Novarra Position Paper for the W3C/OpenAjax Workshop on Mobile Ajax

Novarra is a provider of content transformation solutions for adapting Internet content for display on mobile devices. Among our products are a content transformation server that sits between the WWW server and the mobile device; and WWW browser clients that run on the J2ME and BREW platforms.

Novarra has an interest in the future of Ajax and Mobile Ajax for the following reasons:

- The use of Ajax on standard (i.e., non-mobile) web sites has become more common.
- It appears that the use of Ajax on sites designed for mobile use will become more common.

Novarra is especially interested in how Ajax (and mobile Ajax) sites and content transformation can co-exist and cooperate to provide a good mobile experience.

Ajax on Standard Web Sites

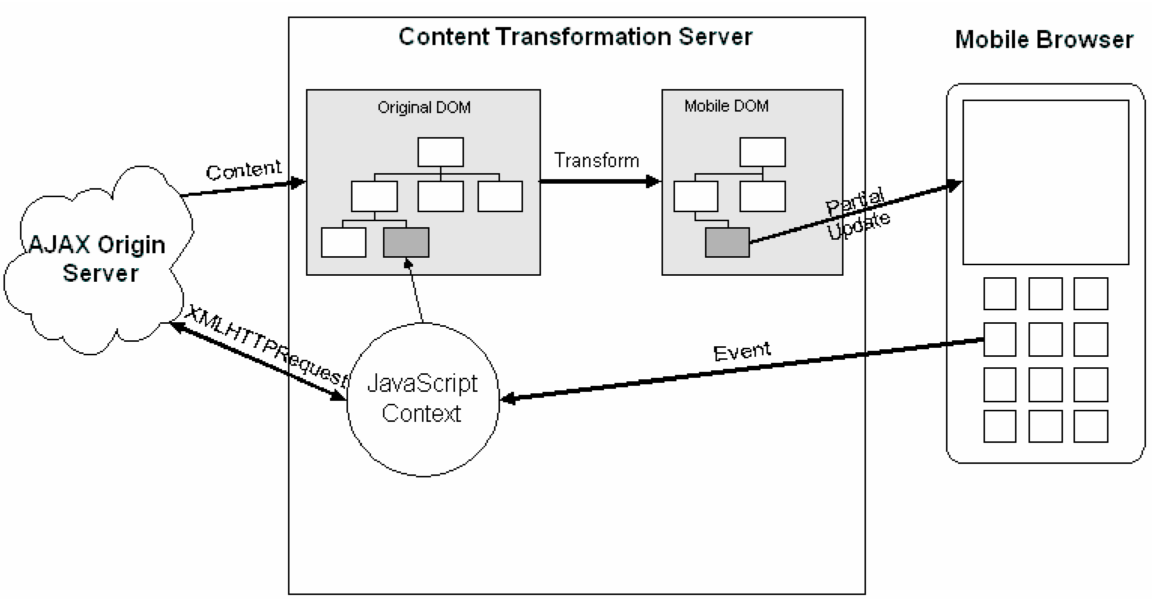
In the last two years the use of Ajax on standard web sites designed for desktop browsers has increased dramatically. This causes a problem for content transformation (CT) servers. CT servers today are mainly focused on adapting the HTML and CSS of a web page into a form that can be displayed in a useful and visually pleasing manner on a mobile web browser. Some JavaScript can be run on the CT server, such as document.write calls, certain handlers (onLoad, onClick, onSubmit), etc. However, dealing with Ajax and the XMLHttpRequest object (normally the core JavaScript object for Ajax) for a CT server is problematic because:

- Many, if not most, mobile web browsers don't support the XMLHttpRequest object. Another method that the browser supports (e.g., using IFRAMEs) must be used to request dynamic data to update the web page.
- Many mobile web browsers don't have the capability to update a portion of the currently loaded page dynamically. When there is a change, the entire page needs to be reloaded. This makes interactive Ajax pages difficult to handle in reasonable manner.
- Each request that is sent using the XMLHttpRequest object requires four hops instead of two—request from mobile browser to CT server, request from CT server to origin server, response from origin server to CT server, CT server to mobile browser. This can decrease the responsiveness of the web page if it makes a lot of Ajax requests. Of course, the CT server can perform some of the computation required to update the web page, so this could minimize any network delay caused by using a CT server.
- The DOM of the original page (held on the CT server) is different from the mobile DOM on the mobile web browser. For mobile web browsers that accept

partial updates to a web page, this requires that the CT server map what has changed in the original DOM to the mobile DOM and transfer just those changes to the mobile web browser.

- Many mobile web browsers do not have pointing devices and therefore cannot handle some common Ajax techniques such as mouse over and mouse dragging.

One possible solution to the above problems is to use custom mobile web browsers (such as the ones provided by Novarra) to implement Ajax-like support on devices that don't have mobile browsers powerful enough to handle Ajax content. These custom mobile web browsers can be designed specifically to work with the CT server, so they can minimize many of the problems mentioned above. For example, dynamic updating of content could be implemented on the custom mobile web browser without requiring a full JavaScript implementation to be available on the device. (See figure below.)



Ajax on Mobile Sites

Although still not common, the introduction of more advanced mobile web browsers such as the Access NetFront browser and the Safari browser on the Apple iPhone and high-end Nokia phones makes it clear that more Ajax web sites will be targeted toward mobile phone browsers. These browsers have more-or-less full HTML, CSS, and JavaScript capabilities, including the implementation of the XMLHttpRequest object.

As more and more mobile devices with advanced mobile web browsers become available, content developers will want to develop content that takes advantage of the Ajax capabilities of these browsers. However, there will still be a sizeable percentage of users with devices that do not have Ajax-capable browsers. A content transformation solution for Ajax will let mobile content developers avoid writing versions of their content for both Ajax-capable and non-Ajax-capable browsers.

The CT server problems listed above for standard web sites with Ajax will also apply for mobile web sites with Ajax. Some rules or best practices for development of mobile Ajax applications could mitigate some of these problems and make the handling of mobile Ajax sites significantly easier than standard desktop Ajax sites. These best practices could include standard labeling of content and/or JavaScript functions.

In summary, while the use of Ajax for mobile devices is still fairly light, its usage will become much more common in the future. Since there will be devices with non-Ajax enabled mobile browsers in heavy use for the foreseeable future, CT servers will be needed to bridge the gap. Guidance in this area from the W3C could help improve the mobile Ajax experience for a wide segment of users.