Open2Test Test Automation Framework for OpenScript (Web) – Introduction

Version 1.0

January 2010

Table of Contents

1. Preface

Automation testing is an upcoming field that draws maximum benefits with minimum effort. The benefits of automation testing can be seen in its ability to increase the efficiency of resources, test coverage, and the quality and reliability of the software.

While there are several frameworks that provide support for automated software testing, this document introduces one particularly effective type: the **Open2Test Test Automation Framework.**

In the Open2Test Test Automation Framework, the discrete functional business events that make up an application are described using keywords. This approach fosters code reusability, optimum utilization of the tool, and greater productivity.

2. Framework Overview

2.1. Introduction to Framework

Automation testing requires a well-defined approach, based on a comprehensive framework, in order to get maximum benefits.

A **framework** is a hierarchical directory that encapsulates shared resources, such as a dynamic shared library, image files, localized strings, header files, and reference documentation in a single package.

There are various frameworks available for automation, such as:

- Test Script Modularity Framework
- Test Library Architecture Framework
- Data-Driven Automation Framework
- Hybrid Automation Framework
- Keyword-Driven Automation Framework

Keyword-based test design and test automation is based on the idea that the discrete functional business events that make up any application can be described using short text description (keywords). By designing keywords that describe those discrete functional business events, testers begin to build a common library of keywords that can be used to create test scripts.

The Open2Test Test Automation Framework is a keyword-driven automation framework that works with Oracle OpenScript. This framework allows testers to develop test cases in .csv format (using Microsoft Excel) and a list of keywords. When the test is executed, the framework processes the .csv file and calls functions associated with the keywords entered in the .csv file. These keyword functions in turn perform specific actions against the application under test (AUT). The framework interprets the keyword and executes a set of statements in the program.

With this framework in place, applications can be automated without starting from scratch. Testers simply use the application-independent keywords and create extra application-specific keywords.

2.2. Framework Features

In addition to standard features such as performing operations and verifications on the objects, the framework includes other sophisticated features, including

1. **Use of variables**: Variables can be defined and used across the generated test script. This can be used to capture runtime values, which can be reused as input elsewhere during test execution.
2. **Conditional checking**: Conditional constructs such as 'if' can be implemented using keywords to handle different flows based on various conditions.
3. **Data-driven testing**: This framework supports data-driven testing by importing data from an external data sheet.

4. **Calling functions and reusable actions:** Common functions or actions can be triggered through keywords. This framework supports an OOPS approach. This increases the reusability of functions, which in turn reduces the unnecessary repetition of steps.
5. **Keyboard inputs:** This plug-in is included to perform keyboard actions on the specified test object.
6. **Date/time functions:** The date/time function plug-in is included to perform different operations on date/time.
7. **Exception handling:** Runtime errors can be effectively handled and reported using this framework.

2.3. Framework Benefits

Reusability:

The Open2Test Test Automation Framework is an application-independent framework that deals with all possible actions and verifications that can be performed on an object. Therefore, the code for the same object can be used across different applications.

Duplication of work is minimized at every level. For instance, a user might have to perform a certain action on an object of a similar class (e.g., clicking a button) repeatedly. This can be in the same test case or in a different application altogether. In both cases, the same code can be reused.

Optimum utilization of the tool:

The framework has the advantage of using keywords as the input for triggering an action. This well-built framework uses the features of the tool effectively. For instance, a common object repository created for an application can be used in all the test cases of the application.

Less effort:

The effort involved in coding and reviewing is minimal when compared to other frameworks, since a good percentage of coding is done within the framework. The tester simply has to enter the keywords, reducing the time required for coding. Recording is also not required as the global repository is used. The amount of rework required for migrating from one application to another on the same platform is reduced because the code remains the same.

Increased quality:

The scripts are of uniform quality since they make use of the same code.

Greater productivity:

Open2Test Test Automation Framework provides both qualitative and quantitative benefits for automation and is highly productive compared to any other framework. This framework also addresses the ongoing maintenance of the test scripts in a cost-effective manner.

Maintenance:

Simple modifications to the application can be easily handled in the code. The changes will be done only in the external file containing the code and the scripts need not be changed.

No coding skills required by the end user:

No coding skills are required to automate and review the scripts. The scripts are user-friendly with good readability. Scripts can be interpreted easily by a person who does not have complete knowledge of the tool.

Return on investment is high:

Although the initial effort for building a framework is high, in the long run, the return on investment will also be high because of the reusability and optimum utilization of the tool through the script.

3. Framework Architecture

3.1. Framework Architecture

Architecture forms the foundation of any software application. It should be robust enough to handle the desired functions efficiently and effectively. With this approach, the goal is to develop an application-independent, reusable keyword framework that can be used directly across any application without spending any extra time on it.

In order to make all the components of the system work in sync, it is important to define the components and its functionalities, as well as the binding relationship between them.

An Automation Framework Architecture comprises the following components:

- **Input Layer**
  1. Object Library
  2. Keyword Test Script
  3. DataBank
- **Framework Layer**
  1. Framework Engine
  2. User-Defined Functions
  3. Common Functions
- **Output Layer**
  1. Actions on AUT
  2. Results Report

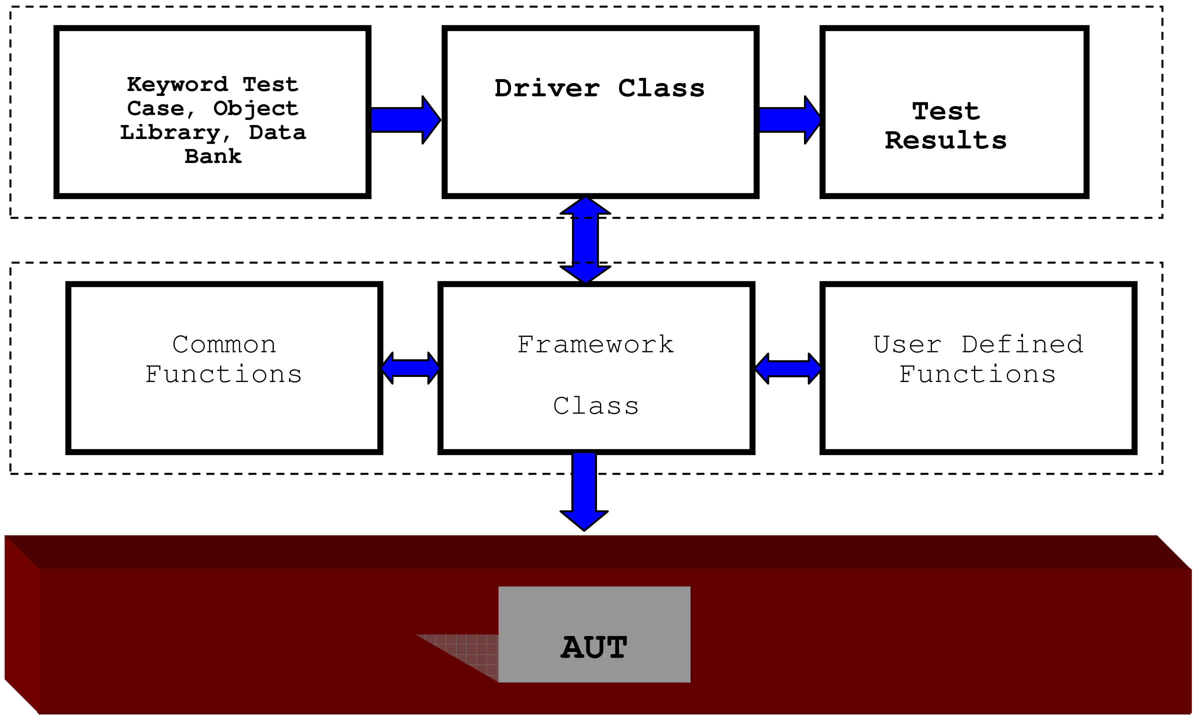

Figure 1: Framework Architecture

3.1.1. Object Library

The OpenScript Web Functional Test Module is an application module that supports functional testing of Web-based applications. It uses the Web Document Object Model (DOM) and the DOM Recorder to automatically capture Web page objects, actions, and navigations. Using the 'Inspect Path' feature of OpenScript, DOM Path of the required Web Page elements can be captured and stored in a defined **'Object Library'**. For Open2Test Test Automation Framework, 'Object Library' acts as an object reference for the Keyword Test Script.

The framework performs the actions on the objects defined in Object Library, as specified in Keyword Test Script.

3.1.2. Keywords Test Script

Keywords from the Test Case sheet trigger specific functions in the framework to perform a specific operation on the desired object in the application. The keywords test script sheet is loaded in the OpenScript in .csv format, from which the framework fetches the data table records. The data table records contain the keywords that describe the desired actions. This also provides additional data needed as input to the application, the benchmark information to verify the state of components, and the application in general.

Ideally, the keyword (vocabulary) should be written in such a way that the keywords are recognizable and suitable for manual testing. Retaining this crossover capability allows us to create one common test case for both automation and manual purposes. It is this feature that makes the Keyword-Driven Framework powerful.

3.1.3. External Test Data

External test data is given as inputs to the test scripts to perform the same operations on the application using different sets of data. This spreadsheet holds multiple combinations of inputs to be fed to test the application. External test data can also be given as an input sheet during checking operations.

3.1.4. Framework Engine

The framework engine forms the backbone of the automation framework. All the coding logic, in the form of Java methods, is stored in the framework engine. The framework engine has been designed in modular fashion, with effective exception handling. Addition and deletion of functions makes the framework flexible enough to customize it for any other application.

3.1.5. User-Defined Function

Application-specific functions can be designed and seamlessly integrated with the Open2Test Test Automation Framework as a user-defined function (UDF). Since the UDF feature has been designed as a separate component, updates to UDF have no impact on the framework engine.

3.1.6. Common Function

The common functions (CF) are the functions that are reusable across all platforms, like File System Operations. These functions are application-independent, and do not depend on the technology that has been used to develop the application. Separating the common functions from the shared module ensures maximum utilization of reusable scripts, and in turn reduces the maintenance effort of scripts.

3.1.7. Reporting

After execution of the test script, it is necessary to get the results of the execution. Open2Test Test Automation Framework has been designed to provide a detailed test report, which aids the effective debugging. The report details pass/fail status of test steps, test description, and exception occurred. This helps in performing effective analysis on the execution report.

3.1.8. Actions on AUT

As per the actions defined in Keyword Test Script, the framework engine simulates the user actions on the objects, with the reference from the Object Library. As the Open2Test Test Automation Framework supports various functions like Loop, Condition, and String

Operations, in addition to Action keywords, test applications can be automated effectively for functional testing.

4. Conclusion

Analysis is an important and time-consuming phase in automation testing. However, in the long run, the time spent will be useful during the regression phase. In order to keep up with the pace of product development and delivery, it is essential to implement effective, reusable test automation. The Open2Test Test Automation Framework provides a way to drive productivity and foster code reuse — ultimately enhancing the quality of resulting software.