AJAX in Widgets and Web UIs

Position Paper by Guido Grassel
Nokia Research Center, Helsinki, mailto:guido.grassel@nokia.com
W3C / Open AJAX Alliance Workshop on Mobile AJAX [1]

We define a Web application as an application developed using HTML, CSS, JavaScript, XmlHttpRequest (XHR), DOM. The latter technologies are generally subsumed under the term AJAX. We differentiate three main use cases for Web applications and AJAX:

- Browser-based Web applications
- Widgets
- Web User Interfaces

A Widget is a single purpose Web application, of which at least parts are locally installed on the computer or device. A Widget run-time environment usually lacks the browser chrome. User interfaces developed using Web technologies, Web UIs, are basically larger Widgets, typically consisting many interlinked screens to provide a multitude of features to the users. Ultimately a Web UI provides the entire user interface of a device or computer.

Combination of browser chrome and Web application's own user interface elements, such as menus, forms, and content, can be confusing for inexperienced (=average) users who cannot tell the difference. For this reason we see absence of a Browser chrome as an advantage of Widgets and Web UIs compared to Browser-based Web applications. In our demonstration at XTech 2007 [2], we demonstrated two further advantages of Widget compared to Browser-based Web applications on mobile devices: A Widget may add its own custom menu items into the options menu and re-define softkeys. A Widget may implement its own, application-specific navigation scheme and disable the Browsers Web viewing and navigation method, such as Nokia Web Browser for S60 features thumbnail overview of the page (Mini Map), Visual History, and virtual pointer.[1]

Available Widget run-time environments loosen the tight sandboxing of Web applications running in the browser. They allow functionalities such as client-side persistent storage, interfacing to local applications or the operating systems, and execution of locally installed JavaScript code (i.e. JavaScript code installed as part of Widget installation to the device) in addition to JavaScript from one or several Web-sites. They allow cross-site scripting if only because the same origin rule cannot be applied to locally installed JavaScript code.

[1] I am happy to repeat the main points of my XTech 2007 speech and show the demo running on a Nokia device, if the workshop program committee sees value in this.

Mashing-up client functionality and data with Web-based services gives doubtless great new service opportunities. For instance, we demonstrated a Mobile Phonebook mashup of client contacts, Yahoo People Search and Google Maps in our presentation at the XTech conference [2]. However, opening the browser sandbox without putting adequate alternative protection mechanisms into place poses a security thread to users. Users put trust into the security of Web browsers, they expect that the browser protects their data and device. We argue it is hard to explain to average users that Widgets are less secure than browser-based Web applications and they should be more careful when installing a Widget to their device or computer than when using a Web application in the browser. Web run-time manufacturers and Widget portal operators must take Widget security. This requirement is especially important for Widgets on mobile devices, because of the increased level of trust most users put into these devices compared to desktop computers.

The security situation of Web UIs is similar to Widgets. A Web UI integrates a wide range of on-device functionality and Web-based services. For instance, a part of the Web UI may provide the UI for on-device contacts while another part integrates the user's Web mail account. It is useful in this case that on-device contacts can be used to select the recipients of an e-mail message. Hence, JavaScript code needs to have access to the contacts database interface. However, some JavaScript code included in a malicious e-mail message should certainly not have access to the same JavaScript interface.

We foresee that specifications on new JavaScript interfaces to on-device resources will eventually be desirable. At the same time the security models used by Widget and Web UI run-time environments should be defined. However, we believe that the mobile industry has not, yet, gained sufficient experience to start specification work. Therefore, we recommend that neither W3C nor the Ope AJAX Alliance undertake any specification work in this area in the near future.

[1] W3C / AJAX Alliance workshop on Mobile AJAX, workshop home page: http://www.w3.org/2007/06/mobile-ajax/

[2] Guido Grassel: Mobile phonebook mashup application developed using Web technologies, XTech, 15-18 May 2007: http://2007.xtech.org/public/schedule/detail/210