

# WebMakeup: Empowering Users to Mod Websites

Oscar Díaz<sup>1</sup>, Iñigo Aldalur<sup>1</sup>, Cristóbal Arellano<sup>1</sup>, Haritz Medina<sup>1</sup>,  
Sergio Firmenich<sup>2</sup>

<sup>1</sup> University of the Basque Country (UPV/EHU), San Sebastián (Spain)  
(oscar.diaz, cristobal.arellano, inigo.aldalur)@ehu.es,

<sup>2</sup> LIFIA, Universidad Nacional de La Plata and CONICET (Argentina)  
sergio.firmenich@lifia.info.unlp.edu.ar

**Abstract.** Modding refers to the act of modifying hardware, software, or virtually anything else, to perform a function not originally conceived or intended by the designer. The rationales for modding should be sought in the aspiration of users to contextualize to their own situation the artifact at hand. Websites are not exception. *WebMakeup* targets mod scenarios where web pages are turned into canvases users can tune to account for their situational, idiosyncratic, and potentially, short-lived needs. By clicking, users turn DOM nodes into widgets. Widgets can next be rearranged, deleted, updated, or stored for later reuse in other pages. In addition, widgets can be involved in “blink” patterns where interactions with a widget might affect the related widgets. This empowers users to tune not only *what* but also *when* content is to show up using AJAX-like approaches. *WebMakeup* is publicly available as a Chrome extension.

## 1 Goals

Web content should be unleashed from how it is canned by Web masters. Modding refers to the possibility of users to tune web content and interactions to fit their own patterns. This vision accounts for a post-production user-driven Web customization. Being user driven, appropriate abstractions and tools are needed. As an example, consider *www.tvguia.es*. This website provides the channel grid plus the-movie-of-the-day recommendation (see Figure 1 (left)). A user might just focus on some few channels, hence a thorough channel grip might imply a lot of scrolling if your favorite channel is down the grip. In addition, content from other websites about the recommended movie might be of interest. Figure 1 (right) depicts a modded version: channel “*La 1*” is removed whereas additional content about the recommended movie is obtained from *www.filmaffinity.com*.

The bottom line is that mod scenarios are characterized as being idiosyncratic, situational, and, potentially, short-lived, aiming not so much at synergistically combining third-party data but improving the user experience of existing websites. Since these scenarios are very dependent on Web consumption habits and user interests, modding necessarily has to be do-it-yourself (DIY).



Fig. 1. *Tvguia* before (left) and after (right) being modded.

Mashup Feature Checklist		Mashup Tool Feature Checklist	
Mashup Type	UI mashup	Targeted End-User	Non Programmers
Component Types	UI components	Automation Degree	Semi-automation
Runtime Location	Client-side only	Liveness Level	Level 3
Integration Logic	UI-based integration	Interaction Technique	WYSIWYG
Instantiation Lifecycle	Short-living	Online User Community	Private but sharable

Fig. 2. *Characterizing WebMakeup*.

## 2 Proposed Solution

*WebMakeup* is an editor for Web modding. Figure 2 characterizes *WebMakeup* as a UI mashup tool which pivots around the notion of “widget”. Next, we describe the solution with the help of the *tvguia* example. A demo video is available at <http://onekin.org/downloads/public/WebMakeup/video.mov>.

**Widget Creation.** *WebMakeup* is a plugin for *Google Chrome*. Its installation is reflected by the *WebMakeup* button at the right of the address bar. On clicking this button, a scrollable menu pops up. Click “*New*” (see Figure 3): (1) the current page is turned into an editor canvas where the pointer is turned into a camera, (2) a grid-like structure is interspersed on top of the current DOM tree, and (3) two tabs pops up (i.e. the *piggyBank* tab and the *patterns* tab). Broadly, a widget is a singularized DOM node with an additional functionality provided through a decorator. Widgets can be internal or external. The former are those obtained from the website being modded (through the *camera* pointer). By contrast, external widgets are those obtained from somewhere else, and left in a “piggy bank” (available through the namesake tab). External widgets can be obtained from a single element or as an aggregate of different elements (i.e. HTML fragments) all coming from the same page. External widgets are created through the right-click contextual menu.

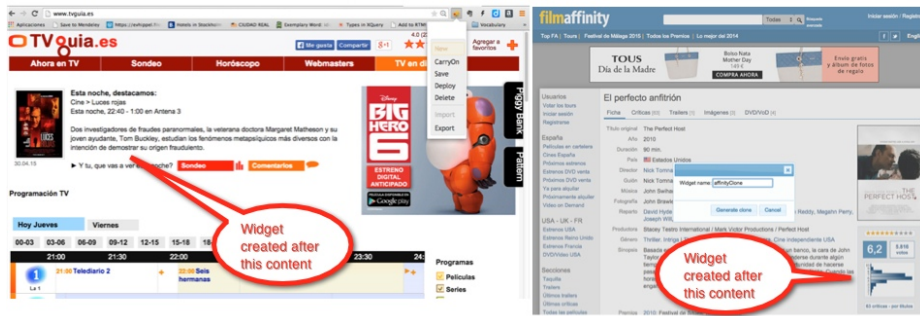


Fig. 3. Widget creation: internal widgets (left) versus *external* widgets (right).

**Widget operations.** Once created, widgets can be deleted, move around or parameterized. Widgets can have parameters and a state (i.e. visible or collapsed). Parameters are automatically derived from the underlying HTML fragment and it includes labels, entry form parameters, refresh polling frequency, etc. Parameter assignment can be by value or by reference. By value refers to the user manually providing the value. By reference involves the system automatically retrieving the value by applying an XPath upon the hosting page at runtime. XPaths are derived from user interaction upon the host page at parametrization time. Users do not need to know XPath. Click twice for widgets to show up their parameters.

**Widget Animation.** Widgets can be in two states: visible or collapsed. *Blink* relationships can be set between widgets so that interactions upon a widget can impact another widget's state. *Blinks* can be set manually or semi-automatically. The former involves drawing a pipe from the triggering widget to the triggered widget *à la* Yahoo Pipes (see Figure 4). The default triggering event is *click*, though users can select other DOM events. Semi-automatically obtained blinks are thought to ease the definition of common patterns of blink relationships that involve several widgets. Here, users first choose the widgets (ctl+alt), and next, click on the selected pattern available in the namesake tab. *WebMakeup* automatically generate the blinks that jointly account for the pattern at hand.

### 3 Level of maturity

WebMakeup is available at the *Chrome Web Store*: <https://chrome.google.com/webstore/detail/alnhgodephpjnaghlcemlnpdknhbhjj>. Usability studies have been conducted and reported at [1].



Fig. 4. Setting blinks between widgets. Widget below will be visible after clicking on the widget above.

## Acknowledgments.

This work is co-supported by the Spanish Ministry of Education, and the European Social Fund under contract TIN2011-23839. Aldalur has a doctoral grant from the Spanish Ministry of Science & Education.

## References

1. DÍAZ, O., ARELLANO, C., ALDALUR, I., MEDINA, H., AND FIRMENICH, S. End-user browser-side modification of web pages. In *Web Information Systems Engineering - WISE 2014 - 15th International Conference, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part I* (2014), pp. 293–307.