Differences in Automated Testing on MeeGo and Android Mobile Platforms

Oleksandr Kachur, Vladimir Sayenko
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
{alexander.kachur, visank}@gmail.com

Abstract

The main target of this report is to investigate possibilities of automated black box host based functional testing on MeeGo and Android platforms, for third-party applications. The comparison of automation testing approaches is considered. The host based architectures for open source mobile environments are summarized.

Index Terms: MeeGo, Android, testing, black box, automated testing, host based testing

I. Introduction

Quality assurance became the main part of a development process. In huge systems where the price of a bug is big, the test process provides information about product quality to the other departments. Being integrated into the continuous integration system the automated tests make possible to perform a fast feedback in reports about functionality that had been broken [1]. Project management makes plan based on this reports and developers fixes bugs that had been found. Automated testing is a powerful approach that provides to the quality assurance department actual data about the product state.

II. Main part

In this work automated testing process, based on black box host based approach, was chosen. To investigate capabilities of each platform it is sufficient to make these steps:

1. Choose application for different platforms with similar functionality.
2. Develop for each platform automated test, which will perform next action sequence:
    a. Start application.
    b. Execute some operations.
    c. Check that all operations were executed successfully.
    d. Close application.
3. Analyze test results.

The application “calculator” had been chosen as application under test (AUT). This application is chosen because of simple functions and its high source code availability for both platforms.

The implementation for MeeGo was based on decisions that were previously made for Android. The reason is that there are a variety of possible ways to implement this

solution on MeeGo platform. Figure 1 shows developed architectures for both platforms. Architectures depicts as UML deployment diagrams.

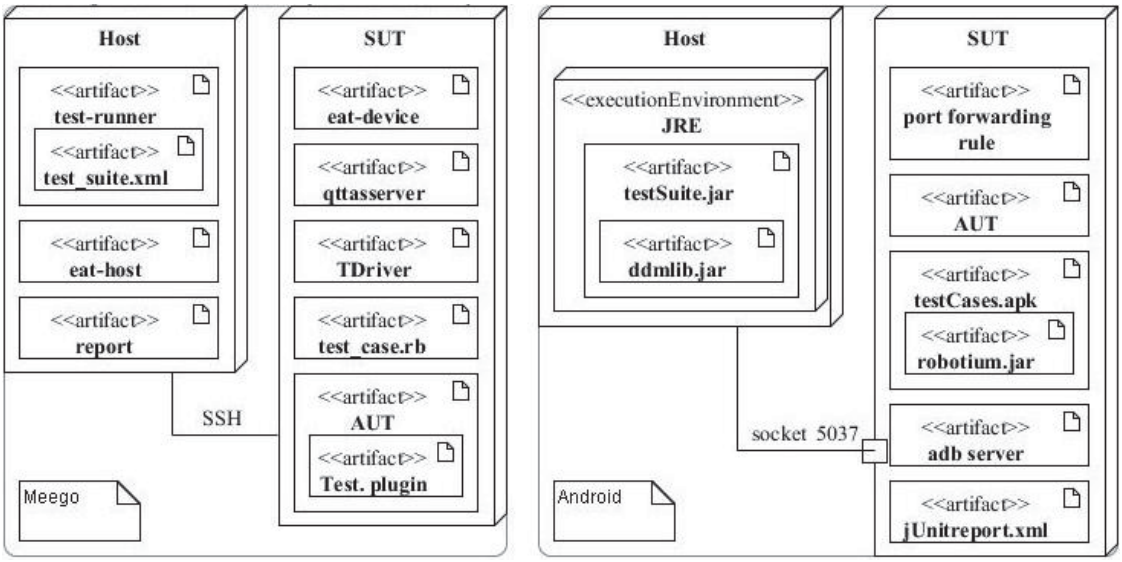

Fig. 1. MeeGo and Android host based testing architectures

Also a comparison of selected attributes for both platforms had been performed. Table I shows the result of this comparison [2, 3].

TABLE I
Comparison of MeeGo and Android testing features

	Android	MeeGo
Execution on real device	Impossible to interact without PC	+
System interaction	DDMS	bash
AUT interaction	Instrumentation	TDriver
Test plan execution	-	Testrunner
Semi-automated testing	-	+
Verify images	-	+
Unit testing	JUnit	Test::Unit, MiniTest
Monkey testing	+	+
Reporting	No out-of-box solution. Few open-source projects to generate junit report.	Testrunner (QA reports format) TDiver (several formats [4])

III. Conclusion

So now it's possible to conclude, that MeeGo provides for testers more flexible solution with wider functionality than Android does. But due to problems with environment deployment (there is no official support for AMD processors and non-Intel video cards) we'll look forward to get more user-friendly SDK.

References

[1] Joel Spolsky, "Daily Builds Are Your Friend," http://www.joelonsoftware.com/articles/fog0000000023.html
[2] Meego wiki, "Quality assurance (QA) tools," http://wiki.meego.com/Quality/QA-tools
[3] Android dev guide, "Testing fundamentals," http://developer.android.com/guide/topics/testing/testing_android.html
[4] Forum.Nokia, "Testability driver," https://projects.forum.nokia.com/Testabilitydriver/wiki/FeatureDocumentation