Overview of the HbbTV compliant browser upgrade on Android based DTV platform

Milena Milosevic, Krsto Lazic, Branimir Kovacevic, Nenad Jovanovic, Marko Kovacevic
Department of Computer Engineering and Communications
Faculty of Technical Sciences
Novi Sad, Serbia

***Abstract*—Upgrade to the new version of software is resource and time consuming process. Manufacturers try to optimize this process in any way. This paper presents upgrade of the HbbTV software solution as a system which relies on web technologies. Since web browser is one of the platform dependent components of the HbbTV software solution, the paper focuses on the web browser's update on Android based DTV platform. With the Android upgrade, web browser also changes. These changes can be major changes and to achieve faster releases and better performances, manufacturers have to choose which browser to choose, existing or the upgraded one. Two default Android web browsers are analyzed and compared. Web browsers' overview of supported features and performances related to the HbbTV is given.**

Keywords-digital television; HbbTV; Android; web browser

I. INTRODUCTION

Demand for digital entertainment is one of the fastest growing in the CE (Consumer Electronics) market. Compared to mobile devices, tablets and PCs, where user experience is interactive, traditional television offers static user experience. Nowadays, same user interface is expected on all connected devices, including DTV (digital television) devices (TVs and STBs (set-top boxes)). More interactions and social features are requested in addition to watching TV, and thus DTV devices are moving forward from static to interactive user experience. In the fast growing DTV market there are variety of devices, technologies and standards offering interactivity and entertainment.

Hybrid Broadcast-Broadband TV (HbbTV) standard [1] provides integration of broadcast DVB (Digital Video Broadcasting) services and broadband IP network. The integration provides better user experience through services like digital teletext, EPG (Electronic Program Guide), and program non-related services, such as video on demand, catch-up TV, interactive ads, voting, personalization, games or social networking.

The traditional TV software had small significant updates over time. With the appearance of the Smart and Connected TVs, TV market rapidly changed [2], with constant need for upgrades and new outcomes. For example, Android based DTV platform needs an upgrade each time new Android major version is available.

Software solution of DTV architecture is complex. Its integration with new Android version and upgrades are time and resource consuming processes for CE manufacturers.

The mentioned changes in DTV software development influenced emerges of traditional TV software solutions with rapid development model. Rapid software development model is also a choice in web browser development. Google's Chrome applied this model from the beginning, while Mozilla Firefox transitioned to this model [3] [4].

This paper presents and analyzes upgrade to the new version of the Android platform through upgrade of the HbbTV software solution as a complex software system which relies on web technologies. HbbTV applications which have become widespread are executed in the web browser. Web browsers are customized to support HbbTV standard requirements. There is more than one web browser on which HbbTV compliant web browser can be based. There are open source, proprietary and hybrid browsers (half open source, half proprietary). Some browsers have support for different operating systems.

Android platform has its default web browser. This work focuses on HbbTV solution based on the default web browser. With every new Android release, update of default web browser is present. Updates can be minor and customization of that browser for HbbTV purposes goes smooth, but sometimes updates lead to some major changes. One of such updates with major changes is transition from WebKit-based web browser to Chromium-based web browser which came about in Android release 4.4.x. These changes impact on the porting of the HbbTV solution and thus impact on the whole DTV solution upgrade in terms of time and resources.

Question is how to achieve rapid release development from one to another version, now and in the future? We propose choosing one web browser and making it HbbTV dedicated. Two Android default browsers will be discussed and analyzed as HbbTV candidate browsers: WebKit-based browser on which existing HbbTV solution relies, and Chromium-based browser which is going to have greater support in the future. Advantages and disadvantages of both solutions are presented.

The rest of this paper is organized as follows. Section 2 gives related work. HbbTV overview and features that should be supported by web browser are given in third section, while section 4 gives comparison of these two browsers regarding HbbTV requirements. Last section concludes the paper.

This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under Grant 32034.

II. Related Work

Modern web browsers are compared by many criteria. Anand and Saxena [5] compare Chrome, Firefox and Internet Explorer in terms of CPU utilization, disk usage, multiple tab performances, compatibility with modern web technologies (HTML5 and CSS3) and execution time of JavaScript engines. Nielson, Williamson and Arlitt [6] give overview of the benchmark performance test results for four browsers (Internet Explorer, Safari, Firefox and Opera). Their focus is mostly on JavaScript, rendering and AJAX performances. They also emphasize the choice of the operating system which may affect the test results. All authors agree that there is no 'one browser for all'; which browser is better depends on its use and purpose.

Web browser is HbbTV compliant if it supports some predefined features. Features which need to be implemented on the Java based devices in order to support HbbTV functionality are described in [7]. Some of these features are web browser related: support for spatial navigation, additional DOM events, support for JavaScript plugins required by the HbbTV 1.1.1 standard [1] etc. More specific overview of these and additional changes necessary for the implementation of the HbbTV required functionality on the Android based DTV devices is given in [8]. HbbTV 1.2.1 standard [9] requires support for one more JavaScript object (application/oipfSearchManager). One implementation of this embedded object is described in [10], with accent on the search performances. HbbTV 2.0 specification is not yet published and should offer a solution for synchronized IP and DVB reception, companion screen feature and transition to some of the HTML5 features as new widely adopted technology. Technology development of HBB-Next project and its contribution to HbbTV 2.0 (support for second screen, media synchronization and user identification) are presented in [11].

III. HbbTV Browser Requirements

HbbTV standard is based on the existing standards and web technologies (Fig.1). It further profiles them and introduces few novel solutions.

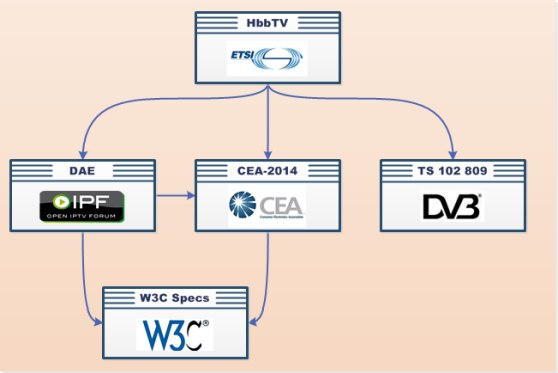

Figure 1. HbbTV and existing standards overview

CEA-2014-A (also known as Web4CE) [12] refers to the problem of displaying web pages on the DTV devices. It defines CE-HTML (Consumer Electronics HTML) language which is based on the following web languages:

- XHTML 1.0 transitional/strict,
- Ecmascript 262, 3rd edition,
- DOM Level 2.0,
- CSS TV Profile 1.0,
- XMLHttpRequest.

CE-HTML also specifies some additional features. HbbTV standard addresses on the following additional features: support for text input using remote control (Multi-Tap), definition for key codes which are sent to the applications (solves diversity codes problem on the DTV devices) and MIME type for CE-HTML applications (application/ce-html+xml).

CEA-2014-A, along with the CE-HTML, also defines embedding of non-linear audio/video content in an application (using JavaScript plugin). It specifies DOM event handling (for example key events) and still image formats (JPEG, GIF, and PNG).

DAE (Declarative Application Environment) [13] defines embedding of linear audio/video content in an application and specifies JavaScript API for applications running in a DTV environment. It specifies navigation using CSS3 directional focus navigation while the nav-up, nav-down, nav-left, nav-right properties are used by the applications.

ETSI-TS 102 809 [14] defines DTV transport of information about applications and transport of applications via DSM-CC standard. HbbTV browser has to support DVB URLs.

Along with the existing technologies HbbTV standard defines the application lifecycle and gives some recommendations. HbbTV browser should have cookie support and should not offer a history UI (user interface) for HbbTV applications.

HbbTV 2.0 specification announces the use of HTML5 features. HTML5 standard is not yet defined, but is being widely adopted technology by all major browsers. Main features which HTML5 brings are:

- abstracting the content from the hardware (audio, video and other components),
- WebSocket – an asynchronous protocol providing full-duplex communication over a single TCP connection. Server can send messages to the client at any moment,
- WebRTC – enables real-time communication, for example: voice call, video chat and P2P file sharing without plugins,
- localized storage.

Data exchange between application and DTV middleware in the existing solution is realized via JavaScript plugins (HbbTV embedded objects). Plugins are implemented using NPAPI (Netscape Plugin Application Programming Interface) [15] as a cross-platform plugin architecture. NPAPI supports initializing, creating, destroying and positioning plugin content,

scripting, printing, full-screen plugins, windowless plugins and content streaming. For HbbTV embedded objects significant value in browsers has scripting support, asynchronous callback support and object array support.

IV. Comparison of Web Browsers

Android version 4.4.x introduced Chromium-based default web browser. Android web browser in older versions has been based on the WebKit. Accordingly, HbbTV software solution in versions older than this has been based on the WebKit engine. With the new Android upgrade and emerge of Chromium-based browser the question is which web browser provides good performances and cost-effective updates. Two solutions arise: preserving WebKit-based browser and its porting of native parts on the upgraded Android version or switching HbbTV solution to the Chromium-based browser (making it HbbTV compliant).

WebKit-based browser is HbbTV compliant, as part of the existing HbbTV solution. Support for CE-HTML, DOM events handling, OIPF MIME types, CSS 3.0 navigation properties and DOM extension with HbbTV required objects (KeyEvent, KeySet and Channel) has been implemented. HbbTV JavaScript object for communication with DTV middleware are implemented using NPAPI. WebKit-based browser supports NPAPI architecture by default. It is dependent on native Android APIs. With the upgrade of Android these APIs has changed. Transition of HbbTV solution relying on WebKit based browser to Android 4.4.x requires significant effort on porting this browser's native part.

Chromium-based browser, compared to WebKit-based browser, is already integrated with the native APIs on the upgraded Android version. This default browser is not HbbTV compliant. HbbTV browser related features such as DOM object extension with new objects, CE-HTML parsing, support for OIPF MIME types, CSS 3.0 navigation properties and specific handling of DOM events need to be implemented. Cookie support exists but there is no support for DVB URLs.

In the default Chromium-based browser plugins are disabled. HbbTV requires plugins' support, so Chromium-based browser needs to be reconfigured to support plugins. Plugin support is exposed through PPAPI (Pepper Plugin API) [16]. NPAPI in this browser is deprecated and will be excluded in the future. HbbTV JavaScript plugins are implemented using NPAPI. Since PPAPI doesn't provide API for scripting objects, plugins can't be rewritten using PPAPI. Instead, NPAPI support has to be revived in the Chromium-based browser.

Main question is which solution leads to faster HbbTV release. WebKit-based browser is already HbbTV compliant and needs be ported on modified native Android API. Chromium-based browser runs on Android 4.4.x and newer versions as default browser, but needs to be modified to be HbbTV compliant. Along with development effort, web browser performances and support for HbbTV relevant and next generation features are important. Authors of the [5] conclude that Chromium's weak point is multi-tabbing. This feature is irrelevant for HbbTV browser; standard specifies only one active application at the moment.

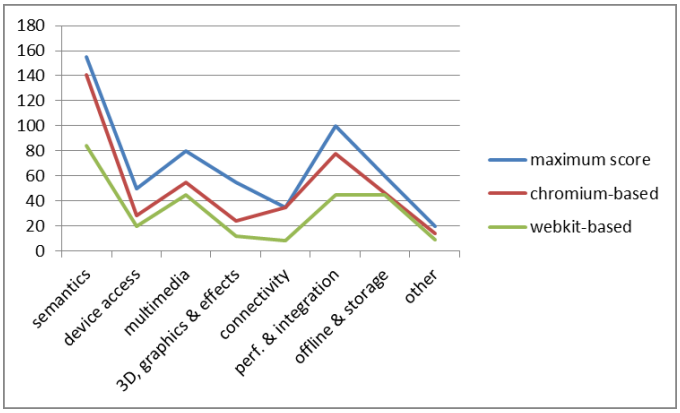

Figure 2. HTML5 support test results

HbbTV 2.0 introduces some of the next-generation features. HTML5 support is one of them. Tests for HTML5 browser support already exist. They test currently specified features by the HTML5 specification and tests change accordingly to the specification. One such test [17] is executed on both browsers. This test validates support for HTML5 semantics, device access, performance and integration, multimedia, 3D graphics and effects, connectivity, offline and storage and other. Some non-HTML5 features are also validated but are not included into final result. Results for these two browsers are depicted in the Fig. 2. They show how many scores browser gets out of maximum.

Results depicted in Fig.2 indicate that Chromium-based browser has better overall HTML5 support than the WebKit-based.

HbbTV applications communicate with the DTV middleware and achieve its functionalities via JavaScript plugins. Browser's JavaScript engine performances influence the performances of the HbbTV applications and thus on the user experience. The benchmarks measure performances of the JavaScript engine.

SunSpider [18] as JavaScript benchmark validates the core JavaScript language. It focuses on the actual JavaScript developers' use today trying to avoid microbenchmarks. Tests are executed multiple times to determine measurement error range. Results (Fig. 3) are given in milliseconds.

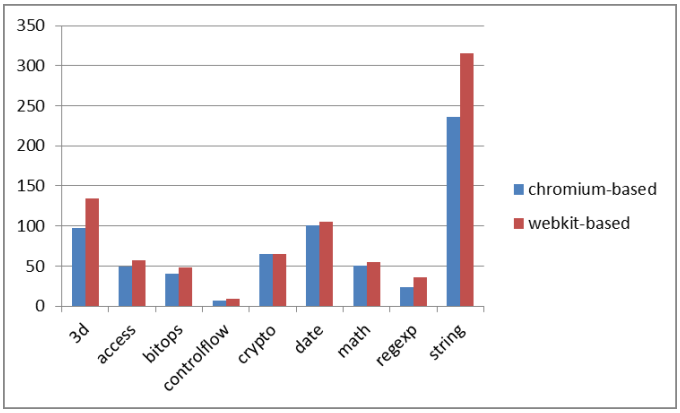

Figure 3. SunSpider benchmark test results

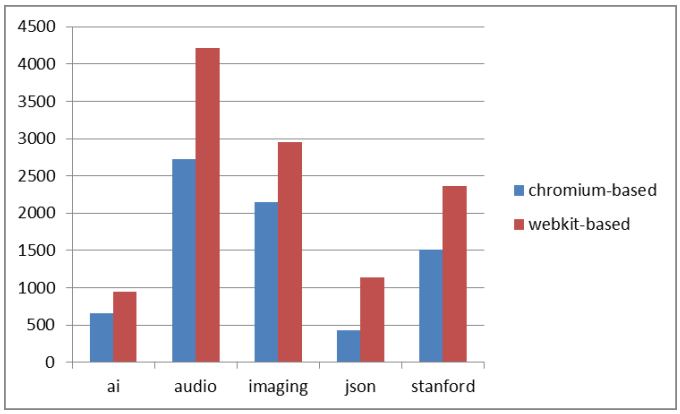

Figure 4. Kraken benchmarks test results

Kraken [19] is JavaScript performance benchmark which measures the speed of several use cases from real-world applications and libraries. It is based on SunSpider. It includes an implementation of the search algorithm, audio processing usage, image filtering routines, JSON parsing and cryptographic routines. Results (Fig. 4) are also reported in milliseconds.

V8 Benchmark Suite [20] reflects pure JavaScript performance. Results are shown in Fig. 5 and higher scores mean better performances.

All tests are executed on the reference platform and are here to give mutual performance overview of these two browsers. From the test results Chromium-based browser is considered as browser with better JavaScript performances, but in the real world it is hard to see JavaScript performance difference between these browsers. Overall user experience considering speed is satisfying for average user.

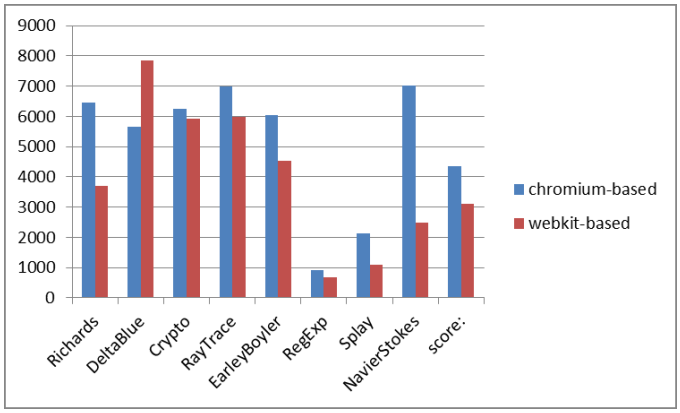

Figure 5. V8 Benchmark Suite results

V. CONCLUSIONS

This paper analyzes problems of the HbbTV browser upgrade which comes with the Android platform upgrade. Overview of the HbbTV compliant browsers based on two major Android web browsers is given. HbbTV required features and browser support for them is presented. The work focuses also on the support for next-generation features such as HTML5 and gives comparison of the browsers' JavaScript performances. To achieve faster HbbTV software releases next step would be choosing one web browser, making it as much as possible platform independent and its use in HbbTV solution across all Android versions.

REFERENCES

[1] ETSI TS 102 796: Hybrid Broadcast Broadband TV; V1.1.1 .

[2] IDATE: "TV 2010, Markets & trends, facts & figures".

[3] O. Baysal, I. Davis, and M. W. Godfrey, "A tale of two browsers," Proceeding of the 8th working conference on Mining software repositories , ser. MSR '11, 2011, pp. 238–241.

[4] F. Khomh, T. Dhaliwal, Y. Zou, and B. Adams, "Do faster releases improve software quality? An empirical case study of Mozilla Firefox," Proceedings of the international working conference on Mining software repositories, Jun 2012.

[5] V. Anand, D. Saxena, "Comparative study of modern Web browsers based on their performance and evolution ," Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on, Dec. 2013.

[6] J. Nielson, C. Williamson and M. Arlitt , "Benchmarking Modern Web Browsers ", Proceedings of IEEE HotWeb 2008, October 2008.

[7] N. Kuzmanovic, V. Mihic, T. Maruna, M. Vidakovic and N. Teslic, "Hybrid broadcast broadband TV implementation in java based applications on digital TV devices", IEEE Trans. Consumer Electronics, 2012, pp.1056-1062.

[8] M. Milosevic, S. Markovic, B. Mlikota, M.Zivanovic and B.Prtvar, "Porting of HbbTV functionality on DTV platform based on Android OS" , TELFOR 2012.

[9] ETSI TS 102 796 : Hybrid Broadcast Broadband TV; V1.2.1.

[10] M.Milosevic, B.Mlikota, M.Radonjic and T. Maruna, "HbbTV 1.5 additions: one design, implementation and integration " , ICCE Berlin 2014, in press.

[11] M.O van Deventer, J.J de Wit, M. Guelbahar, B. Cheng, F. Gomez Marmol, C. Köbel, C. Köhnen, G. Rozinaj, and B. Stockleben: "Towards next generation Hybrid broadcast broadband, results from FP7 and HbbTV 2.0", IBC2013 Conference, Amsterdam, Netherlands , 2013.

[12] CEA-2014 revision A: Web-based Protocol and Framework for Remote User Interface on UPnP™ Networks and the Internet (Web4CE) .

[13] Open IPTV Forum Release 1 specification, volume 5: Declarative Application Environment v1.1.

[14] ETSI TS 102 809: Digital Video Broadcasting (DVB); Signaling and carriage of interactive applications and services in hybrid broadcast - broadband environments; V 1.1.1.

[15] NPAPI documentation : Gecko Plugin API Reference available at https://developer.mozilla.org/en/Gecko_Plugin_API_Reference, accessed 5th Sept..2014.

[16] Pepper Plugin API documentation: available at https://code.google.com/p/ppapi/, accesse 5.9.2014.

[17] HTML5 Test : available at https://html5test.com/, accessed 5th Sept. 2014.

[18] SunSpider 1.0.2 JavaScript Benchmark: available at https://www.webkit.org/perf/sunspider-1.0.2/sunspider-1.0.2/driver.html , accessed 5th Sept. 2014.

[19] Kraken JavaScript Benchmark (version 1.1) : available at http://krakenbenchmark.mozilla.org/kraken-1.1/driver.html, accessed 5th Sept. 2014.

[20] V8 Benchmark Suite - version 7: available at http://v8.googlecode.com/svn/data/benchmarks/v7/run.html, accessed 5th Sept. 2014.