Automation and Framework Design & Development

Submitted By

Shreyas M Modi

13MCEC10

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

April 2015

Automation and Framework Design & Development

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Shreyas M Modi

(13MCEC10)

Guided By

Prof Dhaval Jha

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2014

Certificate

This is to certify that the major project entitled **"Automation and Framework Design & Development"** submitted by **Shreyas M Modi (Roll No: 13MCEC10)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him 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 Dhaval S Jha
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
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, **Shreyas M Modi**, Roll. No. **13MCEC10**, give undertaking that the Major Project entitled "**Automation and Framework Design & Development**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** 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:

Place:

Endorsed by

Prof Dhaval S Jha

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof Dhaval S Jha**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His 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.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Shreyas M Modi

13MCEC10

Abstract

With increasing demand of advanced technology lots of new application, products and services are created. But after creation of these applications each needs to be checked manually to make sure that application is working well before delivering to the customer. Manual checking of applications also involves some other complications like human errors and time. For solving such problems automation is required to automate a task by developing codes and scripts which performs same task by system without Human intervention.

Framework is base for any development process. With the help of proper framework design and development re-usability can be achieved with minimal cost of bugs. Also Framework helps in making development process faster and smooth because it makes each component of development clearly differentiated.

Abbreviations

ORMS	Oracle Retail Merchandising System
OPOS	Oracle Point of Sell
OATS	Oracle Applications Test Suit
HP QTP	HP Quick Test Professional
HP QC	HP Quality Control
OTM	Oracle Test Manager

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General

As working for Oracle Retail, opportunities of working on large systems are very high but application development takes lots of time. After development of product and before delivering to the customer it is really very necessary to check each of the functionality is working as expected or not. For this purpose proper checking of each functionality is required to be done. But if there are thousands of products and each with different versions containing different hundreds of functionality then process takes even more time then development. Also these complete process needs to be properly organized otherwise there are chances of human errors.

When lots of work or same repetitive work is given to human then human can get tired and there is always chance of human error which leads complete system to failure. Also task done by human are very less to be organized and well structured.

1.2 Testing

As testing plays very important part in SDLC (Software Development Life Cycle) it is necessarily be performed with lots of accuracy and perfection. **Unit Testing** is referred as checking individual components or functions to be checked. **Integration Testing** is referred as testing performed after combining individual functions or components. **Smoke Testing** is referred as checking of main functionality of system.

1.3 Automation

Automation is basically performing a task by computing system without any human intervention. So automation reduces manual work of human.

1.4 Framework

Base architecture of system is known as Framework. After construction of proper framework human requires less efforts for setting up system properly.

1.5 Scope of Work

Scope of this paper is limited to automate testing process of Oracle Retail products and also develop proper framework by making proper design structure. Also solving current issues in Oracle retail related to automation and framework. And then compare different types of framework including newly created. Then decide which framework is more suitable for which kind of scenario.

Chapter 2

Automation

2.1 General

Performing any set of task by computing system without human intervention is known as automation. Specifically when testing process is automated then referred as Testing Automation.

2.2 Need of Automation

Repetitive tasks are very cumbersome in nature and takes lots of human time and efforts when performed manually.

Now if we consider any task with some small changes to be performed hundred or thousand times, problems can be

- Human will get tiered
- Physically also not possible because for each executions human needs to interact with system
- There is always chance of error where human is involved
- Switching from one task to another will take extra time
- Speed will be slow
- Less Performance because of less re usability etc. . .

Now consider a system which can do all this work with single click without human intervention, and this is the reason why we need automation.

2.3 When to Automate?

Performing test automation for following projects will lead to more productivity.

- For large size and complex projects
- Large projects which require testing frequently the same area
- Projects whose requirements do not change frequently
- Stable Software with respect to manual testing

2.4 How to Automate?

Following are the basic steps for automation.

- Identify parts which can be automated.
- Select proper automation tool for test automation.
- Generate test scripts in specific language which is supported by selected automation tool.
- Execute the developed test scripts.
- Generate result report for executed test scripts.
- Find out any chances of bug or performance issue from the result report.

2.5 Evaluation Process

When any process is automated then initially we need to make system train. For this purpose we need to train system to identify objects in the system which are known as **Trained Objects**.

Later on these objects can work on similar kind of systems to perform automated tasks. After task gets automated, it is necessary to check initially that whether the generated output is correct or not. This can done by comparing output of generated version with the golden version which is predefined by team working on system manually[1].

Also every testscript will generate a result file which can be in html or xml formate and this file will contain details about test script execution. If driver script is being invoked

and it calls other test scripts then there will be only single result which will involve result about all test cases called by driver script.

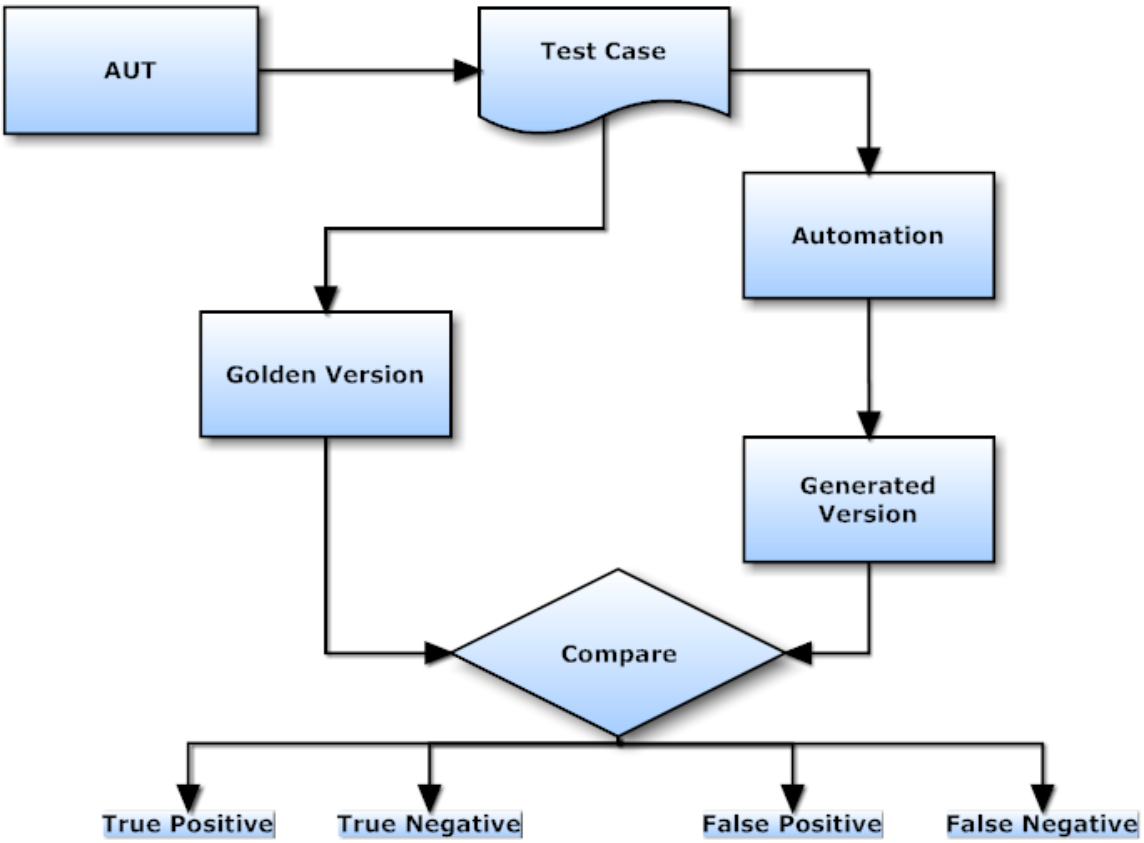

Figure 2.1: Evaluation Process

Chapter 3

Framework

3.1 General

Starting with non technical example, if we are constructing a building then its structure is the most important part upon which complete building is going to be built. And this base structure is called Framework.

3.2 Need of Framework

Technically Framework is set of predefined rules and structure which wraps over the application for better and stable development of system.[2]

- Makes everything structured so easier to work with complex technology
- Ties together a bunch of components to work together
- So implementation code need to follow some specific set of rules which promotes consistent coding, fewer bugs and flexible application
- Everyone can know structure so easy to update anything etc...

3.3 Types of Framework

3.3.1 Data Driven Testing Framework

Data driven framework is where the input and expected output values or results are stored in a separate data file (generally in a tabular format) so that parent driver script can execute all the test cases with different variations of data.

Data can be about input to be provided and after computation match actual output to the expected output so expected output can also be provided.

3.3.2 Modular Testing Framework

Modular testing framework implements the concept of abstraction which involves creation of individual modules of application under test.

These generated modules are reused to form larger test cases module. So it generates abstraction layer for a component to hide that component from other applications. Example of such framework can be to use same function like **Login** in multiple test cases. So each test case uses the same function which reduces the effort of automation engineer. Also we can have different parameters passed to functions based on requirements.

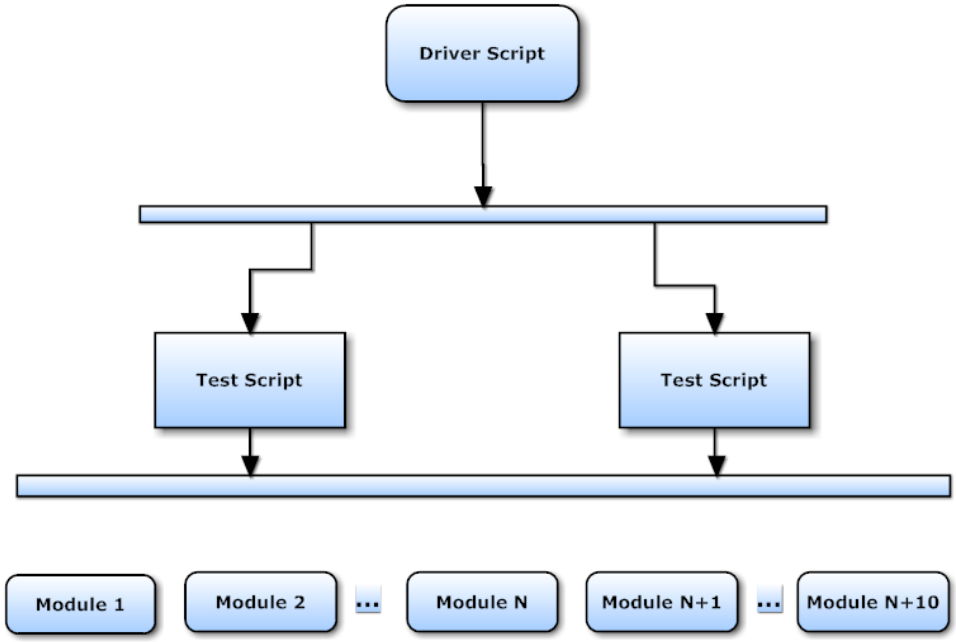

Figure 3.1: Modular Framework

3.3.3 Keyword Driven Testing Framework

Keyword driven framework is an application independent framework which utilizes the data tables and self describable keywords to explain the actions with navigation to be performed on the application under test. So not only data about test is kept in the file which is tabular but also the navigational steps telling what to do in application under testing is in the test scripts is put in external input data file. These navigation are called **keywords**. Keyword driven testing is extension of data driven testing.

Figure 3.2: Keyword-Driven Framework

3.3.4 Hybrid Testing Framework

Hybrid testing framework is the combination of modular, data-driven and keyword driven testing frameworks. But it increases more complexity.

Chapter 4

Case Study of Automation: Server Restart

4.1 General

Oracle has almost all of its products to be uploaded on several servers. Clients gets access to oracle products via these servers. Also internal development work is also carried out in different servers. Due to some maintainability oracle needs to close down server and start up again. These maintenance work may be due to Hardware and Software changes.

4.1.1 Problem after Server restart

Every time when server needs to be shut down because of some maintenance work, each link of oracle products and each configuration links of server needs to be checked manually to check

- Whether product or service is ONLINE or OFFLINE
- Whether credentials are working properly or not
- Whether product or service is launching or not
- Report needs to be generated for each product or service

Time taken: 2 Days approximately with lots of Employees manually checking each product and service availability.

There are thousands of links needs to be checked manually after server restart so chances of making is very high by human.

4.1.2 Objective

Automate process of manual checking by developing proper **Framework design & development**. Each product links or service links needs to be taken one by one and checked accordingly. Also needs to use multiple machines for faster work to be done. At the end of process result needs to be generated which contains either check has passed or failed and if it is failed then where it is failed.

4.2 Approach

There are products which require specific java versions and also some specific system requirements, so complete process is divided among several systems with same methodology and framework also for achieving task to be done faster.

For example ORMS (Oracle Retail Merchandising System) requires java version 7.45(32-bit) and OWMS (Oracle Ware House Management System) requires java version 7.65(64-bit), so both the products can not be checked on same machine. But OSIM (Oracle Store Inventory Management) and OWMS has same requirement, so can be checked on same machine. So based on compatibility of products working environment, machines are separated accordingly.

4.2.1 Framework Design

Framework designed for automation for checking of products after restart of Server contains following basic components

- **Input Sheet file:** Sheet file contains each product or service link with user name and password which gets loaded dynamically by script.
- **Scripts:** Scripts are files of codes created in HP QTP and OATS, which integrates all the components and work for specific goal.
- **Library:** Library is set of functions which are used to perform automation of task.
- **Object Repository** Object Repository is collection of objects which are captured for automation purpose for further identification of product elements. Object repository is collection of **Trained Data**.

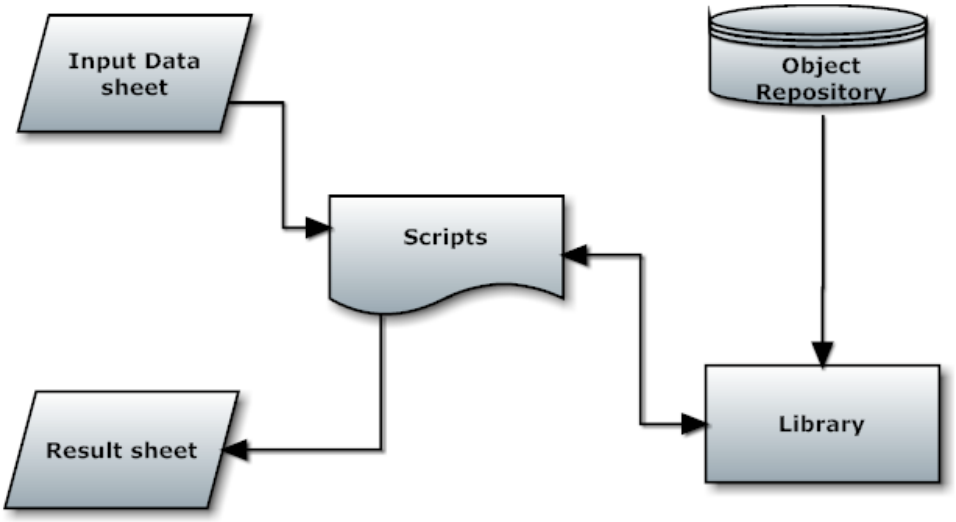

Figure 4.1: Server Restart Framework

- **Result Sheet** Result sheet contains 3 types of output for each product or service based on some criteria.

Result Sheet contains following schema

Table 4.1: Schema of Result Data Sheet

Field Name	**Type**
Product Name	Text
URL	Text
User name	Text
Password	Text
Result	Pass/Fail
Location	Text

Location field can have different text based on passing or failing of valid on particular page.

- Result 0 means script is not able to open log on page it self
- Result 1 means log on page is coming but either Home screen is not coming or user id and password is wrong
- Result 2 means successfully logged into product and Home screen has loaded successfully

In case any product or service is not available after restart of server then failing scenarios are also handled so that script do not stop in between and continue execution by reporting particular product as a failure in result sheet.

Chapter 5

Understanding Existing framework and Technology

5.1 General

Before creating any new thing we need to understand what is the potential of existing framework and design. Also need for creating new framework must be clear by identifying problems with the existing framework.

5.2 Framework Design

Product under consideration for this paper is mainly Allocation. **Allocation** is one of the oracle's latest product which provides facilities of prediction and inventory allocation between stores and warehouse. Previous Allocation called Allocation 13.2.8 was created with web technology. Which involves basic technology tool set and web kind of application for client.

5.2.1 Allocation

Allocation is one of the latest product of Oracle which mainly focuses on Predication and allocation of inventory. Allocation has 3 main modules which are as following.

- Standard Allocation
- What if Allocation
- Scheduled Allocation

5.2.2 Foundatioin Data

Every application requires some basic data available to use, this data is known as **Foundation Data**. For allocation foundation data is generated by application known as ORMS (Oracle Retail Merchandising System).

5.2.3 Script Upgrade

For this application version Smoke test cases were implemented without any kind of framework, which is also called as No-Framework architecture with simple record and playback.

This design was improved by us as a data driven framework. As it was becoming to much difficult to change any data in test script we have re-framed architecture to be data driven and kept input and output files separate.

We have found some scripting issues and need to fix those and also found miss behaving objects which were also fixed. After performing all this if result doesn't come as expected then we need to report a bug with proper detailed information to the QA team.

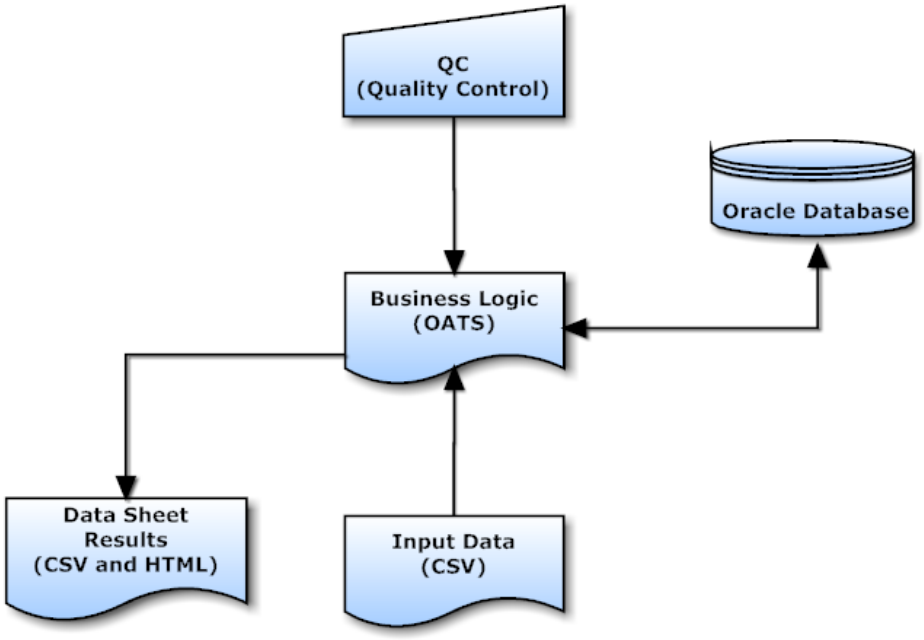

Figure 5.1: Data Driven Framework (Improved Allocation 13.2.8)

5.2.4 Problem

Main problem with this scripts was its framework as it has only data driven framework.

- Scripts are very complex and not much developer friendly

- Single chance required to be done in each and every script
- No separate library which can provide common platform to perform same type of task
- Object Paths are embedded into script
- Single change in object path requires to make change in every script
- Database connection requires same code being repeated for every script
- Change in Query needs to be done in each script
- Understanding of test case becomes very less as every thing at single place makes it much more complex for understanding

Chapter 6

Framework Design and Development

Oracle retail has developed a major product called **Oracle Allocation**[3]. Allocation is helpful for making decision for items to be where at which time. Also how much quantity of items should be where.

6.1 Motivation

First and most important question is why we need framework. And simplest answer to this question is for robust development of script we need proper base which is known as framework. If we have no framework then it looks like script generation is taking very less time but if we consider large number of test cases then we need to record every object from scratch. And if there is any change in application then we need to make change in every script which is very time consuming process. Also in scripts we need to find each and every individual objects and corresponding line of codes.

6.2 Objective

We are suppose to design proper framework for allocation application testing and also develop the same. we are also required to automate testing process with the help of scripting. Design of framework should be very optimistic and flexible with change in the system. Single change at one place should reflect it everywhere in system. Also each component need to be separate but work in relation with each other. Each element of framework should be highly robust and should follow concepts of OOPs.

6.3 Foundation data generation

Oracle Allocation requires foundation data which is generated by RMS (Retail Merchandising System). RMS has its own framework for generating foundation data[4] for allocation

Basically there 2 types of data or tables:

- **Controller Data:** Data generated for automation purpose and testing
- **Application Data:** Data generated by actual application

So we have done generation of foundation data with the help of automation scripts. RMS[4] contains 4 Phases. Each phase has several batches or machines. Each machine contains various number of scripts for generation of specific data.

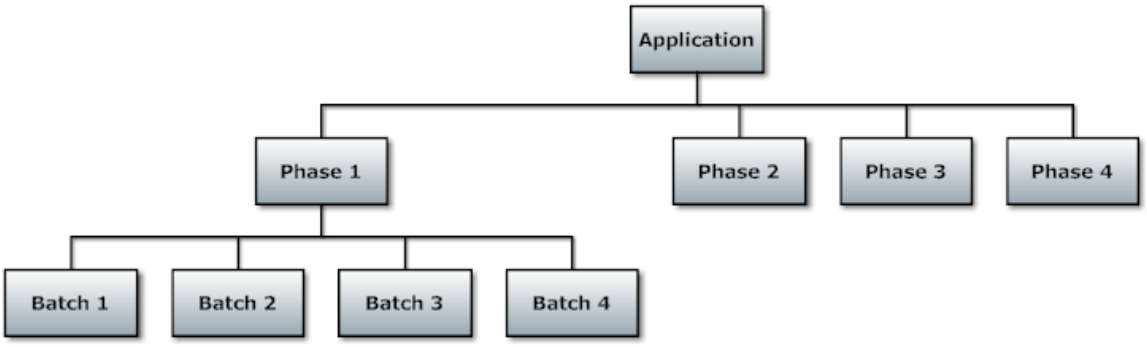

Figure 6.1: Application Phases Architecture

Main purpose for making such partition is making parallel processing within a phase because each batch will have multiple scripts running under it. So making such parallel design is helping in making foundation data generated faster with multiple systems at a same time.

This phase scripts will take data from controller tables and then from the automation scripts it will generate appropriate data. Foundation data for RMS will be generated from sql files. There are 3 SQL files generated for foundation data for RMS.

- Drop all controller tables
- Create new controller tables
- Insert data into controller tables

Scripts will access these tables and will generate foundation data for allocation. And these generated data will be used by allocation irrespective of environment for further test case processing. Internal design of each phase is as follows.

Phase	Machine 1	Machine 2	Machine 3	Machine 4
Phase 1	SOB	Company	Customer	Banner
	Org Unit ID	Chain	Buyer	Channels
	Transfer Entities	Area	Merchandiser	Transfer Zone
	GL Cross Reference	Region	Division	Vat region
		District	Group	Vat Code
		Cost Zone		Vat Rates
Phase 2	Dept	Warehouse	UDA	Partners
	Class	Warehouse Add Batch	Diff Type	Suppliers
	Sub Class	Stores	Diff ID	Supplier Sites
		Store Add Batch	Diff Group	
		Price Zone		
Phase 3	Item Creation 1	Item Creation 2	Item Creation 3	Item Creation 4
Phase 4	Simple Packs Creation 1	Simple Packs Creation 2	Simple Packs Creation 3	Simple Packs Creation 4
	Complex Packs Creation 1	Complex Packs Creation 2	Complex Packs Creation 3	Complex Packs Creation 4
	Item List 1	Item List 1	Item List 1	Item List 1

Figure 6.2: Internal design of Phases

6.4 Framework Components

For better development we need better design and architecture. Framework defines how components can interact with each other. We have decided to go with hybrid framework do you many reasons of betterment over existing framework but biggest reason to take up Hybrid framework is its robustness and single point change will reflect everywhere. Another advantage of this framework is its supporting tool which is Oracle Application Test Suit(OATS). Reason for moving to this tool is, it is fast compare to HP QTP(Quick Test Professional) and it is best suited for Hybrid framework due to design of application tool. List of Modules and sections are generated in the tool which makes easy to find any part of test case in the tool. As show in figure there are multiple components involved in complete framework. **Resources** involves important components. Environment variables and constants defines which environment to access and what is port number etc. **Configuration Files** involves credentials for access and also details about database connectivity.

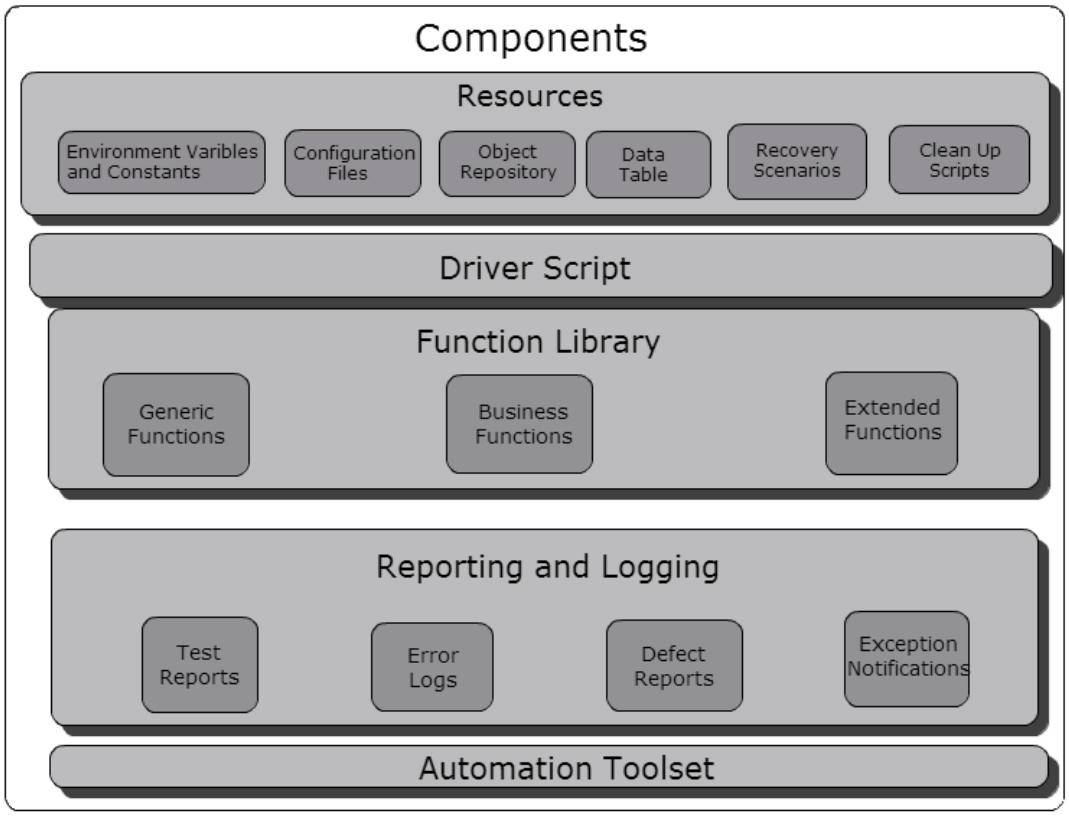

Figure 6.3: Framework Components

Object repository maintains trained objects with specific attributes for further automation.

Data table maintains actions and navigation to be performed and expected data.

Recovery Scenarios are the scripts which gets executed when any pop up comes up and which is not expected, for such situations it should not stop the execution so recovery scripts get executed automatically depending on the situation.

Cleanup scripts are used to clear data for example a new client is created for testing purpose then that data needs to be deleted so that it do not make any problems to actual system with dummy data.

Driver scripts are the parent scripts which invokes the other scripts of automation.

Function Library includes different types of functions like

Generic functions which are not application specific functions,

Business functions are application specific functions and

Extended functions are extensions to the business functions.

Reporting and Logging are also different types of scripts which are used for generation

of reports and log data. These reports are required to be maintained for verification with Quality Analyst. Also those are useful for finding bug and debugging.

6.5 Framework Design

Every element of design is as important as other elements. Following diagram shows design of Proposed hybrid framework.

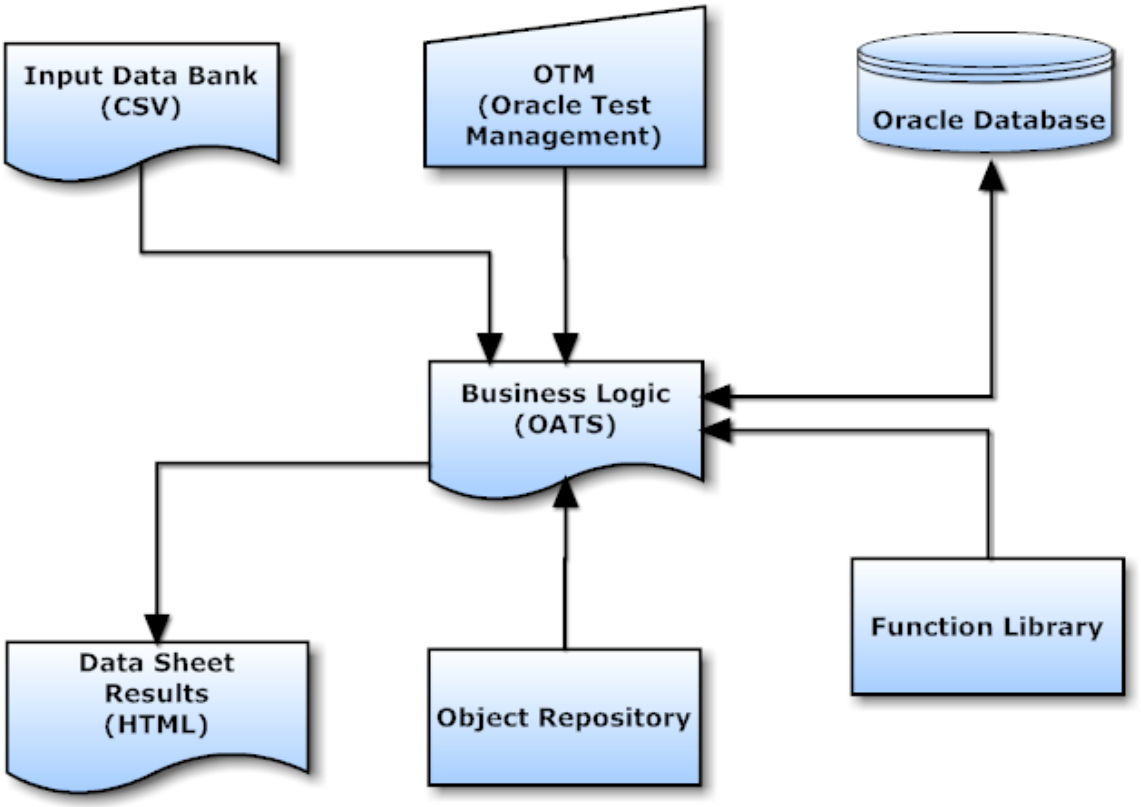

Figure 6.4: Hybrid Framework Design

OATS is a oracle tool which integrates all other components used for development of scripts. Result will be generated in the form of HTML graphical view or in XML form. Object Repository is collection of Object Paths.

6.6 Interaction Model

Following diagram shows how components interact with each other. Each test case will only directly interact to Database and Data bank. Function Library is virtual entity which is consisting of multiple libraries. These libraries have associated Object Repository. These Object Repositories contains object paths being used by that function library. We have created different function libraries which contain different types of functionality

within that. These libraries will be associated with the test for making use of functions within those.

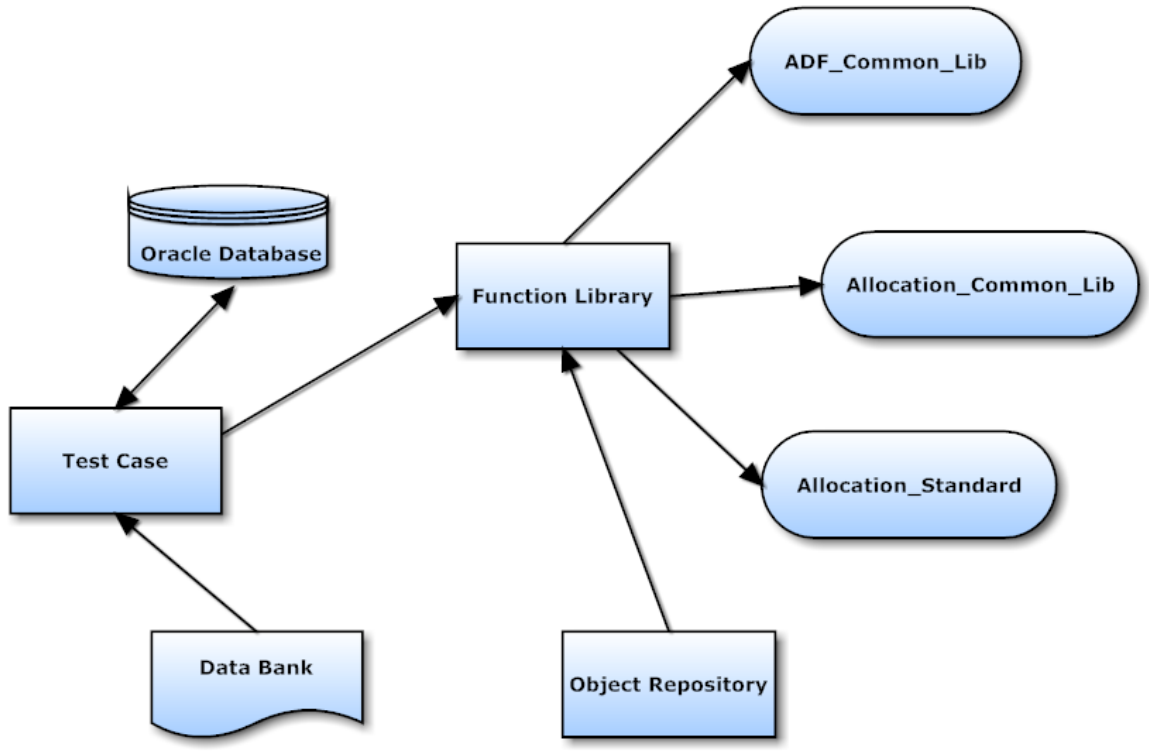

Figure 6.5: Hybrid Interaction Model

There are 4 different types of function library.

- ADF_Common_Lib
- Allocation_Common_Lib
- Allocation_Standard_Lib
- Allocation_Batch_Lib
- Common_Lib

ADF_Common_Lib contains functions which can be used by all the adf based applications as there are different technologies being used like ADF,Forms,Java and Webs. ADF_Common_Lib specifically contains functions for log in and log out.

- **ADF_Logon** function is used by all ADF applications for performing log in to application.

- **ADF_Logout** function is used by all ADF applications for performing log out from application.

Allocation_Common_Lib contains functions which provides functionality of type keyword driven as well as data driven framework. This library contains functions which works stand alone and directly interacts with application UI with the help of ADF functions which interact with application using Object Paths. Following are few explanations

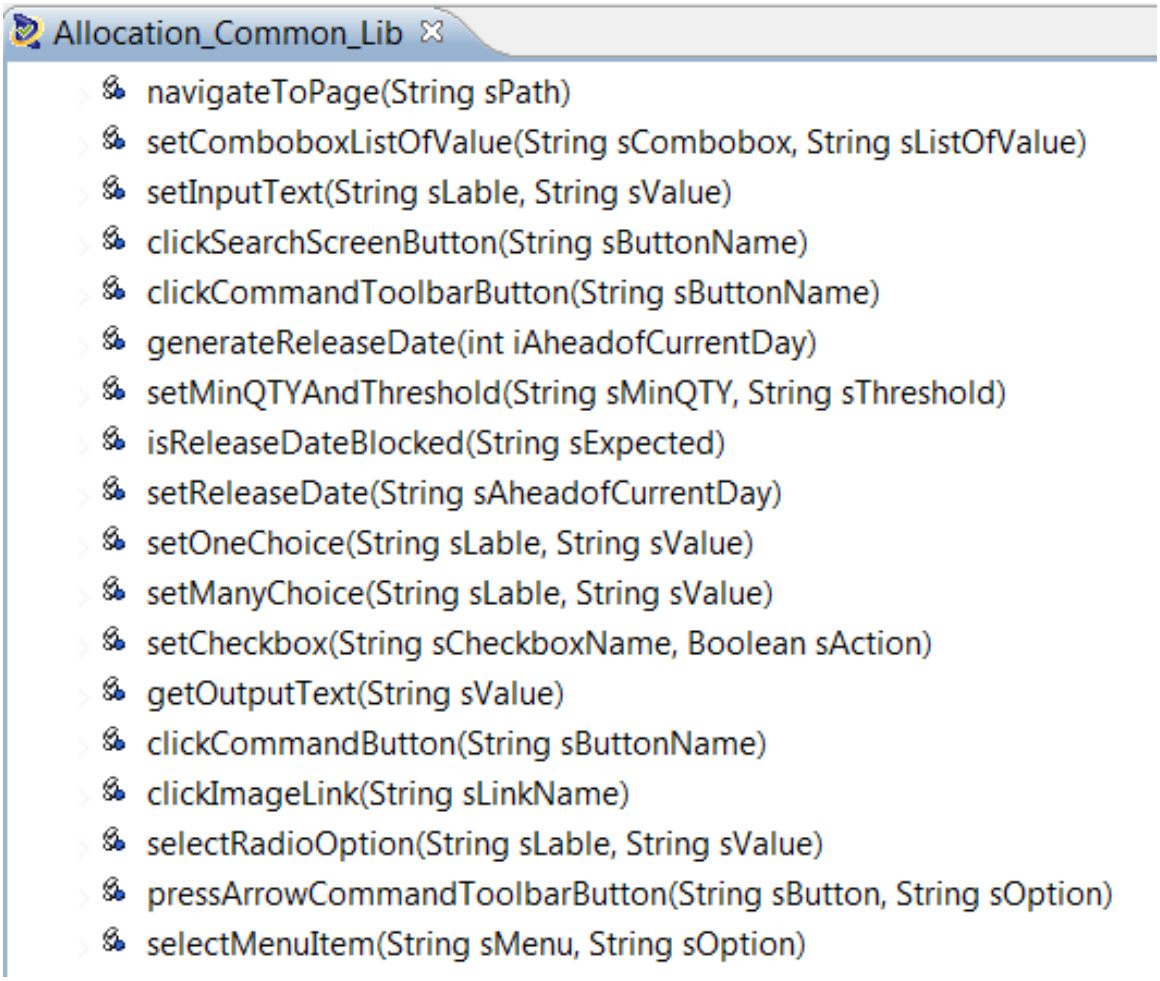

Figure 6.6: Allocation_Common_Lib OATS

of functions.

- **navigateToPage(sPath)** function is used to navigate from main screen to different screens of applications.
  eg navigateToPage("Tasks− > Create Standard Allocation")
  eg navigateToPage("Tasks− >Allocation Foundation− >Manage System Options")

- **setInputText(sLabel,sValue)** with the help of this function we can set any text box value by specifying label name of text box and value to be filled in to text

box. So we don't require to record any new text box object and we can directly use key-value pair and reuse this function to fill data in screen.
eg setInputText("Item","195674")
eg setInputText("Department","3419")

- **getAllocationNumber** will fetch allocation number from screen by rending screen with different tabs of region and will split currently generating allocation ID.

- **checkNotification(sAllocationNumber)** with this function we can use result retrieved by getAllocationNumber and use this number to get notification verification.

- **generateReleaseDate(iAheadofCurrentDay)** will generate a release date for allocation transfer from warehouse to store.

Allocation_Standard_Lib contains functions which can perform set of tasks. This lib uses functions from Allocation_Common_Lib and Common_Lib and perform actions which contains set of different actions.

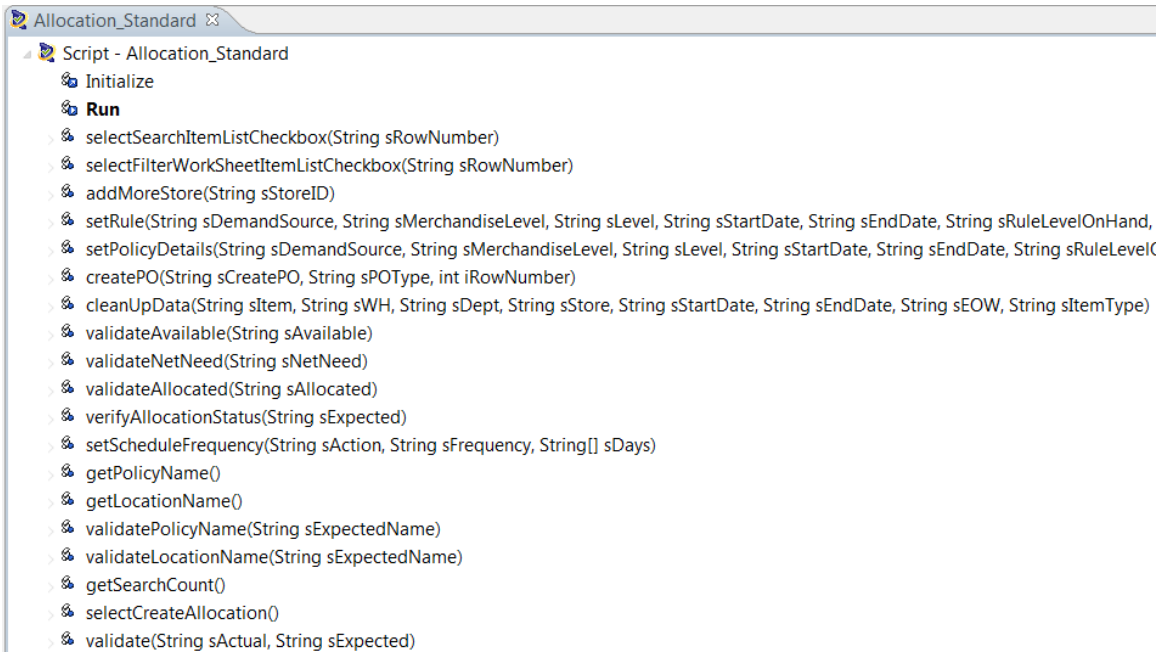

Figure 6.7: Allocation_Standard_Lib OATS

- **addMoreStore(sStoreID)** funcation will add new story in allocation details. This function takes only store id as a parameter and will add this store to list of alloca-

tions store. This stores will be considered at the time of inventory movement from warehouse to stores.

- **cleanUpData** will be executed at the starting of every script. Depending on the type of item and pack being allocated it will clear data from database and also add accordingly data to database.

- **setRule** is used for specifying level of allocation, qty to be considered, hierarchy of distribution etc. Also we need to specify start date and end date of historical data to be considered for prediction of allocation. We can also specify whether to consider stock on hand or not and stock in transit or not.

- **getPolicyName** is used to generate new rule name which can be used to set in creating new policy rule. This function will fetch name from database and then generates new name which does not exist in database.

- **setScheduleFrequency** is useful for set scheduled allocation in which we can specify days for transfer of items and date range.

- **validateAvailable(sAvailable)** will fetch available qty from screen and then matches it will expected qty of available. Even if this does not match it will continue process and goes for next validation also it will notify user about qty values of expected and actual.

Allocation_Batch_Lib library is used for executing shell batches on server machine. This library contains functions which can send commands to be executed on server machine of RMS. As database of Allocation flows from RMS, we need to send shell command to RMS Application server. Functions will take commands and execute it there also comes back with generated output.

- **fetchEnvData** is used for fetching data from assets attached to batch script. It will fetch hostname, putty user name and password,profile,batch directory etc.

- **connectSSHCommand** uses details fetched by fetchEnvData and uses this details for connection and reaching upto specified directory for execution of batch.

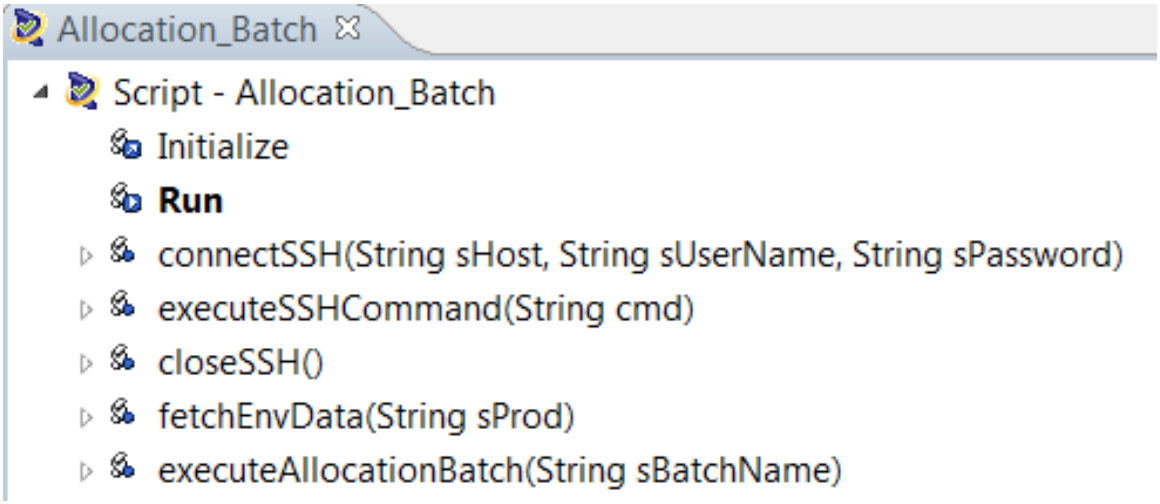

Figure 6.8: Allocation_Batch_Lib OATS

- **executeSSHCommand(cmd)** first uses fetchEnvData and then makes connection using connectSSHCommand then commands are passed to this function and it will execute those commands. We can also pass parameters with space in between.

Chapter 7

Result and Comparison

7.1 Result Generation

Result generated by OATS is it self well descriptive. Every step being performed by script will be recorded by OATS with comments if provided within info. Also every step specifies either it has passed or failed. If it has failed then there is reason attached with steps for failure. Information values added to scripts are visible in result. We can also expand each and every steps to sub-steps for further clarification.

Result can be in HTML view or XML view. Whenever there is databank access it will specify which items from databank is being used.

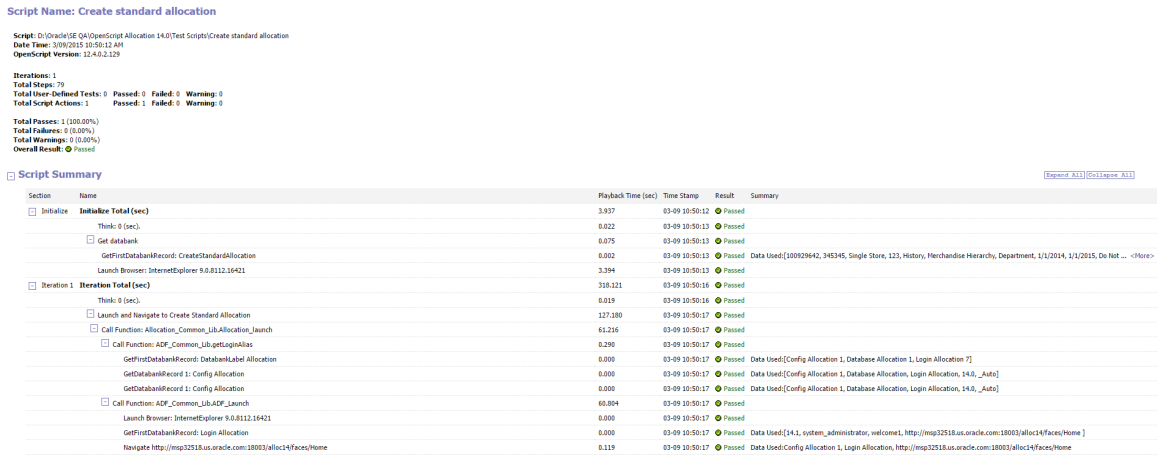

Figure 7.1: Result in HTML view

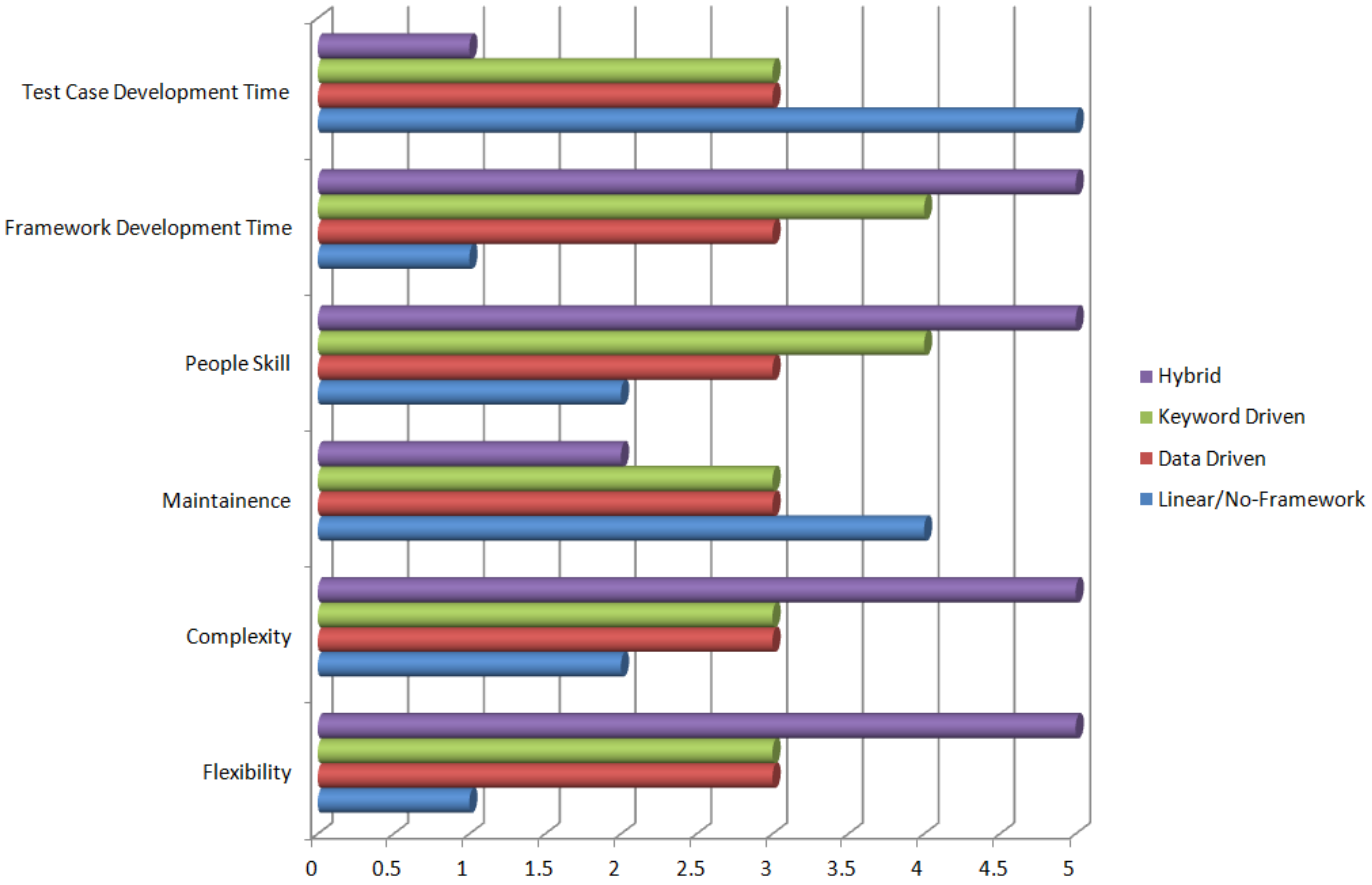

Figure 7.2: Comparison of Frameworks

7.2 Comparison

After observing and developing different types of framework we have come up with comparison of these frameworks. This comparison is mainly based on further development of test cases by using framework. Also it considers relative time complexity for development of framework and further test cases.

As shown in comparison, different frameworks are compared with each other with respect to specific elements. If we consider hybrid framework then its development is boundless and it totally depends on developer for how to build it.

We have also taken up graphical comparison of different frameworks by conducting internal survey within out team of automation and framework development. Here we have taken scale of 0 to 5 which shows higher number means higher value of that particular entity. For positive kind of elements we have taken hybrid framework at scale of 5 and made comparison with other frameworks. As comparison can always be in relative terms and can have relative aspects only.

Element	Linear	Data driven	Keyword	Hybrid
Approach	Simply record and Playback	Record but keep input data in databank	Pair of key and value like which action to perform with which value	Combination of multiple frameworks to make work together but imagination of architechture is required
Skills requirement	Only manual knowledge of application and basic skills required eg. How to use system	Moderate level of knowledge is required	Requires good level of skills to form a key-value pair	Requires high level of programming skills and analytical skills for finding patterns in recorded data and generalization
Development time for framework	Proper knowledge with manual things this can directly work on test cases	Requires time to make data separate and integrate with application	More time required compare to data driven framework	Significantly large amount of time is required for Identification of components and design Also finding patterns in object takes time
Complexity	Low	Medium	Medium (more than data driven)	High
Test case development time	High because every test case is started from scratch	High	Medium	Low, once library and base scripts are ready
Flexible with change	Not At all	Medium	Medium	High
Advantage	Simplest and easy	Moderate complexity and separate data	Moderate complexity and key value pair	High flexibility and depends on creator (boundless)
Disadvantage	Not flexible at all	Less flexibility	Flexible only in terms of order	Very complex design and requires high programming skills and more time to develop

Table 7.1: Comparison of Frameworks

Chapter 8

Conclusion

Specifically about Automation, it has its own advantages and disadvantages. Automation requires more time of development and more man power and skills compare to manual work. But once it is done properly, efforts become very line below the man power and efforts required by manual persons. Also there is not scope of human errors in Automation. Develop automation scenarios for one environment and we can use it on multiple other environments. Scenarios like where we have already published product to client then client reports some bugs, now for checking this bugs are real or false warning by customers, we can use automation to perform this check. After execution if there is failure we can go for analysis and find out what exactly is causing problem to system. But for idea case automation should only be carried out for long term productivity of system and maintenance check.

After designing and developing different types of framework for Automation, we found that these frameworks are suitable in different scenarios.
Linear/No-framework is useful when we know there will be no change in system or application for a long period of time or we know that particular automation is required very fast and for very short amount of time and after that it is of no use then go for Linear/No-Framework.
Data-driven Framework is most situated in cases where we have more number of input and output data and which might required to get changed from environment to environment. Keeping data separate will also provide proper readability.
Modular Framework is used when specific set of actions are required to be performed

repeatedly within other test cases. Modular framework may not have data separate. Keyword Driven Framework is used when we can have pair of key and value. Like a system we provide a key it will give us a value. And with different key it will act accordingly. Hybrid Framework is most suitable for very large products and with long time support. It will take highest amount of time and man power for development but results are worth of its price. Maintainability is advantage as it makes every component separate and manageable properly. Also test case development becomes very fast as everything is already ready and we need to directly connect those dots and use framework design as per requirement. And development of hybrid framework is bound less as it is totally upto developer about how to design and build it because there can be multiple possible Hybrid frameworks.

So each of the framework design is suitable depending on the product size, man power, time of development available and skills.

References

[1] S. R. Shahamiri, W. M. N. Wan-Kadir, S. Ibrahim, and S. MohdHashim, "An automated framework for software test oracle," *Information and Software Technology*, Mar. 2011.

[2] M. S. Azad M. Madni, Barry Boehm, "n-tiered test automation architecture for agile software systems," *Procedia Computer Science 28*, Mar. 2014.

[3] O. India, "Oracle retail allocation, the right product, in the right place, at the right time." Online; Last accessed 12-May-2015.

[4] O. India, "Oracle retail merchandising system, manages the fundamental processes of a retail business." Online; Last accessed 12-May-2015.