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

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

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


