Mobile e-Services Using HTML5

Karl Andersson and Dan Johansson

Pervasive and Mobile Computing Laboratory

Department of Computer Science, Electrical and Space Engineering

Luleå University of Technology

SE-931 87 Skellefteå, Sweden

{karl.andersson,dan.johansson}@ltu.se

Abstract—New mobile networks and new devices like smartphones and tablets are rapidly changing opportunities for public sector units delivering smart, mobile e-services to their citizens. Moreover, the upcoming HTML5 standard allows for cross-device and cross-browser support making service development and deployment much more easier than before at lower costs. This paper analyzes the most important features of HTML5, CSS3, and WebGL and their applicability for mobile e-services in the public sector. Also, we present a novel architecture for mobile e-services using HTML5, mashups, and RESTful web APIs as important building blocks.

Keywords—*Mobile e-Services; HTML5; CSS3; WebGL; Mashups; RESTful Web APIs*

I. INTRODUCTION

The evolution of mobile networks has been going on for decades now and users may today gain access to local, metropolitan, and wide area networks at low cost with the IEEE 802.11, IEEE 802.16, and the GSM/UMTS/LTE technology families being most widely spread. The latest step in this chain of development activities is the advent of 4G technologies offering peak data rates reaching 1 Gb/s for stationary scenarios and peak data rates around 100 Mb/s for mobile scenarios. Two candidate systems exist today: the LTE-Advanced [1] technology standardized by the 3GPP and the IEEE 802.16m [2] technology standardized by the IEEE.

Also, new types of handsets have recently been delivered to the mass market including, most notably, smartphones and tablets. They typically offer very easy-to-use interfaces and opportunities for the end-user to download new functionality in terms of apps from the associated ecosystems for mobile applications. While business users typically use their handsets for PIM (Personal Information Management) type of tasks like sending and receiving emails, interacting with their calendar, and accessing their contacts database consumers typically use their handsets for gaming, watching videos, listening to music and radio, and interacting with their friends using social media type of applications. Yet another popular application area is Mobile e-Services where citizens easily can interact with government agencies and municipalities requesting service and delivering required information in a timely and efficient fashion.

Main actors present on this arena are Apple and Google with their iOS and Android systems and the ecosystems Appstore and Android Market respectively. Microsoft is also trying to address this market with Windows Phone 7 and

Windows Phone Marketplace. This new landscape has opened up tremendous opportunities for app developers having the possibility to market, sell, and distribute their software very easily. Also, music, videos, and other media-oriented material can easily be spread via these new channels.

Since the associated development platforms vary (from Objective C, through Java to Silverlight) it is hard to deliver apps that operate cross-device over platform borders. Although products like PhoneGap [3], trying to overcome this problem, exist many developers are stuck having to develop their apps three times. Public sector entities typically have limited budgets and also cannot rely on citizens having access to a specific platform and thus need a suitable cross-device capable application platform. Web based solutions have been a common way to overcome this problem, but the current standards (HTTP 1.1 [4] and HTML 4.01 [5]) have been very much oriented towards the classic web application metaphor typically modeling an application as a discrete series of web pages. Hence, the possibilities for efficient deployment of highly interactive and usable applications have been very limited unless using plugins or non-standard web APIs.

HTML5 [6] is the new upcoming standard for content structure and presentation on the World Wide Web. Among many other interesting features, HTML5 is being designed with the consideration of being able to run on power-constrained devices such as smartphones and tablets.

This paper analyzes the most important features of HTML5, CSS3, and WebGL and their applicability for mobile e-services in the public sector. Also, we present a novel architecture for highly interactive cross-device capable mobile e-services using HTML5, CSS3, WebGL as important building blocks.

The rest of the paper is organized the following way: Section II surveys related work. Section III presents the most important features of HTML5, CSS3, and WebGL. Section IV outlines our model architecture for highly interactive, cross-device capable mobile e-Services, while Section V lists typical requirements from Mobile e-Services and analyzes how new features from HTML5/CSS3/WebGL meet those requirements. Finally, Section VI concludes the paper and indicates future work.

II. RELATED WORK

Wright [7] discussed the eventual merge of the web and operating systems and pointed out that future users of

smartphones and tablets will increasingly experience these devices through the filter of a web interface. Grady [8] described JavaScript programming using a functional style. Also, a brief introduction to the HTML5 canvas element and its programmatic control under JavaScript was given. Lerner [9] introduced and discussed the features of cross-window communication, web sockets and web workers in HTML5. Suhonos [10] described the implementation of a location-aware mobile search application using Z39.50 and HTML5.

Johnson et al. [11] studied the scalability of a few web-native information visualization methods including SVG, HTML5's Canvas, and native HTML in comparison to Java. The authors report that Java had the best performance, while SVG had adequate interactivity for medium or smaller datasets. Also, they conclude that SVG's vectorized format appears to have a slight advantage over HTML5's Canvas' immediate mode approach. Moreover, they claim that SVG can handle very large visualizations if carefully handled. However, neither SVG nor Canvas is ideal if large data visualization must be combined with interactive performance.

Martinsen et al. [12] studied execution behavior of four JavaScript application classes. They reported just-in-time compilation often increases the execution time for web applications, and that there are large differences in the execution behavior between benchmarks and web applications at the bytecode level.

Chen et al. [13] implemented a framework for browser based multiplayer online games and studied its performance and feasibility. They analytically showed that a Three.js 3D Engine and WebSocket based multiplayer online game could easily handle interaction of a small group of users.

Cazenave et al. [14] experimented with Timesheets.js, which is a JavaScript library for publishing multimedia web documents that take advantage of the new features of HTML5 and CSS3. XSLT transformations for converting OpenOffice Impress presentations into web formats were implemented.

Moreno et al. [15] presented an overall study of HTML5 in relation to accessibility requirements and developed an accessible HTML5 Media Player. They concluded that HTML5 provides accessible solutions, but it is still under development and not available in all browsers yet. They suggested using a combination of both HTML5 and Flash.

Rogers et al. [16] described developing portable mobile web applications by taking advantage of HTML5 and JavaScript libraries.

Taivalsaari et al. [17] overviewed the evolution of the web and argued that new emerging standards HTML5 and WebGL will further eliminate remaining limitations, and transform the web into a full-fledged application platform. Moreover, the authors foresee later versions of HTML to virtualize the underlying operating system capabilities even better, as well as ensure that the necessary security mechanisms are in place.

Taivalsaari et al. [18] stressed the fact that the web is becoming a very important platform for software applications and that the emerging standards of HTML5 and WebGL are paving the way for this development. The authors predict that

there will be a shift from binary applications to dynamically delivered web applications.

III. HTML5, CSS3, AND WebGL

HTML5 [6] is not yet finalized as an official standard, but is being embraced by the browser vendor community to an increasing extent [19]. Together with CSS3 and WebGL the most interesting features include resolution-dependent bitmaps, native support for video, local storage, web workers, offline capabilities, geolocation support, new input types, and web sockets. This section describes those features briefly.

A. Resolution-dependent Bitmaps

HTML5 allows for graphs being rendered, game graphics to be presented, and visual images displayed on the fly using the canvas element. The canvas is a rectangular area on the web page being manipulable by JavaScripts through the Canvas API.

B. Native Support for Video

A new element video is being defined in HTML5. The goal is to enable the web page creator to embed video content without relying on third-party plugins such as Apple QuickTime or Adobe Flash. Three groups of codecs and containers are typically supported by HTML5 compliant browsers: Theora+Vorbis+Ogg, H.264+AAC+MP4, and/or WebM.

C. Local Storage

Similar to cookies HTML5 Storage provides a way for web sites to store information on computers and retrieve it later. Web sites can access the local storage with JavaScript after the page has been loaded.

D. Web Workers

By using web workers browsers may run JavaScripts in the background in a standardized way. Web workers are either dedicated workers or shared workers. Web workers can communicate with other web workers and through message channels.

E. Offline Capabilities

In the case of connectivity loss, HTML5 provides means for users to continue working with their web services. This is mainly handled by the HTML5 cache feature allowing creators of the service to specify a manifest including lists of the files needed for the web service to work offline. Technically, this is implemented so that the web browser keeps copies of the files involved.

F. Geolocation Support

Strictly speaking not being part of HTML5, but rather a standard submitted by the Geolocation Working Group, the Geolocation API provides means of allowing a web page to query the browser of the current location of the device.

G. New Input Types

HTML5 defines a bunch of new input types including search boxes, spinboxes, sliders, color pickers, telephone numbers, web addresses, email addresses, calendar date

pickers, months, weeks, timestamps, precise absolute date/time stamps, and local date/timestamps.

H. Web Sockets

Web sockets, standardized by the IETF [20], provide means for multiplexing bidirectional full duplex communication channels over a single TCP connection. This way, real-time web applications can easily be created.

I. CSS3

Cascading Style Sheets (CSS) came along to enable separation of document content from document presentation and was standardized in its first version back in 1996. The last version, CSS3, is completely backwards compatible with previous versions. Moreover, CSS3 is split into modules and over fifty modules were published by the CSS Working Group. Selectors Level 3, Namespaces, and Color have all reached W3C Recommendations level.

J. WebGL

Web Graphics Library (WebGL) is a JavaScript API aimed for rendering interactive 3D graphics in web browsers.

Yet another important building block in the HTML5/CSS3/WebGL environment is the ability to check for feature support in a specific browser. Modernizr [21] is an HTML5 JavaScript detection library that detects support for many HTML5 and CSS3 features. By calling that library a global object called Modernizr is created containing a set of Boolean properties for each feature it can detect.

IV. MODEL ARCHITECTURE

In order to speed up development and deployment of new mobile e-Services we propose a layered model architecture, see figure 1, where devices and networks form the two basic layers taking care of end-user interaction and connectivity respectively.

Moreover, we propose a service support layer to handle common tasks like identity management, asset management, location management, and community management in a standardized and uniform way.

The actual e-services are placed at the e-Service Layer utilizing the services provided by the other layers.

Figure 1. Proposed Architecture Model

Most e-Services already today use the support layer entity Identity Management where electronic identities (eIDs) play a central role. The European initiative STORK [22] plays a vital role in relation to this aiming at establishing a European eID

interoperability platform where citizens ultimately will be able to establish new e-relations across borders just by presenting their national eID.

The support layer entities Asset Management and Location Management were designed to enable easy integration of a municipalities' and governmental agencies' assets and their locations with geolocation capabilities in the end-users' browsers.

Finally, the support layer entity Community Management allows for integration with social media applications and communities like Facebook, Twitter, LinkedIn, etc.

With the proposed architecture, web mashups can easily be implemented combining different types of data from multiple sources and presenting the information to the end-user in the browser.

Moreover, we also propose using RESTful (Representational state transfer) web services [23] having interaction between layers with stateless resources rather than messages being sent between the layers. REST uses the well-known verbs of the HTTP protocol including GET, POST, PUT, and DELETE. One of the most important concepts of REST is resources being referenced to using global identifiers (normally URIs in HTTP). Each resource can be accessed through its actions.

V. ANALYSIS

Mobile e-Services should be easy to use, available cross-device, and allow for cross-browser support. Moreover, mobile e-Services should be fully personalized and have their security handled using centrally issued electronic identities from independent entities like banks, telecom operators, or tax authorities. They should ideally also make use of the end-user's location whenever applicable.

In our study we analyze how existing e-Services can utilize features from HTML5/CSS3/WebGL making those services truly mobile. A selection of existing e-Services available in Skellefteå municipality, Sweden [24] where studied, see figure 2.

Figure 2. Existing e-Services available at www.skelleftea.se

TABLE I. HTML/CSS3/WebGL FEATURES IMPLICATION FOR MAKING SELECTED E-SERVICES MOBILE

<i>HTML5/CSS3/WebGL Feature</i>	<i>Building Permit Application e-Service</i>	<i>Compulsory School e-Service</i>	<i>Kindergarten e-Service</i>	<i>High School e-Service</i>	<i>Job Application e-Service</i>	<i>Parking Payment e-Service</i>	<i>Procurement e-Service</i>
Resolution-dependent bitmaps	High	Low	Low	Low	Low	Medium	Medium
Native support for video	Medium	Low	Low	Medium	Medium	Low	Medium
Local storage	Medium	Medium	Medium	Medium	Medium	Medium	Medium
Web workers	Medium	Low	Low	Low	Medium	Low	Medium
Offline capabilities	Medium	Medium	Medium	Medium	Medium	Medium	Medium
Geolocation support	High	Low	Low	Low	Low	High	Low
New input types	High	High	High	High	High	High	High
Web sockets	Low	Low	Low	Low	Low	Medium	Low
CSS3	High	High	High	High	High	High	High
WebGL	High	Low	Low	Low	Low	High	Medium

The selected e-Services include Building Permit Application Handling, Compulsory School e-Service, Kindergarten e-Service, High School e-Service, Job Application e-Service, Parking Payment e-Service, and Procurement e-Service. They are all high-volume e-Services and were designed for a variety of different target groups.

Building Permit Application e-Service normally allows citizens to apply for extension and establishment of new buildings on their properties over the web. Typical input can be plans, sketches, photographs, and other information needed for the clerks to take their decisions on whether to approve the building or not.

Compulsory School e-Services provide means of easy-to-use communication between teachers and the pupil's parents informing them about upcoming activities, schedules, and related stuff. Also, there are often modules for teachers assessing each pupil's progress and to document goals and targets for each pupil and making that information available to the pupil's parents.

High School e-Services normally deal with application handling and management of related stuff like bus passes. Also, there is often a module for following up attendance of each pupil at school.

Job Application e-Services are becoming common both in the public and private sectors. Typically, citizens may apply for jobs within a municipality or government agency and upload their CV and other personal data.

Parking Payment e-Services are popular as a timely and easy-to-use alternative to paying the parking fee by cash or credit card. The user typically enters the identity of the car, the parking lot identity and the duration of the stay.

Procurement e-Services are also becoming more and more popular both in public and private sectors. Businesses may easily submit their proposals and can download the information to each tender.

Table 1 lists new features in HTML5/CSS3/WebGL on one axis and the selected e-Services on the other axis. Each combination is analyzed in terms of the applicability (High, Medium, or Low) of HTML5/CSS3/WebGL feature making each service mobile.

Resolution-dependent bitmaps are of most interest to e-Services to graphics-intensive services, especially when the end-user may benefit from constructing images interactively.

Native support for video is of interest where text-based communication and content is not good enough and where video (live or play backed) can increase the end-user experience. We argue that e-Services for building permit applications, high schools, job applications, and procurement may benefit from this feature to some extent. One may also argue that e-Services related to parking may utilize native video support for remote surveillance of the car, but of course privacy issues need to be considered in that case.

As mentioned above, local storage provides means of storing end-user data in the browser. This is an interesting feature if the connection is not stable. Typically, one should design mobile web applications, so that any data is stored locally if the connection to the server is broken. For building permit applications, job applications, and procurement processes one may also argue that non-finished applications and proposals may benefit from being stored locally before final submission stage.

The use of web workers provides, as mentioned already, a way of running JavaScripts in the background independently of any user interface JavaScripts. This allows for long-running scripts not being interrupted by other JavaScripts that respond to clicks or other user interactions, and allows long tasks to be executed without yielding to keep the page responsive. However being a very interesting feature, we argue this new functionality not being so very much of help for making e-Services truly mobile. On the other hand, we argue that the e-Services for building permit applications, job applications, and procurement may benefit from using this feature. This is true when a large volume of data is being uploaded to the web server and hence the functionality of web workers can play a certain role.

Offline capabilities are on the general level very interesting making web applications available even if the web server is not responding or the connectivity is broken temporarily. This is valid for all types of e-Services and is key to making them mobile. However, since connectivity is not often a problem we argue the offline capability being of "Medium" interest.

Geolocation support is of very much interest where information on the user's current location enhances the end-user experience. For the selected set of e-Services this applies to building permit applications where end-users may walk around their premises and track the location. This data can then be used as input for the building permit application. Also, the argument applies to parking payment e-Services where the driver is helped to a large extent if the service automatically can detect the location of the end-user.

Web sockets open up for real-time applications on the web not having the browser to repetitively ask the server whether an event has occurred or not. On the general level, this is an extremely interesting feature, but we argue only the parking payment e-Service can benefit from this new feature. By using web sockets an application designer may easily implement functionality having an e-Service to indicate free parking spaces and when the deadline for picking up the car approaches.

CSS3 controls the style and layout of web pages. In particular, its media query features make web applications mobile enhanced very easy. Hence, we argue all e-Services can elaborate this functionality to a high degree when making an e-Service mobile.

Last, we argue WebGL functionality to be of special interest to Building Permit Application e-Services and Parking Payment e-Services. With the possibility of rendering 3D graphics directly in the browser, virtual reality models can easily be implemented and made interactive. Also, procurement type of e-Services may benefit from this new standardized feature.

VI. CONCLUSIONS AND FUTURE WORK

We conclude that HTML5 together with CSS3 and WebGL will be of great interest to future mobile web application developers and e-Services creators in particular. This is mainly because of this technology's cross-device and cross-browser support, but also its general feature of allowing web apps to behave very much like native apps. By going through a number of standard e-Services, we have shown a great potential for this technology when making e-Services mobile.

We intend to continue this work and perform an in-depth study where features in HTML5/CSS3/WebGL and their applicability to other e-Services than the ones studied in this paper also will be considered.

Moreover, we plan to design and evaluate more of service support layer functionality. The current interest for support of open data in the public sector will be a driving force around that.

Last but not least, we plan to perform end-user studies with a focus group testing new mobile e-Services. Results from that study will be used as input to the design and evolution of modern, mobility-enhanced e-Services.

ACKNOWLEDGMENT

We want to thank Per Dinborn, Skellefteå municipality, for providing valuable feedback on the manuscript and details on their ongoing work in the area.

This work was supported by the NIMO (Nordic Interaction and Mobility Research Platform) project [25] funded by the EU Interreg IVA North program.

REFERENCES

- [1] 3GPP TS 36.300, Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN), Overall description, Stage 2, March 2012.
- [2] P802.16.1/D5, IEEE Draft Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems, March 2012.
- [3] PhoneGap. Available at phonegap.com on July 30, 2012.
- [4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol — HTTP/1.1, IETF, RFC 2616, June 1999.
- [5] HTML 4.01 Specification. W3C Recommendation, December 1999. Available at <http://www.w3.org/TR/html4> on July 30, 2012.
- [6] HTML 5. A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft, March 2012. Available at <http://dev.w3.org/html5/spec/spec.html> on July 30, 2012.
- [7] A. Wright. Ready for a Web OS? *Communications of the ACM*, Volume 52, Issue 12, pp. 16–17, December 2009.
- [8] M. Grady. Functional programming using JavaScript and the HTML5 canvas element. *Journal of Computing Sciences in Colleges*, Volume 26, Issue 2, pp. 97–105, December 2010.
- [9] R. M. Lerner. At the forge: communication in HTML5. *Linux Journal*, Volume 2011, Issue 202, February 2011.
- [10] M. J. Suhonos. Building a Location-aware Mobile Search Application with Z39.50 and HTML5. *Code4Lib Journal*, Issue 10, June 2010.
- [11] D. Johnson and T. J. Jankun-Kelly. A scalability study of web-native information visualization. *Proceedings of Graphics Interface (GI '08)*, Windsor, Ontario, Canada, May 28–30, 2008.
- [12] J. K. Martinsen, H. Grahn, and A. Isberg. A Comparative Evaluation of JavaScript Execution Behavior. *Proceedings of 11th International Conference of Web Engineering*, LNCS 6757/2011, Paphos, Cyprus, June 20–24, 2011.
- [13] B. Chen and Z. Xu. A Framework for Browser-based Multiplayer Online Games using WebGL and WebSocket. *Proceedings of 2011 International Conference on Multimedia Technology (ICMT 2011)*, Hangzhou, China, July 26–28, 2011.
- [14] F. Cazenave, V. Quint, and C. Roisin. Timesheets.js: Tools for web multimedia. *Proceedings of the 19th ACM International Conference on Multimedia (MM'11)*, Scottsdale, AZ, USA, November 28–December 1, 2011.
- [15] L. Moreno, P. Martínez, A. Iglesias, and M. Gonzalez. HTML5 support for an accessible user-video-interaction on the web. *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer interaction (INTERACT'11)*, LNCS 6949/2011, Lisbon, Portugal, September 5–9, 2011.
- [16] R. Rogers. Developing portable mobile web applications. *Linux Journal*, Volume 2010, Issue 197, September 2010.
- [17] A. Taivalsaari and T. Mikkonen. The Web as an Application Platform: The Saga Continues. *Proceedings of 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2011)*, Oulu, Finland, August 30–September 2, 2011.
- [18] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen. The Death of Binary Software: End User Software Moves to the Web. *Proceedings of 2011 Ninth International Conference on Creating, Connecting and Collaborating through Computing (C5 2011)*, Kyoto, Japan, January 18–20, 2011.
- [19] caniuse.com, available on July 30, 2012.
- [20] I. Fette and A. Melnikov. The WebSocket Protocol. IETF, RFC 6455, December 2011.
- [21] Modernizr. Available at modernizr.com on July 30, 2012.
- [22] STORK. Available at eid-stork.eu on July 30, 2012.

- [23] R. Fielding. Architectural Styles and the Design of Network-based Software Architectures. Doctoral thesis, University of California, Irvine, CA, USA, 2000.
- [24] www.skelleftea.se, available on July 30, 2012.
- [25] www.nimoproject.org, available on July 30, 2012.