Supporting Mobile Web Augmentation by end users

Gabriela Bosetti[1], Sergio Firmenich[1, 2], Gustavo Rossi[1, 2], Marco Winckler[3]

[1]LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Argentina
{gabriela.bosetti, sergio.firmenich, gustavo}@lifia.info.unlp.edu.ar
[2]CONICET, Argentina
[3] ICS-IRIT, University of Toulouse 3, France
winckler@irit.fr

Abstract. This article presents MoWA Authoring, an End User Development platform supporting the improvement of existing –usually third party– Web applications with mobile features. This enhancement is carried out by the addition of specific behaviours, mostly dependent on context values. The tool assists the user in the construction of applications by easily selecting the components that fit his needs. A series of forms allows selecting sensors, context values of interest for the application and digital counterparts, to define the augmentation layer. This layer can be composed of augmentation units, called augmenters, which are configurable through widgets that can be placed over the presentation layer of any Web application.

Keywords: Web Augmentation, Mobile Web, End-User Development

Introduction

A single Web application not always fits a precise user's need; lacking some required information or features. Web Augmentation (WA) [1] allows users to manipulate the Web according to their own requirements. In turn, mobile devices bring a new challenge: creating more comprehensive experiences by taking advantage of the devices' capability of sensing the context. Many sites already take advantage of such features or provide a native mobile counterpart, but we can still find many of them with a poor –or no– mobile Web counterpart. Mobile WA can help to add such features when required and End User Development (EUD) [2] techniques may help to add them on-the-fly by the users themselves. Although some approaches exist for augmenting Web applications [3] and even with mobile features [4][5], these are aimed and limited to developers or users with programming skills. Similarly, in [6], the system supports the automatic triggering of adaptations when a concrete context of use is perceived, but it does not contemplate EUD techniques either. A vast number of authoring tools allow the creation of mobile applications from desktop [7], native mobile [8] or Mobile Web environments [9], but none of them generates pure Web applications. All of them rely on some native component for their execution instead of reusing a standard and popular Web browser to which the user is possibly used. With this aim, we present Mobile Web Augmentation (MoWA) authoring [10], an EUD approach

empowering end users to create MoWA applications through a form-based process combined with composable visual widgets and live programming capabilities. In [10] we demonstrated its feasibility by conducting an experiment with 22 end users; results showed that the approach was not only feasible but promising, since users were able to complete, in average, 84% of the requirements of the experiment.

MoWA Authoring: the support tool

In our approach, augmentations are executed by MoWA applications, which are loaded and run by a Web browser weaver, deployed as a Firefox for Android extension. Such applications are notified by sensors when the context changes, and are capable of manipulating the Document Object Model (DOM) of any Web page through specialised augmenters that know how to interpret the perceived context values. At execution time, the sensors observe changes in the user's context. The common way is through the device's physical sensors, as shown in Figure 1. This way, when a change in the context is perceived, the sensed value is notified to the MoWA sensor, which in turn notifies to all its subscribed applications. Applications can be instantiated with different execution strategies that execute the augmenters under different criteria.

Broadly speaking, the process of creating an application consists of four steps, shown at the left of Figure 1. First, the user selects the sensors that the application will be subscribed to. He must also define which context values are of interest for the application, as a particular 3-axes value for the orientation. Then, he should define which will be the digital counterpart(s) to associate with the defined context values of interest, e.g. all the articles in an online newspaper. Finally, he must select and configure the augmenters of his preference, in charge of manipulating the DOM of the underlying application in consequence of its purposes and the sensed values.
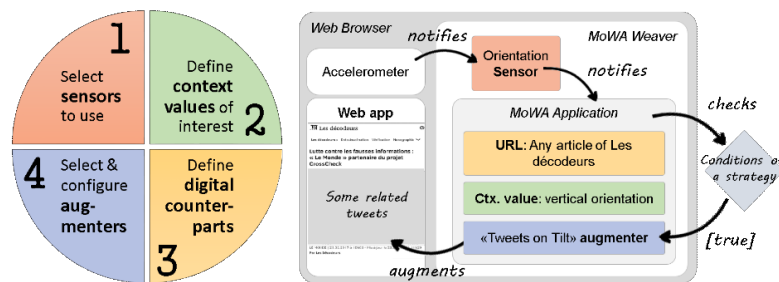

Fig. 1. Main stages of the authoring process and the expected application execution

A motivational scenario

Wayra is a regular reader of *Les Decodeurs*, a section of *Le Monde* specialised in fact-checking and contextualization of popular assertions. She reads it daily on her phone, and sometimes she wants to know what people say about a concrete topic. In Figure 2,

Wayra is reading about "CrossCheck"(a), but the article has just two comments at the bottom of the page (b). She wants to read more opinions so, in this kind of situation, she used to copy the title, open a new tab, visit Twitter and perform a search using the copied text as keywords. However, this process became tedious for her and she eventually stopped doing it. She would like to retrieve tweets in the same context of the article easily. Besides, the extra interactions make her feel uncomfortable to perform it with a single hand, and she would like to use single-handed tilting gestures to show and hide an extra section with related tweets at the top of the article, as in steps "c" and "d".

Fig. 2. The original mobile Web site and the augmented version

MoWA authoring empowers Wayra to create her own solution; Figure 3 shows the screenshots of the main stages of the process. It starts when the user opens the tool in the browser (0) and the full-screen mode is enabled. He chooses to create a new application and sets up some base data. Then, he selects his required sensors (1), and manages the context values of interest according to the selected sensor, as an orientation in (2). In such a concrete case, values for the axes are automatically updated; the user should move the device to the desired position and save it. Then, he defines at least one digital counterpart in (3) and eventually sets up the augmentation layer (4).
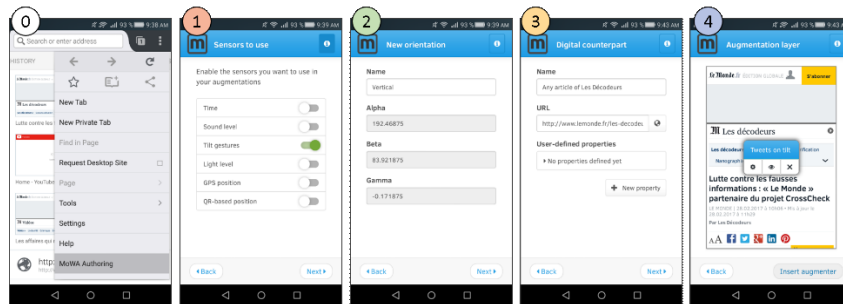

Fig. 3. Main stages of the authoring process through the tool

He can use as many augmenters as required. The last screenshot shows that an instance of the *TweetsOnTilt* augmenter has been created. This augmenter is capable of

showing related Tweets through a sliding down effect that –as a physical metaphor of an object sliding on the surface– makes the interaction memorable for the user. The augmenter just requires the user to pick a DOM element for generating an XPath, so each time the document is loaded, the DOM element can be retrieved and its *textContent* attribute can be used as the keywords to search on Twitter.

Conclusions and Further Work

This paper presents MoWA Authoring, a platform supporting an approach for enhancing Web applications with mobile features. The tool currently supports a set of extensible and specialised sensors, context types and augmenters for covering scenarios in different domains. We are currently extending the platform with new augmenters to provide the end users with a broader spectrum of specific behaviours and making them available from a repository for easy maintenance. For more information about the project, please visit the site of the project[1].

Acknowledgments

References

[1]   O. Díaz and C. Arellano, "The Augmented Web: Rationales, Opportunities, and Challenges on Browser-Side Transcoding," *ACM Trans. Web*, vol. 9, no. 2, pp. 1–30, May 2015.

[2]   H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-User Development: An Emerging Paradigm," in *End User Development*, Dordrecht: Springer, 2006, pp. 1–8.

[3]   O. Díaz, C. Arellano, and M. Azanza, "A language for end-user web augmentation," *ACM Trans. Web*, vol. 7, no. 2, pp. 1–51, May 2013.

[4]   D. Carlson and L. Ruge, "Ambient Amp: An open framework for dynamically augmenting legacy Websites with context-awareness," in *2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014, pp. 1–6.

[5]   W. Van Woensel, S. Casteleyn, and O. De Troyer, "A generic approach for on-the-fly adding of context-aware features to existing websites," in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia - HT '11*, 2011, p. 143.

[6]   S. Ceri, F. Daniel, F. M. Facca, and M. Matera, "Model-driven Engineering of Active Context-awareness," *World Wide Web*, vol. 10, no. 4, pp. 387–413, Oct. 2007.

[7]   F. Corvetta, M. Matera, R. Medana, E. Quintarelli, V. Rizzo, and L. Tanca, "Designing and Developing Context-Aware Mobile Mashups: The CAMUS Approach," Springer International Publishing, 2015, pp. 651–654.

[8]   F. Pittarello and L. Bertani, "CASTOR: learning to create context-sensitive and

[1]   http://www.goo.gl/kQyKe8

emotionally engaging narrations in-situ," in *Proceedings of the 11th International Conference on Interaction Design and Children - IDC '12*, 2012, p. 1.

[9] C. Cappiello, M. Matera, and M. Picozzi, "A UI-Centric Approach for the End-User Development of Multidevice Mashups," *ACM Trans. Web*, vol. 9, no. 3, pp. 1–40. 2015.

[10] G. Bosetti, S. Firmenich, S. E. Gordillo, G. Rossi, and M. Winckler, "An End User Development Approach for Mobile Web Augmentation," *Mob. Inf. Syst.*, vol. 2017, pp. 1–28, 2017.