System for automatic testing of Android based digital TV receivers

Marko Kovacevic, Branimir Kovacevic
University of Novi Sad
Faculty of technical sciences
Novi Sad, Serbia
kovacevic.mare@gmail.com, brankomir@gmail.com

Dejan Stefanovic, Vukota Pekovic
RT-RK DOO
Novi Sad, Serbia
dejan.stefanovic@rt-rk.com, vukota.pekovic@rt-rk.com

***Abstract*—With constant increasing of digital TV receivers complexity and constant need for reducing the product time-to-market - the development of reliable and effective system for automatic testing of digital TV receivers becomes highly desirable. This paper proposes system for automatic testing of Android based digital TV receivers which is able to cover the entire process of product testing – from requirement definition, through verification test plan creation, automatic test cases execution and testing reports generation. System consists of web tool responsible for requirement and test management and stand-alone application responsible for automatic tests execution based on Black Box Testing methodology.**

Keywords-digital tv receiver, android, black box testing, set-top box, automated testing

I. Introduction

Modern receivers of digital TV signal offer a lot of different functionalities to end users, with a tendency to increase these functionalities with each new generation [1]. Standard set of TV features such as live TV channels playback (DVB-T, DVB-C, DVB-S), electronic program guide (EPG), Teletext, Subtitles, MHEG, personal video recorder (PVR), picture-in-picture (PiP) is extended with introduction of the Internet in the sphere of digital television. New features such as IPTV channels playback, hybrid broadcast broadband TV (HbbTV), video on demand (VOD), catch-up TV emerged.

With increasing hardware capabilities modern receivers become capable of executing complex applications written in Java programing language and besides traditional TV receivers based on Linux operating systems, new generation of TV receivers based on Android operating system appears.

With introduction of Android to digital television [2][3] - set of TV features is expanded with new features supported within Android OS. Besides TV functionalities Android OS offers a lot of interactive and non-interactive services and applications that can be installed and executed on the digital TV receivers.

With such increased complexity functional testing of TV receivers became very challenging and time demanding task. A minimal set of test cases required for testing of such complex system is 2.000. Some real examples (Android based Set-top box) showed that it takes approximately 5-10 minutes to carry out one test case manually. So execution of test suite that contains 2.000 test cases requires more than 20 working days. If it is assumed that in development phase of the project software release is typically launched each two weeks it is obvious that even in this stage manual testing is not acceptable. Manual testing is time-consuming and requires a lot of human resources. In later stages of project - software releases are more frequent and it is required that tests are executed frequently and repetitively so the human effort must be even more increased.

Therefore, manual testing of Android based digital TV receiver as a time consuming, human effort consuming and prone to errors due to manual mistakes, needs to be replaced with effective and reliable automated test system.

Currently available systems for testing of digital TV receivers are handling only test management process which facilitates manual testing or are capable to automatize small set of test cases needed for testing complex digital TV receivers.

One of the attempts for creating system which is able to cover all segments of verification and testing process of TV receivers is presented in [4]. This solution handles test management, test execution and reporting process during TV receivers' verification and testing.

This paper presents improvement of previously mentioned solution in the section of test management, reporting process and extension of test execution part with support for Android-specific feature testing. Test management and reporting are integral part of web tool so access to tests and results of test execution is possible from anywhere, at any time. Solution is able to cover the entire process of digital TV receiver testing – from initial requirement definition, through creation of verification test plan, automatic test cases execution and generating reports of testing. It is capable to automatize large part of the tests needed for reliable and effective testing of digital TV receivers based on Android OS.

The rest of the paper is organized as follows: Section II gives general overview of the proposed system for automatic testing. In Section III web-tool responsible for requirement and test management, test plan management and reporting is presented. Section IV gives detailed description of module for automatic execution of test cases using Black Box Testing methodology (BBT). The conclusions are drawn in Section V.

II. The Proposed System for Automatic Testing

General overview of the proposed system is depicted in Fig. 1. Central part of the system is server which is accessed by clients that may require different services, such as requirement management, test management, test plan management, automatic test execution and report generation. Web based client application is used to provide access to all services of the server except automatic test execution using REST (Representational state transfer) services exposed by server. For automatic test execution, stand-alone Executor application is used together with appropriate hardware for test execution. Application uses test plans made through web based client application to autonomously execute tests. After tests are executed, results of execution can be obtained through web based client application. Executor application uses SOAP (Simple Object Access protocol) web service exposed by server to retrieve test plan information and submit execution results.

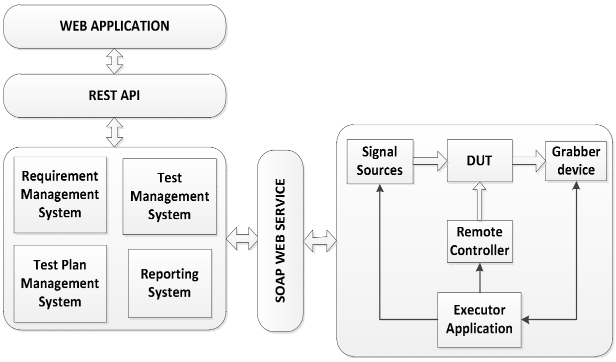

Figure 1. General system overview

III. Requirement and Test Management

Requirement and test management is implemented as web application which consists of four main sections (Fig 2.):

- Requirement management
- Test management
- Test planning
- Reporting

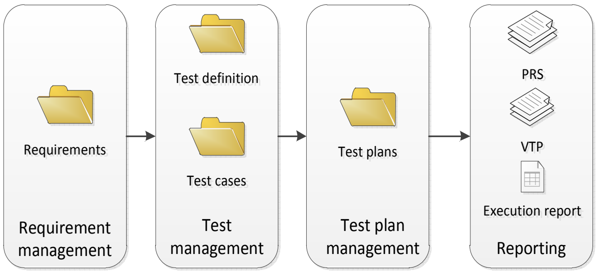

Figure 2. Requirement and test management application

Each section is represented as a top level node in a project tree. Project tree is the main interaction point with user, with all important application functions exposed in form of context menu options applicable on tree nodes.

Requirement management section is responsible for creation, modification and maintenance of project requirements specifications and project requirements. It provides options such as creating/editing/deleting of requirements specifications and requirements, assigning test definition to requirement, assigning documents to requirements and requirements specifications.

Test management section is responsible for creation, modification and maintenance of test suites, test definitions and test cases. It provides options such as creating/editing/deleting of test suites, test definitions and test cases, establishing relation between requirement and test definition, assigning documents to test suites and test definitions. For automatic and semi-automatic test cases test scripts can be added and modified in embedded Python script editor. Referent picture, video, audio and text files needed for automatic and semi-automatic test cases can be assigning to each test case.

Test plan management section is responsible for creation, modification and maintenance of execution test plans. This module enables composing of test plans by simply dropping test suites from test section into test plan, assigning users to test suites or test cases, assigning configuration for testing to test cases which defines devices and modules necessary for test execution (device under test, signal sources, grabber devices, remote control devices, algorithms, etc.). Execution of manual test cases is also possible within this section. It provides detailed description of test with the name of the test case, list of equipment necessary for test execution, description of test case, instructions how to setup test environment, list of inputs for test, test steps that needs to be executed. Test execution time is measured for each test execution. Outcome of each test may create/update issue in issue tracking systems such as Redmine if required.

Reporting section provides export of all data necessary for Project Requirement Specification (PRS) and Verification Test Plan (VTP) as well as reports of test plan executions. There are two types of test plan execution reports. Report for one test run where summary status of one execution is given, together with following information for each test case: test case name, description, execution result, bug opened in issue tracking system if test is failed, execution comment, duration of test execution, name of the tester/station that executed the test. Another type of report is test plan comparison report where execution status of each test through different test runs can be monitored. This report makes it easier to track project progress and to monitor what features are fixed and what are broken for each software release. It gives summary status of execution results through releases as shown on Fig 3. which facilitates project progress tracking and gives overview of quality and reliability of the device under testing. All results are available through web application or can be exported to different types of documents.

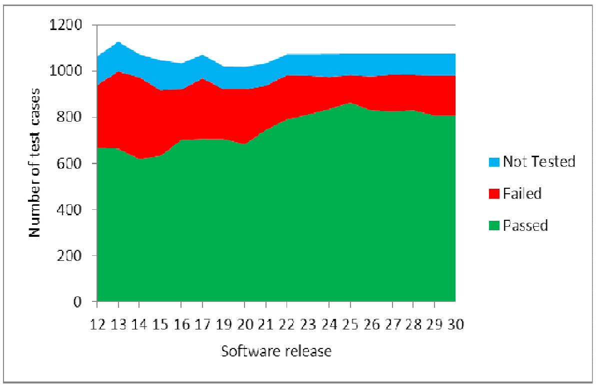

Figure 3. Test execution results

IV. Module for automatic test execution

Executor is stand-alone application used to run automatic tests written in Python scripting language. Application uses test plans made in web application to execute tests. Application uses SOAP web service exposed by server to retrieve test plans and execution resources, as well as to submit execution results.

Black Box Testing (BBT) approach is used for automatic verification of digital TV receivers [5] that is described in this paper. TV receiver is considered as black box with unknown software/hardware architecture. Based on the test inputs, TV receiver produces outputs which are compared to the expected (referent) outputs in order to determine if STB is correctly functioning.

Executor application is able to control a large set of hardware and software modules needed for testing. Minimal set of hardware modules (controlled by Executor) required for testing of Android based digital TV receiver (Fig 4.) consists of:

1) *PC with Executor application*
2) *DVB-T, DVB-C and DVB-S stream modulators*
3) *Realtime audio video capturing device*
4) *Remote controller emulator*
5) *Power switch devices*

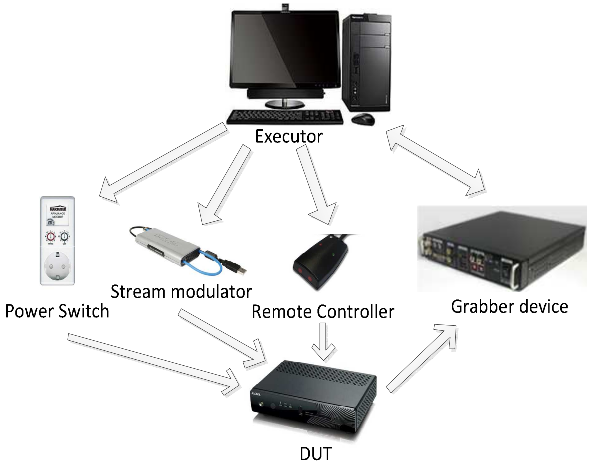

Figure 4. System for automatic testing

DVB-T, DVB-C and DVB-S stream modulators are used as signal generators which provide known input signals. Real-time audio/video capturing device [6] is used for image, video and audio capturing from different output interfaces which receiver might have (HDMI, S/PDIF, CVBS, SCART, COMPONENT, etc.). Device communicates with PC through network interface and transfers grabbed data to PC for further analysis. Remote controller emulator [7] is used to bring the Device Under Testing (DUT) to desired state. Power switch devices are intended for electrical power off/on of the DUT.

Minimal set of software modules (controlled by Executor) required for testing of Android based digital TV receiver consists of:

1) *Picture Block Compare module*
2) *Optical Character Recognition module*
3) *Video Quality Assessment module*
4) *Audio Quality Assessment module*
5) *Audio Compare module*
6) *Android Debug Bridge module*

Picture Block Compare algorithm [8] is used to compare picture captured from DUT output to referent picture. Result of algorithm is percent of matching of two pictures. It supports comparing of full pictures or only regions of the interest. It is commonly used for graphical user interface testing in Android applications, or for assess of image quality in digital television systems [9].

Optical Character Recognition module is used for extracting text from captured images. It is suitable in cases when we need to confirm that certain text appears on the image. It is commonly used for checking if EPG data is displayed correctly, if service list contains correct data, if UI is correctly displayed, if Teletext and Subtitles data are displayed correctly, etc.

Video Quality Assessment module contains a set of video analysis algorithms for analyzing video content in real time. Some of the available algorithms: blocking detection, freezing detection [10], black screen detection, packet loss, blurring, ringing. It is used for video decoding testing, video quality assessment of live and multimedia playback, testing of PVR, VoD or YouTube video playback.

Audio Quality Assessment module contains a set of audio analysis algorithms for analyzing audio content in real time. Some of the available algorithms: audio absence detection, audio level measurement, audio discontinuities (click), audio clipping. It is used for audio decoding testing, audio quality assessment of live and multimedia playback, PVR, VoD, YouTube audio testing.

Audio compare module allows comparison between two audio streams. Actual audio output of the DUT recorded with real-time audio/video capturing device is compared with the referent (expected) output. Signals are compared in frequency domain. It is used for audio decoding testing when expected audio output is known.

Android Debug Bridge (adb) module [11] is used for controlling of adb tool that is available on Android TV receivers. Adb provides many functions that can be used during testing of Android based STB. Some of the functions are: obtaining log data from DUT which makes debugging process easier, performing of factory reset procedure on DUT, checking if DUT has internet connection, performing reboot of DUT, sending key events to DUT, starting of different applications that need to be tested, testing of functionalities such as video, audio and image decoding by starting appropriate application (*VideoView*, *ImageView*) and checking displayed content, pulling the desired files from the board for easier debugging, taking snapshots of user interface , etc.

V. CONCLUSION

This paper proposes system for automatic testing of Android based digital TV receivers using black box testing approach. Solution covers entire process of digital TV receiver testing – requirement definition, verification test plan creation, automatic test cases execution and report generation.

Presented system for automatic testing can be used for functional testing, regression testing, performance testing and stress testing of TV receivers. It improves accuracy, saves time and human effort in comparison to manual testing. It also enables automatic execution of tests that cannot be executed manually (audio/video synchronization tests, testing of playback trick modes such as fast forward and rewind, frame drops detection, measurement of GUI response).

For each executed test case system provides its execution status, captured images, recorded video and audio data and log data obtained during test execution. For failed test cases notification which test step failed and how to reproduce bug is available together with bug opened in issue tracking system. System facilitates project progress tracking and gives overview of quality and reliability of the device under testing – which is used for assessment when is DUT ready for market.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under Grant TR 32029.

REFERENCES

[1] Y. Wu, S. J. Hirakawa, and U. H. Reimers, "Overview of Digital Television Development Worldwide", Proceedings of the IEEE, Vol. 94, No.1, Jan. 2006, pp.8 - 21.

[2] M. Vidakovic, N. Teslic, T. Maruna, and V. Mihic, “Android4TV: a proposition for integration of DTV in Android devices”, 30th IEEE International Conference on Consumer Electronics (ICCE), January 2012.

[3] N. Kuzmanovic, T. Maruna, M. Savic,G. Miljkovic, and D. Isailovic, “Google's Android as an application environment for DTV decoder system”, 14th IEEE International Symposium on Consumer Electronics (ISCE), June 2010, pp. 1-5.

[4] Vukota Pekovic, Nikola Teslic, Ivan Resetar, Tarkan Tekcan, “Test Management and Test Execution System for Automated Verification of Digital Television Systems”, 2010 IEEE 14th International Symposium on Consumer Electronics.

[5] Dusica Marijan, Vladimir Zlokolica, Nikola Teslic, Vukota Pekovic, Tarkan Teckan, “Automatic Functional TV Set Failure Detection System”, IEEE Transactions on Consumer Electronics, Vol. 56, No. 1, FEBRUARY 2010.

[6] “Network Attached Video Capturing device – RT-AV100”, http://bbt.rt-rk.com/en/hardware/rt-av100

[7] V. Marinkovic, I. Kastelan, N. Vranic, Z. Marceta, V. Pekovic, “Protocol independent remote controller emulator for automated testing of multiple DTV/STB”, TELFOR 2012, pp. 1064 - 1067..

[8] D. Marijan, V. Zlokolica, N. Teslic and V. Pekovic, “Quality Assessment Of Digital Television Picture Based On Local Feature Matching”, Digital Signal Processing, 2009 16th International Conference on Volume , Issue , 5-7 July 2009 Page(s):1 – 6

[9] Vukota Pekovic, Nemanja Lukic, Ivan Kastelan, Nikola Teslic, “Platform For STB Video Output Integrity Verification Based On Full Reference Picture Comparison”, 2011 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)

[10] Vladimir Zlokolica, Vukota Pekovic, Nikola Teslic, Tarkan Tekcan Miodrag Temerinac, Video Freezing Detection System for End-user Devices, 2011 IEEE International Conference on Consumer Electronics (ICCE)

[11] Android Debug Bridge,http://developer.android.com/tools/help/adb.html