Understanding Web Augmentation

Oscar Díaz

ONEKIN Research Group, University of the Basque Country (UPV/EHU),
San Sebastián, Spain
`oscar.diaz@ehu.es`

Introduction

The increasing volume of content and actions on the web, combined with the growing number of "digital natives", anticipate a growing desire of more sophisticated ways of controlling the Web experience. Webies 2.0 do no longer take the web as it is but imagine fancy ways of customizing the web for their own purposes. So far, mashups are the forerunner exponent of this tendency where consumers (companies and laymen alike) come up with new applications by synergistically combining third-party resources. This presentation moves the focus to another approach: *"Web Augmentation"* (WA). Rather than creating a new application, WA builds on top of the rendering of an existing website. In some sense, WA is to the Web what Augmented Reality is to the physical world: layering relevant content/layout/navigation over the existing Web to customize the user experience. Unlike mashups, the purpose for WA is not so much coming up with a new application, but framing the new development within the Web experience of an existing website. Since this is achieved by third parties in a non-intrusive way, WA is a client-side technology: extensive use of *JavaScript (JS)* using browser weavers (e.g. *Greasemonkey*) or plugs-in. Rationales for WA include:

– addressing long-tail requirements. Minority usage patterns might not be worth considering in the general release of a website but even so, be catered for as plugs-in to be deployed in an individual basis. An example is *A Bit Better Remember-The-Milk (RTM)* [3]. This plug-in improves the navigation experience of the *RTM* website through a side navigation bar that speeds up specific ways to access to-do tasks. The most of users will be satisfy by the RTM website but this does not prevent long-tail demands from being served by augmenting the RTM website,

– affordance. A company might increase the affordance of its services by transparently embedding its offerings as parts of someone else's website. An example is the *Skype* add-on, [4] a plug-in that turns any phone number found in a web page into a button that launches *Skype* to call that number. The security company AVG provides another example. Its plugin *LinkScanner* [2] scans search results from *Google, Yahoo!* or *Bing*, and places a safety rating next to each recovered link that informs about the trustworthiness of the site,

– end-user customization. Skilful users might also adapt their frequently-visited websites to their specific needs. *BookBurro* [1] is a case in point. This is a plug-in for price comparison at Amazon. *BookBurro*'s developer is a frequent *Amazon* buyer that likes to check other online bookshops before the purchase. From this perspective, WA departs from more traditional personalization scenarios where the website itself either caters for the adaptation (i.e. Web Personalization) or provides the means for register users to configure their Web experience (i.e. Web Customization).

We anticipate a quick eclosion of these "augmentations" as Web users demand more sophisticated ways of controlling the Web experience. As an evidence of this impulse, a repositoy for augmentation scripts, *www.userscripts.org*, holds over 85,000 scripts. Despite these figures, WA is still an art without a clear definition of its aim, good practices or development guidelines.

The presentation advocates for a more rigorous WA development by proposing (1) a set of good practices, (2) an architectural pattern for WA, and (3), a case for an agile approach to WA development. A non-trivial Wikipedia augmentation is used as a running example. In so doing, we hope to pave the way towards empowering users and organization alike with principles and methodologies that make the Web a truly customizable space.

References

1. Andrews, B.: Book Burro (2010),
   https://addons.mozilla.org/addon/book-burro/
2. AVG: AVG LinkScanner - How it Works (2010),
   http://linkscanner.avg.com/ww.sals-how-it-works.html
3. Paprotsky, A.: A Bit Better RTM (2009),
   https://addons.mozilla.org/addon/a-bit-better-rtm/
4. Skype: Skype button in Internet Explorer or Firefox toolbar (2005),
   http://www.skype.com/intl/en/support/user-guides/toolbar?lang=en