RESEARCH ON HTML5 DEVELOPMENT

TECHNICAL REPORT · AUGUST 2011

Jun Steed Huang

Jiangsu University

Advance Data Communication

Lecturer: Professor Jun Steed Huang

8/24/2011
JIANGSU UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
REPORT

RESEARCH ON HTML5 DEVELOPMENT

BY PhD Candidate
PATRICIA GHANN

RESEARCH ON HTML5 DEVELOPMENT

ABSTRACT

HTML5 is a language for structuring and presenting content for the World Wide Web, the core technology of the Internet. It is the fifth revision of the HTML standard which was originally created in 1990 by Tim Lee Banner, standardized as HTML4 in 1997 and has been under development as of August 2011 by, Web Hypertext Application Working Group (WHATWG). Its aims have been to improve the language by providing support for the state-of-the-art multimedia and ensuring that it is easily readable by humans and hence consistently understood by computers and other devices such as web browsers, parsers to mention only but a few. Thus HTML5 is intended to incorporate not only HTML4, but XHTML1 and DOM2HTML predominantly JavaScript. This report tries to examine the goals of WHATWG for introducing HTML5 and the difference between HTML5 and HTML4. It also examines the main features of HTML5 and the significance of these features. Finally the report looks at some of the limitations of HTML5 as compared with other Rich Internet Application or Rich Interactive Application frameworks like Flash, Silverlight and JavaFX and the future of HTML5.

INTRODUCTION

Since it was introduced to the Internet in the early 1990s by Tim Lee Banners, HTML has been in continuous evolution. Some features were introduced in specifications; others were introduced in software releases. In some respects, implementations and author practices have converged with each other and with specifications and standards, but in other ways, they continue to diverge. HTML4 became a W3C Recommendation in 1997. While it continues to serve as a rough guide to many of the core features of HTML, it does not provide enough information to build implementations that interoperate with each other and, more importantly, with a critical mass of deployed content. This also applies to XHTML1, which defines an XML serialization for HTML4, and DOM Level 2 HTML, which defines JavaScript APIs for both HTML and XHTML. Following its immediate predecessors HTML 4.01 and XHTML 1.1, HTML5 development is in response to the observation that, the HTML and XHTML currently in common use on the World Wide Web is a blend of features introduced by various specifications, along with those introduced by software products such as web browsers, those established by common practice and the many syntax errors in existing web documents. It is also an attempt to define a single markup language that can be written in either HTML or XHTML syntax. HTML5 comprises detailed processing models to encourage more interoperable implementations which extends, improves and also rationalizes the markup available for documents and further introduces markup and APIs for complex web applications.

GOALS OF WHATWG AND THE W3C

The idea that HTML's evolution should be reopened was tested at a W3C workshop in 2004, where some of the principles that underlie the HTML5 work as well as other issues were presented to the W3C jointly by Mozilla and Opera. The proposal was however rejected on the grounds that the proposal conflicted with the previously chosen direction for the Web's evolution; the W3C staff and membership voted to continue developing XML-based replacements instead.

Shortly thereafter, Apple, Mozilla, and Opera jointly announced their intent to continue working on the effort under the umbrella of a new venue called the WHATWG. A public mailing list was created and the draft was moved to the WHATWG site. The copyright was subsequently amended to be jointly owned by all three vendors and hence to allow reuse of the specification.

The WHATWG was based on several core principles, in particular that technologies need to be backwards compatible, specifications and implementations needed to match even if it meant changing the specification rather than the implementations and that specifications needed to be detailed enough in order for implementations to achieve complete interoperability without reverse-engineering each other.

The latter requirement in particular required that the scope of the HTML5 specification include what had previously been specified in three separate documents: HTML4, XHTML1, and DOM2 HTML. It also meant including significantly more detail than had previously been considered the norm.

In 2006, the W3C indicated an interest to participate in the development of HTML5 after all, and in 2007 formed a working group chartered to work with the WHATWG on the development of the HTML5 specification. Apple, Mozilla and Opera allowed the W3C to publish the specification under the W3C copyright, while keeping a version with the less restrictive license on the WHATWG site.

Since then, both groups have been working together and the following are their goals for the introduction of HTML5:

- Document real-world browser behaviour
- Document and standardise useful extensions
- Develop practical new features
- Ensure backwards compatibility
- Define robust error handling

The foundation of design and development is organised into three categories and these are:

- ❑ Compatibility
- ❑ Utility
- ❑ Interoperability

Design Principles: Compatibility:

- Support Existing Content
- Degrade Gracefully
- Don't Reinvent the Wheel
- Pave the Cowpaths
- Evolution, not Revolution

Design Principles: Utility:

- Solve Real Problems
- Priority of Constituencies
- Media Independence
- Universal Access
- Support World Languages
- Secure By Design
- Separation of Concerns

Design Principles: Interoperability:

- Well-Defined Behaviour
- Avoid Needless Complexity
- Handle Errors

HTML5 has several goals which differentiate it from HTML4.

The first and most important one is *consistent, defined error handling*. As popularly known, HTML purposely supports 'tag soup', in other words the ability to write malformed code and have it corrected into a valid document. The problem with this is that, the rules for doing this are not documented anywhere. When a new browser vendor wants to enter the market, they just need to test malformed documents in various browsers especially IE and reverse-engineer their error handling. If this is not done, then most of pages would not display correctly. In an attempt to solve this problem, HTML5 has discovered and codify this error handling, so that browser developers can all standardize and greatly reduce the time and money required to display things consistently. Also, in the future when HTML is no longer in use as a document format, historians may still want to read existing documents, and having a completely defined parsing algorithm would make this a possibility.

Secondary goal of HTML5 is to develop the ability of the browser to become an application platform, through HTML, CSS, and JavaScript. Various new elements have been added directly to the language that are currently (in HTML4) Flash or JS-based hacks, such as <canvas>, <video> and <audio>. Useful things such as Local Storage; a JavaScript-accessible browser-built-in sql database, for storing information beyond what cookies can hold, new input types such as date for which the browser can expose easy user interface so that we do not have to use our JavaScript-based calendar date-pickers, and browser-supported form validation would make developing web applications much simpler for the developers and make them much faster for the users because most features would be supported natively, instead of hacked in through JavaScript.

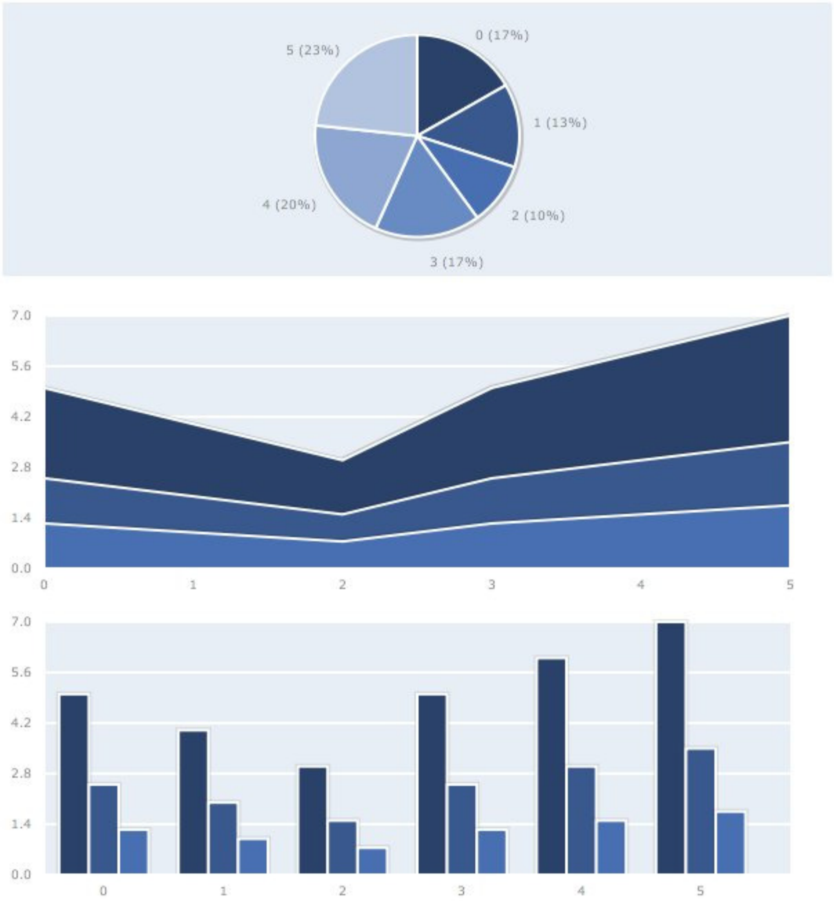
HTML5 also introduces better-defined semantic roles for existing elements. For instance <strong> and <em> now have different meaning and <b> and <i> have vague semantics that would work well when parsing legacy documents. Adding new elements with useful semantics like <article>, <section>, <header>, <aside>, and <nav> would replace the majority of <div> that are used on a web page, making pages a bit more semantic, but more significantly, easier to read. No more painful scanning to see just what that random </div> is closing - instead a designer would have an obvious </header>, or </article>, making the structure of a web document more intuitive.

Significant features that put HTML5 in the limelight

The HTML project working group has released several feature specifications till now. With HTML5 comes the following exciting and significant features: Some of key ones are:

- **Video elements**: This element is to enable designers to build web pages without having to resort to third-party proprietary plug-in software like Flash. YouTube is experimenting with using HTML5 to display video. Built in video controls makes the feature more attractive and easy to use.
- **Application cache**: On the lines of Outlook but minus the email client requirement, one can now store web apps much like emails locally. Google has already made use of this feature replacing its Google Gears mail client with HTML 5.

- **Canvas for Images**: The canvas element that comes with HTML5 help designers to manipulate graphics and photos enabling easy rendering of images. Designers no longer need to use intermediate technologies like jQuery. This saves development time as well as effort. It is a dynamic and interactive with graphics, draws images using 2D drawing API such as Lines, curves, shapes, fills, etc. and it's useful for Graphs, Applications, Games and Puzzles and many more. Below is an example of canvas graphs



- **Geolocation**: The HTML5 Geolocation feature enables location identification using various geo location technologies GPS (Global Positioning system), IP address, RFID (Radio frequency ID), Wi-Fi, Bluetooth MAC address and GSM/CDMA cell IDs. The system is bound by privacy regulations and needs to be authorized by individuals before it can be utilized.
- **Web workers:** This feature enables a web application to execute complex tasks independently without interfering with the performance of a webpage in real-time.

Limitations of HTML5

Browser Support: The main problem with HTML5's acceptance is that only modern browsers support it. That is almost everything except for Internet Explorer. The new version...IE9 offers excellent support, but as of this writing it's not quite out of beta. Even if it were, the majority of people would still prefer to use older versions of IE for quite some time.

The Language is a Spec: Although parts of the language are very stable, the language itself is considered a work in progress, so technically, any of the elements could change at any time. The language is not expected to be completed for several years, which complicates things further. Fortunately, a lot of the language is considered stable and ready to use. This is a great move forward, as a **Graceful Degradation** approach is used in writing HTML. This implies writing HTML that will work with older browsers, but would offer users with more modern browsers an enhanced experience.

Media Licensing Issues: Another ugly fact about HTML5 is that because of licensing issues, rich media has to be compressed in multiple formats in order to be compatible with most browsers. So one needs to probably use something like mp3 audio for webkit browsers (safari, chrome), and ogg for mozilla (firefox) browsers. This is a bit more tedious, but hopefully those issues would be resolved in no time.

Audio and Video Tag Limitations:

As earlier mentioned, one of the biggest additions to the HTML5 specification is the audio and video element tags which allows the individual to embed media files directly into the HTML markup. This allows one to use these media files in a way similar to the way they would use the img tag. Providing the location of the resource is all that is required and additional attributes allow you to create the behavior of the media elements such as streaming. This is a simple and standard way to enhance HTML with media files.

Although HTML takes the right step with the audio and video tag additions, providing a great multimedia experience is more than just playing a video file or a music mp3. Below are some limitations of the audio and video elements and how it compares to Flash and Silverlight:

- *Real-time manipulation and inspection of audio files is not possible.* Individuals can set some additional options like auto play or buffering, however one cannot manipulate the audio stream and provide their own equalizer changes. Flash and Silverlight both can do this. The HTML5 audio tag will simply not suffice for more advance scenarios.

- *The video tag*. The video tag allows the individual to set options like auto play or buffering but more common and advanced features are missing. In Silverlight, you have a wide array of options like applying shaders directly to the video and smooth streaming in HD 1080p. These options allow you to provide a much richer and advanced video viewing experience.

- *No DVR-like and smooth streaming capabilities*. Creating a video viewing website like YouTube is possible in HTML5. And, YouTube is currently experimenting with delivering its own video content through HTML5. However, with Silverlight and Flash creating a rich DVR-like experience with live video is highly possible as compared to HTML5 at the time of writing this report. For example Silverlight's HD smooth streaming allows for picture in picture, pausing live video, rewinding, slow motion and down streaming when bandwidth slows down. Also With smooth streaming, the buffering is minimized and one can jump to different spots of the video almost in real-time. That is the difference between a one-way stream from server to host that is HTML5 progressive download with simple stop, pause and play and a two-way stream the viewer can interact with in other words the DVR-like experience.
- *Content protection and Digital Right Management (DRM) does not exist in the video tag in HTML5.* This is a big problem for hosting copyright or sensitive content that you do not want allow viewers to copy or save to their hard drive. For instance imagine Netflix using an HTML5 video tag for a new Hollywood movie and everyone was able to save a copy to their hard drive? Until changes are made to HTML5, video tag would only be used for non-sensitive media.

Is HTML5 ready for use in Mobile applications?

In order to effectively use HTML5, there is the need to **control the environment** in which it is used. This is due to the fact that support is not as widespread as expected and therefore does not make real sense for it to be heavily used unless; usage can be locked down to certain platforms which have HTML5 support. Fortunately Webkit is leading the way for HTML5, and so can safely focus on devices powered by Webkit. The 3 hottest mobile devices; The Palm Pre, iPhone 3Gs and the new Google Android phone all have browsers that are based off the **Webkit rendering engine**.

Safari is even leading the way on the mobile HTML5 front; the iPhone with the latest software upgrade is the only device that could properly render the <audio> element. Because these devices are so young and all use the same rendering engine, the likelihood of them pushing a rapid software upgrade is pretty high. Currently, one can confidently use many of the HTML5 features in iPhone Web app development and it is expected that Pre and Android would follow in suit.

Future of HTML5

At first glance, with HTML5, the new elements immediately jump out and command attention. The W3C has indeed listened to the community and hence planned for the future when architecting the abundance of new element available. HTML5 provides everything from basic structural elements like <header> and <footer> to others like <canvas> and <audio> that tap into a very powerful API which give designers the freedom to create more user-friendly applications and at the same time distancing themselves from their reliance on Flash for saving data and intense animation. The W3C has taken the best parts from the various Web technologies and rolled them into, what is being dubbed the most powerful markup language to date. Thus the future of the web is sort of standard compliant. HTML5 is definitely going to revolutionize the way web applications are built. It is also anticipated that browser makers would get a consistent implementation going in the near future. Nonetheless there is still the issue of consistency between browsers at the moment (Firefox supports .ogg video, whereas Safari prefers the MP4 format for the tag). These issues need to be ironed out as soon as possible to ensure mass adoption of this technology.

Conclusion

HTML5 is with no doubt the best package ever to be introduced by the W3C. It provides great features at great price. It has made web applications easier in terms of time and money and developers can now develop without the need for bulky IDEs. However there are still issues that need to be critically handled such as the limitations in order for this new technology to gain worldwide adoption and acceptance.

References

- http://lachy.id.au/
- http://whatwg.org/
- http://www.a2zmenu.com
- http://www.diveintohtml5.org HTML5
- http://www.w3.org/html/