Open Source Keyword Automation Framework

By R. Raymond

At the time I authored this posting, 109 results are returned when performing a search for the word 'keyword' on the Stickyminds web site. There is quite a discussion and knowledge transfer taking place on keyword-driven test automation. There are success stories. There are stories of failure. There are descriptions of keyword-driven automation. Some of the articles are new. Some are over 8 years old.

Quoting Dion Johnson in his article "Automation Déjà vu…Again!", "Why do we continue to ask the same questions that we largely asked over a decade ago?" If you read Dion's article you'll notice that I'm taking him a little out of context for the topic in this article, but his question is just as applicable to our context as the question was to Dion's content. One can find articles, whitepapers, and books from ten or more years ago describing keyword-driven test automation frameworks that are as applicable today as when the article, whitepaper, or book was originally written back in the day.

In the whitepaper by LogiGear titled "Achieving the Full Potential of Software Test Automation: The Strategy for Reducing Costs and Speeding Time-To-Market", the authors cite "lack of good test automation framework" as one of the reasons test automation projects fail to achieve their potential. Literature on good test automation frameworks can be found written ten or more years ago.

Modifying Dion's question to fit the topic of this article we should be asking -- with all of the literature and advancement of the science of test automation why does a paper written ten years ago still apply today? Have we in the test automation career field not advanced from the past into the future? Dion was right; we continue to ask the same questions that we asked ten years ago even if he was talking about something slightly different.

Fortunately, time is on our side and has rendered a solution. Open source was not as big ten years ago as it is now and an open source keyword-driven test automation framework just might be the solution.

An open source keyword-driven test automation framework creates the good test automation framework that avoids the reasons automation projects fail to achieve full potential by combining the experience of a community of test automation engineers to create an automation framework that all of the textbooks, whitepapers, and articles say we should create. Whitepapers, articles, and books tell us to create such a framework and provide guidance on how to create an automation framework, but none have actually built the recommended framework and given it to us…until now.

A group of automation engineers at Keane, Inc. working in their Innovez R&D lab, have created a robust, tool-independent, keyword-driven automation framework and are making the framework available to the QA and testing community as open source to take the technology to the next level.

The basic premise builds from the standardization of keywords for common functions across a variety of technologies and has written driver scripts for a variety of automation tools. This design allows testers to create test scripts using keywords in Excel spreadsheets and run those scripts with all of the major test automation tools on the market.

The architecture of the system is shown in figure 1.

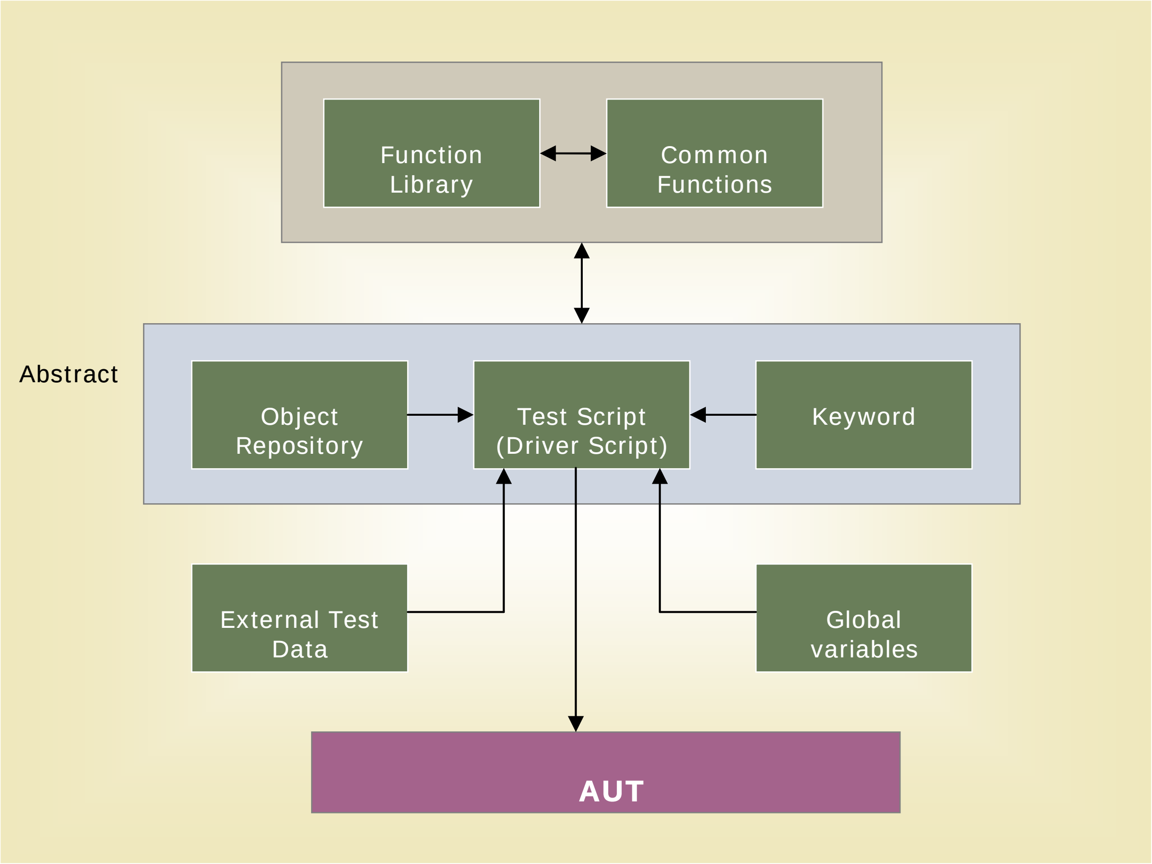

Figure 1. Automation Framework Architecture

The keywords have been standardized across all tools and technologies. The driver script has been written for a specific test tool and technology combination.

The advantages of the framework are several multi-faceted:

Start quickly **The framework code is proven successful, and in using the code on your project. you are starting with instant success without having to spend months or years to develop your own framework.**

Tools agnostic **Test scripts contained in the spreadsheets do not need to be rewritten and will run with a new tool should you change tool vendors. Tools licenses are expensive and it is easy to change tools if you don't have to re-script your automated test cases. Perform object recognition with the new tool to create a new object map, then change object names in our scripts and you are on your way.**

Technology agnostic **If the application under test changes technology such as going from an old VB application to Java, then your scripts are still valid, and all you need to do is perform object recognition and change the object names in the keyword scripts. You'll also need a new driver for your automation tool and Java which you can get from the open source repository.**

Vendor agnostic **The processes for management and deployment are owned by the open source clients.**

The keyword-driven automation framework can be downloaded from www.Open2Test.org under the GPL as open source for use on your projects. The web site describes the framework in detail and the road map for the framework's future. It would be great if you get value from using the framework, but it would be greater if you could become part of the open source community to help maintain and expand the technology.