MAWA: A Browser Extension for Mobile Web Augmentation

Iñigo Aldalur, Alain Perez, Felix Larrinaga

▶ To cite this version:

HAL Id: hal-04215528

https://inria.hal.science/hal-04215528

Submitted on 22 Sep 2023

MAWA: A Browser Extension for Mobile Web Augmentation

Iñigo Aldalur, Alain Perez, and Felix Larrinaga

Mondragon University, Faculty of Engineering, Loramendi 4 Arrasate, Spain
{ialdalur,aperez,flarrinaga}@mondragon.edu

Abstract. In recent years, the usage of mobile browsers has experienced an astonishing growth. Nowadays, most citizens use their mobile phones instead of their laptops to surf the Web for immediate availability. Nevertheless, Web design is performed considering laptops screen dimensions and websites are readjusted to mobile screen resolutions using Responsive Web Design. This conversion to smaller screen resolutions causes some drawbacks to mobile navigation. Web Augmentation is an effective methodology that allows end-users to customize third party websites according to their needs. This technique can mitigate the problems caused by small screen resolutions.

This article introduces MAWA, a Firefox mobile browser extension for Web Augmentation designed to remove and move content in any website using a visual programming technique. The article presents the benefits introduced by MAWA in response to the drawbacks arising when websites are adapted from laptop screen resolutions to mobile dimensions. Evaluation is also introduced in this article where end-users with no programming knowledge have adapted four different websites using the extension. Results show that testers claim that drawbacks presented by mobile browsers are mitigated with the utilization of MAWA.

Keywords: Human-Computer Interaction · End-User Development · Web Augmentation · Customization · Mobile Browser

1 Introduction

In recent decades, there has been an important increase in the use of mobile phones and, as a consequence, a growth in the access to web applications [1]. Several studies predict that the next decade will bring a significant increase in the demand for mobile data [2]. Boehm et al. [3] foresaw in 1995 that in 2005 there would be 55 million of Web practitioners in the U.S. In 2005, Scaffidi et al. [4] estimated 90 million end-users in the U.S by 2012. If we extrapolate this quantity worldwide, we can imagine that nowadays (2020) the real number is uncountable. In fact, today more than half of the Internet traffic is mobile. This increase raises the partaking of end-users in the customization of websites to their desires and necessities [5]. The customisation of web pages allows end users to redefine colours, relocate components, remove features or add new elements.

In other words, to adapt the functionality, content and design to the user's wishes. The customization process is mainly carried out by users with no programming skills, as most web users around the world are mere consumers with limited programming knowledge. A very popular technique for customizing existing third-party websites is Web Augmentation (WA). This technique has been developed as a method to customize or adapt the features of websites without affecting the server-side code [6]. Furthermore, augmentation is not developed by the website owner but by the website user. There are different approaches to realising WA, but one of the most popular is the use of visual programming techniques [7]. Visual programming can be defined as the use of graphics as the programming language [8].

Lieberman et al. [9] define End-User Development (EUD) as "a set of methods, techniques and tools that allow users of software systems, acting as non-professional software developers, to create, modify or extend a software artefact". End-users are able to start with basic adaptation mechanisms and gradually advance to use more powerful adaptation mechanisms without facing insuperable barriers [10]. WA technique is very appropriate because it can be applied progressively. WA can be applied to any website and it is accomplished by end-users. The same drawbacks experience by desktop users affect mobile users. Problems can even increase since mobile versions are, in many cases, adaptations of desktop versions.Consequently, WA not only can be used for desktop web versions, but mobile web versions can also be customized by end-users. Mobile users need tools and strategies to meet their needs when browsing. This paper investigates WA as a mechanism for customising web pages by end users through a Firefox mobile browser extension for web adaptation.

The remainder of the paper is organized as follows. Section 2 analyses the problem analysis including the issue we want to solve, its causes and its consequences. Section 3 discusses related work in order to give the reader an idea of what has been done in this area from WA perspective. Section 4 describes MAWA[1] (Mobile Application for Web Augmentation), a browser extension that customizes mobile version web pages to adapt them to users requirements. Section 5 presents MAWA evaluation and its results based on different questionnaires (NASA-TLX, SUS and their satisfaction), section 6 shows features we would like to enhance about MAWA and section 7 concludes the paper.

2   Problem analysis

2.1   The problem

Most web designers prepare their initial web configuration for desktop resolution and later think about how to adapt this design to the mobile paradigm [11]. This process is frequently carried out with the Responsive Web Design (RWD). RWD is a design and implementation technique that enables web designers and developers to create websites with a flexible approach to adapt their content and

---

[1] https://addons.mozilla.org/es/firefox/addon/mawa/

layout to different resolutions on different devices with the same web experience [12]. This web restructuring is not free. It comes with a computational cost as the initial desktop version has to be adapted to the configuration of the mobile browser. Furthermore, this adaptation causes usability problems such as the display of unnecessary content, the reordering important content and, as a consequence, cluttered navigation and scrolling to find the desired information, among others [13]. These usability drawbacks induce a poor web experience for users.

2.2   Causes

In the beginning, websites were designed for desktop computers because this was the supported platform for browsing the Web. However, over time, technology has evolved and mobile phones and tablets are common platforms for consuming websites via navigation browsers[14]. Due to this natural evolution, web designers had the habit of designing first for desktops computers where large screen resolutions were required. Later, they would design another website for mobile browsers while maintaining the same style [15]. RWD avoids creating two different versions of the same website and was created for desktop-to-mobile adaptation and not vice versa even though it is common to listen that it should be "mobile first design" [16]. Not only website designs are considered first, but also other aspects, such as the WCAG 2.0 guidelines for desktops, which are created only for desktops and designers have to decide how to proceed for mobile consumption [17]. Desktop version design first causes problems and disadvantages to mobile web browser edition [2] that can be mitigated with our proposal.

2.3   Consequences

The negative consequences of the problem previously addressed on users experience are:

- Scroll: scrolling up and down a page without reading the content may be a signal of frustration and lack of confidence, which is a frequent action on mobile browsing [18]. In relation to this, [19] concluded that when users frequently scroll to find irrelevant information in their navigation, they are less satisfied. Furthermore, "the current scrolling method for a mobile device is both time-consuming and fatigue-prone" [20]. So, considering that [21] conducted a study in which they determined that people used the scroll bar on 76% of the pages, and that 22% scrolled to the end regardless of page length. It is extremely important to avoid scrolling in mobile phones.
- Show unnecessary content and hide important content: All mobile web users want to extract the desired information quickly and with minimal mental effort [22]. RWD relocates all content shown in the desktop version hiding some content and maintaining other content visible in the mobile version [12]. Unfortunately, this content may be useless to readers if the desired content is concealed or the user has to scroll to find it.

– Produce cluttered navigation: websites are changed from desktop version to mobile version by adapting the layout and rearranging the content to the size of screen [14]. This step of adapting the website generates different drawbacks such as the obstacle of reading the content correctly from the device [23], which causes more user interaction than for desktop users and consequently the time needed to carry out any task is higher [24]. In addition, it generates more input errors than using the desktop navigation [25].

3 Related work

Bouvin [6] originally coined the term WA in 1999 to describe a tool that "through integration with a Web browser, a HTTP proxy or a Web server adds content or controls not contained within the Web pages themselves to the effect of allowing structure to be added to the Web page directly or indirectly, or to navigate such structure. The purpose of such a tool is to help users organize, associate, or structure information found on the Web. This activity can be performed by a single user or in collaboration with others". Our work focuses on WA and consequently this section analyses the different works accomplished in this concern. On the one hand, an important number of references focus their research in WA for computers. For example WebMakeup [26] is a visual programming Chrome browser extension. End-users can copy different web nodes by clicking on them with the mouse. When using this extension to modify a website, users can insert previously copied nodes from a "Piggybank" by dragging & dropping. Web-Makeup permits updating inserted nodes with actual information or not. Additionally, inserted nodes can be collapsed by clicking on certain node defined by the user. MDWA [27] proposes a novel method for designing WA applications based on client-side and server-side components. They propose a model-driven approach that can improve the abstraction level of client-side and server-side development. They provide a set of tools to design the composition of core applications. These applications present new back-end functionality to support WA. The front-end has the main function of augmenting the web page. [28] shows a tool that proposes a data acquisition system capable of capturing the user interaction in web interfaces. Subsequently, these interactions can be reproduced automatically without any human action, which can be a method to reduce the time to perform an activity. Similarly, Excore [29] is designed to add content from repetitive web searches. It detects a keyword on a defined website, automatically carries out the web search on a different website and inserts the desired content on the target site. This automation is defined by an end-user through a visual programmable Chrome browser extension. Chudnoskyy et al. [30] take a step forward in creating web adaptations by assisting users with recommendations and automatic composition. It provides a composition environment for end-users supplied by different elements, which can be placed on the canvas, synchronized with each other and shared with other users. [31] customize websites depending on the disability a person has (visual, motor, cognitive or hearing impairments) and the attitude towards computers and the Web older people show. [32] in-

troduces a adaptive hypermedia method suitable for adaptive link annotation. This method shows a combination of direct manipulation and automatic linking annotations that allow users to better control the adaptability of the page. This direct control can better support users' attention to information during a task. [33] has design a system in which the webmaster starts with the existing layout and specifies which elements must be modified. Then, touch-based events will be used as implicit input for the adaptive engine, which automatically modifies, rearranges, and re-styles interactive items based on browsing usage. OFIE [34] relies on Programming by Demonstration to define the appropriate native input, which must be executed automatically when the rule is triggered. This method allows end users to define operations simply by performing required native input interactions on the graphical user interface of the application. End-user does not need any necessary programming knowledge or write/edit any code. Trusty et al. [35] developed ALOE, a Firefox browser extension that augments Web pages by translating all selected English words with their translations to defined languages.

On the other hand, research has also been conducted in WA for mobile phones. For example, PageTailor [36] is a Firefox-based browser that runs on Windows Mobile PDAs and is implemented as a Mozilla plugin API. The tool provides a technique for the selection of objects that by dragging a line over the object selects the web node associated. PageTailor permits the customization of traditional websites mainly by increasing the size of important web elements and diminishing the size of irrelevant content. PageTailor does not use Responsive Web Design. Collapse-to-Zoom [37] is a much simpler tool than PageTailor since it only allows expanding and collapsing web elements. It is also designed for modifying traditional websites in PDAs. If the user taps over an element and drags the pen diagonally to the top, the node is expanded. If the movements is towards the bottom, the element is collapsed. MoWA [5] is a mobile browser extension for augmenting existing Web Applications with information that the original web page does not provide. It adapts the website content depending on information obtained from the mobile sensors (GPS, microphone, gyroscope, light...).

Independently on the target aim by these applications (desktop or mobile), their final objective is to enable users in adapting their favourite website to their needs. In our approach, the scope focuses on mobile browsing and the main objective is to ease end users navigation experience by adapting websites to their needs using WA.

4 Mobile Websites Web Augmentation

How often do you use your mobile phone browser to look for information on the web? Nowadays it is more common to use mobile phone browsers than desktop browsers[2]. Therefore, it is utmost importance to adapt mobile version websites to the needs of users by avoiding or minimizing the consequences presented in

---

[2] https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet

section 2. We will illustrate the use of MAWA with some examples with the conference website.

MAWA is a Firefox mobile browser extension for WA designed to remove and move content from any website through a visual programming environment. MAWA abstracts technical details into a visual domain-specific language that facilitates end-user involvement. The goal of this extension is to avoid scrolling, cluttered navigation and the removal of unnecessary content. In MAWA there are two different phases: (1) the editing phase, in which the end-user adapts the website, and (2) the execution phase, in which the system automatically completes the adaption by itself.
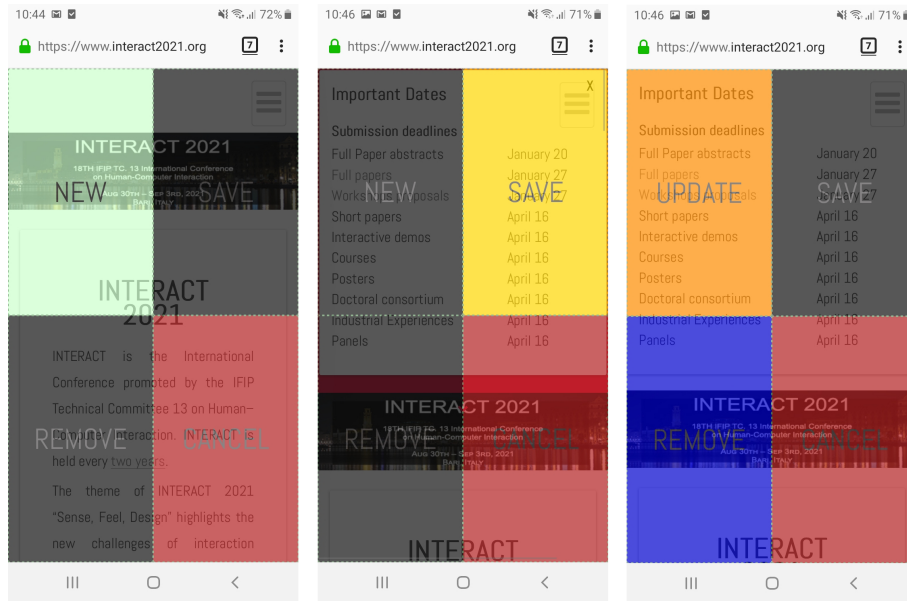
4.1 Edition stage

The edition stage begins when the user clicks on the *new* menu button. First, the user browses a website and if they decide that this must be adapted, they double-click on the screen where the menu appears (figure 1 left). The user then begins to edit the website just clicking on the node they wish to update. Once the user clicks on a node, its background colour is changed adding an X button to the top right and arrows to its top, bottom, left and right. From this point on, three different actions can be performed, (1) delete the node, (2) relocate the node and (3) select a relative node using the arrows. At the moment the web page modification is completed, the end user double clicks on the screen and the second menu pops up (figure 1 middle). When the save button is clicked, the web page is reloaded and the modifications are automatically made visible. To remove the modification or to continue editing the website, the user has to double-click on the screen and the third menu option appears (figure 1 right). Note that each menu button has its own color when active and its color never changes, except for deactivated buttons which are shown in gray.

Remove: MAWA allows you to remove any node from the web to avoid consuming unnecessary content. Websites may present information that is not relevant to the user. This information may not be annoying on computer screens, however, it is frustrating when consumed on small cell phone screens [24] [25]. If the user decides to remove the content of a node, the user must click the X button located in the upper right corner of the node (figure 2 left). Once clicked, the node is removed from the web page (figure 2 middle). If the user is not interested in some of the conference tracks, they would remove tracks using MAWA and only consume the ones of interest (figure 2 right).

Move: MAWA also allows you to move nodes from one point on the web page to another. This is done by dragging and dropping the selected node. When the node is dragged, the background colour where the node could be dropped becomes purple. This colour transformation provides the user with information as to what the new location of the node might be. Once the node is dropped, it is inserted into its new locations and removed from the original position. Figure 3 shows at the top left how the user has selected the important dates of the conference. The same figure shows how the user has relocated the node to the top of the web site (figure 3 top right). Figure 3 shows at the bottom how the

Fig. 1. MAWA menus: (left) initial menu, (middle) menu after edition stage, (right) menu after execution stage
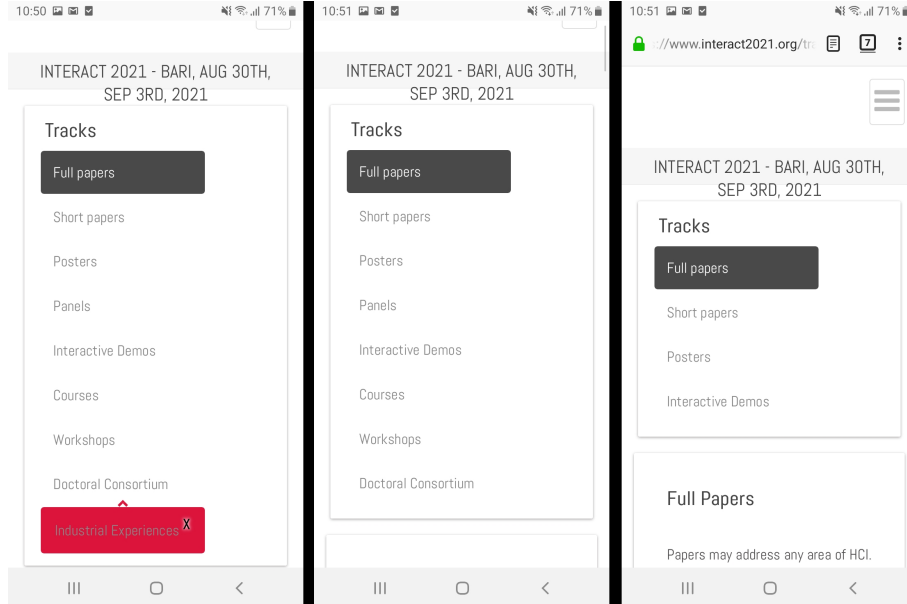

Fig. 2. Removing nodes with MAWA: node selection (left), node removal (middle), removing unessential tracks for the user (right)

important dates are located at the top right of the laptop version. However, in the mobile version it is located at the bottom of the page. All important information for a user should be at the top of the page because it avoids scrolling by avoiding the drawbacks explained in section 2. Figure 3 demonstrates with an example how MAWA helps to avoid scrolling in case essential information is at the bottom and, consequently, also to avoid cluttered navigation.

Node selection: Users have to select a node in order to move or remove it. Simply click on the node to select the desired element. However, some nodes are not so easy to select due to their particular definition. For example, the body of the page is one of these elements. If we want to select the body, it is complicated to click in the upper left edge of the page to select it. Other examples are selecting complete tables or interactive nodes when the user does not want to perform any action on the node. For this reason, MAWA includes an arrow on the top, bottom, left and right of the selected node. If the user clicks the arrow at the top, the parent node is selected, the arrow at the bottom selects the first child, the arrow on the left the previous sibling node and the arrow on the right the next sibling node. If the node has no children, the bottom arrow will not be included. Similarly, if the node has no siblings before, the left arrow will not be included and if the node has no siblings after, the right arrow will not be included.

4.2   Execution stage

One of MAWA's goals is to reproduce the user's creation steps one by one. For this reason, MAWA saves information about each modification made by the user in the editing process. The automation process finds the first modified element and reproduces the user's actions by deleting and relocating all nodes. This process continues until the last edited node in the same order as the user followed in the editing steps. At the end, the layout will be the same one the user had at the end of the editing process. The main drawbacks of the automation process are website updates. Website updates can cause MAWA to not be able to find the modifiable nodes. As a consequence, the automation fails. This failure can occur in two different cases. The first is when the node to be deleted or moved has been deleted from the website or relocated. The locator (mechanism used to locate nodes within a website) is not able to find the node. In this case, if the node is to be deleted, the user does not perceive any error. Nevertheless, if the node is to be moved, the user may notice that an important element is missing. The second case is when the node selected as reference to relocate the node has been removed or relocated. In this case, the system is not able to relocate the node and it seems that the node simply has been removed from the website. If one of these cases occurs, the end user could update the adaptation double-clicking on the screen and selecting update (figure 1 right).
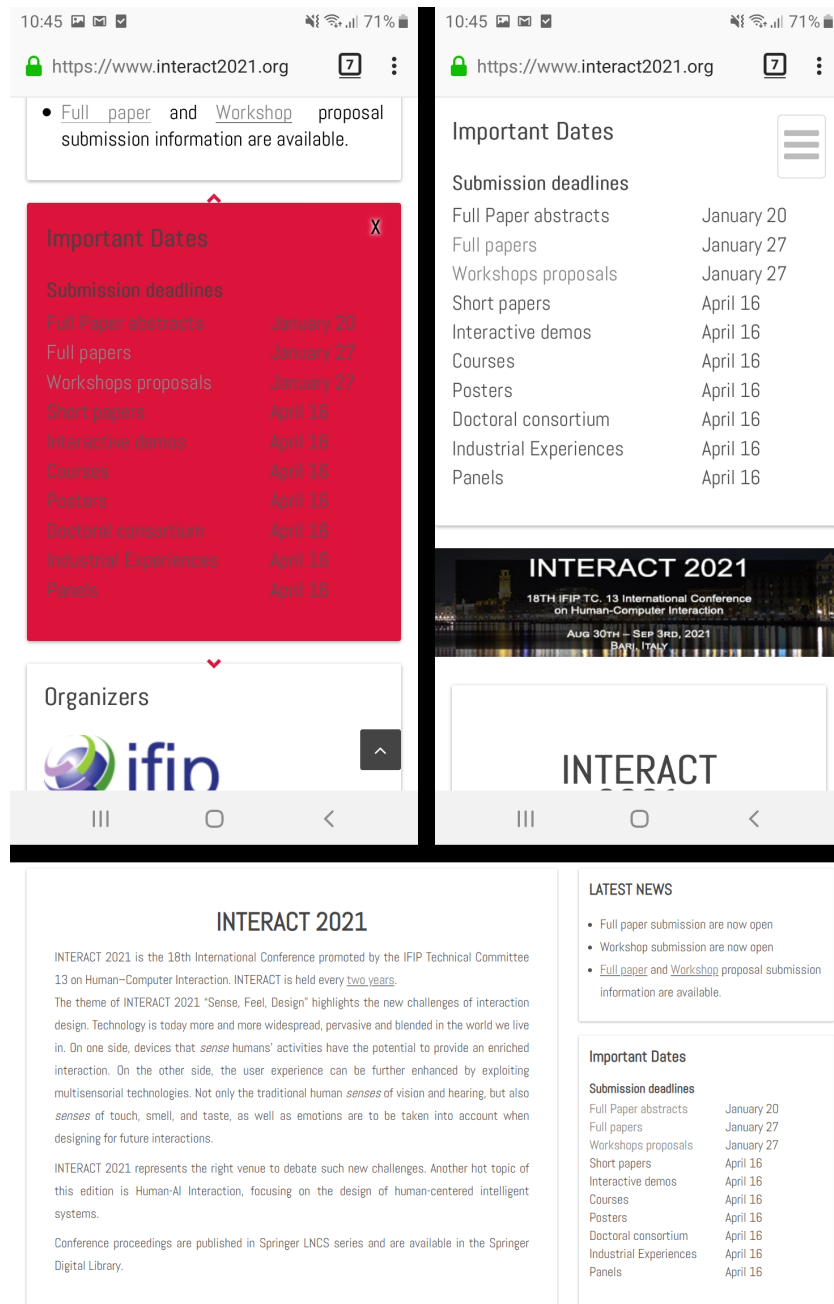
Fig. 3. Moving nodes with MAWA: selected node before being moved (top left), the node relocated on top of the website (top right), selected node on the laptop version (bottom)

4.3   Architecture

MAWA architecture has been designed taking into account the information the system must store during the editing process and the information needed during the execution process. We take advantage of the browser structure to store this information in it. The architecture is illustrated in figure 4.

Each adaption registers an ID created exclusively for it and the URL in which the adaptation should be executed. When a web page is loaded, the system checks whether this URL has been used in an adaptation. If this URL is found, the system starts the whole process to perform the complete adaptations. Otherwise, the system waits until another web page is loaded. MAWA allows to have more than one adaptation to the same domain but only one for each subdomain.

Each adaptation has one or more nodes that will be modified in the editing process. All nodes will be identified by an ID. Each node will be populated with its location point on the page and the actions that MAWA should complete when the adaptation is enacted.

The location point is populated with different locators. A Web locator can be defined as a mechanism for uniquely identifying an element in the Document Object Model (DOM) [38]. As explained before, website updates pose a major drawback for WA tools, since nodes might no longer be found. To address this drawback, based on some previous works using multiple locators to extend the life expectancy of locators [39][40][41], three different locator algorithms have been implemented. These locators are based on Xpath and generate different results when they fail, being able to use another Xpath to try to find the element. If one of them finds the element, we can use the new information to repair the previously broken Xpaths. Over time, the website developers will update the website and, therefore, these three Xpaths will fail. For this reason, an algorithm able to regenerate Xpaths has been implemented. This idea has been extracted from previous works [42][43][44]. To generate a new Xpath, MAWA stores additional information of each adapted node and all its ancestors. This information is completed with the all their attribute names and all of their values. When previous Xpaths fail, this algorithm is executed. It generates different Xpaths with the stored information until one of them locates a single node inferring that it was the desired node. The evaluation of the algorithm shows that on very few occasions an invalid Xpath is generated.

As a conclusion, this algorithm system improves the life expectancy of the adaptation and the user does not need to re-adapt their augmentations frequently. Some website updates are invisible for users because the website maintains its appearance. However, its structure may have been modified and Xpaths are sensitive to these changes. Website upgrades may create an idea on users that MAWA is of low quality and they might stop using it. Apparently, the website has not been modified but the application does not work properly. This is why it is of utmost importance to implement a robust locator system to improve the quality of the application.
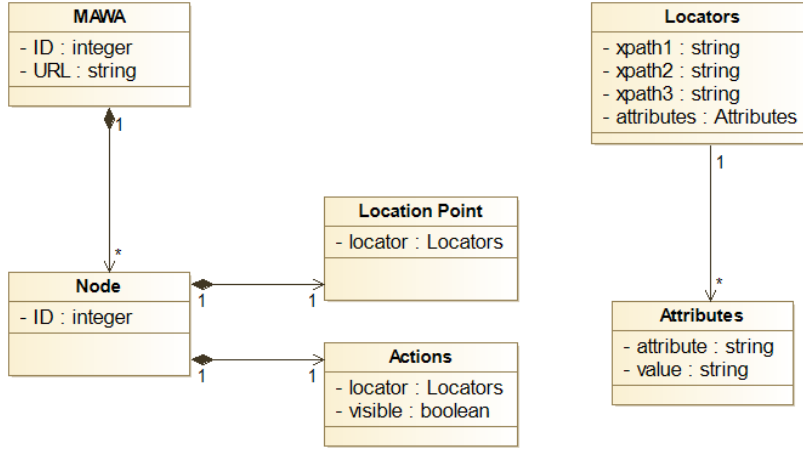
Fig. 4. MAWA abstract syntax

5   Evaluation

The main objective of MAWA extension is to avoid scrolling, cluttered navigation and removing unnecessary content by adapting your favourite or more visited websites. Allowing users to adapt content to their needs improves usability [45]. In order to validate whether MAWA accomplishes its purpose, we have evaluated MAWA following the NASA-TLX (Task Load Index) and SUS (System Usability Scale) questionnaires. NASA-TLX is "a multi-dimensional scale designed to obtain workload estimates from one or more operator while they are performing a task or immediately afterwards" [46]. SUS is used to quickly evaluate how well people comprehend the usability of a software application they are toiling away [47].

5.1   Research Method

Setting. The study was conducted at the Mondragon University (Arrasate - Mondragon, Spain). All participants used their own mobile phones on which they had version 61.0 of the Firefox Mobile browser installed. Some of these phones were Samsung Galaxy S8, Samsung Galaxy J2 Core, Xiaomi Mi10, Sony Xperia M2, iPhone SE, OnePlus 6, Huawei P20...

Procedure. At the very beginning of the evaluation, participants were informed of the purpose of the study and given a brief description of it. Thereupon, an instance of MAWA was presented to exemplify the main functionality of the application. The example consisted of removing and moving some web elements of the university home page simulating that we were students. Unnecessary content for students was removed and essential content was moved to the top. Next, participants were proposed to adapt in a defined manner the websites of Filmaffinity,

Amazon, NBC News and to conclude one of the websites they frequently visit on their phones. Finally, participants were directed to an on-line Google Forms questionnaire[3].

Subjects. Sixteen people participated in the evaluation and 62,5% of the participants were male. Regarding their age, all of them were between 18-58 age range and most of them were between 30 and 39 years old. The participants came from Arrasate-Mondragon and the nearby towns. Most of the subjects were married and with a degree from Mondragon University but nobody had a technical knowledge, the aim of the evaluation was to test MAWA with end-users. Most of the subjects were working in different fields when the evaluation was conducted. These fields were finance, construction, teaching, agriculture or sports. 50% of the subjects had never used an editing program such as Photoshop. 87,5% of the participants have installed at least one browser plug-in in their computer with an average of 3,3. In addition, 43,8% of them have installed at least one plug-in in their mobile browser with an average of 1,2. 68,8% of the participants spend less than 30 minutes a day on the Internet on their laptops and additionally, 56,3% of them are connected to the Internet for between 30 minutes and 1 hour on their mobiles.

Instrument. A questionnaire was utilized to gather the users' experience in the evaluation. The questionnaire was composed of four sections; background, their perceived workload (NASA-TLX questionnaire), usability (SUS questionnaire) and satisfaction. Satisfaction was measured by different questions with a 7-point Likert scale (1=completely disagree, 7=completely agree).

Data Analysis. Descriptive statistics have been used to characterize the sample and evaluate the participants' experience using MAWA.

5.2   Results

All users were able to finish the proposed tasks and, at the end of the evaluation, they answered the proposed questionnaire. The modifications made to the Filmaffinity, Amazon and NBC News web pages were guided with a document that included clear images and texts with the actions that users had to complete. Nevertheless, at the end, all subjects were able to modify and additional website, which was one of their favourite web pages on their own. The results of the evaluation are the following.

NASA-TLX results: NASA-TLX results are reported in the table 1 and summarized in the figure 5. NASA-TLX is used to evaluate the perceived workload during a task. The reason for using this questionnaire in this evaluation is to be informed about the subjects' feelings during the exercise. We wanted to be sure that the evaluation was balanced in order to obtain objective results about MAWA. A Likert scale between 0 and 10 was used for the answers.

---

[3] https://docs.google.com/spreadsheets/d/1RKx0xwsZ32-flCtRGUYF1-B0ddfmJ5xyTpqW0gNEEhQ/
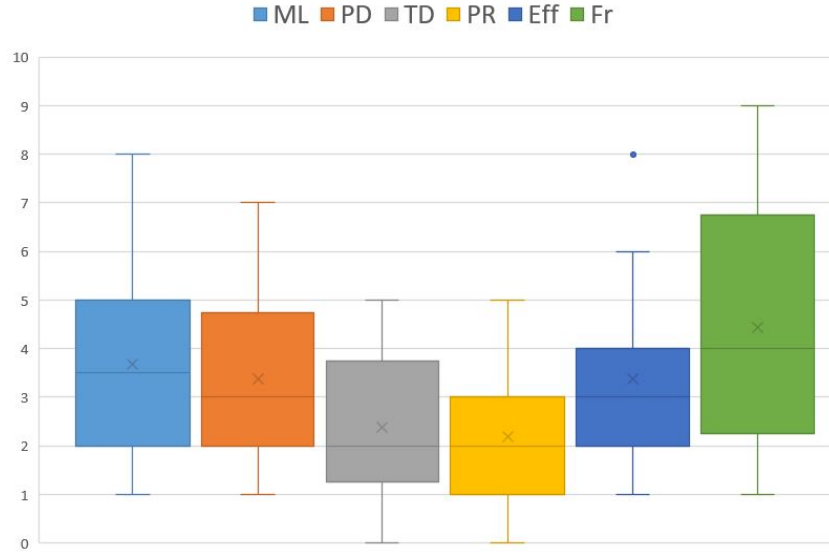
Fig. 5. NASA-TLX scores

– Mental Load: is used to determine the subject's mental and perceptual activity during the activity. The results indicate that the mental load was low during the activity due to the fact that the subjects' responses are at an average of 3,69. The highest value was 8 for a person and two other people voted 6. Despite this, the rest voted 5 or less and this is because they did not find the exercise difficult.

– Physical Demand: is closely linked to mental demand. This question is related to the physical activity required during the task. The results are very similar to those for mental demand as users had an average of 3,38 for physical demand and the same 3 people only voted more than 5.

– Temporal Demand: measures the pressure felt by the subject with the tasks accomplished in the evaluation. All subjects voted less than 5 and two of them voted 0, i.e., they did not felt pressure at all. The average is also very low with a value of 2,38.

– Performance: is used to know the degree of success of the subjects in completing the task. In general, all users are confident that they completed the task owing to the fact that they have all voted 5 or less on this item. On average, the subjects are satisfied with their result in the evaluation with a 2,19. This means that, they were able to complete the task completely without any shadow of a doubt from their point of view.

– Effort: evaluates the degree of mental and physical demand of the activity. The subjects in general did not claim that the evaluation was exhausting because the average is 3,38. These results ensure that the evaluation was not at all difficult because only two people voted more than 5 in this section.

- Frustration: measures how irritated, stressed and annoyed the subject feels during the activity. Despite the fact that on average subjects voted a 4,43 only 5 people voted more than 5 on this item. Frustration during the activity was superior than mental load, physical demand, temporal demand or effort due to the fact that in the first modification they did not succeed in moving the nodes. Nonetheless, the following adaptations were easier for them, there is no steep learning curve.

The standard deviation measures the amount of variation or dispersion of a set of values. The lower the value, the closer the values are to the average value. Comparing the three questionnaires included in this evaluation, NASA-TLX questionnaire has the most elevated values in all standard deviation questions. This might indicate that not all users felt the evaluation process in the same manner. Some of them might be more stressed and frustrated and on the contrary, some might be relaxed and feel that they did an excellent activity. In spite of this, all users were able to finish all activities.

Based on [48], the value must be above 60 to not consider any issue in NASA-TLX. In our particular case, 33,61 is the value obtained, so we can consider our results as satisfactory.

Table 1. NASA-TLX results (N=16)

Feature	AVG	Med	SD	MAX	MIN
Mental load	3,69	3,5	1,96	8	1
Physical demand	3,38	3	1,86	7	1
Temporal demand	2,38	2	1,63	5	0
Performance	2,19	2	1,32	5	0
Effort	3,38	3	1,86	8	1
Frustration	4,43	4	2,61	9	1

SUS results: The SUS results are reported in table 2 and summarized in the figure 6. SUS is used to measure the usability with 10 questions. This usability scale has been used in this evaluation because we wanted to know the overall usability assessment from the user's perspective about MAWA. For the answers, the SUS uses a Likert scale from 1 to 5.

Users considered the overall usability of MAWA to be acceptable. The odd numbered questions have been highly evaluated and even numbered questions have been evaluated with an inferior mark. As expected as a good result in this questionnaire, the odd marks on average have been evaluated with more than 4 and the even ones with less than 2. Regarding the questions directly, people claim that MAWA is easy to use and that most people would learn to use the extension quickly. Moreover, they consider that the system is correctly integrated, that they feel confident using MAWA and finally subjects would use MAWA frequently. On the other hand, questions with the lower score are because subjects

found the system complex, they think that they need the support of a technician to use MAWA, there are inconsistencies, the system is cumbersome and they had to learn a large number of thing to use the extension. These questions are the negative usability characteristics that a tool can have. For that reason, the sub-standard score means that MAWA is usable from the subject's perspective. In general, the standard deviation in all questions is below (with the exception of question S8). This means that users' opinion is very similar in all questions.

[49] estimates that 65 is the minimum value for which the tool is considered to have no usability problems. The result obtained in the SUS evaluation of MAWA is 82,18 thus, users considered the overall usability of the tool to be almost excellent. A result more elevated than 85.5 means that the tool is excellent from the usability point of view.
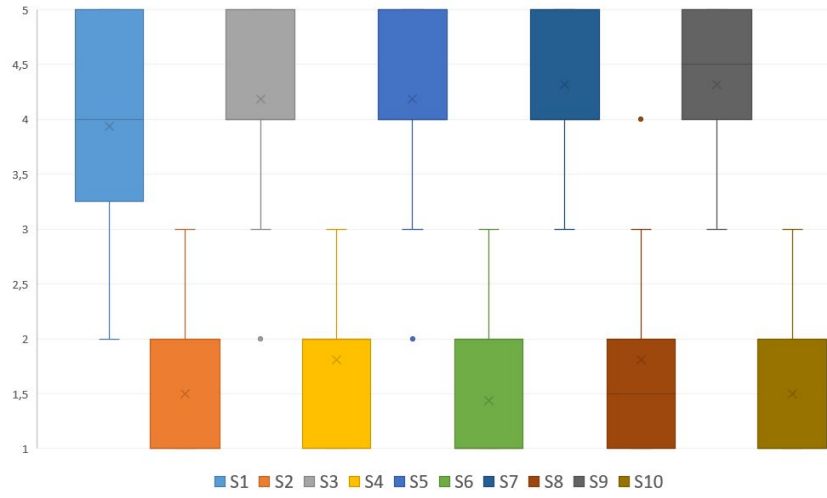

Fig. 6. SUS scores

General questions: The results of the general questions are reported in the table 3 and summarized in the figure 7. These questions evaluate the subjects' posture about the consequences of the problem described in section 2. Additionally, we want to know their opinion on whether the actions in MAWA are intuitive or not and their ease of execution. A Likert scale between 1 and 7 was used for the responses.

Questions 1 to 3 are the most relevant of the article because they would approve whether or not MAWA is suitable for avoiding scrolling, removing unnecessary content and producing comfortable web browsing. Question 1 is about scrolling and subjects claim that MAWA prevents scrolling (mean of 6,37) with a median of 7. Just one person voted 4 and another 5 on this question. This means that,

Table 2. SUS results (N=16)

Items	AVG	Med	SD	MAX	MIN
S1	3,94	4	0,99	5	2
S2	1,5	1	0,63	3	1
S3	4,19	4	0,91	5	2
S4	1,81	2	0,65	3	1
S5	4,19	4	0,91	5	2
S6	1,43	1	0,62	3	1
S7	4,31	4	0,6	5	3
S8	1,81	1,5	1,05	4	1
S9	4,31	4,5	0,79	5	3
S10	1,5	1	0,63	3	1

without any shadow of a doubt, users affirm that MAWA avoids scrolling. The second question inquires if the extension facilitates web navigation and they corroborate that this statement is true. The subjects voted on average a 5,94 out of a maximum of 7 and only 4 people voted 5 or less for this question. Finally, question 3 asks if MAWA facilitates to find desired information on a website and they are convinced that this is also true. The average to this statement is 6,12 and this time also only 4 people voted 5 or less. As a result of these questions, we can confirm that MAWA helps to solve the consequences described in section 2.

Question from 4 to 13 can be classified in pairs. First we asked if a certain action is easy and then if this is intuitive. There are 5 different actions. The first one concerns the selection of nodes by simply clicking on a node (questions 4 and 5). For both questions, the results are almost the same because the subjects consider this action to be both simple and intuitive (average 5,75 and 5,81). The next two questions (6 and 7) refer to the selection of nodes using the arrows at the top, bottom or both sides of a selected node. Both questions have obtained similar answers to those of the previous block (average 5,81 and 5,93). In general, we can affirm that node selection is not a drawback in MAWA because it is easy and intuitive to click on the node or if this method is not possible, to use the arrows around the selected node. Questions 8 and 9 concern node removal. These questions have obtained the most elevated score of the questionnaire with an average 6,62 and 6,69 and the median is a 7 for both. The result determines that the way in which users can remove non-essential content from the website is elementary. Questions 10 and 11 enquire how easy and intuitive is node relocation. This time, there is a difference between both results because there is a 0.5 point difference between the two answers (average 6,37 and 5,87). Despite the fact that node removal is intuitive, the subjects feel this action to be more basic than intuitive. Finally, question 12 and 13 refer to the MAWA action menu. The extension designed for the laptop browser provides a menu, which is not possible for mobile version because extension icons are hidden. Double-clicking to show the menu is easy for users and intuitive in spite of the fact that this statement

has the lowest mark in the questionnaire (average 5,62 and 5,25). Compared to all previous results, in this occasion, the standard deviation indicates that results have been more distributed (1,31 and 1,23).
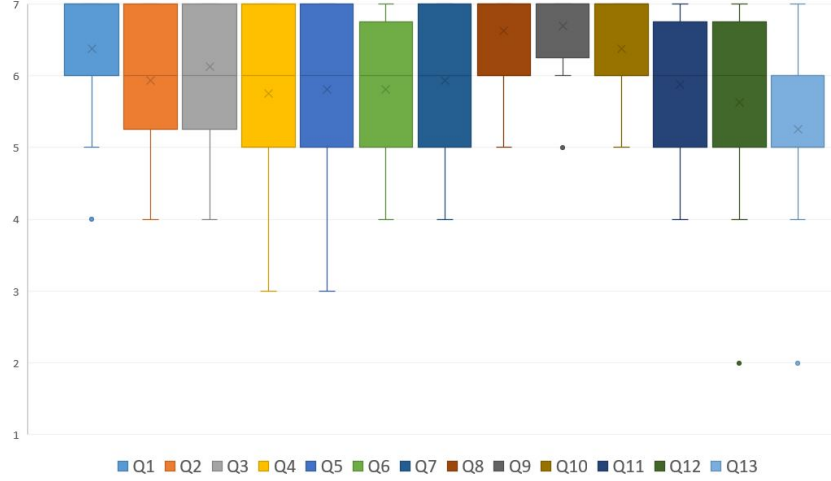
Fig. 7. General questions scores

6   Future work

In the future, the first goal is to be capable of adding content from different websites in order to avoid tab switching. The main reason why we would like to include this feature in MAWA is the annoyance caused by the screen resolution. Adding a new tab in the mobile browser needs more actions than computer browsers. Additionally, as presented in this article, looking for information is frustrating. Based on some previous works that have presented tab switching as a drawback in Web navigation [29][50][51], MAWA will include a method to add content from other websites besides the actual actions. Furthermore, in the evaluation we asked about subjects' opinion about web browsing on mobile and laptop and they answered that on a Likert scale of 1 to 7 (average 5,94, median 6) that mobile browser navigation is more cumbersome than laptop browser navigation. Therefore, we feel that this additional feature would help to facilitate mobile web navigation.

On another note, we would like to improve the robustness of the locators due to the fact that we are not the owners of the websites we adapt and when they are updated, locator mechanisms used might fail. Despite the fact that we use different locator mechanisms in MAWA, locators end up failing. This may produce the feeling that the extension is not first-rate and users might uninstall

Table 3. General question results (N=16)

Items	AVG	Med	SD	MAX	MIN
Q1	6,37	7	0,88	7	4
Q2	5,94	6	0,99	7	4
Q3	6,12	6	0,95	7	4
Q4	5,75	6	1,29	7	3
Q5	5,81	6	1,22	7	3
Q6	5,81	6	0,98	7	4
Q7	5,93	6	1,12	7	4
Q8	6,62	7	0,62	7	5
Q9	6,69	7	0,6	7	5
Q10	6,37	7	0,81	7	5
Q11	5,87	6	0,88	7	4
Q12	5,62	6	1,31	7	2
Q13	5,25	5	1,23	7	2

the extension. We would like to develop a locator mechanism capable of adapting to any website upgrade.

7   Conclusions

We have presented MAWA, a Firefox mobile browser extension for WA designed to remove and move content on any website. Users do not need any programming skills to fulfil these actions as these modifications are realized by user actions (click and drag & drop). The purpose of this extension is to mitigate the consequences of adapting websites designed for desktop to mobile browsers, as explained in the section 2. These consequences are scrolling, producing cluttered navigation, displaying unnecessary content and hiding important one. The presented evaluation confirms that MAWA is able to diminish these three drawbacks. Scrolling is avoided by relocating the web content for example, moving some important content from the bottom of the website to the top. In order to shun consuming unnecessary content, the user can remove these nodes from the website by simplifying the web information. The consequence of both actions is to facilitate web navigation and information search. Furthermore, the evaluations validate that MAWA is usable and that the actions that users must carry out to adapt their websites are intuitive and easy to accomplish.

References

1. Nurshuhada, A., Yusop, R.O.M., Azmi, A., Ismail, S.A., Sarkan, H.M., Kama, N.: Enhancing performance aspect in usability guidelines for mobile web application. In: 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS). pp. 1–6. IEEE (2019)
2. Johnson, T.A., Seeling, P.: Desktop and mobile web page comparison: characteristics, trends, and implications. IEEE Communications Magazine **52**(9), 144–151 (2014)
3. Boehm, B.W., Clark, B., Horowitz, E., Westland, J.C., Madachy, R.J., Selby, R.W.: Cost models for future software life cycle processes: COCOMO 2.0. Ann. Software Eng. **1**, 57–94 (1995)
4. Scaffidi, C., Shaw, M., Myers, B.A.: Estimating the numbers of end users and end user programmers. In: IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2005), 21-24 September 2005, Dallas, TX, USA. pp. 207–214. IEEE Computer Society (2005)
5. Bosetti, G., Firmenich, S., Gordillo, S.E., Rossi, G., Winckler, M.: An end user development approach for mobile web augmentation. Mobile Information Systems **2017**, 2525367:1–2525367:28 (2017)
6. Bouvin, N.O.: Unifying strategies for web augmentation. In: HYPERTEXT '99, Proceedings of the 10th ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots, February 21-25, 1999, Darmstadt, Germany. pp. 91–100. ACM (1999)
7. Aldalur, I., Winckler, M., Díaz, O., Palanque, P.A.: Web augmentation as a promising technology for end user development. In: Paternò, F., Wulf, V. (eds.) New Perspectives in End-User Development, pp. 433–459. Springer International Publishing (2017)
8. Myers, B.A.: Visual programming, programming by example, and program visualization: a taxonomy. ACM sigchi bulletin **17**(4), 59–66 (1986)
9. Lieberman, H., Paternò, F., Wulf, V. (eds.): End User Development. Human-Computer Interaction Series, Springer (2006)
10. Spahn, M., Dörner, C., Wulf, V.: End user development: Approaches towards a flexible software design. In: 16th European Conference on Information Systems, ECIS 2008, Galway, Ireland, 2008. pp. 303–314 (2008)
11. Punchoojit, L., Hongwarittorrn, N.: Usability studies on mobile user interface design patterns: A systematic literature review. Adv. Human-Computer Interaction **2017**, 6787504:1–6787504:22 (2017)
12. Marcotte, E.: Responsive web design: A book apart n 4. Editions Eyrolles (2017)
13. Mahajan, S., Abolhassani, N., McMinn, P., Halfond, W.G.J.: Automated repair of mobile friendly problems in web pages. In: Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018. pp. 140–150. ACM (2018)
14. Frain, B.: Responsive web design with HTML5 and CSS3. Packt Publishing Ltd (2012)
15. Pinandito, A., Az-zahra, H.M., Fanani, L., Putri, A.V.: Analysis of web content delivery effectiveness and efficiency in responsive web design using material design guidelines and user centered design. In: International Conference on Sustainable Information Engineering and Technology (SIET). pp. 435–441. IEEE (2017)
16. Nebeling, M., Norrie, M.C.: Responsive design and development: Methods, technologies and current issues. In: Web Engineering - 13th International Conference,

ICWE 2013, Aalborg, Denmark, July 8-12, 2013. Proceedings. vol. 7977, pp. 510–513. Springer (2013)

17. Tigwell, G.W., Menzies, R., Flatla, D.R.: Designing for situational visual impairments: Supporting early-career designers of mobile content. In: Proceedings of the 2018 on Designing Interactive Systems Conference 2018, DIS 2018, Hong Kong, China, June 09-13, 2018. pp. 387–399. ACM (2018)

18. Aula, A., Khan, R.M., Guan, Z.: How does search behavior change as search becomes more difficult? In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010. pp. 35–44. ACM (2010)

19. Lagun, D., Hsieh, C., Webster, D., Navalpakkam, V.: Towards better measurement of attention and satisfaction in mobile search. In: The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014. pp. 113–122. ACM (2014)

20. Liu, C., Liu, C., Mao, H., Su, W.: Tilt-scrolling: A comparative study of scrolling techniques for mobile devices. In: Intelligent Computing Methodologies - 15th International Conference, ICIC 2019, Nanchang, China, August 3-6, 2019, Proceedings, Part III. vol. 11645, pp. 189–200. Springer (2019)

21. Radeckỳ, M., Smutnỳ, P.: Evaluating user reaction to user interface element using eye-tracking technology. In: Proceedings of the 2014 15th International Carpathian Control Conference (ICCC). pp. 475–480. IEEE (2014)

22. Setlur, V., Rossoff, S., Gooch, B.: Wish I hadn't clicked that: context based icons for mobile web navigation and directed search tasks. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI 2011, Palo Alto, CA, USA, February 13-16, 2011. pp. 165–174. ACM (2011)

23. Mustonen, T., Olkkonen, M., Häkkinen, J.: Examining mobile phone text legibility while walking. In: Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004. pp. 1243–1246. ACM (2004)

24. Andreadis, I.: Comparison of response times between desktop and smartphone users. Mobile research methods **63** (2015)

25. Yesilada, Y., Harper, S., Chen, T., Trewin, S.: Small-device users situationally impaired by input. Comput. Hum. Behav. **26**(3), 427–435 (2010)

26. Díaz, O., Aldalur, I., Arellano, C., Medina, H., Firmenich, S.: Web mashups with webmakeup. In: Rapid Mashup Development Tools - First International Rapid Mashup Challenge, RMC 2015, Rotterdam, The Netherlands, June 23, 2015. vol. 591, pp. 82–97. Springer (2015)

27. Urbieta, M., Firmenich, S., Bosetti, G., Maglione, P., Rossi, G., Olivero, M.A.: MDWA: a model-driven web augmentation approach - coping with client- and server-side support. Softw. Syst. Model. **19**(6), 1541–1566 (2020)

28. Fernández-García, A.J., Iribarne, L., Corral, A., Criado, J., Wang, J.Z.: A flexible data acquisition system for storing the interactions on mashup user interfaces. Computer Standards & Interfaces **59**, 10–34 (2018)

29. Aldalur, I., Perez, A., Larrinaga, F.: Customizing websites through automatic web search. In: Human-Computer Interaction - INTERACT 2019 - 17th IFIP TC 13 International Conference, Paphos, Cyprus, September 2-6, 2019, Proceedings, Part II. vol. 11747, pp. 598–618. Springer (2019)

30. Chudnovskyy, O., Nestler, T., Gaedke, M., Daniel, F., Fernández-Villamor, J.I., Chepegin, V.I., Fornas, J.A., Wilson, S., Kögler, C., Chang, H.: End-user-oriented telco mashups: the OMELETTE approach. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume). pp. 235–238 (2012)

31. Kurniawan, S.H., King, A., Evans, D.G., Blenkhorn, P.: Personalising web page presentation for older people. Interact. Comput. **18**(3), 457–477 (2006)
32. Tsandilas, T., Schraefel, M.M.C.: User-controlled link adaptation. In: Ashman, H., Brailsford, T.J., Carr, L., Hardman, L. (eds.) HYPERTEXT 2003, Proceedings of the 14th ACM Conference on Hypertext and Hypermedia, August 26-30, 2003, Nottingham, UK. pp. 152–160. ACM (2003)
33. Leiva, L.A.: Restyling website design via touch-based interactions. In: Bylund, M., Juhlin, O., Fernaeus, Y. (eds.) Proceedings of the 13th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2011, Stockholm, Sweden, August 30 - September 2, 2011. pp. 599–604. ACM (2011)
34. Bellal, Z., Elouali, N., Benslimane, S.M., Acarturk, C.: Integrating mobile multimodal interactions based on programming by demonstration. International Journal of Human–Computer Interaction pp. 1–16 (2020)
35. Trusty, A., Truong, K.N.: Augmenting the web for second language vocabulary learning. In: Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011. pp. 3179–3188. ACM (2011)
36. Bila, N., Ronda, T., Mohomed, I., Truong, K.N., de Lara, E.: Pagetailor: reusable end-user customization for the mobile web. In: Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys 2007), San Juan, Puerto Rico, June 11-13, 2007. pp. 16–29. ACM (2007)
37. Baudisch, P., Xie, X., Wang, C., Ma, W.: Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In: Feiner, S., Landay, J.A. (eds.) Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, Santa Fe, NM, USA, October 24-27, 2004. pp. 91–94. ACM (2004)
38. Ricca, F., Leotta, M., Stocco, A., Clerissi, D., Tonella, P.: Web testware evolution. In: 15th IEEE International Symposium on Web Systems Evolution, WSE 2013, Eindhoven, The Netherlands, September 27, 2013. pp. 39–44 (2013)
39. Almendros-Jiménez, J.M., Tedesqui, A.L., Moreno, G.: Annotating "fuzzy chance degrees" when debugging xpath queries. In: Advances in Computational Intelligence - 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, Puerto de la Cruz, Tenerife, Spain, June 12-14. pp. 300–311 (2013)
40. Biagiola, M., Stocco, A., Ricca, F., Tonella, P.: Diversity-based web test generation. In: 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering ESEC/FSE 2019 August 26–30, Tallinn, Estonia. pp. 231–242 (2019)
41. Zhang, Y., Pan, Y., Chiu, K.: A parallel xpath engine based on concurrent NFA execution. In: 16th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2010, Shanghai, China, December 8-10. pp. 314–321 (2010)
42. Kirinuki, H., Tanno, H., Natsukawa, K.: COLOR: correct locator recommender for broken test scripts using various clues in web application. In: 26th IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2019, Hangzhou, China, February 24-27. pp. 310–320 (2019)
43. Song, F., Xu, Z., Xu, F.: An xpath-based approach to reusing test scripts for android applications. In: 14th Web Information Systems and Applications Conference, WISA 2017, Liuzhou, China, November 11-12. pp. 143–148 (2017)
44. Hammoudi, M., Rothermel, G., Stocco, A.: WATERFALL: an incremental approach for repairing record-replay tests of web applications. In: Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2016, Seattle, WA, USA, November 13-18. pp. 751–762 (2016)

45. de Santana, V.F., de Oliveira, R., Almeida, L.D.A., Ito, M.: Firefixia: an accessibility web browser customization toolbar for people with dyslexia. In: International Cross-Disciplinary Conference on Web Accessibility, W4A '13, Rio de Janeiro, Brazil, May 13-15, 2013. pp. 16:1–16:4 (2013)
46. Hart, S.G.: Nasa-task load index (nasa-tlx); 20 years later. In: Proceedings of the human factors and ergonomics society annual meeting. pp. 904–908. Sage publications Sage CA: Los Angeles, CA (2006)
47. Brooke, J.: Sus: a retrospective. Journal of usability studies **8**(2), 29–40 (2013)
48. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: Advances in psychology, vol. 52, pp. 139–183. Elsevier (1988)
49. Brooke, J.: Sus: a "quick and dirty'usability. Usability evaluation in industry **189** (1996)
50. Díaz, O., Sosa, J.D., Trujillo, S.: Activity fragmentation in the web: empowering users to support their own webflows. In: 24th ACM Conference on Hypertext and Social Media (part of ECRC), HT '13, Paris, France - May 02 - 04, 2013. pp. 69–78 (2013)
51. Zhang, H., Zhao, S.: Measuring web page revisitation in tabbed browsing. In: Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011. pp. 1831–1834 (2011)