УДК 004.415.53

NATIVE APPLE TV AUTOMATION

Serdziukou P.S., Savenko A.G.
Institute of Information Technology of Belarusian State University of Informatics and Radioelectronics

Connected TV devices are very widespread in the Western market and have become part of the daily life. The number of sold set-top boxes has been growing steadily, and has doubled over the past 4 years. With the growth of sales of such devices, the demand for developing high-quality applications for TV platforms and, accordingly, test automation of these applications is growing. The largest media corporations like HBO, NBC, Netflix and Viacom have their own applications for connected TV devices for round-the-clock access to their content.

Connected TV is a device that's connected to the internet and accelerates the delivery of streaming video content. They provide entirely new experience for television viewers by delivering interactive features such as smart video, games, video streaming in addition to regular television content.

Apple has taken care of creating test automation frameworks for the entire ecosystem of their devices. XCTest library makes it possible to implement unit tests for iOS, tvOS, watchOS and OS X platforms.

XCUITest – A UI testing framework that is built on top of XCTest. It includes additional classes (such as UIAccessibility). The tests are packaged in a test ipa (iOS packaged application) runner that executes the tests on the AUT (application under test) ipa [1]. You can check XCUITest work principle on the picture 1.

XCTest is easy to learn and support testing framework. Apple introduced command line tools for Xcode allowing to execute XCTest tests using, for example, shell scripts. It also brings a potential in seamless continuous integration. For instance, widely used CI systems, such as Jenkins, have a lot of plugins for Xcode integration.

Picture 1 - XCUITest work principle

Nevertheless, XCTest framework requires to write tests in Objective-C or Swift, which are not popular among automation engineers, due to narrow platform orientation of this languages. Also, tests should be written in the same code base as the application, what can

become a problem in widely separated teams. In fact, this code requirement makes it impossible to name XCUITest as black box testing solution and whiter one. Automation and QA engineers need to have a solution, which can help them to test application in more natural way and with no need in the source code.

Engineers from Facebook already resolved this issue for iOS platform and this solution is WebDriverAgent [2]. It is HTTP server that launches on the iOS device and proxies XCUITest framework to provide the means to execute any suit of commands on mobile devices, such as iPhone and iPad. There are a lot of enterprise solutions which works with this framework, for example Appium – the most popular aggregator of mobile test automation frameworks [3]. Also, WebDriverAgent implements major part of the JSON Wire Protocol, what permits easy integration with Selenium based solutions, for example Appium used this feature within their framework. Perhaps, it should be mentioned that this solution supports only iOS device family.

Now our own input will be described. We spent several months to expand WebDriverAgent on tvOS platform [4]. Main logic was written in Objective-C as it is the most commonly used language for Apple orientated framework. If you write libraries using Swift, you will spend more time to support them, because this language may obtain huge changes (even lexical) from release to release. The main different with mobile platform was related to type of user interaction with the device. While interacting with connected TV-device, you need to use remote control as focuse engine, on the other hand on mobile devices user needs only his or her finger as a controller. The main feature of my solution is that it automatically explicitly navigates to target focusing element, there is no need in explicit navigation commands execution, like up, down, left, right. So, it also improves user experience for this framework.

As WebDriverAgent is a HTTP server, you need to execute HTTP commands somehow. That's why I have create Java client for this implementation of WebDriverAgent [5]. It wraps routes requests execution with well-known Selenium based API. Also, it provides logic for driver session setup and management.

As a result, we have ready to work solution for native Apple TV automation. Our WebDriverAgent based solution may save thousands of man hours spent on the manual tests execution. Moreover, it takes teams to a whole new level of Continuous Integration, Continues Delivery and Deployment processes for tvOS devices. It needs to be mentioned, that all the listed solutions are open source and free to use.

References

1. Nuriel, R. XCUITest – The Emerging iOS UI Test Automation Framework / R.Nuriel // Perfecto [Electronic resource]. – 2018. – Mode of access: http://blog.perfecto.io/mobile-application-testing/xcuitest-the-emerging-ios-ui-test-automation-framework/. – Date of access: 14.11.2018.

2. WebDriverAgent / Facebook // Github [Electronic resource]. – 2018. – Mode of access: https://github.com/facebook/WebDriverAgent/. - Date of access: 14.11.2018.

3. Appium: Mobile App Automation Made Awesome // Appium [Electronic resource]. – 2018. – Mode of access: http://appium.io/. – Date of access: 14.11.2018.

4. WebDriverAgent / P.Serdziukou // Github [Electronic resource]. – 2018. – Mode of access: https://github.com/shvul/WebDriverAgent/. – Date of access: 14.11.2018.

5. WebDriverAgent Java Client/ P.Serdziukou // Github [Electronic resource]. – 2018. – Mode of access: https://github.com/shvul/wda-java-client/. – Date of access: 14.11.2018.