ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Mobile Test Automation Framework for Automotive HMI

Rajkumar J Bhojan[1], K.Vivekanandan[2], Subramaniam Ganesan[3]

Solution Test Architect, Wipro Technologies, Detroit, MI, USA[1]

Professor, BSMED, Bharathiar University, Coimbatore, India[2]

Professor, Electrical & Computer Engineering, Oakland University, Rochester, MI, USA[3]

Abstract: As the mobile applications and mobile users are growing rapidly, it is indeed for researchers and testing experts to come up with effective verification techniques to ensure reliability of these mobile applications. In order to mitigate mundane manual testing on Mobile application, we have come up with a customized mobile test automation framework using SeeTest and Selenium TestNg flavor. In this paper we describe two popular cases of mobile applications for automotive testing, sample regression test cases on real devices, Framework for handling complex automotive test cases, and an example automated test.

Keywords: Mobile Test Automation, automotive testing, framework, Regression Testing

I. INTRODUCTION

Gartner stated that, by 2015 mobile application development will outnumber PC projects by a ratio of 4-to-1. Gartner also predicts that emerging mobile applications, systems and devices are transforming the application development space rapidly, and are one of the top three CIO priorities at the enterprise level [1]. According to Forrester Research Inc., by 2016, smartphones and tablets will put power in the hands of a billion global consumers [2]. In real time projects, manual testing tasks, such as extensive low-level interface regression testing, can be laborious and time consuming to do manually. Moreover, a manual approach might not always be effective in finding certain classes of defects. Test automation offers a possibility to perform these types of testing effectively. Once tests are automated, they can be run quickly and repeatedly over the time. This can be a cost effective method for regression testing of software products that have a long maintenance life. Regression testing includes rerunning previously-completed tests and checking whether program behavior has changed and whether previously-fixed faults have re-emerged.

II. CHALLENGES IN AUTOMATED TESTING IN MOBILE APPLICATIONS

Some of the Smart Phone device limitations with respect to automation are stated below.

1. Small screen size
2. Limited CPU power
3. Limited RAM size
4. Limited secondary storage
5. Limited battery life
6. Cumbersome input UI

As described in [3] mobile applications differ from traditional desktop applications. The following table summarizes various peculiarities and implications on Testing.

Table 1 – Mobile App Characteristics

Mobile Apps Distinctive Characteristics	Implications on Testing
Connectivity	Functional, Performance, Security, Reliability testing through different networks
Convenience (Quality Design)	GUI Testing
Supported Device (Diversity of physical device and operating systems)	Test Matrix based on Diversity Coverage testing
Touch Screens	Usability and Performance testing
New Programming Language	White Box and Black Box testing, Byte-code analysis
Resource constraints	Functional and Performance monitoring testing
Context Awareness	Context dependent functional testing

According to authors, Hyungkeun Song, Seokmoon Ryoo, Jin Hyung Kim, Mobile Test Automation is the most difficult part because features and functionalities are different for each platform. So, framework should be designed to integrate them. If same function exists on iPhone and Android, applications can be tested using the same interface. Even though the testers don't know what kind of mobile platform will be tested, they should confirm the test result. They also stressed that testing efficiency will be improved if it is an integrated environment to support the testing of various platforms. [4]

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Why Mobile Test Automation?

As mobile applications are becoming increasingly sophisticated, they significantly increase the requirement for functional and non-functional tests. In order to tackle this, test organizations are increasingly exploring alternatives to traditional manual testing. Automated testing is a highly effective approach to mobile application quality assurance that can provide significant business returns, provided it is implemented by using the right tools and architecture, factoring in cross-platform challenges.

The following specific types of testing also need to be automated:

Cross-platform compatibility testing: This is necessitated by the growing number of handsets and platforms.

User experience testing: Most testing organizations have limited experience with design and execution of usability tests.

Field or network testing: Testing must be performed in a geographically distributed environment to account for a variety of network types.

Structural challenges: Integrating mobile testing toolsets into existing IT systems is a crucial hurdle. [5]

The tools such as Eggplant, Jamo, SeeTest, ZAP, SilkTest, MonkeyTalk, Robotium and TestQuest are being used for Mobile test automation [5]. In order to achieve a common platform independent framework, we need to select any one of the tools from the above list.

III. INTRODUCTION TO AUTOMOTIVE HUMAN MACHINE INTERFACE (HMI)

According to authors [6], the software part directly interacting with the user of an infotainment system is called an HMI (human-machine interface). HMIs are difficult to specify, implement, and test, since their static and dynamic properties are very complex. They also stressed, that car infotainment systems are an important part of modern cars, providing radio functionality, media playback, navigation, and even telephony. They are expected as part of the standard instrumentation in high-priced cars, but are increasingly finding their way into the medium-priced and even low-priced market segments as well [6]. Before a new HMI version is actually delivered to the OEM, several test activities take place at the supplier's site. These tests typically include module / component tests, integration tests, as well as system tests. As described in [7], the automotive testing opens up new research areas in the following model based Human Machine Interface (HMI) testing approach:

- Designing a test-oriented Human Machine Interface (HMI) specification model
- Identification of test generation criteria and test generation methods specific for infotainment HMIs
- Definition of methods and interfaces for automatic test execution, observation and verification of the SUT (system under test)

IV. MOBILE APPLICATIONS FOR AUTOMOTIVE DEVICES

Most of the major car manufacturers like GM, Ford, Lexus, Nissan, Audi, etc., use mobile applications as their enhanced features. These mobile applications are available for iOS, Android, Windows and BlackBerry devices.

The following steps show how to connect with Cars and what type of tests can be done using Mobile applications: (Figure 1)

1. Log on to mobile application to ensure a secure connection between your vehicle and your mobile device.
2. Monitor Fuel and Oil levels: The app lets you check your current fuel economy and oil life, making it easy to monitor these critical indicators of your vehicle's health.
3. Check Tire Pressure: Properly inflated tires can improve the safety and fuel efficiency of your vehicle. With these apps, you can check the current and recommended air pressure for each tire on your vehicle in real time
4. View Hands free calling information: Apps lets you quickly check that you can look up your vehicle's phone number or keep tabs on how many minutes you have left and when they expire.
5. Control your vehicle remotely: Use mobile apps to activate remote features of your vehicle. Now you can start your engine, lock and unlock your doors, and flash your horn and lights from anywhere.

The steps involved in mobile vehicle communication are described in the following flow chart in Figure 1.

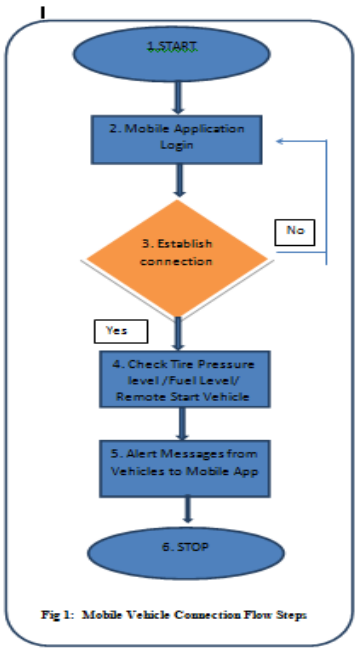

Fig 1: Mobile Vehicle Connection Flow Steps

The following Figure 2 shows the Mobile app connectivity with Vehicles.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

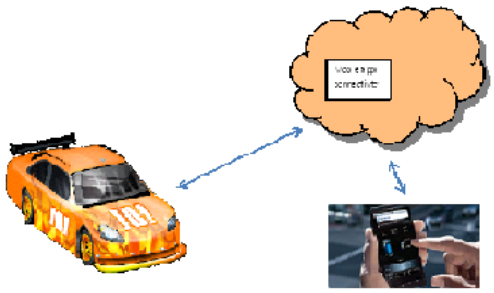

Figure 2. Mobile App Connection

IV. SAMPLE REGRESSION TEST CASES ON REAL DEVICES

For the purpose of framework development, we also collected some of the following real time test cases for mobile test automation on intelligent devices.

S.No	Test Case Descriptions
1	Verify user request to establish a connection with his vehicle
2	Verify the subscriber "Remote" Command is successful
3	Subscriber information and vehicle data could not be fetched_ErrorCode101
4	Verify Mobile app is active on device: Perform a honk horn.
5	Perform Lock/Unlock command for the Latest model vehicle which is in Disabled state

Table 2 – Test Case Descriptions

VI. FRAMEWORK APPROACH

We propose an N-tier architecture framework for implementing mobile testing on automotive systems as shown in Figure 3. The selected automation tool will interact with mobile applications in either IPhone or Android or Blackberry mobiles. The application in turn communicates with the automotive controller systems of the cars. The goal of framework is to focus on the following automated tests on the device. [8]

a. Functional Test

Functional testing is how well the mobile application executes the functions it is supposed to execute, including user commands, data manipulation, searches and business processes, user screen and integrations. It verifies that each function of the mobile application operates in conformance with the requirement specification. In our model test cases, when a user clicks “Lock” button, it will lock the specific car.

b. SOA(Service Oriented Architecture) / web Service Tests

Whenever a service consumer has access to the contract (WSDL) file of a given Web service, then, that particular service can be invoked via client application. In our example, mobile application is a client which interacts with a specified car engine. Whenever the subscriber information is not matching, then service cannot be invoked.

c. Performance Testing

Performance testing generally describes the processes of making the device and its server as efficient as possible, in terms of response time, machine resource usage, and server request handling.

d. Device Testing

Device Testing includes verification and validation of hardware devices of a Car which includes services like Lock, Unlock, honk, start, stop, etc. If a user locks his car after giving his PIN number, his car should lock the doors. An alert command should appear after the lock. PIN authentication should also be happened while the user clicks lock or unlock the car.

The proper alert message has to be given whenever user is giving wrong commands. For example, if the car is already unlocked and user is trying to unlock the car, an alert message has to be shown as "Already unlocked". If the fuel is empty or the car engine has a problem or car battery is dead, it should not allow the user to "remote start", app has to give a message "Cannot be started at this moment".

e. Security Testing

Authors Michael Becher, et al, stated that a security violation occurs when an app performs an action beyond the permissions granted to the app at install-time by the underlying smartphone platform. For example, if an app accesses sensitive data for which it is not granted permission, this is a clear security violation. [9]. Authors Peter Gilber and Byung-Gon Chun also stressed that Smartphones and "app" markets are raising concerns about how third-party applications may misuse or improperly handle users' privacy-sensitive data. In order to enhance mobile device security, thorough validation of apps applied as part of the app market admission process. They also recommend that automated validation of smartphone apps at the app-market level is a promising approach for drastically improving the security of smartphones [10].

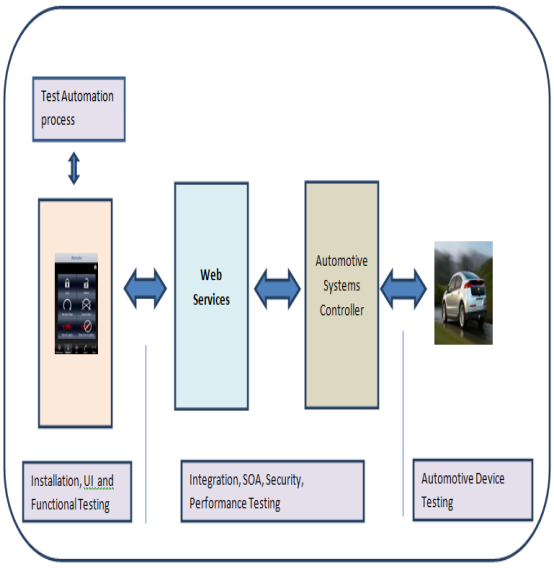

Fig. 3 An N-tier Architecture Frameworks for mobile testing

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

VII. FRAMEWORK FOR HANDLING COMPLEX AUTOMOTIVE TEST CASES

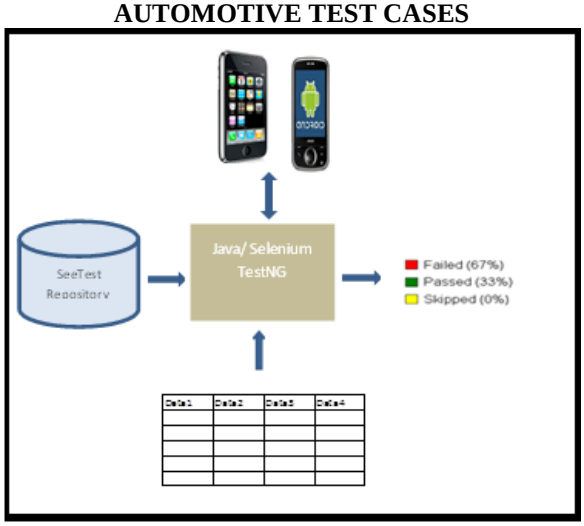

Fig. 4 Framework for Test Execution

Our Mobile Test Automation framework has been designed for automating automotive HMI functional test cases. The framework broadly covers the following high level approach:

1. The scripts should be reusable across different variants of the Mobile devices. There should be minimal script modification because of variation of the hardware characteristic of the devices.
2. Script maintenance should be low and scripts should be user friendly
3. The automation script should be reusable across different version of the mobile applications.
4. The automation framework should run under real-world conditions. In other words, the automated scripts of the application are executed in actual mobile device, in real environment, using the processor and memory of the device and not using any kind of emulators.
5. The framework should also support test automation on real car devices and get results from them.

As shown in the Figure. 4, we selected appropriate test cases after thorough analysis and feasibility for the automation. The repository used for defining objects by one of the following identification methods: Native, Web-DOM, Text and Image. The object repository is the place where SeeTest recognizes and stores information such as native properties, text, values, etc. [11].With this object repository, the tool identifies the mobile application and executes the business flows as per the functions in the test script. If any one or more of the object's property values in the mobile application differs from the property values stores in the object repository, the tool will not identify the object and the script will fail.

We extracted the application elements that we want to run a test and set name for each and every objects. Once it is extracted, the elements will show up in the object repository. The generated scripts were exported to a Selenium TestNg project [12]. On top of these exported scripts, we added relevant java code based on the test case steps/flows in Eclipse ide. We also modified scripts with additional features like taking data inputs from excel sheet and merging one or more executable methods and sending the reports in an html format. The external test data is given as input to the test scripts to perform the same operations on the application using different set of data. We used spreadsheet which holds multiple combinations of inputs to be fed to test the mobile application. As we start executing the scripts, the controller goes on IPhone/Android/Blackberry devices, which are connected through USB provisions. In reporting section of the framework, after execution of the test script, it is necessary to get the results of the test execution. The reports are customized in the Selenium TestNg framework such that the summary report and the detailed report are stored in html format. This provides report details for which test scripts have failed or passed while running a test suite. The summary report provides details of execution, duration, test start time and end time. The detailed reports describe exceptional cases handled, steps passed, and steps failed. Then the TestNg framework will generate XSLT & HTML reports with screen shots as shown in Figure 5.

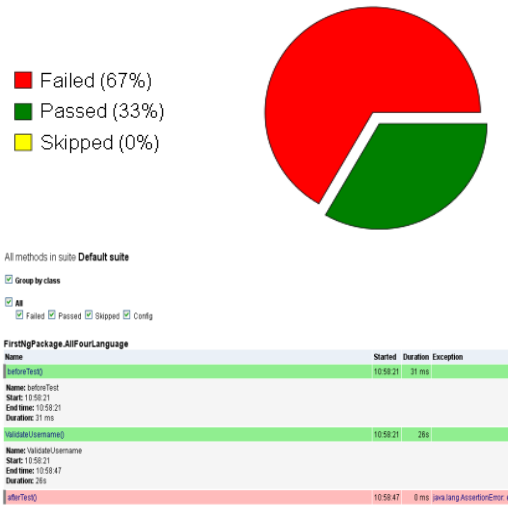

Fig. 5 XSLT & HTML Reports generated after the execution

The following Figure 6, shows Screen Shots of OnStar Remote Link application.

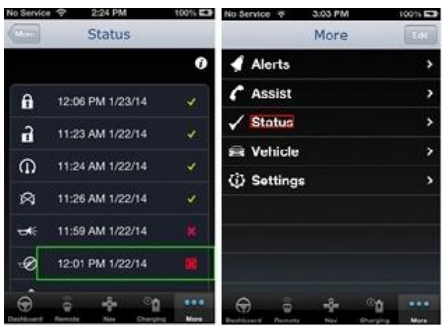

Fig. 6 Test Run Screen Shots of the application on real device

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

VIII. CONCLUSION AND FUTURE WORK

As the mobile applications and mobile users are growing rapidly, it is indeed for researchers and automated testing experts to come up with effective verification techniques to ensure reliability of these mobile applications. As development cycles become shorter over time, the need for test automation in regression testing is mandatory, particularly when using agile (scrum) development methodologies [13]. In the world of mobile applications, Mobile Test Automation is critical to ensure application quality. This framework forces encapsulation by hiding technical detail for manual team reduces the complexity and provides ease of handling test execution. We believe our framework can easily be extended to find a broader range of testing on Mobile test automation for automotive testing.

REFERENCES

[1] Susan Moore Gartner, "Cloud, Mobility and Open Source will drive Application Development Market", Integration Summit-Sydney, November 2012, http://www.gartner.com/newsroom/id/2131115
[2] Ted Schadler, John C. McCarthy, "Mobile Is the New Face of Engagement," Forrester Research, February 2012
[3] "Software testing of mobile applications: Challenges and future research directions," http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6228987
[4] Hyungkeun Song, Seokmoon Ryoo, Jin Hyung Kim, "An Integrated Test Automation Framework for Testing on Heterogeneous Mobile Platforms", First ACIS International Symposium on Software and Network Engineering, 2011.
[5] http://www.utest.com/mobile-app-testing
[6] Steffen Hess, Anne Gross, Andreas Maier, Marius Orfgen, "Standardizing Model-Based In-Vehicle Infotainment Development in the German Automotive Industry", Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12), October 17–19, 2012, NH, USA
[7] Linshu Duan, "Model-Based Testing of Infotainment Systems on the Basis of a Graphical Human-Machine Interface", Second International Conference on Advances in System Testing and Validation Lifecycle, 2010.
[8] Pradeep Kumar, Ramakrishnan "Selecting the Right Mobile Test Automation Strategy: Challenges and Principles", Sep 2012, www.cognizant.com/.../Selecting-the-Right-Mobile-Test-Automation-Strategy
[9] Michael Becher, Felix C. Freiling, Johannes Hoffmann, Thorsten Holz, Sebastian Uellenbeck, Christopher Wolf "Vision: Automated Security Validation of Mobile Apps at App Markets", IEEE Symposium on Security and Privacy, 2011, DOI 10.1109/SP.2011.29
[10] Peter Gilber and Byung-Gon Chun, Vision- Automated Security Validation of Mobile Apps at App Markets ", MCS'11, ACM 978-1-4503-0738-3/11/06, 2011.
[11] http://experitest.com/
[12] http://testng.org/doc/index.html
[13] Mike Cohn, "Succeeding with Agile: Software Development using Scrum", Pearson Education, 2009. ISBN-10: 0321579364.

BIOGRAPHIES

Rajkumar J.B, is a Solution Test Architect, Wipro Technologies, Detroit, MI, USA. He has over 20 years of experience in both IT and Academics. He holds M.Sc., (Physics) from Bharathidasan University, MCA., from Bharathiar University and M.Phil (Computer Science) from Manonmaniam Sundranar University, India. Currently he is pursuing Ph.D (Computer Science) from Bharathiar University, India. He has executed IT projects in diverse geographies including India, APAC & USA. He is a Certified Scrum Master and has rich experience in Agile/scrum Methodologies. He is a member in IEEE and ACM.

Dr.K.Vivekanandan, Professor, has been working with Bharat hiar School of Management and Entrepreneur Development, Bharathiar University for the past 25 years. He has guided 15 Ph.Ds. and Published 25 articles in reputed International Journals. He received his Masters and Ph.D (Computer Science) from Bharathiar University, Coimbatore, India. He has a long term association with National University of Rwanda for collaborative research work in Information System.

Dr. Subramaniam Ganesan is a Professor of Electrical and Computer Engineering, Oakland University, Rochester, MI, 48309, USA. He has over 25 years of teaching and research experience in Digital systems. He served as the chair of the CSE department from 1991 to 98. He is with Electrical and Computer Engineering Department since 2008. He received his masters and Ph.D. from Indian Institute of Science (IISc) Bangalore, India. He worked at National Aeronautical Laboratory (NAL) India, Ruhr University, Germany, Concordia University Canada, and Western Michigan University before joining Oakland University in 1986.