

HTML5 in Mobile Devices – Drivers and Restraints

Antero Juntunen, Eetu Jalonen, Sakari Luukkainen
HICSS 2013 – Hawaii International Conference on System Sciences

Introduction

- Current issues for native applications:
 - Fragmentation, between and within OSES
 - Rigid revenue sharing models
- Web-based mobile applications may address these issues
- HTML5 is an evolution of the previous standards and provides certain features to the browser that are typically associated with desktop-style software
- Goal: identify drivers and restraints for HTML5 in mobile devices

Framework

- Derived from a literature review on technology evolution
- Relevant theories include technological discontinuities [1], disruptive innovations [2], diffusion of innovations [3], platform theories [4], etc.
- Categories:
 - Added value
 - Ease of experimentation
 - Complementary technologies
 - Incumbent role
 - Technological performance

HTML5 Basics

- HTML: Markup Language for creating Web pages
- W3C and WHATWG standard bodies working together, first combined HTML5 standard in 2007
- Work in progress, recommendation planned for 2014
- HTML5 used as a blanket term for HTML5, JavaScript and CSS3
- HTML5 is about web applications and multimedia
- HTML5 in mobile applications is geared to provide a good alternative to native applications in as many devices as possible

HTML5 features for mobile devices

Feature	Comment
Multimedia	< <i>video</i> > and < <i>audio</i> > tags, support for both media formats without 3 rd party plug-ins.
Hardware integration	Access to mobile device features such as <i>GPS</i> , <i>accelerometer</i> , <i>microphone</i> , <i>camera</i> , etc.
Device adaptation	Modifying the page based on the device's <i>screen size</i> , <i>keyboard type</i> , etc.
User interactions	Support for <i>touch</i> and <i>speech</i> interaction, also <i>haptic feedback</i> (vibration).
Data storage	Data can be stored <i>offline</i> within the <i>browser</i> or on the underlying <i>filesystem</i> , though there is also a simple <i>key-value</i> based <i>database</i> .
Network	Cross-domain requests with <i>XMLHttpRequest</i> . <i>Server-Sent Events</i> or <i>Push Events</i> for sending data to HTML5 applications even when the page is not active on the browser. <i>WebSocket</i> [21] allows for more efficient data transfer, based on a TCP stream (two-way).
Widgets	HTML5 applications can be run off-line with the <i>ApplicationCache</i> feature, but also shared as archive files that can be unpacked and deployed in the same way as more traditional applications as per the <i>W3C Widgets</i> family of specifications.

Added value

- For the end users
 - No manual installation or update of an application
 - A unified user experience for multiple devices and platforms
 - HTML5 applications can better mimic the user experience of native applications
 - Offline data caching of HTML5
- For the developers
 - Cross-platform development (reduced OS fragmentation)
 - Web applications not tied to app stores: revenue sharing
 - More visibility for certain applications in web searches, etc.

Ease of experimentation

- How easy is it for developers to use HTML5 and how does HTML5 affect the software development process?
- HTML5 builds on existing knowledge of web technologies: easy transition for web developers
- Intrinsic advantages of running applications on the web:
 - Ease of deployment
 - Speed and ease of updating applications
 - Not tied to the approval processes of application stores
- On the other hand: Sufficient server hardware and bandwidth required (cloud one option)

Complementary technologies

- Adequate browser support a prerequisite for mobile HTML5 applications
- Platform vendors may control browser development

Feature	Safari on iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser	Nokia Browser	Internet Explorer	Opera Mobile	Opera mini	Firefox	webOS Browser
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones Tablet	MeeGo - NS Symbian	Windows Phone 7.5	Android & Symbian	Java iOS Android	Android, MeeGo	HP Phones HP TouchPad
Versions tested	3.2 to 6.0	1.5 to 4.1	18	1.0	6.0 to 7.0 1.0 to 2.0	1.2 ^3 to Belle	9	11 to 12	5 to 7	6 to 15	1.4 to 2.0 3.0
Application Cache <small>W3C API Offline package installation.</small>	✓	✓ 2.1+	✓	✓	✓ 6.0+	✓		✓		✓	✓
Web storage <small>W3C API Persistent and session storage.</small>	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓	✓	✓		✓	✓
Web SQL storage <small>W3C API (no active) Persistent SQLite storage.</small>	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓		✓			✓
Geolocation <small>W3C API Geolocation & tracking using GPS, cells or Wi-Fi.</small>	✓	✓ 2.0+	✓		✓ 6.0+	✓ Belle+	✓	✓		✓	✓
Multimedia <small>W3C API Video & Audio Players.</small>	✓	✓ 2.3+	✓	✓	✓ 7.0+	✓ Belle+	✓	✓		✓	✓
Server-Sent Events <small>W3C API EventSource pattern to maintain the connection to the server open.</small>	✓ 4.1+		✓		✓ 2.0+	✓		✓		✓	
Web Sockets <small>W3C API Newbidirectional protocol over HTTP.</small>	✓ 4.2+		✓		✓ 6.1+	✓		✓		✓ 7+	✓ 3.0.5+

Incumbent role

- Main incumbent players in the mobile application market: platform vendors (e.g. Google, Apple)
- Application store benefits:
 - Simplicity of monetizing applications
 - Visibility (potentially) through application store
 - Usability of native applications
- Web application benefits:
 - Not tied to application store policies
 - Flexible revenue models
 - Wide set of option for deployment (traditional website, cloud, deployed as an application)

Technological performance

- HTML5 still a work in progress
- Issues
 - Adapting the web application view to the conventions of a particular platform
 - Browser compatibility
 - Browser performance
- One solution: frameworks such as PhoneGap and Titanium SDK
 - Provide access to internal APIs of mobile platform but providing them in a platform-independent way

Conclusions

Dimension	Driver	Restraint
Added value	Cross-platform compatibility (D1)	User experience compared to native apps (R1)
Ease of experimentation	Cheaper, more flexible development and deployment (D2)	
Complementary technologies		Browser support (R2)
Incumbent role	No reliance on restrictive policies (D3) Flexible revenue models (D4)	Infrastructure and marketing expenses (R3)
Technological performance		Performance compared to native apps (R4)

Practical examples

- Financial Times moving from AppStore to an HTML5 application
 - Restrictive policies of Apple cited as a reason
- Facebook used to have an HTML5 application wrapped as a native application, but it released a new, native app
 - “Fast, reliable experience” the reason

References

- [1] P. Anderson and M.L. Tushman, “Technology Cycles, Innovation Streams and Ambidextrous Organizations”, In *Managing Strategic Innovation and Change*, Oxford University Press, New York, 1997.
- [2] C.M. Christensen, *The Innovator’s Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston, 1997.
- [3] E.M. Rogers, *Diffusion of Innovations*, Free Press, New York, 2003.
- [4] T. Eisenmann, G. Parker, and M.W. Van Alstyne, “Strategies for Two-Sided Markets”, *Harvard Business Review*, 84 (10), 2006, pp. 92–101.