

Open Web Platform – Relentless Progress

Guenter I. Klas
Nov 04, 2012

Once a year, the World Wide Web gathers in the form of its ‘member companies’ and experts to meet, take stock, look into the future and debate what the focus areas shall be to improve the web. The organisation undertaking this effort is the World Wide Web Consortium, W3C. End of Oct 2012, the latest gathering took place, this time in France. Some of the discussions stay private to the consortium members, others are public. Here some opinions on the public aspects.

Not too far-fetched is the bet that the web community talked about the hot topics in the industry. But then, what’s the relevant industry? The IT industry? The Web 2.0 industry? The software industry? The mobile telecommunications industry? The publishing industry? Confused? No surprise. The web can be regarded as an enabler, boosting opportunities for many different industries and sectors. Thus, what’s considered a hot topic may well depend on what the companies from different industries perceive as creating new opportunities for them. One significant enabler stands out, HTML 5, the successor of the still widely available HTML 4.01.

HTML 5 is the way to create the web pages and web applications of the near and far future. Near future because it’s *already* partly implemented in several web browsers. Partly because we don’t know what the whole is. The scope of the ‘whole’ seems to be a moving target, and growing by the day. Far future because the HTML 5 W3C standard is far from completed yet. HTML 5 has a relatively long history inside and outside W3C. The original community that thought about upgrading HTML 4.01 to HTML 5 at a time when W3C was moving into a different direction is still working on HTML 5 outside W3C in a group called [WHATWG](#) [1]. Though attempts have been made to reign in the rebellious (as seen from a W3C standardisation perspective), concerns were recently raised that the merger of the two groups’ output isn’t completed yet and the [risk of a fork in HTML 5 specifications](#) isn’t completely buried either [2]. Given the importance attributed to HTML 5, this sounds confusing, baffling and to some extent worrying.

Focusing on the positive, let’s acknowledge that HTML 5 improves the user’s web experience in many respects including the use of audio, video, and graphics animation. It is also supposed to make the life of web developers much easier.

It’s important to say that not all the laurels should be given to HTML 5 which is ‘only’ a new version of the mark-up language HTML 4.01. Actually the praise needs to go to “[HTML 5 & Friends](#)”, as Paul Rouget and Geoffrey Dorne call it on the Mozilla developer network [3]. Only the collection of HTML 5 and its friends like CSS3 and Javascript make the whole experience increasingly exciting. That overall collection is often simply referred to as *HTML 5* or *HTML 5 technologies* (reason being that the collection tends to become overwhelming...).

This habit is good and bad: Good as it just throws everything in one pot and more practical people wouldn’t need to worry about a cumbersome classification of the many technology efforts into certain groups of specifications (“Is Web Workers part of HTML 5 or part of CSS3 or part of Javascript ...?”). Bad in the sense that throwing everything in a single bucket renders the overall picture opaque. The bucket of many, many tens of specifications looks messy, like the wild growth of 100s of herbal plants in a wide and big ceramic pot.

A set of technologies like “HTML 5 technologies” is one thing. But what’s called **the Open Web Platform** (let’s call it OWP for short here) is even bigger and bolder. The OWP is a programming environment just like say the ones built around Java for mobile driven by Oracle, or Java for Google’s

Android platform or Objective-C for Apple's iOS operating system. However, the OWP builds on a family of technologies that emerged around the web browser.

Goals: The OWP aims at enabling *richer* websites and web applications, *more interactive* ones and ones that *work across different operating systems and devices*. As the industry embraces the idea of a rich web platform, all industry players of name and status in this field work on a version of rich web platform, which they claim to be open (less encumbered with intellectual property rights). As a result we find today implementations from Google, Apple, Microsoft, Mozilla etc. They are all exciting, but they typically only support partial interoperability.

For example, an HTML 5 application written for Apple's Safari browser didn't work on my Mozilla/Firefox browser at the time of writing this paper. Very annoying. The error message I received was "download the Safari browser" and after having done so the next error message popped up urging me to download Apple's Quicktime Player. Oh dear, is that the Open Web? W3C's job is to reign in the players and get them to agree on a series of joint specifications. That should bring interoperability and cross-platform/browser capability to the market eventually. Hopefully.

OWP buildings blocks: HTML 5 is a component enabler of the OWP. The OWP includes a number of other specifications that aim at certain improvements or new capabilities. What might be the other elements of the OWP? Well, just look at other computing platforms which are called "native platforms" when they are created with other means: operating systems for desktops, laptops and tablets like Windows 8 or Mac OS (Snow Leopard, Mountain Lion,...) and smartphone operating systems like iOS (5, 6, ...) and Android (Ice Cream Sandwich, Jelly Bean, ...). People need means to display stuff like graphics application programming interfaces (APIs), graphics animation support, ways to interact with users e.g. through touch screens, APIs for applications to access features of the operating system and underlying device hardware like the camera, a near-field communications chip, the gyroscope, geolocation data etc. That gives a hint of what *at least* is in scope of OWP.

An OWP needs to be able to interact with audio and video in an integrated and seamless way. Not just offering the user to click on a link such that a new browser tab opens and the video is played there by a video player plugin, whose latest version you first need to download. That was yesterday. An OWP allows playing audio and video without extra plugins. It also allows to programmatically controlling the audio and interacting with it (e.g. to play music in the browser and let pictures dance and move in the rhythm of exactly that tune). An OWP will support screens of all sizes (from your watch to the smartphone to a big TV screen) and make use of the hardware features offered by different device types (phones, tablet computers, cameras etc.). Finally, an open web platform will feature improved communication mechanisms and channels such that one browser platform can talk in fresh ways to web servers and directly to other browser platforms.

The multiple faces of OWP: The Open Web Platform is first a vision. Second it is a growing reality in terms of various implementations following roughly the same or a similar blueprint (see Google, Apple, Microsoft etc.). Third it is a conglomerate of standardisation efforts to keep a lid on fragmentation and push the 'open agenda' (everybody can contribute, no nurturing of a particular commercial eco-system like Apple's, no license fees). Specifications created by the W3C shall enable any number of standards-compliant and therefore interoperable implementations.

W3C standards are royalty free. Thus, for many parties who watch their costs and margins, there is an incentive to use W3C specifications to build new devices and new services for consumers and enterprises. Equally there is an incentive for established parties like Apple, Google, Microsoft to protect their flourishing eco-systems through differentiation and better performance. This may lead them to divergence from the "standards for all", which in turn undermines the vision of W3C (openness, interoperability). Differentiation and competition has the benefit that it spurs and accelerates innovation. Thus, the battle of the big players has a good side as well. Although speed is not necessarily an attribute that can be associated with the HTML 5 standardisation effort in W3C, royalty-free standards are a characteristic of W3C.

Complexity: Unfortunately the modern and emerging web platforms are not exactly simple. They consist of a couple of key building blocks where each one is a moving target and each is being described by a countless number of books for developers, web pages of proponents' developer networks and specifications of W3C. The main building blocks are centred on HTML 5, CSS3, Javascript and SVG. Where HTML is a language to define the mark-up of a document (titles, headers, body, footer, tables, input forms etc.), CSS is a language to define style (formatting, colours, shades and the like). Javascript is a programming/scripting language and SVG is a language for creating 2D scalable vector graphics and images. Below tables capture some of the buzzwords and relate them to their purposes.

HTML 5 feature	HTML 5 area supporting the feature
Drawing graphics via scripting, enabling photo galleries etc.	Canvas 2D
3D graphics and 3D animation, enabling simulations, games.	Canvas 3D, leveraging WebGL
Embedding audio media in web pages and controlling/manipulating them via program code	Audio element
Embedding videos in web pages, styling them and controlling/manipulating them via program code	Video element
Built-in availability of royalty-free audio and video codecs	Open codecs
Various others	

CSS3	CSS3 area supporting this feature
Image transformations (rotate, scale, skew,...); smooth image transitions (e.g. for picture shows)	CSS Transform, CSS Transition
Tailoring the presentation of content (e.g. pictures, text) to a range of output devices (e.g. phone vs. desktop) without need to change the content itself	CSS Media Queries
Shadows for graphical elements and text	Box Shadow, Text Shadow
Use of compressed fonts that lead to faster download of pages with text; fonts include information about where they come from	WOFF (Web Open Font Format)
Ability to drag and drop items from one place in a web site to another place. Also works across web sites or web applications and from a computer desktop to a web page.	Drag and drop
Various others	

Javascript related	Area supporting this feature
Ability for web content to run scripts in background threads without interfering with the user interface or slowing down the interaction with the user (no more waiting...)	Web Workers
Various application programming interfaces (APIs). They allow web pages or web apps to access information on the device itself, which in the past was hidden from web	Geolocation API: users can provide their location to a web page or web app, File API: gives a

browsers.	web app access to the file system, Camera API: allows a web page to control the device camera, APIs for touch events from touch sensitive screens, API for device orientation (landscape, portrait)
Ability for web pages/web apps to store information on devices far beyond cookies	LocalStorage
Ability to store large amounts of structured data in the browser suitable for high performance searches locally on the device	IndexedDB
Ability to use a web page/app also when being offline and not connected to the Internet	Application Cache (app cache)

Other 'friends' of HTML 5	Area supporting this feature
Ability to have a permanent communication session in place between a web page/app and a server for exchange of general data (focus is on non-HTML data)	Web Sockets
Communications means (e.g. phone call, video conference) directly initiated and controlled out of the browser	WebRTC
Ability for a web server to push information about events to clients (historically it has been the browser that initiated a request to a web server)	Server-sent events

Getting the feel of it: Of course, one gets a better feel of HTML 5 & friends' capabilities by looking at the numerous demos and showcases on the web. For a game experience of HTML 5, see e.g. [BrowserQuest](#) [4]. For a quick interactive tour through HTML 5, it's worth to pay a visit to [Apple's HTML 5 showcase](#), which however may only run in their Safari browser (here we go with the open web) [5]. In case you are familiar with Firefox, Mozilla's [demo studio](#) offers a selection of HTML 5 demos [6]. Here are my favourites:

- The web is colourful and nicely animated with [3D Image Transitions](#) [7].
- The web is not only 2D anymore but in sophisticated, interactive ways [3D](#) [8]. It enables [high-resolution 3D graphics](#) [9], and [fast animation](#) [10] thanks to WebGL.
- The web now enables [full-screen animation](#) as in [11].
- Audio is now much integrated into a website experience. An example is [audio read-along](#) [12]. A further example is a VJing tool for real-time visual performance, in which you can animate a picture and sync opacity, scale and rotation of the picture to the sound of the music (though this only seems to work on Firefox). The same demo shows the use of [HTML 5 drag and drop](#), through which one can pick and choose the desired picture from another application (like Windows Explorer) and drag and drop it straight into the HTML 5 application [13].
- The concept of web workers is shown in a [demo](#) [14] where animated balls are randomly moving and colliding. Each ball has an associated Web Worker task that re-computes the ball's speed, position, acceleration.
- For a rather comprehensive "try it yourself" overview of HTML 5 and its friends go to the [Mozilla developer net](#) [3].

Two newer ideas are subsumed in the following buzzwords System Applications and Web Intents. They drew an interested audience at the TPAC 2012 event.

System Applications: Today, web applications are often not as powerful as native applications. The idea is to use web technologies to create system level applications which integrate closely with a target operating system. On a smartphone a system level application would be e.g. the build-in phone dialler or the address book. If anybody could just write a phone dialler, offer it for free download to phones and people would download and install this web-based phone dialler naively trusting the creators, such an application could run havoc on a phone. Therefore, the goals include besides defining Javascript programming interfaces, which clarify the potential interaction with a host operating system like Linux, also specifying the run-time environment and a security model.

Example areas include telephony, messaging, calendar, contacts, systems settings and network interfaces. APIs for the latter one would e.g. allow manipulating radio interfaces of a device (cellular, WiFi), listing available networks, showing current signal strength, and enabling/disabling the radio connections. A further example is a secure elements API, which allows interaction with hardware (e.g. the phone's SIM card) for tamper-proof storage or cryptographic operations.

There is a difference between a web application trying to *request* a phone call and a web application trying to *manage* all the phone calls itself, to route an incoming call to voice mail etc. Similarly there is a difference between a web app reading an entry in the user's address book compared to a web app administrating and managing the whole contacts list itself. The difference is between 'application level' and 'system level', respectively. Say we use a smartphone and visit the Facebook website. That interaction happens on application level and the Facebook web page won't get permission to just access our private phone address book. This is taken care of by the browser's security model. However, say we click on the web-based phone dialler on the smartphone. That dialler shall of course have access to our full address book and be able to interact with the cellular chip set via the operating system without prompting us repeatedly for permission. A *new security model on system level* needs to cater for this.

The [system applications effort in W3C](#) is in a very early stage [15]. System applications is also what Mozilla's Firefox-OS project needs and what network operators might want to shape in their interest. System application security models and APIs will play a role when it comes to creating smartphones which are based on Firefox-OS and which might be customised for the taste of a service provider's target consumer market.

Web Intents: Web Intents is a mechanism for client-side-initiated service discovery in the web and inter-application communication across the web. Typically a service somewhere in the web would register its ability to handle an action on the user's behalf. Web applications would start an action like "sharing" and the system would find the appropriate list of candidate services suitable for the user. As [W3C detail](#) in [16], "An Intent is a user-initiated action delegated to be performed by a service. It consists of an "action" string which tells the service what kind of activity the user expects to be performed (e.g. "share" or "edit"), a "type" string which specifies the data payload the service should expect, and the data payload itself."

Some client-side program code (say my browser page in which I can organise my holiday pictures) requests an Intent be handled (the intent being to share a picture with social networks). For this, I would simply click on a button on my pictures administration page. This Intent data is then passed to the browser, which then offers me as user to select which of the suggested candidate services I want to use. In the background my browser has already discovered which services out there in the big Internet can support my Intent. Upon my selection (say I choose Facebook) the Intent data is passed to the Facebook server to work on it (i.e. share and display my picture).

As W3C further say "an Intent is like the dual of a hyperlink. With a hyperlink, the source page specifies the exact URL to be navigated to. With Intent, the source page specifies the nature of the

task to be done, and the user can select any of a number of applications to be used to complete the task.” This sounds a bit like micro-outsourcing on the Internet. Not a bad idea.

As part of the whole story, web/cloud/Internet-based services first need to inform and declare towards browsers that they are able to handle certain Intents (that’s called registration). Second, my client web page in the browser needs an API to dispatch an Intent for handling ‘somewhere in the cloud’ (that’s called invocation). Third, someone on the client side (either I or the browser) needs to decide which service “out there” shall be selected to do the job (called Selection). Fourth, the work order (Intent) must be delivered to the service (called Delivery). Finally, the service out there needs a way to respond upon having finished its job (called Response).

In short Web Intents defines a standardised way of service discovery and a kind of light-weight remote procedure call mechanism for web apps. This is great, so of course, not very new. As we all know, there is nothing new under the sun, and yes, remote procedure calls have been around for decades and so have mechanisms for service discovery. Yet again a slightly different way of solving a problem ...

Still Web Intents have clearly the potential to make the browser-centric user experience we all have much richer. SonyMobile e.g. suggested that web apps should not only be able to discover services in the Internet/Cloud, but also [Web Intent-capable services in the local network](#) (e.g. in your house) [17]. SonyMobile provided a [video demo](#) for this [18]: A smartphone user would look up some video clips on the web, then realise that the screen size on the phone prevents a good experience and then decide to ‘outsource’ the presentation of the video to a device with a larger screen (e.g. a laptop or TV monitor). Such use cases can be implemented on the basis of today’s existing UPnP and mDNS enabled consumer devices. At TPAC 2012, NTT demonstrated how a web application (running in a browser) showing YouTube videos uses [Web Intents to find a TV monitor in the local network](#) and then controls the playing of the video on the remote monitor [19].

Web Intents could revive the interaction between browser-enabled devices (like smartphones, tablets) and all sorts of consumer devices including machines, ranging from computer monitors, TV sets, public display facilities, projectors to household appliances like fridges and washing machines. Exciting times lie ahead then, also because such ideas have been around for quite some time and still solutions for above use cases don’t seem to be widespread yet.

Moving into new markets: The open web platform has emerged as an evolution of the Internet browsers and today the various platforms are largely accessible via browsers running on consumer and enterprise devices like smartphones, tablets, laptops and desktops. However, there is some reason to assume that this computing platform finds its way into different vertical markets. One is [automotive](#) for in-car entertainment and automotive services [20]. Another one can be publishing of e-books and e-book readers and a final example relates to reinventing the TV set.

Challenges: One of the few challenges is that the Open Web Platform is designed by multiple architects in parallel and implemented by companies according to their taste when they feel like doing it. That makes convergence on common specifications difficult. The first efforts on HTML 5 were started in 2004. When the TPAC 2012 conference shut its doors, W3C had confirmed [Plan 2014](#) as [announced in Sept 2012](#) to complete HTML 5.0 by 2014 [21], [22]. That’s only ten years after the first work was started on HTML 5 (yes, outside W3C). Fortunately the browser industry has implemented large parts of it already.

Despite the attractiveness of the subject (HTML 5 technologies), finding common ground appears sometimes difficult or impossible. As mentioned, W3C and the rebellion WHATWG have decided to disagree and settle on an interesting concept: WHATWG will continue working on the evolution of HTML 5 as a living standard whereas W3C’s HTML 5.0 standard shall be considered as a snapshot of things in time, agreed upon after heavy battles and let’s assume some compromises too. By the time the snapshot is declared finished in 2014, the ‘real HTML 5 world’ will have already moved on

another few miles and new capabilities will be reflected in products and implementations. The HTML 5 working group in W3C has indeed taken on board a very tough job.

The open web platform is like a multi-story building, continuously under construction with the first couple of floors planned to be ready in 2014, with a labyrinth of staircases, rooms, web kitchens, dark corners, light suites, software labs and loud bars. In recognition of this and that a few software developers found themselves lost in the maze of techniques and specifications, W3C has convened a community site called WebPlatform.org which at the time of TPAC 2012 was still in alpha stage [23]. It's supported by the big web technology players and shall document the web platform, free of particularities of browsers, proprietary platforms and regardless of brands. In the end, the multi-story building gets a reception with someone handing out leaflets to visitors: How to find your way around the (amazing) maze.

For further reading, I recommend [W3C's wiki on the open web platform](#) [24] and Mozilla's [introduction to the wider family of HTML 5-related technologies](#) [25].

References

- [1] Home page of the WHATWG community: <http://www.whatwg.org/>
- [2] Opinion about the risk of HTML 5 forking: <http://www.theverge.com/2012/7/22/3175248/html5-fork-w3c-whatwg>
- [3] HTML 5 & Friends, an overview from Mozilla: https://developer.cdn.mozilla.net/media/uploads/demos/p/a/paulrouget/html5-dashboard/demo_package/index.html
- [4] BrowserQuest from Mozilla: <http://www.youtube.com/watch?v=kYcNJQ3Y6Sg>
- [5] Apple's HTML 5 showcase: <http://www.apple.com/html5/showcase/threesixty/>
- [6] Mozilla Demo Studio: <https://developer.mozilla.org/en-US/demos/>
- [7] 3D Image transitions to try out: <https://developer.mozilla.org/en-US/demos/detail/3d-image-transitions/launch>
- [8] Demo about snappy trees and canvas: <https://developer.mozilla.org/en-US/demos/detail/snappytree/launch>
- [9] 3D graphics at its best: <https://developer.mozilla.org/en-US/demos/detail/3d-grapher/launch>
- [10] Particle animator: https://developer.cdn.mozilla.net/media/uploads/demos/b/o/boblemarin/5cfea13ba1397f696bea7b2ff62c0188/fluid_1339407870_demo_package/index.html
- [11] Animated flight through a city: https://developer.cdn.mozilla.net/media/uploads/demos/p/a/paulrouget/a980ed07dc6c3e323b81b172613e1b30/flight-of-the-naviga_1304581547_demo_package/index.html
- [12] Audi read-along demo: <https://developer.mozilla.org/en-US/demos/detail/html5-audio-read-along>
- [13] HTML5 drag and drop, audio processing: https://developer.cdn.mozilla.net/media/uploads/demos/s/p/spite/dc461502a5dfbfa8585c27a5a0a9804/html5-vjing-tool_1312317583_demo_package/index.html#
- [14] Web Worker demo: <https://developer.mozilla.org/en-US/demos/detail/balls-with-gravitation/launch>
- [15] Charter of the group standardising System Applications APIs in W3C: <http://www.w3.org/2012/09/sysapps-wg-charter>

- [16] Web Intents as described by W3C: <http://www.w3.org/TR/2012/WD-web-intents-20120626/>
- [17] Suggestion to use Web Intents for local network service discovery:
[http://www.w3.org/wiki/WebIntents/SonyMobile - Local Network Service Discovery](http://www.w3.org/wiki/WebIntents/SonyMobile_-_Local_Network_Service_Discovery)
- [18] Video demo of Web Intents: https://docs.google.com/file/d/0B-2pb_m94nPXRGV5LTRvM0pLaUU/edit?pli=1
- [19] Session on Web Intents at TPAC 2012: [http://www.w3.org/wiki/TPAC2012/session-WebIntents-local-services#Web Intents and Web Intents for local services](http://www.w3.org/wiki/TPAC2012/session-WebIntents-local-services#Web_Intents_and_Web_Intents_for_local_services)
- [20] W3C Web & Automotive workshop 2012: <http://www.w3.org/2012/08/web-and-automotive/>
- [21] W3C HTML 5 - Plan 2014: <http://dev.w3.org/html5/decision-policy/html5-2014-plan.html>
- [22] W3C plans on HTML 5: <http://arstechnica.com/information-technology/2012/09/w3c-announces-plan-to-deliver-html-5-by-2014-html-5-1-in-2016/>
- [23] W3C community site for developers: <http://www.webplatform.org/>
- [24] W3C Open Web Platform wiki: [http://www.w3.org/wiki/Open Web Platform](http://www.w3.org/wiki/Open_Web_Platform)
- [25] Mozilla overview about the wider family of HTML 5 related technologies:
<https://developer.mozilla.org/en-US/docs/HTML/HTML5>