pass="10" fail="0">Critical Tests</stat>
  <stat pass="10" fail="0">All Tests</stat>
    </total>
  <tag />
- <suite>
  <stat pass="10" id="s1" fail="0" name="QA SIM 14.1 SmokeTest Demo_Old">QA SIM 14.1
    SmokeTest Demo_Old</stat>
    </suite>
    </statistics>
  <errors />
    </robot>
```

Figure 6.2: SYNERGY Result File for Existing Scheme



```
    <?xml version="1.0" encoding="UTF-8" ?>
- <robot generator="Robot 2.8.1 (Jython 2.5.2 on java1.7.0_65)" generated="20150305
    15:25:32.659">
- <suite source="C:\Users\amanik\SynergyWorkspace\SE_QA_SMOKE_TEST_INTEGRATED\Test
    Scripts\QA_SIM_14.1_SmokeTest_Demo.html" id="s1" name="QA SIM 14.1 SmokeTest
    Demo">
+ <test id="s1-t1" name="001_SIM_Help">
+ <test id="s1-t2" name="003_Validate foundation data">
+ <test id="s1-t3" name="011_DSD Receiving against a new PO-SIM PC">
+ <test id="s1-t4" name="013_RTV -SIM PC">
+ <test id="s1-t5" name="016_Store to store transfer-SIM PC">
+ <test id="s1-t6" name="019_Transfer Request-SIM PC">
+ <test id="s1-t7" name="022_Inventory Adjustment-SIM">
+ <test id="s1-t8" name="032_Lookups-SIM">
+ <test id="s1-t9" name="035_JAVA Version used for Deploy">
+ <test id="s1-t10" name="038_JAVA Version in the Client (Laptop)">
+ <kw type="teardown" name="SIM_CommonLib.Close SIM">
  <doc />
  <metadata />
  <status status="PASS" endtime="20150305 15:38:46.495" starttime="20150305 15:25:32.679" />
    </suite>
- <statistics>
- <total>
  <stat pass="10" fail="0">Critical Tests</stat>
  <stat pass="10" fail="0">All Tests</stat>
    </total>
  <tag />
- <suite>
  <stat pass="10" id="s1" fail="0" name="QA SIM 14.1 SmokeTest Demo">QA SIM 14.1 SmokeTest
    Demo</stat>
    </suite>
    </statistics>
  <errors />
    </robot>
```

Figure 6.3: SYNERGY Result File for Proposed Scheme



	TIME ELAPSED (min:sec)		
Nos. Of TestCases	QTP + FOUNDATION DATA SCRIPT	SYNERGY(with existing Functionality) + New Foundation Data Script	SYNERGY(With Proposed Functionality) + New Foundation Data Script
10	00:38:31	00:19:10	00:13:14
50	1:29:11	1:07:22	00:54:11
100	2:00:09	1:53:02	1:40:12
150	2:55:33	2:42:00	2:30:01
200	3:49:59	3:29:23	3:20:59

Figure 6.4: ResultComparison Table

Chapter 7

Conclusion

7.1 Conclusion

The main objective of this project is to test the whole Store Inventory Management automatically. The need is to execute the existing test scripts written using VBScripts for automating the regression testing of the OR-SIM product using the automation tool called HP-QTP. As HP-QTP have some issues, to overcome those issues, all the test cases need to be migrated from HP-QTP to Oracle's newer software SYNERGY. Automating the test suite of the whole SIM product using SYNERGY gives a better Automated system in terms of time taken for execution.

For optimization purpose, new Scheme has been proposed for writing optimum Function libraries and Foundation Data Creation Scripts, in terms of space and time which improvises the automation speed.

The execution result comparison between existing scheme and proposed scheme shows that after addition of new functionality, the time taken for automation testing of AUT is quite less hence, achieves better performance in terms of time taken.

References

[1] ”A complete guide to working on Quality Center with QTP”. Learn QTP. Retrieved 2011-01-27.

[2] A. Bertolino, ”Software Testing Research: Achievements, Challenges, Dreams”, In Future of Software Engineering(FOSE’07), 2007

[3] Solution brief: Oracle Retail Stores Solutions

[4] HEWLETT-PACKARD QuickTest Professional Validated Integration with Oracle E-Business Suite 12.1

[5] A. Pillai, ”Basics of QTP Framework”

[6] Kris Lange, ”Oracle Retail Store Inventory Management”

[7] HP-Quick Test Proffessional Tutorialspoint