

AJAX for Mobile Devices

An overview of the status quo, and a proposal for web application platform industry standards

Larry Lieberman, Senior Program Manager Lead, Microsoft Corporation
9/16/2007

Historical perspective

Microsoft “powered” devices

Over the last 12 years, Microsoft Corporation has developed a successful operating system for mobile devices based on Windows CE, currently known as Windows Mobile. The Windows Mobile deliverable consists of (1) a particular set of Windows CE components, (2) a logo test kit requirements document, (3) a set of light weight middleware applications, and (4) commercialization components, such as drivers and value-added services provided by OEMs and Mobile Operators. Microsoft licenses the Windows Mobile OS to third parties, (generally OEMs and MOs), who commercialize and sell the operating system as part of a device and service offering.

Microsoft entered the embedded device space with Windows CE in 1992, after many years of creating desktop operating systems. Microsoft’s desktop strategy centered on the idea of “platform,” and we brought this idea to the device space. The notion of a vibrant and compelling third party software ecosystem was perceived a necessity for success. That said the small, handheld device is not a PC, and the limitations of the battery powered device, with a small screen, and intermittent and variable network connectivity, make a successful software ecosystem a difficult proposition.

Web applications

The last five years have seen the growth of “Web 2.0” and AJAX as a new approach to developing applications on the desktop, hosted in the web browser. Before the rise of AJAX, web “applications” were extremely limited in the types of interactivity that they could deliver. Web pages hosted content, and while web sites quickly acquired the ability to dynamically adjust that content based on user input, (search engines are a great example of this basic behavior), creating a rich user experience that users have come to expect from an application has, until recently, been something of a challenge.

The set of practices that have been grouped into the concept of “AJAX”, have changed this situation. On the desktop, there are many web applications that demonstrate the ability to fulfill user’s expectations of interactivity and dynamic presentation. The other part of the equation is the notion of the web service. A cloud hosted interactive data stream ready and waiting to satisfy the requests of a willing user base.

Into this environment, more powerful mobile phones are becoming more common. Many phones ship with a web browser. Those browsers are becoming more powerful. Faster data connections are becoming available and affordable. Mobile Operators are scrambling to provide new value propositions that drive users to get and use mobile data plans. The stage is set for AJAX to fuel the next mobile platform, and in order for that happen, the industry must drive toward standards.

The problems

Customer knowledge

The PC evolved as a multi-purpose device; it doesn't do anything by itself until you add software. The device market on the other hand, evolved from appliances like the transistor radio. Devices were designed, built, and consumed primarily with one use in mind. People had to figure out how to add software to their PC in order to get it to do what they wanted, (and many would argue this model has failed, since many people just use the pre-installed software). In the device world, people purchase a device that satisfies the key uses they care about, and generally they are unaware that they can add aftermarket applications at all. They are aware that they want their device to "do" certain things, but generally they don't know that they can add software to make their device do these things. To summarize this class of problems:

1. Mobile phone users don't understand the concept of "applications"
2. Even if they do, finding applications is hard
3. Even if they find the application, they have to figure out how to install the application
4. Mobile Operator security concerns may make installing and running third party applications difficult
5. Poorly written third party applications might have a destabilizing influence on the phone, and support call generating influence on the mobile phone user

Platform fragmentation

For the developer writing software for a mobile device, one of the biggest challenges is the question: "what type of device do I want to target?" Some platforms are easier to write for than others; while other platforms may have more "sockets" or installed end users. It's clearly not the PC world; there are several different operating systems in the marketplace, each with its own set of tradeoffs. Further, there are distinctly different types of users.

The recommendation

Web-based applications present the beginning of a solution to these problems. Customers have come to understand the web browser paradigm: they know how to find things on the Internet. Applications that run in the web browser don't have to install locally on the device. They are hosted on the server and run in client script. Applications that run in the browser and adhere to enumerated web standards have a greater likelihood of running on a wide variety of devices.

On the desktop, web application development didn't explode until the number of available web browsers, with their various format differences, started to coalesce into a few solid market leaders. At this point, developers learned that they could create web based applications, and have their applications behave appropriately regardless of which browser was used.

A strong standardization effort has the potential to energize mobile developers around the web application platform by vastly improving the reach of web applications across a broad range of target mobile device types.

The challenges

AJAX isn't a standard today, but rather a set of practices that often vary from browser-target to browser-target. Further, web browsing on the mobile device has not yet gained the ubiquitous acceptance that it has on the desktop. Finally, there are hurdles on the mobile phone device that don't exist on the desktop, at all, and until these are resolved, it will be difficult for mobile AJAX to gain the momentum necessary to start the network effect.

A mobile application, at a minimum, has four fundamental requirements: (1) rendering, (2) event handling, (3) scripting, and (4) asynchronous data exchange, all of which are generally available today with the current sets of AJAX functionality. Beyond these requirements, however, there are centralized capabilities that would vastly improve the adoption of a mobile platform:

1. Phone specific functionality. The ability to access features that are specific to the device itself. For example: does this device have access to the user's GPS location coordinates? If so, a web based application might greatly benefit from having access to it. Likewise, applications may want to be able to initiate a phone call, or access a local address book.
2. Disconnected and transient state. Web based applications on the desktop have already attempted to handle the notion of network disconnect. On a device, it is much more likely for the network connection to be intermittent and transient (moving from network to network). Therefore, it is critical that any web application platform to seamlessly transition from online to offline, with little to no disruption of the user's experience.
3. Device Capabilities standards. The mobile device industry is one of wide variety. Today, there are some methods available for a developer to determine if the device he or she is developing for has a certain-sized display, however these are far from standard. Ideally, a developer should be able to get a wide variety of information back about the device through a standard "capabilities" mechanism, and then act on that data in order to provide a best of breed experience for that device.
4. Security Sandboxing. Today there are a number of different techniques available to ensure that untrusted applications are able to perform meaningful amounts of functionality on a mobile device. These need to be standardized to support a common model and experience.
5. First class citizenship for web application. Web applications should be save-able as favorites or shortcuts directly accessible from within the mobile device shell.

Conclusion

This is an exciting time. Web applications have the potential to increase the capabilities of the mobile device world exponentially. In order to spur this innovation, we should act sooner rather than later to provide strong, meaningful standards-based solutions to the common problems we face.