Hybrid Test Automation Framework for managing Test Data

Kamini (Simi) Bajaj
School of Computing Engineering and Mathematics
Western Sydney University
Australia
k.bajaj@westernsydney.edu.au

Abstract

Test Data is an essential element for both, Manual and Automation Software Testing and more so in Automation Testing where data drives the test execution. Test Data Management (TDM) is an important ingredient for testing however test automation has been proven to successfully test the data intensive applications. Hence, we would primarily focus TDM for Test Automation in this research. The purpose of TDM is to improve the effectiveness and reduce the time and cost of testing. However, it is essential to identify the components of TDM that contribute towards fulfilling the purpose. Test Data Management involves systematic approach in creating and managing test data (test scenarios, test cases and test data being used). In order to succeed, it requires strategy and a tool to enable teams to create and manage test data in effective and efficient ways. Many hybrid frameworks and tools have been deployed that permit data driven scripts to carry out keyword driven testing. However, it is identified that most of them have lacked correctness and coverage of components of TDM. In this paper, we introduce test automation framework called TAFPro that works as a test harness for IBM test automation tools and offers increase in efficiency and reduction in application and life cycle costs. Since organizations are moving towards open source tools we aim to make the TAFPro tool independent for Java based test tools and ran a CASE study with Selenium successfully for TDM and achieved the goal.

Keywords: Hybrid Automation Framework; Test Data Management, Hybrid framework for Selenium, Test Automation

1. Introduction

Test Automation is conducted with the help of software tools that control the execution of tests for assessing the behavior of an application under test (AUT) by comparing actual outcomes with expected outcomes. It forms an integral part of
testing life cycle for data intensive applications or applications that require repeated testing of scenarios via test cases.

The input data required to execute the test cases is called test data. This data may be entered manually by the tester or read by the AUT at the time of executing the test cases from some predefined data files. Test data may be available in various forms such as a data file that may be loaded into the application or an external data source as tables. It may be in different formats such as Excel sheets, XML, system collected data, or as a database. Preparing appropriate test data is an important part of the test setup. Lack of systematic approach while building test data during test design and execution may lead to missing some important conditions in test cases.

The process of organizing correct combinations of data in the right format and all the required combinations as suited to the AUT is called test data management (TDM). TDM plays a significant role in the software test life cycle. The amount of data that is required to be generated for testing applications is varied and depends upon the nature, size, and complexity of the application. The efficiency of the application is determined by the amount of time it spends in processing the data and generating reports. It is important to test various combinations of test data checking all the possible conditions that the application may face during production. Hence, managing various combinations of test data is an inherent part of test automation. In order to maximize their return on investment in automation, organizations setup special teams for provisioning test data.

Today businesses are embracing Agile application development and delivery, and as a result there is an increasing need for continuous testing. This shift requires rapid access to the appropriate test data that must be available to test continuously. Managing such test data amplifies the quality of test efforts to perform the two types of test automation. Functional testing, wherein order to test all data intensive features, TDM enables not only testers but business users to spend more time on testing and less on creating data at run time. This is achieved by extracting test data that is available to act as input for data driven testing. Second one being regression testing. In the changing development environment with new features, enhancements, defect fixing; regression testing enables teams and business users to uncover new defects, in the functional and non-functional areas of the application. Hence, it is important that the test data be readily available for automated regression suites driving the need for well-organized test data management.

The purpose of this paper is to primarily outline the inference of research on various test data management tools and frameworks.

Test Data management is best achieved with the help of test automation frameworks. We conducted a survey of test automation frameworks including hybrid tools available in industry and academic research arena. There are several challenges in adopting such framework for test data management. This paper discusses those challenges and in response, recommendations on how to leverage better management of data is proposed. A gap has been identified in the existence of such frameworks for test data management. A custom framework is developed for each test automation project which requires huge amount of efforts and lacks reuse. This research aims to
We introduce test automation framework called TAFPro that leverages the TDM capabilities. TAFPro is a hybrid test automation framework that focuses on reducing the cost and time of test execution with flexible options for TDM and filtering with IBM Rational Functional Tester test automation tool. We further aim to enhance the capability of TAFPro and this project conducted research, analysis and implementation of TAFPro to integrate with Selenium, an open source test automation tool for effective test automation with greater test coverage with varying data. We chose Selenium since most organizations are moving from paid to open source tool in particular Selenium for test automation.

The paper is structured as follows: Section 2 cover the concept, requirements and challenges of Test Data Management. Section 3 highlights how TDM works in automation and how the current TDM challenges are different to the ones identified by TestPro. Various Test automation frameworks are referred to in Section 4 along with frameworks for TDM in automation and introduction to TAFPro test automation framework and how it addresses the challenges of TDM for efficiency and scalability of automation scripts with reduction in test maintenance by a user friendly non-technical interface. A case study on test hybrid automation framework to work with Selenium is discussed and project implementation and results are reported in Section 5.

2. Test data management

2.1 Requirements of test data management

To ensure high quality application delivery, there is a need to have complete, realistic, secure, correct and accurate test data as input during test execution. Generating such quality test data is a complex task, but doing so will assist in ensuring total application quality. In order to create and manage the required test data, there are two primary areas of consideration. First, test data set-up, which involves creation of baseline test data. Secondly, test data format, which involves generating data that can be used by testers while performing any types of testing such as functional, regression, performance or load testing.

Test data creation follows typical practices, however each of those practices come with their own challenges [1]. Test data is created and generated either by copying data from production or manually creating the data sets which could be master data, user generated or external data. Test data management involves entire cycle of creating of test data that offers a comprehensive coverage to validation of functions and applications using that test data.

In preparing test data for effective test data management, the following requirements play a very important role such as identification/generation/creation of test data, identification of test data element that can be used among multiple application to reduce redundancy and increase consistency, test data masking,
prioritization of test data, refreshing test data that help improve the efficiency of test process by maintaining test data within the baseline, application of business rules to test data, import and export of test data in various formats, plug-in support by the test tool, test script language support, report and metric generation, test result comparison to identify any inconsistencies and errors during test execution.

The most efficient way to compare results is by deploying an automated capability to compare the baseline test data against results from successive test runs to monitor the essential components of performance such as accuracy as well speed. Automating the comparisons helps the test team save time and identifies problems in the process of test data management.

2.2. Challenges in test data Management

Test data management is a comprehensive area that supports handling vast amount of data collection, input, storage and processing. Data and its quality is critical to successful testing of any application[2]. It also comes with many challenges that have been reported by many authors in the lines of errors in capturing, variety, security, integrity, coverage and many more [1, 3]. In this sub-section, we havelisted some of the challenges in acquiring test data suited for appropriate test data management for improve testing pace and quality while minimizing cost as shown in figure1.

![Fig. 1. Test data management challenges](image)

### 3. Test data management in automation

Though we can use vast amount of data to test applications using manual testing, it is very time consuming. For regression testing it is difficult to achieve the level of intensity required to successful run the tests. Test automation provides a consistent, repeatable, quick and cost-effective technique for software quality assurance. The major advantage offered by test automation is in the area of regression testing.
Test Automation has its own testing life cycle. Thus, involves capturing user actions outlined in test cases as scripts, running those scripts multiple times to test that functionality of the application under test. Testers need a collection of test data to test various combinations of data sets. In order to conduct test automation, variety of tools such as IBM rational functional tester, HP QTP, Selenium, TestComplete and many more are available. These tools offer a repeatable and efficient testing solution. The benefits of test automation tools are significant namely repeatability and accuracy, effective regression testing, maximum test coverage, thorough and efficient testing and breadth of testing.

To successfully run the tests using automation the quality of test data plays a very important role[4]. Test data management has been reasonably addressed in research and commercial world. It has been identified that lack of test data management would pose issues in test automation[3, 5-8]; however, the focus has been on the quality of data that scripts take as input. The challenges identified are: correctness of data, regulatory compliance, quantity, correct format, access to data sources, coverage, masking and clear data requirements. Various TDM solutions have been suggested to address these issues[6, 7, 9-11].

Apart from the issues addressed already there is another issue related to TDM. The test data must be defined and organized before the test execution can take place. Most of the test tools can connect to a data source to read the data for test coverage. One of the limitations of the existing test automation tools identified is the lack the user friendliness if the user wants to modify the data and filter out some data while conducting the tests. In order the make changes to test data, modification to the input data source must be done. Changes to input data source can only be carried out with technical expertise and hence this cannot be achieved by the end user.

4. Test automation frameworks

Test Automation Framework is set of components that form the building blocks to provide the basis of test automation. An automation testing framework provide support for automated software testing by providing set of assumptions, concepts, practices and libraries. Testers as well as developers are able to write efficient scripts with the help of framework The framework can help make the process of script writing and defect finding efficient and faster[12]. The framework simplifies the automation effort and promotes consistency.

Correctly implemented Test Automation Framework can further improve ROI by reducing the test effort and costs, support and maintenance effort and costs.

The benefits of automation framework are as follows: 1. It accelerates the of test automation tool implementation. 2. It improve the return on investment of the automated tool. 3. It simplifies test automation tool interface which make it easier for business users to participate in testing. 4. It improves the efficiency and thoroughness of test process by capturing and maintaining application data as well as the context. 5. It reduces the maintenance effort for test scripts.
4.1. Role of frameworks in test data management

Since TDM has become an integral part of testing strategy for organizations, it is vital to manage it with frameworks. Frameworks can become an effective and efficient execution harness for test automation tools that manage the test data for execution of the scripts with wider coverage and variations. Since initial test data is difficult to manage by the end users, frameworks play an important role to manage test data.

We conducted research to identify existing frameworks that can be used and implemented to provide improve, accelerate and simplify the interfacing needs of the test automation tools. A gap has been identified where not many such frameworks exist. Most of the test automation projects develop these frameworks individually for each project. Such need for developing a framework increases the effort required within the project.

Few of the frameworks used in the industry are: TestNG offers data driven testing framework that offers simplifies testing needs from unit testing to integration testing[13]. The tester must understand the annotations and XML in order to successfully use it. QuickTest Professional allows linking the scripts to tableof datarepresented as xls files that contain rows of data. One of the drawbacks is that the cost associated with QTP is very high. Watir is a driver of automating browser testing using variety of frameworks such as RSpec, Cucumber and Test/Unit[14]. TestLeft is a test automation framework for functional automation testing[15].Common automation testing framework conducts automation testing for any kind of E-commerce applications using Selenium Software Testing Tool[16]. Selenium, the most widely used open source data driven test automation tool which uses the same or different scripts with varying data sets achieved through programming which is very time consuming.

All of these tools address test data management as stated in Section 3, however to use them successfully for varying data sets technical knowledge is required.

In this paper, we present a test automation framework TAFPro that provide an execution harness for IBM Rational Functional Tester scripts to carry out robust, low maintenance automation, TAF Pro is not a test automation scripting tool. It is an execution tool. TAFPro has features and functionalities namely test planning, execution system, results reporting – aimed at basic analysis, data management and filtering and data delivery to multiple people.

TAFPro does not touch or interact with the application, RFT does. There is a definite demarcation line. TAFPro talks to RFT. RFT talks to the GUI. The GUI talks to the application. The Application talks to the database and so on.

4.2. Tool dependent Java based Test Automation Framework

Test automation maintenance effort is reduced, cost is lowered and efficiency improves with proper planning, design and architecture of the framework. The primary benefit is to design and deliver a framework that offers easy implementation,
speed, performance, maintainability, scalability and able to easily use multiple data sources to perform test automation.

TAFPro Automation Framework is one such solution that offers all the listed benefits. It contains features such as Planning, Execution System, and Results reporting—aimed at basic analysis, Data management and filtering and delivery of data to multiple people and provides a framework to build robust, low maintenance automation. TAFPro is not a scripting tool hence it does not Record or debug a script, solve why a script is broken or not function, interact with the application—objects or carry out version control.

Fig. 2. TAFPro with IBM Rational Functional Tester

TAF Pro instructs RFT to run a script with a nominated set of data. It does not influence the test in any way. It will send and instruction with some data if needed, and will collect the result from RFT and optionally a message if the script has included a return message. Pass or Fail, TAF Pro will make no decision, it will simply report that in a user-friendly way. TAFPro improves application deployment, upgrades efficiency and reduces project and application life cycle costs as shown in figure 3.
It provides an interface that simplifies test automation and test data management for business users. It enables testers to develop and execute their own test suites without the need to write new test scripts.

It is an integrated solution with the IBM testing tools such as Rational Functional Tester and Rational Quality Manager. This integration of TAFPro with these tools enhances the automation capabilities and increases the business value obtained from test automation significantly. However, the fact that TAFPro is limited to IBM testing tools only whereas organizations are now moving towards open source tools requires further work to extend the framework capability to wider platforms. After this great contribution to academia and industry it is vital to offer a framework that is flexible to adapt to changing needs of the organizations.

We then did the survey of the tools and identified Selenium being the most adapted test automation tool by the organisations and also being referred to in the research domain[12, 16-18]. Selenium is an open-source powerful and flexible tool that supports automation testing of web-based applications. It has core selene languages that look very similar to html and is bundled with all further offering like IDE, RC, Grid. Selenium RC allows to run tests written in programming languages to run from anywhere. Selenium 3.0 APIs offer very simple Mobile apps automation testing for IOS and Android (Appium).

We conducted detailed research and analysis to increase the strength of the test data management handled by TAFPro to contribute to the research community as diverse tools and solutions are available however none of them offer the test data management capability beyond quality of test data that is input to the tool.
5. Case study in test automation framework

With an aim to explore and enhance the test data management capabilities of the TAFPro test automation framework we initiated a project with TestPro and applied for a grant with NSW government. After successfully receiving the grant we conducted research on the various test automation tools and frameworks existing in the academic as well as industry to verify the need and justify the project objective.

A detailed comparative analysis was done with all the major data management tools available. We concluded that there is no other tool that offers similar test data management capabilities as offered by TAFPro. It combines elements of data-driven and keyword-driven frameworks, which makes it a hybrid automated testing framework. Test development and execution in TAFPro adopts a role-based approach.

Hence, we decided to disintegrate TAFPro from IBM products and make it tool independent to link to any Java based test automation tool. For the purpose of this case study we tested the new TAFPro with Selenium. We chose Selenium since it supports various browsers, operating systems, programming languages, and testing frameworks each with a different approach to supporting test automation for web-based application testing.

The architecture of the new TAFPro with Selenium integration is given in figure 4.

![Fig. 4. TAFPro with Selenium](image)

We requirements of the new TAF pro-Selenium integration were identified using Agile Scrum development. Few of the example requirements are given in figure 5.
The main functions of the TAFPro for adding test cases and scenarios to test execution suites conduct the test data management are as follows:

**Planned Scenarios:** Planned Scenarios are used to organise Actions into Test Cases and Scenarios that can be combined to create Execution Suites. A key element of getting the most out of TAF Pro is creating Actions that are reusable. Planned Scenarios contain one or more Test Cases and Test Cases contain one or more Actions. A Scenario or Test Case without any content is considered incomplete.

**Execution Suites:** The Execution Suites panel is used to create and execute Execution Suites. Execution Suites consist of one or more Planned Scenarios containing Test Cases and Actions. Each Action in an Execution Suite needs to be linked to a Data Group which will provide the data required to determine the test completed. An Execution Suite can be executed when all Scenarios are complete and all Actions are linked to Data Groups.

**Execution Results:** Displays the results available after running an Execution Suite. The user can drill down to review the input and output data for each line of the test as well as seeing any system messages sent from the Test Scripts. Results are colour coded green for success, red for failure and orange for a warning.

**Suite Filters:** The filters that have been defined for the Execution Suites are catalogued in this window. They are grouped under the Execution Suite name. Filters are used to create subsets of the Data Group lines during Test Execution.

**Action Data:** The Action Data panel displays the data contained for the currently selected Data Group or Action. Selection of the Action or Data Group can occur from several places in TAF Pro. The data shown will either be the first few lines, all data in the Data Group, or the data corresponding to a filter if the filter is selected.

Implementation of these requirements took few weeks followed by rigorous testing involving running the tests using selenium scripts and TAFPro.
results of the tests showed that the new TAFPro was able to achieve the objectives of improved scalability and efficiency in execution of tests and scripts built with Selenium; reduction in challenges associated handling of test data used during testing; ease of linking of test scripts to form scenarios and suites of associated tests.

With this implementation, we were able to minimize the impact of the framework on script development; provide a non-technical user interface that offer the use of framework for test execution from a business perspective as opposed to purely technical testing; flexibility in data selection and association with the filtering capability reducing the test maintenance overload.

6. ACKNOWLEDGMENT

We thank NSW Department of Industry for the Grant TECHVOUCHERS and the industry partner TestPro for continuous support in making this project a success.

7. References


Authors

Kamini (Simi) Bajaj

Academic Course Advisor (Information Systems Program) in School of Computing, Engineering and Mathematics at Western Sydney university. Simi has specialised in SDLC, Project Management, Test Strategy, Test Planning, Test Design, Test Execution, Defect Management, Test Management, Process Management and Implementation since 2004 and has a wealth of experience in all areas of the project and product life cycle and testing working with numerous industry sectors.

Simi has been doing research in the areas of Software Development, Software Testing, Information systems, Cybercrime, Phishing and Information Security. She has also acted as session chair in conference and published papers in local and mostly in International conferences. She has been reviewing papers for conferences and journals since 2004.