

HTML5 and Companion Web Technologies as a Universal Platform for Interactive Internet TV Advertising

Emmanouil Perakakis
Dept of Commerce and Marketing
TEI of Crete
Ierapetra, Greece
mperakakis@staff.teicrete.gr

Alina Levinenko
Dept of Commerce and Marketing
TEI of Crete
Ierapetra, Greece

Georghita Ghinea
School of Information Systems,
Computing and Informatics
Brunel University
Uxbridge, UK
george.ghinea@brunel.ac.uk

Abstract— In this paper, we propose, design and develop a prototype interactive TV Commercial (iTVC) for the Internet Connected TV using only web technologies instead of propriety APIs or Adobe Flash, which are the most common methods up to date. The main advantage of this method is that developers can use a technology that is, or will be in the near future, compatible with most, if not all, different connected TV platforms, as the adoption of HTML5, CSS3 and JavaScript is constantly gaining ground among Connected TV manufactures. We will show that, using these technologies, all of the features of a typical interactive Ad (Advertisement), such as navigation, video, audio, animation etc can be easily achieved and therefore it is a much better choice for authoring universal interactive Advertisements that will work in most platforms instead of re-developing the same Ad for each platform.

Keywords- *Interactive TV Advertising; TV Commercials; HTML5; CSS; JavaScript; Connected TV; Smart TV; Cross-platform*

I. INTRODUCTION

Interactive Advertising is one of the most interesting new features of Interactive TVs, especially for professionals in the marketing business. This new platform offers a variety of new possibilities for the marketing of a product which were never possible before on traditional TVs, such as instant purchase (T-Commerce), detailed on-demand product descriptions, newsletter subscriptions, social media interaction, longer presentations and games, are a few example, that offer much more user engagement than Traditional TV Commercials (TVCs).

Developing Interactive Advertisements for the Connected TV currently is neither easy nor straight-forward. This should not come as a surprise since this technology is very new and there is a plethora of platforms struggling to gain market share over an extremely heterogeneous area. Google TV, Samsung Smart Hub, LG Smart TV, Apple TV, Nintendo Wii (Opera), Sony PS3 are just a few of these platforms that use very different technologies to offer the Connected TV internet experience. In order to develop an application (App) or an Interactive Advertisement for each of these platforms the developer must use dedicated APIs for each device. Since there are no market leading platforms yet (i.e. adopted by most

users), and also no standards have been set, it can be an expensive and daunting task to re-develop the same iTVC for each platform using different APIs and even Programming Languages.

However, recently, there is a clear movement towards the adoption of the familiar standard web technologies of HTML, CSS and JavaScript for every Connected TV device either through their web browsers or even in their application development core. As W3C states: *There is a strong growth in the deployment of devices that integrate regular Web technologies such as HTML, CSS, and SVG, coupled with various device APIs* [1]. By adopting these technologies into the TV area, a new universal method for developing Apps and iTVCs seems closer than ever. The purpose of this paper is to research whether these technologies are suitable and can offer all the required features needed for developing complete Interactive Ads compatible with most TV platforms.

II. THE CONNECTED TV PLATFORMS

Interactive Internet TV systems represent an extremely heterogeneous area, very different from the other platforms that have experienced rapid evolution recently, such as smart phones. While smart phones, have 5-6 main different platforms, with clear market leaders of Google Android OS and Apple iOS, followed by RIM's Blackberry and recently Windows Phone, in the TV area things are quite different. Currently, internet TV platforms are being developed by service providers (Hulu, Netflix etc), traditional TV manufacturers (LG Smart TV, Samsung Smart Hub, Philips SmartTV), internet service companies (Yahoo TV, Google TV), computer manufacturers (Apple TV), PC software developers (Boxee, Kylo.tv, Windows MCE), TV channels (HbbTV), set-top box manufacturers (Roku, TiVo), Media Player manufacturers (Western Digital, Iomega) and even game console manufacturers (Sony PS3, Microsoft XBOX 360, Nintendo Wii).

A. Development Tools and Browsers

Most of these platforms have APIs for developing platform specific applications. To go through each and every of these platforms would be a research in itself, but on Table 1 there is a

small but representative sample with common features of some of the internet TV platforms as an example of the diversity between them. It is important to note about Flash support that Adobe itself has stated that it is not recommending its use on the browsers of TV devices although it does support it [2].

B. Remote Controls

Since the iTV platforms are interactive, there must always be some device that is used to control the device and the content. In this area, there is again an absence of standards and each device uses quite different remote controls, which vary from PC-like full-qwerty keyboards to very simple TV-like remotes, and also, game controllers in the case of game-console devices. It is also important to note that most platforms have available Mobile Apps to fully control them through a Smartphone.

TABLE I. SAMPLE OF TV PLATFORMS & WEB STANDARD COMPATIBILITY

| | Google TV | Samsung smart HUB | Boxee | LG Smart TV | Nintendo Wii |
|-----------------|-----------------------|-------------------|------------------|------------------|-------------------|
| Availability | US | 120 countries | World-wide | 80 countries | World-wide |
| App Development | JAVA/HTML/CSS/J S API | HTML5/ XML/JS API | HTML5/ JS | Adobe AIR/ Flash | C/C++ |
| HTML | FULL HTML5 | FULL HTML5 | Limited HTML5 | HTML4.0 1 | Limited HTML5 |
| CSS | CSS3 | CSS2, CSS TV | Partly CSS3 | CSS3, CSS TV | CSS2 |
| JS | YES | 1.8 | YES | 1.6 | YES |
| Flash | YES | YES | YES | No | Yes |
| Remote Control | Touchpad/ Keyboard | TV Remote | Remote with Kbrd | TV Remote | Motion Controller |

III. INTERACTIVE TV ADVERTISEMENTS

It has been now 70 years since the very first TV commercial appeared in between a baseball match in the United States displaying a simple static image and some text: “America Runs on Bulova Time” [3]. The evolution of this initial idea has come to the common 30-second TVCs a viewer experiences every day on the TV. More visually and emotionally appealing than ever, TV commercials however seem to struggle to find their way within the context of new technologies, such as interactive digital TVs, DVRs and Internet Connected TVs.

Interactive TV Ads though, are much newer than traditional TVCs. These initially appeared on non-internet interactive TVs and, lately, on Internet Connected TVs. The main difference from the 30-sec traditional TVCs is that they add the element of interactivity with the user/viewer as opposed to the passive watching of a Video Commercial.

Although new on TV, Interactive Ads have been around on the computer platforms for a while, mainly on the web. A

typical web interactive ad starts with a Banner on a website which the user can click (Entry Point), if he chooses to, and be transferred onto the advertiser’s website, which is called the Landing Page. The Landing Page can be either a web page which is part of the advertiser’s website, or, an independent micro-site. *A Micro-site is a multi-page ad accessed via click-through from initial ad [...] but has access to more information from the advertiser than a display ad allows* [4]. The micro-site is solely dedicated to the product being advertising, as opposed to the general advertiser’s website which possibly has more products on display. Interactive Ads are also present on Mobile Platforms.

These are very similar to the PCs but can also be found inside Apps, apart from the web browser (e.g. Apple’s HTML5 based iAd), and probably have even more in common with Interactive TV Ads.

A. Features of Interactive TV Ads

Similarly to the Interactive Ads found on the PC web-browsers and Smart Phones, an iTV ad also has two main parts:

i) *The Entry Point*: This can be (a) a graphical banner on a website being visited on the TV web browser, (b) a banner (static, animated or video) inside a TV App or (c) a TVC which looks like a typical 30-sec spot and is displayed to the viewer during a commercial break, before a film begins, during “pause” etc; This TVC an action, that if the users chooses to take (e.g. “press the X button on your remote for more”), will continue to the part (ii).

ii) *The Landing Page/Micro-site*: This is very similar to a typical TV App and is the main advantage that interactive TV ads have over the traditional Ads. On the product mini-site the user will be able for example to further explore its features, be more engaged with it and even purchase it directly (t-commerce).

By studying a number of interactive TV ads [5][6][7] these are the features that a typical TV Micro-site can have:

- It is intensively branded and graphically heavy
- Is visually interesting with animation and graphics
- Has a music soundtrack and other audio (e.g. narration)
- Can include a game or a contest
- Can include a signup form (e.g. for newsletter)
- Includes textual information about the product
- Has a small menu limited to 3-6 options
- Contains Social Engaging elements (Likeboxes etc)
- Can include videos such as the TV ad of the product or a video feature tour
- Has extensive information about the product
- Can include t-commerce option (e.g. order online)
- Allows user interaction/navigation

IV. PROBLEM AND PURPOSED SOLUTION

As there are no universally accepted specifications for developing Interactive Advertisements for TV and, also, no single Connected TV platform has managed to gain considerable market share it is currently very difficult to design and develop iTVCs compatible with all platforms. Ads have to be redesigned for each platform, requiring time, knowledge and money in order for the Ad to have greater exposure to a wider audience.

In our work we **used the latest web technologies of HTML5, CSS3 and JavaScript to create a single Interactive Ad to target all different platforms**. Additionally, to make it more compatible regardless the limitations of support for some platforms (e.g. for HTML5), we used a JavaScript detection of features and provide a “fallback strategy” for the missing features.

The advantages of this solution are:

- Universal cross-platform compatibility.
- Adaption for different input devices (remote controls) can be programmed.
- All the features of current TV Ad platforms (discussed in section III.A) can be accurately reproduced with HTML5/CSS3/JS.
- HTML5 has now native video support, a vital feature for a TV Ad [8].
- Personalization, a very interesting and effective feature of iTV Ads [8] can be added.
- It does not require any platform-specific technical Skills: Developers don't have to learn new languages/technologies.
- Even more features can be added, such as personalization of ads, multi-lingual, mash-ups (e.g maps, social plugins) etc.
- More control of the Ad can be on the advertiser's side since the ad can reside on the advertiser's website.
- Easily setup Analytics for the Ads, as these will reside on a web server and can use normal Web Analytics services.

V. INTERACTIVE ADS USING WEB TECHNOLOGIES (HTML5 / CSS3 / JS)

A. Pre-HTML5

Maybe, it would be interesting to ponder why HTML and web technologies have not been adapted until now for interactive TV ads and what has changed that enables a move towards this direction. This is easy to answer as HTML and related technologies on previous versions (HTML 4.01, CSS2) lacked some vital features that are essential for the TV experience. To begin with, there was no native support for video, audio or animation/drawing. Even text was very limited to a very small number of fonts while text effects, such as shadows, or text transformations were not possible. JavaScript

had also a number of issues, as it behaved quite differently on different systems, making it virtually impossible for a cross-platform experience using the same code. Another thing to notice is that the available hardware on TVs and set-top boxes was not powerful enough to support the aforementioned features. But now, web technologies such as HTML5 allow for much richer, more TV friendly web content, such as video, that can be embedded and controlled much more easily and without the need for any third-party components [8].

B. The Web Technologies

Although, mainly for marketing purposes and as advertised by Apple's Steve Jobs, HTML5 has become the only strong buzzword to include these new web technology features. So, when a non-technical person refers to HTML5, it actually means the combination of three main technologies: HTML5 for structure, CSS3 for presentation/style and JavaScript for Interactivity/Animation. None of the above web technologies is of much use as stand-alone but by combining the three of them, it is possible to create very sophisticated pieces of software.

C. The new features of HTML5 (CANVAS, VIDEO, AUDIO etc) and how these can be utilized to design interactive TV Ads

It is important to note that HTML5 is not a finished Recommendation and is not expected to be completed before 2014 [1]. However, as the industry progresses really fast, it has already been implemented in most major browsers, and is currently being adopted by connected TV devices as well. The official latest Editor's Draft version is published online from W3C [10].

- *Video*: The new <video> tag is probably the most important new feature of HTML5 for the TV. It is possible using this single tag to embed a video onto the page. There is however still an issue with this as there are currently two different video formats which are supported on different browsers: (a) H.264 Baseline profile in an MP4 container and (b) VP8 in a WebM container, or Theora in an Ogg container. This is a patent issue but the workaround for cross-platform support is to include both video versions. It is also noteworthy the preload attribute for pre-loading videos allowing a smoother playback.
- *Subtitles*: Complementary to video on TV, it is very common to have subtitles, either on different languages, or for accessibility or even as part of the creative (e.g. captions with only music as soundtrack). This is now possible using the WebM format and the <track> element. The track element allows authors to specify explicit external timed text tracks for media elements [10].
- *Audio*: The <audio> tag is similar to video but for playing audio. It is another very important feature that was lacking in prior versions of HTML. Where, on a PC website, it would be possible to be silent, this is not the case on TV. Most TV content has some kind of soundtrack, in the form of music, narration or sounds. Audio used to be reproduced on the web using flash or other 3rd party plug-ins but now it's embedded into native HTML5. Synchronizing audio with graphics, video and interactions

- (e.g. push of a button) is also easy, using JavaScript Events or Timers for time-based sync.
- Canvas:* Some argue that the <canvas> tag is the most important new feature of HTML5. It's basically an area in the browser where the developer can draw graphics or produce animation using JavaScript with the provided API. A vital feature for creating real-time graphics, used for animation or game design.
- Interaction Design:* HTML is a markup language so it does not offer much interactivity on its own. However, JavaScript can offer all the needed features to add interactivity to the interactive Ads. Using JavaScript is even more enhanced by the many good-quality libraries, like jQuery, that offer many cross-platform functions for producing the most common functionality with only few lines of code. Some example uses of Javascript in this context are: Animation, Key Press captures, transition effects etc.
- Text / Typography:* Written text, although probably the most common form of communication on the traditional web on a PC, has a lot of differences on TV. For one thing, reading long chunks of text on TV is not very appealing, and can be boring and tiring. Text exists on TV but usually as (a) subtitles to narration or different languages (b) small chunks of text, usually no more than 2-4 lines. The fonts are also quite big, in order to be easily read when sitting far from the screen. Text, although it existed on previous HTML versions, it is enriched quite a lot in the new version. This includes new fonts, beyond the extremely limited 5-6 font with @font-type tag of CSS3 and also with text effects, so common on TV such as shadows and borders (text-shadow), rotation (box-rotate) and even gradients and advanced masks (-gradient). Text can also be animated using JavaScript.
- Graphics Effects (CSS3):* TV is a graphically rich medium, so the new effects of CSS3, such as shadows, rotation, gradients, opacity etc can be very useful for the presentation of content. Although some may argue that this was also possible in the past using pre-rendered graphics, it is not the same with the real-time option. Especially on a dynamic Ad where, for example, a different message might be desired for different people.
- Other Features:* HTML5 and CSS3 features are by no means limited to the above. These are however the most obvious ones to be used for an interactive TV Advertisement. Other features include many new semantic tags for clearer and more specific content structure, and Local Storage for easily storing values even when the browser is closed or refreshed. As the Recommendation is not finalized yet, it is still possible for other features to be added (or removed) before the finalized version.

D. Universal Compatibility

As HTML5 is not a completed framework but a work in progress, not all of its features are supported across all devices. This is a huge problem for creating a TV Ad that is compatible with all devices, as purposed in this paper.

This problem can be addressed using JavaScript to detect compatible tags for the current browser and provide fallback strategies for when a feature is not available on the current device. For example, if the particular device lacks video support using the HTML5 <video> tag, an image could be displayed, or even a flash video, if the device supports it. Since JavaScript is supported on all devices, this technique will ensure that the Interactive Ad will be viewable across all platforms.

- Use of Fallback solution for unsupported features using Modernizer.js:* Since all platforms have their own unique characteristics, browsers, and, hardware it is impossible to share the exact same support for HTML5 features. So, it is important, in order to have a smooth universal experience to use "fallbacks" for features not supported by a device. This is very easy to accomplish using the javascript library "modernizr.js" [12] which can find the features a browser supports and allows the developer to have alternatives for the unsupported ones. So, for example, if the full-screen video feature is not supported, an alternative image can be displayed instead of an empty area.
- Compatibility and performance:* Using this fallback technique it is possible to make the commercial compatible to virtually all platforms. However, due to the major differences in the hardware, it is important to also test the iTV Ad to the actual devices to ensure a smooth user experience, and see its performance. If the performance is not satisfactory, some features of the Ad may be turned off or adjusted to achieve the optimal user experience on each device.

VI. THE PROTOTYPE iTV COMMERCIAL

As a proof of concept, an iTVC was developed, using only web technologies. As mentioned before, the TV Ad had to include two parts: (a) The Entry Point, which was chosen to be a Video similar to the traditional 30-second spots but with the added choice for the viewer to press a button to continue to (b) The Branded Microsite/Landing Page, a website with information about the product where the user can navigate and see its several features.

A. The "30-second" video calling user to action

The first part of the iTVC could be either a clickable banner or a "30-second spot". A 30-second spot was chosen for this prototype but replacing it with a banner is very easy and would not make any difference to the 2nd part.

Technically, the 30-sec spot is a normal HTML5 page that has a full screen video in the background using the new <video> tag, a music soundtrack and some car sound effects using the <audio> tag [9], and, a sequence of text sentences on the foreground which are layered and faded-in and out on top of the video. The fade-in and fade-out effects are easily produced using the jQuery library which animates the CSS3 opacity property [11].

During the spot (which could be played, for example, during a commercial break) the viewer watches part of the TV spot but he/she has the option to press a button on the remote

control in order to continue to the second part. The button can easily be traced using JavaScript onKeyPress events. If the user presses the button the commercial will not stop but it will only hide the text “Press [M] button for more” (as a visual indication that the command has been accepted) and will redirect to another HTML page/file after the end of the 30-second spot: the Landing Page.

B. The Landing Page/micro-site

This is the main interactive part of the iTVC. It is composed of navigation menus, controlled using JavaScript events, and a number of screens for the main page and each menu choice. Technically, it is a single HTML5 page with a number of main layers (DIVs) that are hidden or shown depending on the user’s actions using JavaScript events. Each of these main layers contains all the elements of each screen. A standard menu is displayed on the main screen where the user can choose the product he is interested in.

The background for all the screens is a video sequence of time-lapsed clouds in order to have a continuous sense of motion. It is interesting to note that this feature of background video was not present in any of the commercial Interactive Ads that have been examined in our research which indicated that maybe this was not feasible in these platforms. Since it is a new HTML5 feature it is also possible that it will not render on all Connected TV systems. A fallback strategy was included using the Modernizr.js library Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. for showing static clouds as a background for the unsupported devices. It would also be an alternative to use JavaScript to animate the background or even use Adobe Flash for the platforms that support this.

For each other static graphical element, PNG images were used, in order to include 8-bit transparency that was needed and positioned with CSS on the appropriate positions. As not all TV platforms share the same screen resolution, relative width and height HTML/CSS attributes were used to ensure that the graphics would be scaled accordingly.

The menus can be controlled using the arrows on the remote control, where the KeyPress events are traced with JavaScript and current menu choice is highlighted by changing the CSS properties of box-shadow and background-color.

VII. TESTING ON DIFFERENT PLATFORMS

A small initial number of representative devices were chosen for the tests and, using their web browsers, the web page containing the ad was opened. The systems that were available for testing were: the *Nintendo Wii* (*Opera browser*), a *Windows 7 HTPC connected to TV*, a *Sony Internet TV* and a mini-mac running the *Boxee internet TV software* (also available as a set-top box).

Overall, the system runs smoothly on all the test devices. It worked at its best on the Chrome HTPC since this browser has probably the most HTML5 features up to date and also provides excellent performance. Nintendo Wii, was the device with the most unsupported features, and actually one of the main reasons that it was chosen for this experiment: to test the

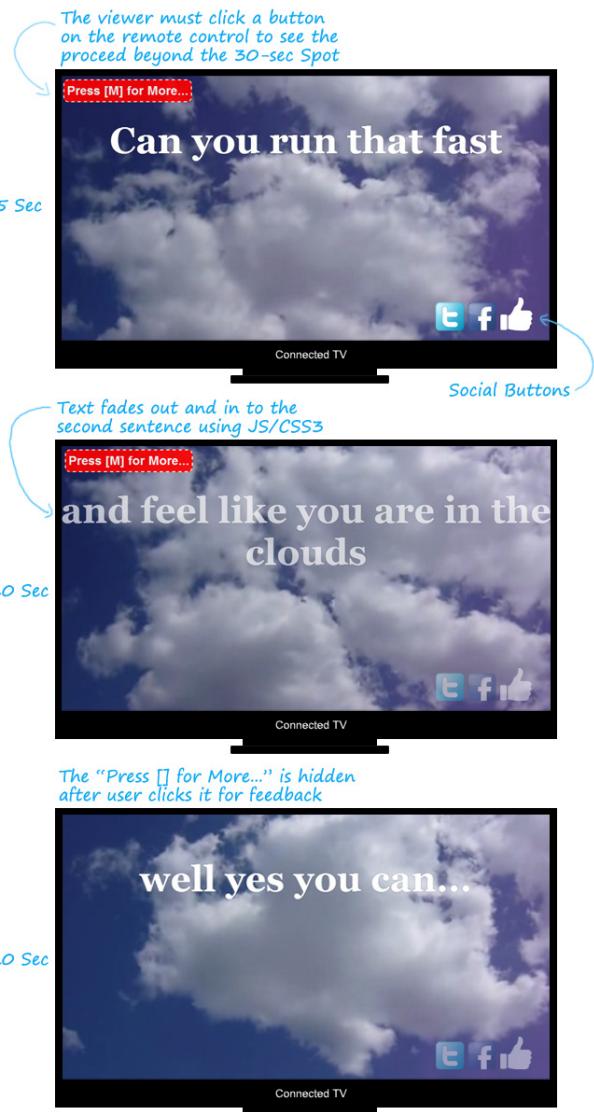


Figure 1. The “30-sec Spot” in HTML5

success of the fallback functions. In this case, the fallback functions worked smoothly and proved their ability to be able to run the ad on relatively limited systems. Of course, there differences with the fully-supported version, such as some limitations on CSS3 effects, a somehow slower performance, and also special keystroke events had to be traced to use the Wii remote. For the next device, the Sony Internet TV browser was quite limited, and a static background had to appear instead of a video but the system was functional in overall. Finally, on the Boxee device, the system worked quite smoothly, and the performance was similar to the Chrome version. The only problems were that it was trickier to run 2 difference files of audio and video simultaneously and also, since the arrow keys were used by the device for keyboard simulation, different keys had to be used for navigation.

So, the Interactive Ad succeeded in working adequately on all tested devices, while the fallback strategy proved vital for its success.

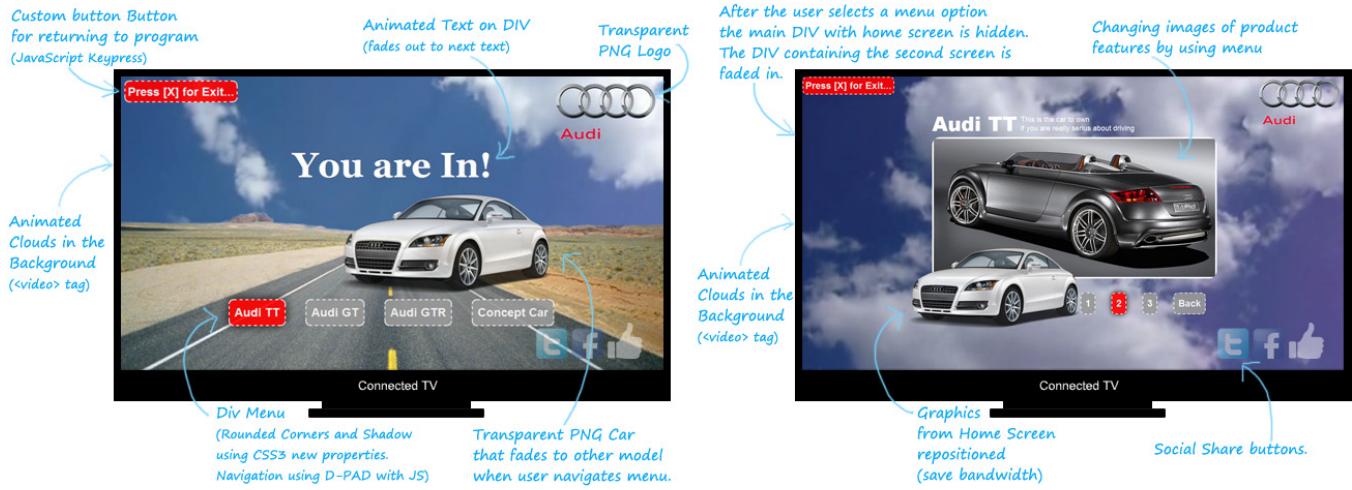


Figure 2. The Landing Page / Branded Microsite in HTML5

VIII. CONCLUSIONS AND FUTURE WORK

For this paper we examined the ability of the latest versions of web standard technologies, specifically HTML5, CSS3 and JavaScript, to create complete feature-rich Interactive Advertisements compatible with most Connected TV platforms. By examining the support features of connected TV platforms for these standards, and seeing a number of example Interactive TV Ads, it was concluded, that all the required features can be easily and accurately replicated using these technologies. To test the idea, an example prototype Interactive Ad was developed using the aforementioned web technologies.

The main benefits as seen in our research and practically through the prototype were:

- *Simple coding without need for specialized tools.* Even on a simple text editor is adequate, or, if desired, any web-site development environment.
- *Rapid development:* It was very easy and fast to develop the Advertisement, while it was not needed to learn any new obscure APIs, different for every device.
- *Cross-platform compatibility:* The same Ad, is compatible with all devices with a web browser. There are however limitations on some platforms due to the limited support for HTML5 and CSS3. Still, the prototype works on these, by replacing the unsupported features with supported ones.
- *Contains all the elements of an interactive TV Ad such as Video, Audio, interaction, animation, social buttons etc.*
- *Can be changed at real-time:* Since the Ad is a simple HTML page, it is possible to reside on any web-server. This means the advertiser can change it anytime. This approach however, will raise network-related concerns, such as bandwidth and security that have to be carefully examined. For example, on a network speed glitch the Ad might freeze and annoy the viewer, so this issue should be addressed using, for example, caching techniques.
- *Can support Analytics:* Analytics are a sought out functionality for marketing professionals and Since, the proposed Ad is a webpage, it is easy to track user interactions even with existing services (e.g. Google Analytics).

For the example prototype a very simple interactive Advertisement was created, but it did have the most common features that these Ads share. However, it could easily be expanded to include more demanding features, such as in-Ad games, forms and even more social networking and mash-ups.

Forms have been possible even before the latest web-standard versions and, with JS and CSS3 can now become more user-friendly and appealing. Social Networks include HTML5/JS APIs so these can be included too. Finally games, although very limited with previous HTML versions, can now be developed with HTML5 and JavaScript, reaching quality on a par only previously possible with Flash Games.

REFERENCES

- [1] W3C, "Web and TV", 2011, Available at: <http://www.w3.org/standards/webofdevices/tv>. Accessed in February, 2012.
- [2] R. Lawer, "Not just mobile: Adobe is abandoning Flash on TVs as well", GigaOM, 2011, Available at: <http://gigaom.com/video/flash-tv-future/>
- [3] J. Sivilka, "Soap, Sex, and Cigarettes: A Cultural History of American Advertising", Second Edition, 2011, p.196.
- [4] Interactive Advertising Bureau, "Glossary of Interactive Advertising Terms v2.0", Available at: <http://www.iab.net/media/file/GlossaryofInteractiveAdvertisingTerms.pdf>, Accessed in February, 2012.
- [5] DIRECTV, "Dynamic Ad Unit Campaigns", Available at: <http://www.directvadsales.com/Advertising.aspx#0-5>. Accessed in February, 2012.
- [6] Rovi Corporation, "Rovi Advertising Service factsheet", Available at: <http://www.rovicorp.com>. Accessed in February, 2012.
- [7] YuMe, "Example Ads with YuMe platform", Available at: <http://www.yume.com/content/ad-gallery-connectedtv>. Accessed in February, 2012.
- [8] F. Daoust, P. Hoschka, C. Z. Patrikakis, R. S. Cruz, M. S. Nunes, D. S. Osborne, "Towards Video on the Web with HTML5", Position Paper, NEM Summit, 2011, Available at: <http://www.w3.org/2010/Talks/1014-html5-video-fd/video-html5.pdf>
- [9] K. Chorianopoulos, G. Lekakos and D. Spinellis, "Intelligent user interfaces in the living room: Usability design for personalized television applications", Proceedings of the 2003 International Conference on Intelligent User Interfaces, 230–232. ACM Press, 2003.
- [10] I. Hickson, "HTML5 - a vocabulary and associated APIs for HTML and XHTML", Editor's Draft, W3C, 2012, Available at: <http://dev.w3.org/html5/spec/Overview.html>
- [11] jQuery JavaScript library, Available at: <http://jquery.com>
- [12] Modernizr JS library, Available at: <http://modernizr.com/>