Mobile Ajax and Application Adaptation

Volantis Systems Ltd. Position Paper

W3C/Open Ajax Alliance Workshop on Mobile Ajax, September 2007

Rhys Lewis, Ph.D
W3C Technical Architecture Group Member
Chief Technology Officer, Volantis Systems Ltd.
1 Chancellor Court
Occam Road
Surrey Research Park
Guildford GU2 7YT
United Kingdom
+44 (1483) 739775
rhys@volantis.com

Background

New devices that can connect to the Web continue to appear regularly. Most of these devices can be used while the user is mobile. Newer devices often have more advanced capabilities than the more traditional computer systems that are commonly used for Web access. For example, a significant number of new devices now have built-in support for the Global Positioning System [1] for determining their location.

The diversity of capability of such devices has been a challenge for authors of Web sites and applications. Not only do such devices vary in the markup languages and styling mechanisms they support, there are very significant physical differences between them too. Screen sizes, keyboard capabilities and the presence or absence of a pointing device are just a few of the many differences that can occur between devices.

Until recently, the browsers available on mobile devices have had far more restricted function than their counterparts on more traditional computing platforms. The limited resources available on mobile devices have made highly functional browsers difficult to develop. However, advances in device capabilities and in browsers the last year or so are changing this picture. Not only are there now browsers that can deliver a usable Web experience on mobile devices, those browsers provide client-side environments that can support the kinds of highly interactive applications that have made such an impact on traditional uses of the Web. In particular, the availability of reasonable amounts of storage and of client-side programming environments, such as JavaScript, have enabled Ajax-based applications to start to appear.

The Web Experience

The availability of browsers that can provide the function necessary to render existing Web pages is a major step forward. However, the ability to render the existing pages is just one aspect of the use of the Web for mobile applications. Differences in the physical capabilities of devices, and particularly in their displays and input devices, often make the use of traditional Web sites rather awkward. Although access to existing sites is valuable, their assumptions about the user's environment often reduce the user experience to utilitarian. It is rarely a delight.

Increasingly, applications are being created that specifically address the needs of mobile users. In some cases, such applications rely on client-side code modules created specifically for each kind of device. While this sort of approach enables very high quality user interfaces to be created, it is very expensive for developers. Many different versions of an application need to be created in order to support the wide diversity of mobile devices in the market place. Not only are there software differences between devices, there are significant hardware differences too. These affect the way that user interfaces need to be constructed. The challenge for application developers is to be able to create a compelling user experience on a wide variety of different devices without costs escalating.

Web technologies, and the Ajax approach in particular, seem able to provide the basis for lowering the cost of development of applications that can span a range of devices. Approaches, such as application adaptation, have been very successful in recent years in allowing Web sites to be created that can work well on a range of devices. Work is underway to extend the capabilities of such sites to include the kinds of interaction commonplace in desktop Ajax applications.

Application Adaptation

One approach to helping authors with this level of diversity in capability is based on content adaptation [2]. Authors create content in a device-independent manner, using markup languages such as DIAL [3]. Together with appropriate styling information and media, these materials are used by adaptation systems to create representations of Web pages suitable for the device accessing the site. Volantis has many years of experience in creating systems that enable content adaptation to be used to support literally thousands of different types of device.

Content adaptation has been very successful as a technique for allowing specially authored Web pages to be accessed on a huge variety of different devices. However, to date the focus of the approach has been on ensuring that pages can be rendered appropriately so that they are usable. With increasing capabilities in devices comes increasing interest in the ability not only to render pages but to enable novel, distributed applications. Volantis is already pursuing solutions that support such environments.

One of the interesting features of device-independent authoring approaches is their ability to retain the author's intent. Whereas languages such as HTML tend to focus on presentation, newer markup languages tend to retain more semantic information. The similarity between the challenge of delivering sites to a wide variety of different devices and the challenge of supporting users with disabilities has often been noted. The role of semantics in possible solutions has also been described (see for example [4]). A significant amount of semantic information is needed in order to render a page appropriately for a particular device and user.

Content adaptation addresses the diversity in device capabilities associated with page rendering. By extending the concepts of content adaptation, it is possible to adapt entire applications, not simply their content. We call this process *application adaptation.* It is the next logical step beyond content adaptation.

In application adaptation, not only is the rendering of content adapted for specific devices, the application behavior is as well. Authors specify application behavior in some abstract manner, perhaps as a set of policies that could differ between different delivery contexts. Adaptation processing then generates the appropriate combination of markup, scripting, styling and whatever else is required to implement that behavior on a specific device. This approach is an extension to the existing mechanisms used for content adaptation in which aspects of styling, layout and use of media are specified by so-called policies that authors use to tailor the resulting materials for different delivery contexts.

Application adaptation offers the possibility that for a significant range of features authors may be able to specify content, look, feel and behavior all without recourse to explicit programming.

Application Representations

Clearly, a key aspect of support for application adaptation is a representation that captures the author's intent not only for the look and feel of the application but also for its behavior. Some mechanisms that might provide the basis for this are already starting to appear. In the W3C, XForms [5] defines some key capabilities for application construction. The XML Binding Language [6] is another specification that may have a role to play in more abstract definitions of applications. It provides a way to link user interface artifacts with content models. It has been used, for example, to add novel user interface controls to XForms-based applications. Also, work on stand-alone widgets [7] is progressing. Widgets are units of function that have some associated behavior and semantics beyond simply presentation. Although currently rather limited, widgets point the way to abstractions that capture behavior in addition to presentation.

It is, of course, possible to create markup languages that mimic procedural programming languages. VoiceXML [8] is an example of a successful language that takes this sort of approach. However, Volantis feels that approaches ought to be possible where the intent of an author is captured in a declarative rather than procedural fashion. Languages like State Chart XML [9] seem to point the way

to alternative methods for representing applications in terms of states and transitions. These sorts of representation, which we refer to as 'declarative', seem to be not only suitable for representation in markup but also suitable for generation by graphical tools. For example, the state chart diagrams of UML [10] might provide an interesting graphical representation that could be converted into an appropriate, declarative application markup language. The recent Workshop on Declarative Models of Distributed Web Applications [11] showed widespread interest in this kind of approach.

The Role of Ajax in Application Adaptation

Although an aim of application adaptation is to reduce the need for authors and developers to create code explicitly, this capability itself requires access to client-side programming systems. The result of adaptation is a combination of markup, styling and code that runs in the browser to implement the author's intent. Compelling, highly interactive applications need access to Ajax capabilities whether they are created manually by developers or are the result of application adaptation.

Volantis products have been taking advantage of the increasing availability of Ajax on modern mobile browsers to deliver interactive components that provide enhanced application function and an improved user experience. Even though these components make relatively modest use of browser capabilities, nonetheless they support the ability to encapsulate significant application function in ways that are easy for authors to specify and control. This is an important point. It shows that there is significant value even in relatively limited implementations. Browsers can support these enhanced application capabilities even thought they might not, for example, be able to support Ajax applications that use the popular toolkits and libraries.

Summary

Volantis Systems is keenly interested in pursuing the notion of declarative models of distributed, interactive Web applications. We feel that the approach could provide the basis for widely available, compelling applications available on a wide variety of devices. We also feel that the approach may also enable improvements in access to sophisticated systems for people with disabilities.

The combination of a declarative authoring mechanism, based on current work at W3C, and appropriate Ajax implementations, toolkits and libraries seems ideally suited to providing powerful, interactive applications that are easy to author.

References

[1] Global Positioning System, Wikipedia (see http://en.wikipedia.org/wiki/Global_Positioning_System)

[2] Glossary of Terms for Device Independence, R. Lewis, 2005, (see http://www.w3.org/TR/di-gloss/)

[3] Device Independent Authoring Language (DIAL), K. Smith, 2006, (see http://www.w3.org/TR/dial/)

[4] The Meaning of 'Life': Capturing Intent from Web Authors, R. Lewis, 2006 Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility? (see http://portal.acm.org/citation.cfm?doid=1133219.1133231 – free ACM account required to access the full text)

[5] XForms 1.1, J Boyer, 2007 (see http://www.w3.org/TR/xforms11/)

[6] XML Binding Language (XBL) 2.0, I.Hickson, (see http://www.w3.org/TR/xbl/)

[7] Widgets 1.0, A. van Kesteren and M. Caceres, 2006, (see http://www.w3.org/TR/widgets/)

[8] Voice Extensible Markup Language (VoiceXML) Version 2.0, S. McGlashan et. al., 2004, (see http://www.w3.org/TR/voicexml20/)

[9] State Chart XML (SCXML): State Machine Notation for Control Abstraction, J. Barnett et. al. (see http://www.w3.org/TR/scxml/)

[10] Unified Modeling Language, Wikipedia, (see http://en.wikipedia.org/wiki/Unified_Modeling_Language)

[11]Declarative Models of Distributed Web Applications Workshop Report, D. Raggett, 2007, (see http://www.w3.org/2007/02/dmdwa-ws/report.html)