

The Role of Open Web Standards for Website Development Adhering to the One Web Vision

Serena Pastore

INAF- Astronomical Observatory of Padova – vicolo Osservatorio 5 – Padova - ITALY

ABSTRACT

Information dissemination is essential for every organization that needs to increase its visibility. This is valid for educational or research institutes and businesses. Websites are actually the usual way to promote information considering that modern websites are web-based systems composed by several software layers implementing different functionalities. Nowadays, a plethora of web-based content management systems (WCMS), commercial and open-source, provide similar capabilities. The paper highlights the trends in modern website development considering actual open web standards and the particular features of the mobile framework. Approaches include responsive web design techniques and custom mobile sites. Web content should be accessible everywhere, anytime and with whatever devices, which means it should be visualized and processed in the same way regardless of the type of device used, the physical location of the user, and the communication network used at the time. Mobile users' behavior has a great influence on website development, but open web standards should be adopted in web design to ensure a long-term growth towards the era of the Future Internet.

Keywords: *open web platform, web standards, HTML5, responsive web design, mobile web applications, native applications*

1. INTRODUCTION

Information dissemination is a powerful vehicle to enhance the awareness of an organization with business and educational goals. Although in the last few years the web has become the major medium for promoting an organization and enhancing its visibility, websites are the main tool used to publish content. The website concept actually involves a complex web information system composed of layered software providing a set of functionalities running behind a web server. For the different categories related to the purpose of a site (i.e., institutional, news, blogs, web portal, etc.), modern systems are developed by using web content management systems (WCM) [1] that combine many capabilities needed to reach users. In addition to providing documents, WCMSs offer interactivity and online utilities. Even if most are created with specific goals (i.e. Wordpress [2] was devoted as a blog system, even if it is nowadays used to develop all kind of sites) and there are some categorizations [2], all are essentially used for website development goals, for example, web systems such as Plone [2], Joomla [2], Wordpress, Drupal [2], Liferay [2], and so on. Differences regarding software and implementations (i.e., using different web programming languages such as Python or PHP or Java) do not affect the overall functionalities. All these systems have similar capabilities for managing all categories of media content (including text, audio, video, blogs, news, tweets, etc.) and provide simplified management for authors and editors. The choice of the tool [3] is tied to a subjective

decision [4] based on the language even if commercial or open source solutions are chosen. This paper analyzes trends in modern website development by highlighting the following factors:

- The need to adopt web standards for developing web tools in general and websites and web applications in particular (i.e., to guarantee usability, interoperability, accessibility and so on);
- The adoption of an open web platform that collects open (royalty-free) technologies that enables the web and allows a software component of the web to be implemented without requiring approval or licensing fees;
- -The need to provide content that will be the same regarding the type of information regardless of the type of device used or the communication network, and thus content that could be reused and shared;
- A shift in users' behaviour toward a mobile ecosystem, and the issues related to development in desktop and mobile environments.

Web development tools allow software to be implemented on web architecture based on key technologies and principles: they include protocols such as HTTP, identifiers (i.e., URL, URI and IRI) and other features such as internationalization, meta formats and so on [5]. Unlike other types of architectures, web architecture is

transient and is redefined or replaced within a few years. According to recent research [6] on mobile Internet, accessing the web mainly through mobile devices, we are facing a major trend in a shift toward mobile use of content, and thus mobile browsing is expected to outpace desktop-based access. Starting from these assumptions, the paper describes the actual role of WCMs compared to the open web standards technology evolution that should be at the basis of whatever web design. The need to face the mobile environment is then analyzed by considering users' behavior in the mobile landscape. Some solutions lead to responsive web design that for WCMs means defining various templates. However, this compromise solution fails in customizing the mobile context in the best way. However, the mobile world has various categories of devices, operating systems, and hardware features. The final goal is to provide the same information to all users regardless of the device used, but this implies difficulty in adopting standards privileging the development of custom solutions.

2. METHODS IN WEBSITE DEVELOPMENT

When developing for the web today, three main aspects must be taken into account: the use of WCM as a framework for publishing information and other utilities, the use of open web standards in the design, and the mobile landscape that drives users' choices and behavior.

2.1 Web Content Management Systems

There is some confusion over what is meant by a website and the process behind developing such a tool. The term includes the different kinds of container hosting web content in various formats. We found as synonyms web portal, WCM, and social applications. These terms share software layers separating the backend layer, the frontend layer, and the graphics layer, as shown in Figure 1. The backend includes a database management system (DBMS) where to store content, an application and/or web server for deploying and serving content, the frontend system that manages the web technologies used to structure and present the content, interact with the user, and the graphics layer related to the information strategy and visual design. Usually these layers have different roles in implementing single aspects of website development.

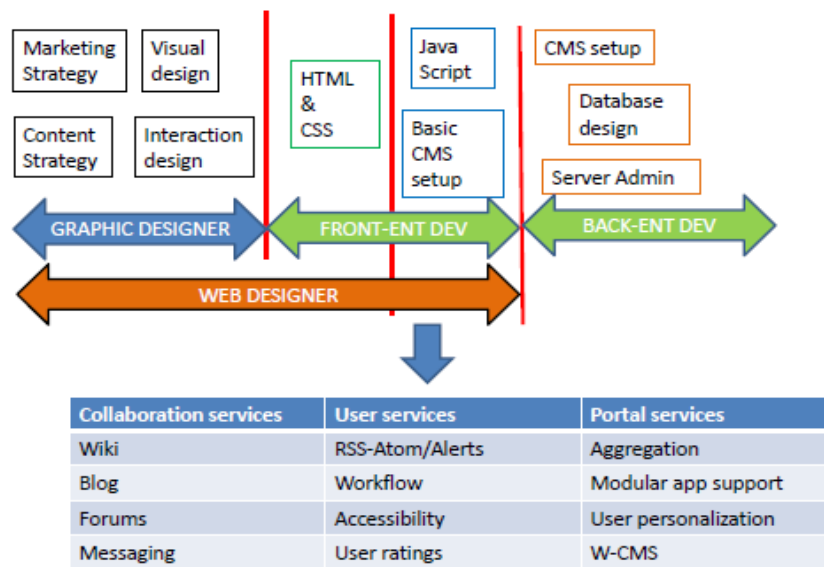


Figure 1. Software layers, roles and services implementations in website (Source Webanta)

There is however a subtle distinction among the individual terms. A portal is defined as a software platform for building websites and web applications. Common uses include websites that require the presentation of different pages depending on a user's login status, or presentation of different pages depending on a user's role or collaboration platform. WCMs is more related to content and on the content publishing process that follows a life cycle from creation to final approval by involving different actors. Finally, a social platform should join each framework with social applications that are web applications leveraging social identity, data, and

features. Often a website is a mix of such functionalities: each WCMs is composed of different modules implementing various features including embedding of user-driven social applications [7] (i.e., Facebook, Twitter, Foursquare, etc.). Categorization of services is shown in Figure 1 in which typical applications are listed according to services for collaboration, users, or portal needs. WCMs are essential for any modern site since they simplify actions and operations for content management. Adopting a specific solution among the various now available should be based on the services and needs required (e.g., institutional or business websites, blog

systems, news-like system, etc.). Instead, the choice seems to be related more to subjective evaluations that to the languages used. Few if any WCM products satisfy all the criteria “out of the box”, and thus, selecting a specific WCM platform should provide the ability to allow for a

custom integration between the WCM system and any back-end processes or systems. The main capabilities required by generic WCM and its components and extensions are summarized in Figure 2.

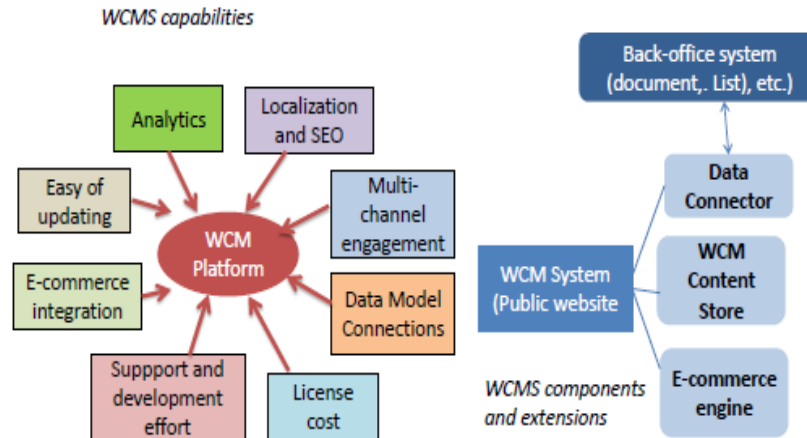


Figure 2. WCMs’ features, capabilities and components (Source Pricecept.com)

Analyzing the language-based aspect, PHP-based systems (i.e., Wordpress, Joomla, Drupal) are considered easier to learn and manage even if they suffer more from security issues. Since the systems are created for different purposes (i.e., blog for Wordpress or news-based systems for Joomla), their use as general sites appears inappropriate requiring adaptation of the information content. Python-based systems (i.e., Plone) are designed to manage information and users’ permission more strictly, but present a high learning curve discouraging adoption. Java-based web systems (i.e., Liferay [1]) suffer probably of the complexity of the software framework that seems too layered for the majority of website goals. All WCMs require special customizations to adapt to specific requirements. WCMs are made of different components (Figure 2). Aside from the core system, many add-ons (often called extensions, components and so on) are developed to provide all functionalities required by modern websites. The primary scope is reflected in the way the system defines content and layout. Joomla is an article-site target script that organizes the content as articles focusing on a layout suitable for a news or blog system. Using this tool with a different scope is possible, but all type of information must be considered as an article with its properties and thus could be difficult to realize. Extensions could be added to Joomla to customize the information, but it is an adaptation, not a native view. Our experience with a more generic CMS such as Plone have given the impression they are more suitable for publishing the different kinds of information that not-target websites have [8]. The adoption of a WCM is certain to be sustained, but its functionalities could not be

limited to provide good editing and workflow capabilities for content authors. Evolution in web technologies should be taken into account especially addressed to standards and the mobile landscape. Web standards dictate the rules that allow for making content accessible and device and location independent, and the mobile landscape is gaining a great role in the computer science panorama introducing the needs of different styles according the device used for their visualization.

2.2 Open Web Standards and the Open Web Platform

Web standards [9] are defined as specifications that help to define an open web platform for application development. HTML5 [10] will be the cornerstone for this platform, if considered as a framework including various web technologies (Figure 3) used to publish information on the web in different formats (website, web application, widgets). The basis is the execution within the browser by its engine without additional downloads. As shown in Figure 3, web design and applications require different components and features.

Each component is described by a specific standard that is developed within an organization, consortium, or groups of organizations (i.e., W3C Consortium [11], WHATWG community [12], etc.). Web standards are guidelines for ensuring that the websites are created in a way to allow the information to be accessible and to as many people in the most efficient manner. Standardization enables search engine exposure, and this is an important factor

considering that search is the lingua franca of the web since users navigate directly through URL, but they use search engines for most of their actions. Although the boundaries of the platform continue to evolve, HTML5 includes, with the definition of a markup language for structuring the content and for defining metadata, languages for presentation and interaction with users. The evolution of HTML after a period that has seen a debate between different specifications carried out by organizations is definitely going toward this version that HTML5 will become a W3C recommendation in 2014. As shown in Figure 3, HTML5 introduces newer technologies and concepts: video and audio elements that allow for embedding multimedia directly in a web page avoiding the use of plug-ins, the canvas element that allows graphics to be rendered on the fly, new forms of

implementation easier to use, and new structural tags to add semantics to the code. The applications caches' aspect allows for developing applications working offline. The main issue is that as usual not all features are yet supported on certain browsers. The new version of the presentation language (CSS3 [13]) needed for the layout and graphical aspect, introduces with special effects some important rules known as media queries [14] that help address different media devices. CSS was designed, since previous versions, with the optics used across many types of media by including some measures of device awareness through media type distinguished in screen, print, handheld, etc. Unfortunately, this classification of devices never really worked on browsers that ignored such specification.

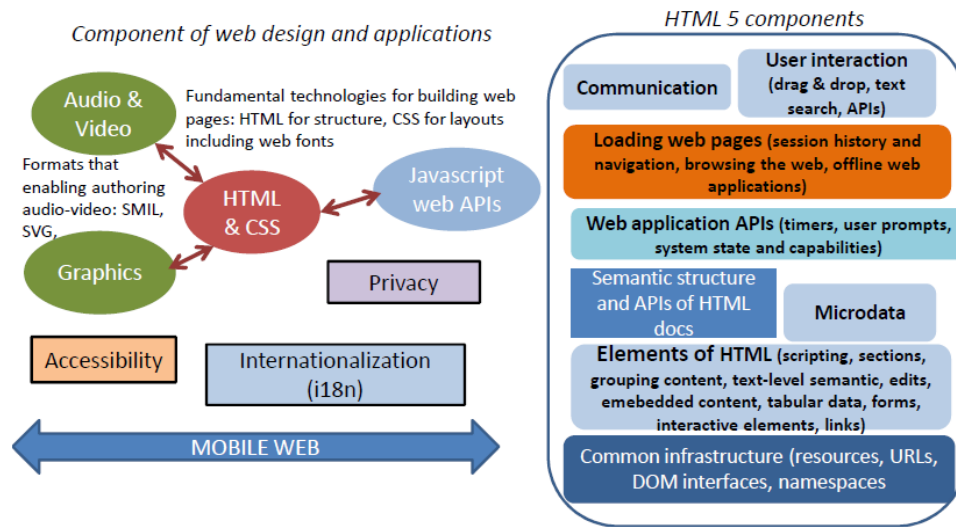


Figure 3. Components of web design and applications and of HTML5 framework(www.w3c.org)

```

MEDIA TYPES
@media print { body { font-size: 10pt } }
@media screen { body { font-size: 13px } }
@media screen, print { body { line-height: 1.2 } }

MEDIA QUERIES
@media screen and (max-width: 600px) { .class { background: #ccc; } }
    max-width: if viewing area smaller than 600px
@media screen and (min-width: 900px) { .class { background: #666; } }
    min-width: if viewing area is greater than 900px
@media screen and (max-device-width: 480px) { .class { background: #000; } }
    Max-device-width: if the max-device-width is 480px

For iPhone
<link rel="stylesheet" media="only screen and (-webkit-min-device-pixel-ratio: 2)"
type="text/css" href="iphone4.css" />

For device orientation
<link rel="stylesheet" media="all and (orientation:portrait)" href="portrait.css">
<link rel="stylesheet" media="all and (orientation:landscape)" href="landscape.css">
    
```

Figure 4. Examples of media types and media queries rules

Media queries improve the features of the media type, act to target devices' classes, and inspect the physical characteristics of the device rendering the work. An effect of how a developer can target different devices is shown in Figure 4. This technique, applied to address different mobile devices, gives greater control over. Separate styles for multiple devices can be stored in a single CSS, and developers can set the condition of particular media features when accessed by a device. This method allows for designing a more compatible website that look goods on any device.

ECMAScript and its dialects [15] is an object programming language used extensively for user interaction. The programs or scripts refer to a code executed by the browser when a page is downloaded, or in response to an event triggered by a user. ECMAScript is widely used in web design due to the introduction of the XMLHttpRequest (XHR) object that allows for server interaction. It is a component of the AJAX (Asynchronous JavaScript and XML) [16] framework that incorporates several technologies for web development. It uses the Document Object Model (DOM) [17] that models a page and defines APIs for interaction with elements, the XHR API that provides scripted client functionality for transferring data between a client and a server. These technologies allow web applications to be developed that can be made available either directly in the browser as a web page or can be packaged and distributed as widgets. In this perspective, the Ajax application, a model of which is shown in Figure 5, has an approach similar to that of applications used on desktops.

model, given it to the view. Finally, the view presents the data to the user. So the model is where all applications' data objects are stored, the view is what is presented to the user and the interaction, and the controllers connect models and views by receiving events and input from views, processing them and presenting an output. The main aspect is that ECMAScript is a standard and several implementations follow its specifications. Moreover, the widespread diffusion of libraries or prototypes containing a set of built-in functions allows for easier development of Javascript-based applications. Each framework (i.e., jQuery, MooTools, Prototype, etc. [19]) offers many functionalities and is customized for the mobile environment. In this perspective, the use of the presented technologies could help to build interfaces that will adapt to a range of devices, which avoids writing unique applications for each device.

2.3 About the Mobile Landscape

The mobile technology landscape is confusing. There are numerous choices from HTML5 technologies, native app methods, and different WCMSs. The starting point is that a mobile experience goes through even the web platform. Web-based operating systems [20] working in the perspective of always-on connections were created born with the Google OS Chrome experience and then exploded with the diffusion of Android-based and Mac iOS-based devices, and have shifted applications to a web experience. Nowadays all devices, even a game console, have web browsers. This software is more than a client application, it is a specific environment realized by an engine able to execute programs. In this context, a website is a collector of information displayed regardless of the device and the communication link.

There are two main approaches: the design of a unique site that adapts to the visualizing devices using specific templates and; a duplication with a custom mobile version. The mobility however is related to the type of device, the communication protocols, the operating system, and the web browser. When developing a website, the information published must be usable and accessible by all users, and thus the constraints introduced by mobile devices and users' behavior [21] should be considered when designing the website.

The main constraints are mobile devices' capabilities in terms of input devices, reduced processing and storing features, screen and software application. We are faced with many devices, input models, and browsers. Web development should take into account the change in interfaces. The evolution of the touchscreen is taking over as the main input device (Figure 6), and the development of web apps requiring input and web navigation are conditioned by this trend. Devices are equipped with web

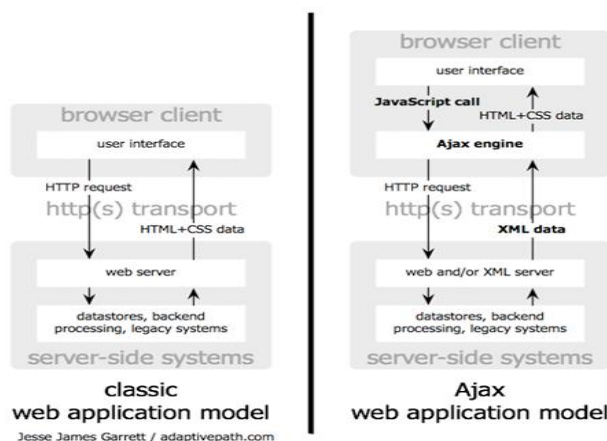


Figure 5. The comparison between application models.

Such applications should follow the Model-View-Control (MVC) pattern [18] that breaks an application down into the data (the model), the presentation layer (the view), and the user interaction layer (controller). In the event flow, the user interacts with the application; the controller's event handlers triggers and then requests data from the

browsers that manifest different behaviors in rendering websites allowing, for example, an adaptation. Among the wide range of devices, at least in Europe, Apple-based

seems to win the battle, and Apple’s mobile browser, Safari, is the most used (Figure 6).

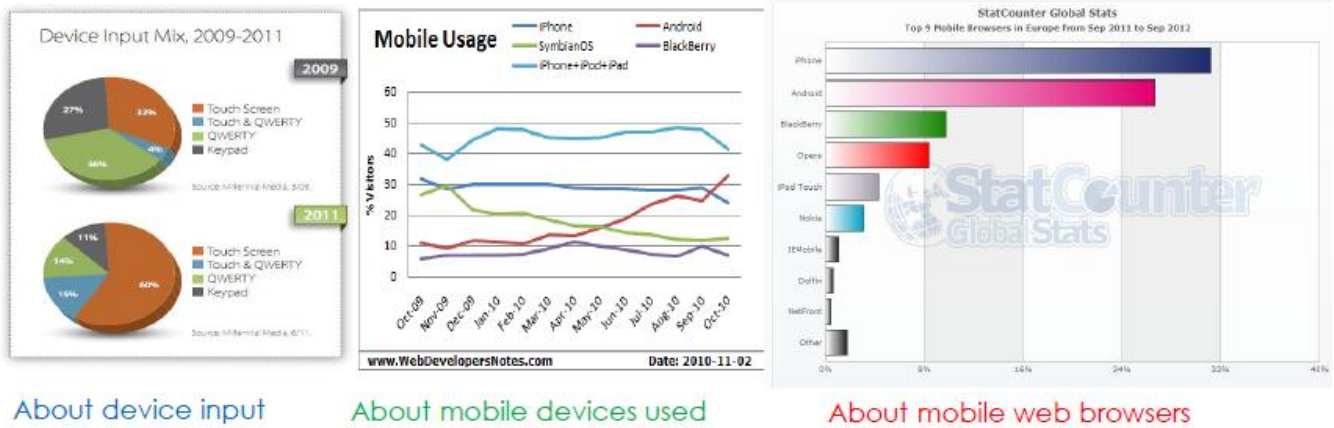


Figure 6. Statistics about device input, devices and mobile web browsers usage

However, this browser is based like many others on the WebKit [22] engine an open source engine that seems to follow newer standards enriching user experience. In a mobile world, web design should emphasize the content: content and styles should be separate entities, and the graphics should be limited to make information more usable for visitors. The aim is an easier and clean navigation that goes directly to the information. The problem is that users do not want to view a mobile site but want to browse a site. Social tools and Web 2.0-based applications still are important, but with the diffusion of mobile devices, micro networks and hybrid applications that run on and offline by incorporating mobile use are becoming the actual platform. Mobile web and smart

devices are changing the way the web is experienced. Users work differently with mobile phones or tablets (Figure 7), and this behaviour should be taken into account in web development.

The main aspects are the time a user stays on a website (reduced respect a desktop and fixes environment), the kind of content searched should be as immediate as possible. Many applications or services are related to other features of mobile devices such as the capability of different sensors (i.e. an accelerometer as motion detector, GPS sensors for location awareness, a noise sensor through the microphone, etc.).

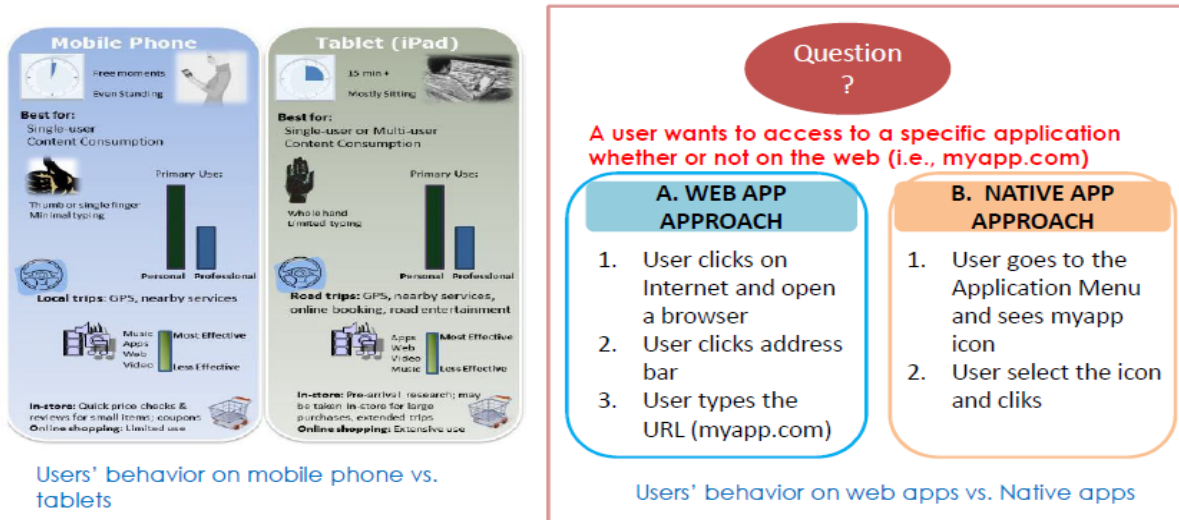


Figure 7. Users’ behaviour on device and on apps

3. ADOPTION OF RESPONSIVE WEB DESIGN VS. NATIVE APPLICATIONS

When considering the context and the need to choose a web standard, there is another question to consider. Mobile applications development is divided according to two main viewpoints: native applications vs. web apps. Much literature explains the pros and cons of the two methodologies [23], and there is a sort of tension between the two approaches. Most of the debate is focused on different technical features that evolve very quickly. Although a mobile application is software written for mobile devices that is specifically designed to run on a device's operating system and machine firmware, a web application is accessed from all web-capable devices and usually does not need to be adapted for different devices. The main issue is that web apps could not access native device features such as GPS, camera, calendar and accelerometer. A native app needs to run on its proprietary operating system platform, but a web app is typically coded in a browser-rendered language by using web technologies. Although architectural differences remain, the user experience provided by both interfaces is blurred, as most native apps use real-time web connectivity and web apps provide offline modes that can be accessed without network connectivity. In some cases, some of these apps are referred to as hybrid apps. We can summarize the main advantages of creating a native app in three steps:

- better control over the UI experience without using frameworks that emulate native artifacts;
- access to platform APIs that might not be available to web apps;
- potentially lower network usage at runtime since a web app might need to load completely at run-time.

With a mobile app, a developer should include the different kind of devices, and targeting a specific category could improve its efficacy. A comparison between the deliver mechanisms of the two aspects is highlighted in Figure 7.

Currently, web apps' advantages lie with the development side by providing direct control over the application's distribution with a lower cost. Moreover, web apps are cheaper and faster to develop and maintain, and they run on common browsers that can be accessed on most web-enabled mobile devices. Device customization is simpler from a developer standpoint. The access to device capabilities is going to be overcome since browser vendors, handset manufacturers and operators are trying to introduce such features via browser application programming interface (API)s. An example is the W3C geolocation API specification [24] that defines a high-

level interface for location information associated only with the device hosting the implementation that allows the use of proximity sensors embedded in most devices. The problem is that native apps are tied to proprietary software, and if the aim is to provide information for the different kind of channels (i.e., desktop and mobile) in a one web optics, the solution goes through the adoption of the responsive web design technique [25]. This decision allows for one design based on standards customized for a mobile or desktop view.

3.1 Responsive Web Design

Responsive design comes from an emergent discipline called responsive architecture that analyzes how physical spaces can respond to the presence of people passing through. In the web context, the term means to treat each different device as facets of the same experience. Standards-based technologies make the results more flexible and adaptive to the media that renders them. The problem is that while it is possible to view a website on a browser of any size, viewing a site designed for a desktop computer on a mobile device can be very problematic. It requires a lot of zooming in and out and scrolling in all directions, which provides a poor user experience. Creating flexible designs that adapt to provide the optimal layout for any browser size embraces the one web goal: that is, a design target for all kinds of devices. It creates a more consistent user experience, results in a future-proof and easy-to-maintain website, and eliminates the need to build separate, device-specific, websites. These objectives should target the audience requirements and the nature of the content. Technically, responsive design is a combination of flexible grids that expand and contract as the page width changes, images that are sized to track with the page width and use of CSS media queries to apply different CSS rules at different page widths. It combines the idea of a fluid site layout (one that contracts and expands with the browser width) with the technique of using CSS media query properties. These settings allow the device to detect the browser widths, change the style, layout, and proportions of content so that users enjoy a seamless experience, regardless of whether they are viewing the site by distinguishing the different categories of devices. The main effect is on the resizing of the content according to the screen dimensions (Figure 8), but with other elements of the language, it is possible to hide or view images when seeing on mobile or desktop devices.

With responsive design, HTML pages serve all screens, and only the CSS changes. The aspect to consider now is how to apply these techniques to WCM, since this is the main way to develop a website. There are two possible solutions: specific responsive templates to apply in a website or the idea of a WCM that is born responsive and

could adapt the view to different needs. Actual applications of this technique seem to prefer the definition of specific templates made up of style sheets and use of media queries. Many WCMSs have specific templates that follow the rules of this technique, but should be taken as the default template. Other solutions could be difficult to implement since the large presence of CSS files on WCMSs and its adoption according to priority. An

available solution following the second approach is a WCMS open source (SQuiz [26]) that is responsive. Even if distributed freely, it comes as a virtual downloadable machine. It is said that its own installations are possible. In any case, some people disagree with the use of such of responsive technology [27] for the aspects related to the “most common resolution used” in mobile devices.

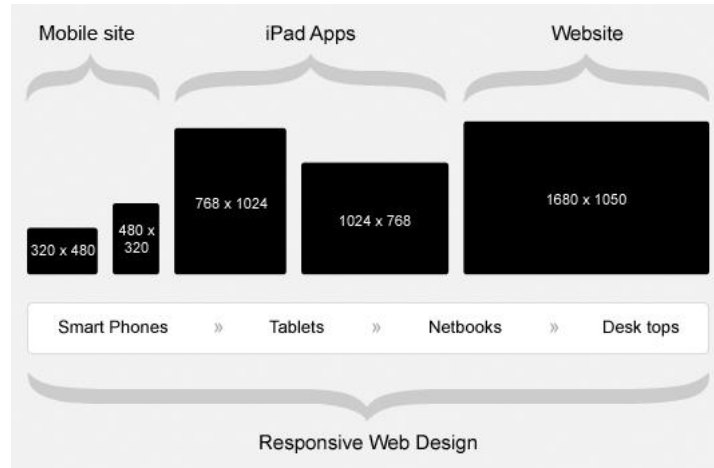


Figure 8. Different screen layout setup for the devices

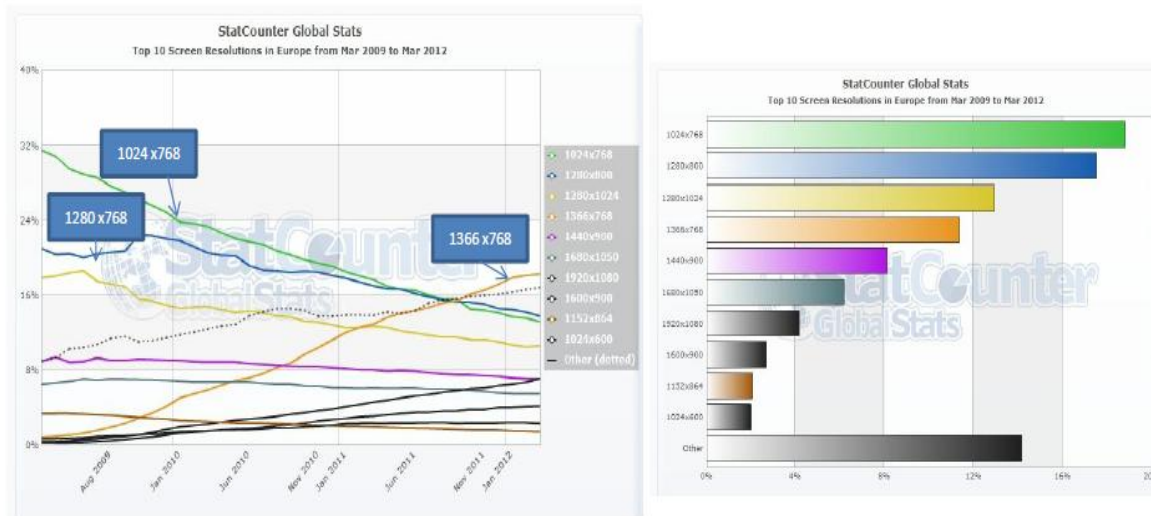


Figure 9 Browser screen resolution statistics in Europe

The most popular screen resolution (1024x768) reveals a downgrade toward the 1366x768 that is the setting of notebooks (Figure 9). Responsive design is an approach for supporting mobile devices, but there are greater differences between “mobile devices” (phone-size vs. tablets-size), and it could be necessary to have one set of HTML pages for phones, and another set for tablets and desktops that help to optimize the single experience. A compromise solution is a mix of the two: responsive design applied to desktop/netbook/tablets-like devices and

a specific design for mobile phones. This means a separate set of pages for a mobile site, but the existence if technologies for making this site mobile from a desktop-view developed with a WCMS, through various framework helps this task. These tools, which are divided into solutions that use an HTML5 approach or native applications and are categorized by considering the level of investment of the organizations, are defined as frameworks for implementing a mobile strategy.

Level of investment	HTML 5 solutions	Native app solutions
Turnkey	Pressly Mobify	MobileRoadie redFoundry
Custom	jQuery Mobile Sencha	PhoneGap FreeRange360

Figure 10. Solutions to “mobify” a website

In any case, these tools do not require code to be written. Examples of turnkey solution are Mobify [28], Pressly [29], or specific WCM mobile plugins, based on the HTML5 platform, implementing a solution adapted to mobile devices and tablets. Aside from these, as custom solutions, jQuery Mobile [19] and Sencha [30] are more complex but allow for better customization. All such products use Javascript mobile web development frameworks that provide excellent user interface controls for different modern mobile devices and share common characteristics based on standards: optimized for touchscreen devices, cross-platform and lightweight. In this way efficiently delivering an excellent mobile Web experience is straightforward. The technique that such tools use in detecting what type of device is requesting content and delivering a customized page for particular screen sizes known as the “if viewport then” technique.

The viewport meta tag resizes the web template to fit in the mobile screen.

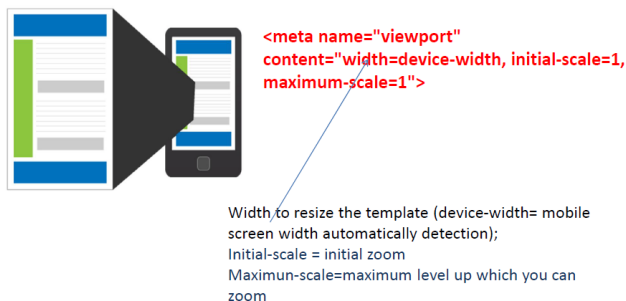


Figure 9 Use of the viewport meta tag

Viewports [31] are important elements of meta tag that allow for setting the browser’s layout relative to device width (Figure 9). On a mobile screen, the visual viewport is the part of the page currently shown on the screen. The CSS layout calculated to the layout viewport, which is wider than the visual viewport. The combination of the viewport meta tag and media queries helps to customize the mobile view by providing an excellent level of control and flexibility.

3.2 Writing for Mobile: Native Apps Solutions

Although web apps could allow writing once and use it in different platforms adhering to web standards, a native

app sometimes has advantages. Fortunately, there is also a mobile framework that allows developers to author native applications with web technologies and access APIs and app stores. The focus is that mobile apps need to access device features (i.e., the camera, the accelerator, etc.), and solutions known as system integrators help support both distribution channels adding a custom experience that will support numerous features and user interface enhancements. Turnkey native apps-based solutions are MobileRoadie [32] and RedFoundry [32], which use an online approach to easily build custom and optimized mobile apps. PhoneGap [34] and FreeRange360 [35] are costumed approaches. Specifically, open-source framework PhoneGap, originally developed by Nitobi, was acquired together with the company by Adobe which contributed the code to the Apache Software Foundation (ASF) renaming it in Apache Cordova [36]. Actually, it is an incubator of the foundation that distributes the original software. It is defined as a platform for building native mobile applications using standards. The interest in native apps is also a question of market and commercial opportunities offered by stores (i.e., Apple store, etc.), but the advent of native apps that follow web standards of the HTML5 framework, contribute to developing interest in these technologies and allows for widespread diffusion.

4. FINAL DISCUSSIONS

Websites remain the primary way to reach users, and WCMs are the ideal tool to use for this aim. However, selecting the right WCM presents several challenges related to an organization’s objectives and the add-ons required. Often it is necessary to separate knowledge management and content creation processes from facilities tied to public-facing websites such as e-commerce systems or integration with social networks since they require different frameworks. The WCM platform should be selected based on flexibility in terms of content updates, content delivery, and custom functionality. The constraints that should be considered differ and include specific aspects relative to the target of the site, the traffic, the investment required, and the attention to the mobile landscape. In any case, the focal point should be adherence to open web standards in a one vision web. Organizations and businesses that move around the web are creating a series of initiatives for adopting such technologies (i.e., the Open-stand.org initiative and the docs.webplatform.org). This approach should be applied when discussing the mobile strategy although we could adopt different solutions when developing mobile apps as web or native apps. The idea is toward a one web vision, and the design principles are to follow open web protocols in developing information used by everyone, in any way, and with whatever device. There are two solutions to this question that go through one website implemented with a responsive design and

two different websites. The first approach embraces the one web vision, but may not be ideal considering different users' behavior in mobile browsing. Very often mobile websites have specific features and are used not for regular browsing but as a gateway to other apps. With a responsive design, it should be possible to cover the tablet/desktop range, and a separate design for phones could offer real optimization. The approach of separate sites could be realized thanks to various software but is distinguished about HTML5-based or native based. Though the issue is related to the mobile vision of the information, different solutions available to "mobify" a website made with WCMs according to the two options all the possibilities to be implemented. The choice could be related to the traffic that a site has: if the majority of a site's traffic is side door from search engines or social sites, the site should embrace mobile web and HTML5. When users arrive via links, the content must quickly load in the mobile browser. Integrator solutions that sound like an ideal solution are quite nascent, and take quite a bit of work to make HTML5 work and look like a native app. Most sites today seem to prefer the native approach even if for commercial purpose related to the fidelity to a specific brand. If it is necessary to have a native app, it should be better to user some integrators with the end result of a better user experience but adhering to web standards.

REFERENCES

- [1] M. Johnston, CMS or WCM – Which is Which, cmscritic.com 2011. Retrieved 10.2.2012
- [2] Real Story Group, Web Content Management Vendors, 2011. url:www.realstorygroup.com/Research/Channel/CMS/Vendors. Retrieved 10.2.2012.
- [3] M. Skeleton, Selecting a web content management system, Priocept.com, 2011. url: http://priocept.com/2011/01/04/selecting-a-web-content-management-system/
- [4] R. Battles, Is a Content Management System right for you?, Jovia Studio Blog, Dec. 2008. url: joviawebstudio.com/index.php/blog/is_acontent_management_system_right_for_you. Retrieved 10.2.2012
- [5] L. Shklar and R. Rosen, Web Application architecture: principles, protocols and practices. 2ed. Wiley. 2010
- [6] Morgan Stanley Research, The Mobile Internet Report, 2009. url: http://www.morganstanley.com/institutional/techrese
- arch/pdfs/Theme_6_Data_Growth.pdf. Retrieved: 14.02.2012
- [7] J. LeBlanc, Programming Social Applications, O'Reilly Media, Inc. 2011. ISBN: 978-1449394912
- [8] S. Pastore, C. Boccato "A web multi-tiered framework for the Italian National Institute for Astrophysics: expanding functionality by incorporating collaborate tools to promote the institute". WSEAS Transactions on Information Science and Applications. Issue 11, Volume 3, November 2006, pp. 2207-2214. ISSN: 17900832
- [9] E. Anderson, V. DeBolt, D. Feartherstone, et. al., InterACT with web standards: A holist approach to web design, New Riders, 2010. ISBN: 978-0321703521
- [10] M. Pilgrim, HTML5: Up and Running, O'Reilly Media, 2010. ISBN: 978-0596806026
- [11] The W3C Consortium, at url: http://www.w3.org
- [12] The WHATWG community, http://www.whatwg.org
- [13] P. Gasston, The Book of CSS3: A developer's guide to the future of web design, No Starch Press, 2011.
- [14] Media Queries: an evolution in web design, Webmaster. Format Blog, at url: webmasterformat.com/blog/media-queries-and-responsive-web-design, 20.01.2012
- [15] A. MacCaw, Javascript Web Applications, O'Reilly Media, 2001. ISBN: 978-1449303518
- [16] N.C. Zakas, J. McPeak, J. Fawcett, Professional Ajax, Wrok. 2006. ISBN: 978-0471777786
- [17] J. Keith, DOM Scripting: web design with Javascript and the Document Object Model, FriendsofED, 2ed. 2010. ISBN: 9781420233893
- [18] E. Gamma, R. Helm, R. Johnson, J. Vlissedes, Design Patterns: elements of reusable object-oriented software. Addison-Wesley Professional, 1994.
- [19] L.M. Orchard, A. Pehlivanian, S. Koon and H. Jones, Professional Javascript Frameworks: prototype, YUI, ExtJS, Dojo and Mootolls, Wrox, 2009. ISBN: 978-0479384596
- [20] G. Numitor, Google Chrome for Android, Flu Press, 2012. ISBN: 978-6201912847

- [21] J.J. Garret, The Elements of User Experience. User-centered design for the web. New Riders Publishing. 2002. ISBN: 978-0735712027
- [22] M. Firtman, Programming the Mobile Web, O'Reilly media, 2010. ISBN: 978-056807788
- [23] Mobile applications: native vs web apps – what are the pros and the cons? mobiThinking blog, url: mobithinking.com/native-or-web-app. Retrieved 2.05.2012
- [24] W3C geolocation API specification, url: dev.w3.org/geo/api/spec-source.html
- [25] E. Marcotte, Responsive web design, A book apart. 2010. Isbn: 978-0-984442577
- [26] Squiz Site, url: <http://cms.squizsuite.net>
- [27] M. Slater, Responsive design is not the only good answer, Webvanta blogs, at url: <http://www.webvanta.com/post/812351-responsive-design-is-not-the-only>
- [28] Mobify, url: www.mobify.com
- [29] Pressly, url: www.pressly.com
- [30] Sencha Architect 2, url: <http://www.sencha.com>
- [31] P. Koch, A tale of two viewports, Quirksmode.org. 2011
- [32] MobileRoadie, url: mobileroadie.com
- [33] RedFoundry, url: www.redfoundry.com
- [34] PhoneGap site. url: <http://phonegap.com>
- [35] Freerange 360, [url:freerange360.com](http://freerange360.com)
- [36] Apache Cordova site, url: <http://incubator.apache.org/cordova>