THE NEW "UNIVERSAL TRUTH" OF THE WORLD WIDE WEB

Alexandru Tăbușcă[1]

Abstract

We all see that the world wide web is permanently evolving and developing. New websites are created continuously and push the limits of the old HTML specs in all respects. HTML4 is the real standard for almost 10 years and developers are starting to look for new and improved technologies to help them provide greater functionality. In order to give the authors flexibility and interoperability and to enable much more interactive and innovative websites and applications, HTML5 introduces and enhances a large set of features, such as new form elements, APIs, multimedia elements, structure and semantics updates. The development of HTML5, started in 2004, is currently carried out by a joint effort of the W3C HTMLWG and the WHATWG organizations. A lot of important companies participate in this effort, including the largest browser developers: Microsoft, Mozilla, Opera and Apple. The specifications of the new "to be" standard is still work in progress and quite a way lies ahead before its completion. Taking into account this fact there is a certain possibility that the features presented below have already been somehow modified or changed even in the near future.

Keywords: html5, cross-platform, css3, JavaScript, mobile application development, flexibility, interoperability.

Introduction

First of all we have to remember several historical, background facts about the well-known and widely used HTML standard. In 1999, the W3C consortium has decided to stop working on the HTML and move the specifications towards the more strict XHTML standard. A group of web developers from Opera and Mozilla did not agree with the approach and chose to present a paper to the W3C, in 2004, arguing that, "*we consider web applications to be an important area that has not been adequately served by existing technologies… There is a rising threat of single-vendor solutions addressing this problem before jointly-developed specifications.*"

The paper suggested several design principles to be followed:

- backwards compatibility and a clear migration path
- well-defined error handling, in the manner of CSS – meaning that an error should be treated as a "ignore unknown and move on" process, oppositely to the XML's strict "stop and solve" error handling
- users should not be exposed to developers' errors
- practical use: every feature that goes into the web-applications specifications must be justified by a practical use case
- scripting is a very important part and should be kept together with the declarative specifications

[1] Alexandru Tăbușcă, PhD Lecturer, Romanian-American University, School of Computer Science for Business Management, Bucharest – Romania

- avoid device-specific profiling
- keep the process open

The document was not sanctioned by the W3C consortium, and as a consequence Opera and Mozilla, and later on joined by Apple, continued to maintain a mailing list entitled "Web Hypertext Application Technology Working Group (WHATWG)". This public location kept track of their work and their proof-of-concept specifications. The proposed specifications extended the HTML4 capabilities and were later called "Web Applications 1.0", under the continued supervision of Ian Hickson[2].

In 2006, the W3C consortium realized that their past approach was wrong and decided to return to the HTML universe once more. They have asked the WHATWG for its specifications in order to use them as the basis of what will grow, evolve and is now known as the HTML5 specifications set.

All application developers currently working within the dynamic world wide web field can see that the new trend of the mobile applications space is more and more approaching the "elusive" HTML5 language – bundled together with JavaScript and CSS3 technologies, all of them collaborating for the creation of web applications and native mobile applications. This fact is very useful especially when dealing with cross-platform development or when working with content that already exists on the web.
This article is going to present some of the best HTML5 centered, cross-platform mobile frameworks that are now being used by developers to deliver native applications experiences on a variety of mobile devices.

Given the marketing waves and the publicity surrounding HTML5 (especially by the iconic Steve Jobs of Apple), it would be easy for the casual IT user to think that now is the moment that one particular technology will finally prevail, that will cook for us, do our homework or walk the dog ☺. Well, actually, HTML5 is not exactly that ☹; the specifications are not even officially ratified as a standard (yet). The specifications continue to approach completion, however, and when used in combination with the good old JavaScript and the also new CSS3, HTML5 can achieve a quite incredible things for an heir of the old HTML4 that we all know and love.

This things are especially true for the world of the mobile devices. A "standard" requirement for any modern mobile operating system and its preferred browser is the capability of understanding modern HTML5 applications. The cutting edge modern mobile operating systems, such s iOS or Android use WebKit as their bases. Likewise, the Canadians of BlackBerry and Americans of HP are also using WebKit and even Microsoft has already released a beta mobile version of Internet Explorer 9 for Windows Phone 7.

This means that out of the box, the nowadays smartphones and tablets support the eye candy and all the user-friendly stuff that make HTML5 so talked about. It also means that

[2] Ian 'Hixie' Hickson - the author and maintainer of the Acid2 and Acid3 tests, and the Web Applications 1.0/HTML 5 specification; known as a proponent of web standards, has played a crucial role in the development of specifications such as CSS

developers can freely use this technology when creating their applications and not have to worry that the device itself will have problems supporting a particular function.
It also means that developers that want to create HTML5 web applications for desktop use, e.g. for Google Chrome Web Store, are able to use the same code when building an application for the new shiny iPad2 or for any other Chinese tablet.

HTML5 introduces a lot of new elements that make it much easier to organize and structure the pages. Lots of old HTML pages include quite a variety of common structures such as headers, footers and columns and today it is quite common to emphasize them by the use of elements, giving each one a descriptive attribute like class or id.
Using these elements is largely due to the fact that HTML4 lacks the strong elements of semantics necessary for the description of those parts in a more specific manner. The new and fashionable HTML5 solves this issue by introducing new different elements for representing each of these types of sections.
The elements can be replaced with the new HTML5 elements as following: <header>, <nav>, <section>, <article>, <aside> and <footer>.

A usual structure for a document should look like the following:

```
<body>
  <header>Alex's Page</header>
  <nav>Menu Items</nav>
  <article>  <section>  Section 1  </section>  </article>
  <aside>Footnote for Section 1</aside>
  <footer>created by Alex, 2011</footer>
</body>
```

Video and audio elements are almost omnipresent within the world wide web today! In order to address their use and make it much more developer-friendly the HTML5 recognizes two distinct tags for those types of content: <video> and <audio>.
During the recent periods of time websites like YouTube, Video Yahoo, MySpace, Metatube, Flickr, Vimeo and tens of others similar ones made it very easy for anyone and everyone to publish and see video and audio. However, since HTML4 actually lacks the necessary means and methods to successfully manage, embed and control multimedia content, many websites are forced to rely on other technologies, such as Flash, to provide the required functionalities. Although it is now possible to embed multimedia content by the use of different plug-ins like QuickTime, Windows Media etc., Flash is currently the most widely used plugin that provides real cross-browser compatible solutions with the desired APIs for developers.

As evidenced by the various Flash-based media players, authors are interested in providing their own custom-designed user interfaces, which generally allow users to play, pause, stop, seek or adjust volume. The general idea taking into consideration by the HTML5 specifications is to provide the same functionality directly in browsers, by adding native support for the management and embedding of video and audio content, and providing scripts to control its playback.

The new video and audio elements of HTML5 are really up to the job. Most of the APIs are shared between these two elements - the only important differences are related to the actual differences between visual and non-visual media.

Both Opera and WebKit have released builds with partial support for the video element. The easiest way to embed a video file is to use a the <video> tag and allow the browser to provide its HTML5 default user interface. The <controls> attribute can also be used, being a boolean attribute that indicates whether the author wishes or not to provide this user interface by default.

```
<video src="movie.ogv" controls width="1024" height="768">
   <a href="movie.ogv">Download</a>
</video>
```

Another attribute that can be used in conjunction with the <video> tag is “poster”. This attribute is being used in order to specify a special image file that is being displayed in place of the video before the video has begun playing. Although there are some video formats that support their own poster frame feature, such as MPEG-4, this provides an alternative solution that can work independently of the video format.

Embedding audio content is also a very simple task with the help of HTML5’s <audio> tag. Most of the attributes are common between the video and audio elements, although for obvious reasons, the audio element lacks the width, height, and poster attributes.

```
<audio src="song.oga">
   <a href="song.oga">Download</a>
</audio>
```

The HTML5 language provides the source element for specifying alternative video and audio files which the browser may choose from based on its media type or codec support. The media attribute can be used to specify a media query for selection based on the device limitations and the type attribute for specifying the media type and codecs. Note that when using the source elements, the “src” attribute needs to be omitted from their parent video or audio element or the alternatives given by the source elements will be ignored.

```
<video poster="poster.jpg">
   <source src="video.3gp" type="video/3gpp" media="handheld">
   <source src="video.ogv" type="video/ogg” codecs=”theora, vorbis">
   <source src="video.mp4" type="video/mp4">
</video>

<audio>
  <source src="music.oga" type="audio/ogg">
  <source src="music.mp3" type="audio/mpeg">
</audio>
```

If the developer wants a tighter control over the user interface implemented in its application, in order to make it fit the overall design of the web page, the extensive API provides several other methods and events to let scripts control the playback of the media. The easiest methods that can be used are “play()”, “pause()” and setting “currentTime” to rewind to the beginning. The following example makes use of these elements:

```
<video src="video.ogg" id="video"></video>
<script>  var video = document.getElementById("video");</script>
<p>
<button type="button" onclick="video.play();">Play</button>
<button type="button" onclick="video.pause();">Pause</button>
<button type="button" onclick="video.currentTime=0;">Rewind</button>
```

Another advantage of the use of HTML5 resides in its “friendliness” when compared to the very strict and harsh syntax and error handling of the XHTML. Differently from HTML4 or XHTML, both of which are defined in terms of their syntax, HTML5 is being defined in terms of the Document Object Model (DOM), in other words as the tree structure used internally by the browser application to represent the document. Considering a very simple document, consisting of a title, heading and paragraph, the DOM tree representation looks like this:

HTML
- ➔ HEAD
  - ➔ TITLE
    - ➔ text content
- ➔ BODY
  - ➔ P
    - ➔ text content

The DOM tree includes a <title> element inside the <head> and a <p> element inside the <body> section.

The advantage of defining HTML5 in terms of the DOM is that the language can be defined independently of the syntax. There are two different syntaxes that could be used in order to represent HTML documents:

- the HTML serialization (taking HTML5 specifications for our example)
- the XML serialization (known as XHTML)

The HTML5 variant refers to the syntax that is primarily inspired by the old SGML syntax well known from the earlier versions of HTML, but now defined to be more compatible with the way browser applications actually do handle HTML in every day use.

```
<!DOCTYPE html>
<html>
  <head>    <title>Alex’s Page</title>  </head>
  <body>    <p>This is some text that just happened to be here.</p>  </body>
</html>
```

The XML serialization actually refers to the syntax using XML 1.0 and namespaces, just like XHTML 1.0.

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>    <title>Alex's Page</title>  </head>
  <body>    <p> This is some text that just happened to be here.</p>  </body>
</html>
```

Browser applications will use the MIME type to distinguish between the two types of serializations. Every document that is published as "text/html" must abide to the requirements of the HTML serialization and every document published with the XML MIME type such as "application/xhtml+xml" must abide the requirements for the XML serialization.

Application developers should choose which serialization type o to use. This choice might actually be influenced by several factors. Developers must not take "by heart" the use of one or the other of the serialization types. Both the HTML and the XML serializations are optimized for different scenarios and can be used successfully in different situations.

When using the HTML there are several factors that can be taken into account:

- the serialization is backwards compatible with all existing browsers
- developers are almost always familiar with the syntax
- the not so strict syntax of HTML helps avoiding the error screens that are quite common for every little mistake made when writing for XML
- developers can choose to omit some tags and attribute values without any consequences to the final content and layout

Despite all the above mentioned advantages of the HTML there still are situations when the more strict and organized XHTML serialization is not only welcomed but almost required:

- strict XML syntax forces developers to write very well-formed and organized markup; this this actually makes the code much more easier to maintain and develop at a later moment
- the code can be directly integrated with any other types of XML vocabularies like SVG or MathML
- the developer needs to allow the use of further XML processing, a fact that is more and more used nowadays for different platforms for editing and publishing processes.

With all the unified capabilities brought about by the use of HTML5 there still are a lot of problems raised by the existence of so many different mobile platforms to take into consideration: Android, IPad, IPhone, BlackBerry, WebOS, WindowsPhone and even Symbian or MeeGo (although they do seem to have lost the battle, at least for the time being).

The same fact that brought the problem was also a catalyst for finding an answer: a whole universe of cross-platform mobile development tools have become available on the

internet. Among these platforms, we have to mention four of them that specifically target HTML5 and JavaScript development.

PhoneGap is another HTML5 applications platform that helps developers build native applications by the use of the mixture of HTML5, CSS3 and JavaScript. The fact that makes PhoneGap somehow unique is that they let the developers create a complete and fully functional mobile application but place that application inside a native "wrapper", so that it can use native device APIs and there be able to be submitted to the Apple's App Store or Android Market.
As a matter of fact, PhoneGap enables mobile application developers to create an application as if they were targeting the mobile browser but with the benefit of being able to get into a "device dedicated" application store.

The *Titanium* platform is designed from the beginning to help web developers create mobile and tablet applications with greatest ease. Over the year, 2010, the platform has seen a lot of improvements and has enjoyed a spectacular growth in number of users. As any other respectable solution of this kind, Titanium is continuously updating and bringing new features and devices support. The platform is developed and maintained by Appcelerator[3], a company that has managed to acquired Aptana[4]; the latest is the owner of the famous and awarded Aptana Studio platform, fact that would ensure that the tools for building its applications will continue to improve and evolve over time.

Rhodes is another mobile application development platform, a Ruby-based framework designed especially to help the developers create native applications for a wide range of devices and platforms. Rhodes is somewhat different for the above mentioned tools – the Ruby[5] programming language is the real workhorse that does it all on the backend side. But, despite being a Ruby tool in essence, Rhodes uses HTML, CSS and JavaScript in its views. That fact brings the possibility to use HTML5 for the interface of the application.
Unify Project is in fact not an integrated platform but more of a set of development tools designed to help developers create smartphone applications by the use of HTML5, CSS3 and JavaScript. Sponsored by Deutsche Telekom, Unify Project is being published under a dual open source license (MIT and Apache version 2.0) and it uses a combination of PhoneGap, Adobe Air, Sass and the Xuooxdoo JavaScript framework.

These four environments for developing mobile web applications are not, by far, the only ones available out there. In fact, there are a lot, and I really mean a lot ☺, of such platforms and tools for every taste and programming background – jQuery Mobile, Sencha Touch, Python, Lazarus, Brew, Java Mobile Edition (ME), .NET Compact Framework (CF), Flash Lite, RhoMobile, SproutCore and others

[3] http://www.appcelerator.com/ - developers of Titanium, considered as one of the top 5 cross-platform mobile application development environments
[4] http://www.aptana.com/ - one of the leading open source IDEs for web applications development
[5] http://www.ruby-lang.org/en/ - Ruby is a dynamic, reflective, general-purpose object-oriented programming language that combines syntax inspired by Perl with Smalltalk-like features. Ruby originated in Japan during the mid-1990s and was first developed and designed by Yukihiro "Matz" Matsumoto.

As long as the browser support of HTML5 keeps improving the differences between developing and using a HTML5 application in a native wrapper or accessing it by a desktop shortcut directly as a browser application are going to faint, and hopefully – one day – disappear ☺

From reading the tech news we can see that quite a lot of important player of the today web, like Apple, YouTube or Facebook, are looking to HTML5 as the future preferred platform that can see a widespread use on the next generation of mobile devices.
In my opinion, regardless of the actual way the application is being developed and used - through a framework, through an application wrapper or as the basis for a mobile web application - HTML5 has all the chances to continue becoming an important driving force for mobile application development.

Bibliography

1. Alex Kessinger - How to Make an HTML5 iPhone App; Published by Six Revisions website; March 23, 2010; http://sixrevisions.com/web-development/html5-iphone-app/
2. Chris Heilmann - Talking about HTML5 games development at MIT in Boston; published by Hacks.Mozilla.org website; January 17, 2011; http://hacks.mozilla.org/2011/01/talking-about-html5-games-development-at-mit-in-boston/
3. Christina Warren - How HTML5 Is Influencing Web App Development; Published by Mashable website; April 17, 2011; http://mashable.com/2011/04/15/html5-web-apps/
4. Daniel Eran Dilger - SproutCore debuts new HTML5 web development tools; Published by AppleInsider website; Aprilie 19, 2010; http://www.appleinsider.com/articles/10/04/19/sproutcore_debuts_new_html5_web_development_tools.html
5. Ko Chi Chung, Chang Dong – Interactive Web-Based Virtual Reality with Java 3D; Published by IGI Global, 2008
6. Lachlan Hunt – A Preview of HTML5 - A List Apart Magazine, December 2007
7. Michael Juntao Yuan - Lightweight Java Web Application Development; Published by Prentice Hall, 2007
8. Peter Bright - Silverlight, HTML5, and Microsoft's opaque development strategy; Published by Ars Technica website; December 18, 2010; http://arstechnica.com/microsoft/news/2010/11/silverlight-html5-and-microsofts-opaque-development-strategy.ars
9. Steve Graham – Web Services with Java. XML, SOAP, WSDL and UDDI; Published by Teora, 2005