IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 14, 2021, accepted July 26, 2021, date of publication August 11, 2021, date of current version August 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104187

Web Augmentation as a Technique to Diminish User Interactions in Repetitive Tasks

IÑIGO ALDALUR, ALAIN PEREZ, AND FELIX LARRINAGA
Electronics and Computer Science Department, Mondragon University, 20500 Arrasate-Mondragon, Spain

Corresponding authors: Iñigo Aldalur (ialdalur@mondragon.edu), Alain Perez (aperez@mondragon.edu), and Felix Larrinaga (flarrinaga@mondragon.edu)

This work was supported in part by the Software and Systems Engineering Research Group of Mondragon Unibertsitatea under Grant IT1326-19, and in part by the Department of Education, Universities and Research of the Basque Government.

ABSTRACT The use of the World Wide Web has experienced extraordinary growth in the last decades. The Web has become the main source of information for millions of users. The number of websites offering content to users is countless. In order to personalise information according to their needs, users often have to visit multiple, unconnected pages. Users perform a number of actions to collect that information that requires concentration. If the number of Web resources is large, the activity becomes unpleasant. The problem increases when these tasks are performed frequently and repetitively. These tasks are time-consuming and lead users to experience frustration and disorientation during the activity, causing a loss of concentration that prolongs the activity over time. Web Augmentation combines different Web technologies to improve user experience on existing pages by adding content from different pages among other benefits. This article proposes Web Augmentation as a technique to reduce user interactions in repetitive tasks. To support the proposal, the paper introduces Excore, a browser extension for Web Augmentation that allows end-users to add content from different resources automatically. The article presents the benefits introduced by this approach as a response to the drawbacks experienced by users while performing their activities on the Web. The architecture of the platform and its operations are described by means of an example. A double evaluation of the extension is addressed, one qualitative and one quantitative. The results show that Excore reduces the number of interactions by 94.45% and the time to complete a task by 80.75%.

INDEX TERMS Web augmentation, automation, end-user development, human-computer interaction, repetitive tasks.

I. INTRODUCTION

In recent years the web has experienced an incredible increase in the number of users. Consuming the information available on the web and tailoring it to user requirements has become an essential part of their lives. The customization of information introduces several problems, as users have to visit and analyze multiple, often unconnected, web pages. Continuous access to new pages, switching between tabs or scrolling through content makes it difficult for users to concentrate and leads to frustration and disorientation. In order to support users in the customization of information, several works have been developed based on the concept of End-User Development (EUD) [1]–[4], [5]. Lieberman *et al.* [6] define End-User Development as ''a set of methods, techniques and tools that enable users of software systems, acting as non-professional software developers, to create, modify or extend at some point in time a software artifact''. Unfortunately, coding is often beyond the reach of many end-users [7], [8]. Although end-users are initially unable to use many functions of these tools, they are eventually able to perform complex actions step by step [9]. In occasions, they may require appropriate tools that allow them to easily create their own scripts [6]. The trend is towards the use of end-user centred design technologies, where users without programming skills can perceive the need to tailor web applications to their preferences. [10]. The increasing volume of content and actions available on the web intensifies the desire to control the web experience. More if we take into account that many applications, once designed for the desktop, have gradually made the transition to the Web [11]. Frequently, in order to perform actions conducted through the Web, several websites are visited [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

In general, these websites are completely independent from each another. Consequently, most end-users renounce to actions that obtain the required information by performing their inter-site explorations. The main drawback is that end-users feel frustration when repetitive tasks are involved in extracting content from different websites. [13] studied that multiple windows and tabs have significant flaws that hinder users' performance. Therefore, the aim is to empower end-users to develop extracting content functionality by themselves. By allowing end-users to adapt web content to their requirements, usability is improved and accessibility barriers can be removed [14]. Web Augmentation (WA) is the most appropriate technique to carry out these actions by end-users.

Bouvin [15] 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 with the effect of allowing structure to be added to the Web page directly or indirectly, or to navigate through such structure. The purpose of this tool is to help users organize, associate, or structure information found on the Web. This activity can be carried out by a single user or in collaboration with others". More recent definitions state that "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" [16]. WA techniques were proposed as a method to extend the features of websites without affecting the server-side code. This process is carried out by end-users and not by the website developers. WA can be triggered by browser add-ons, applets, Javascript code, etc. The question would be why an end user would use a WA tool to adapt a website by adding content from different websites. The answer would be because the end user frequently visits that website and it is somehow incomplete. The addition of content from different websites completes the information needs of users in their repetitive activity. After that, the information extraction process should be automatic, without user interaction.

The main purpose of Web Automation is to automate repetitive tasks, such as navigating web pages, filling in forms and clicking on links [17]. Web automation reduces repetitive end-user interactions and alleviates completing tedious tasks [8]. Furthermore, web automation can save time and prevent errors when end-users have to accomplish repetitive or complex tasks [18]. With this in mind, automation is a necessity for repetitive tasks performed by end-users. Although it is possible to develop automation scripts with a large number of programming languages, this is not an alternative for many users. It is necessary to provide end-users a visual environment that facilitates the creation of the automation information extraction process with basic user actions. This paper studies WA as a mechanism to support end-users to complete web pages information from different websites automatically and in parallel for repetitive tasks by reducing the number of interactions using a Chrome browser extension.

The remainder of the paper is organized as follows. Section II analyses the problem we want to solve, its causes and its consequences. Section III discusses related work in order to give the reader an idea of what has been done in this area. Section IV and section V describe Excore (External Content Retriever), a browser extension for Web Augmentation that inserts automatically content from different web pages for repetitive web tasks. Section VI presents the evaluation and its results and section VII outlines the features we would like to enhance in Excore and the conclusions of the paper.

II. PROBLEM ANALYSIS

Following the Design Science guidelines [19], in this section we want to show the main problem we want to solve, as well as the causes that create this problem and the consequences that the problem provokes. In the evaluation we will try to validate Excore by checking if it reduces or eliminates the causes of the problem or its consequences.

A. THE PROBLEM

Websites have evolved over the last decade, but they do not always provide all the information users need. This forces users to create their own scenarios in which they have to perform the same repetitive tasks to satisfy their needs (e.g. opening new tabs, scrolling, multiple clicks...). Nowadays, users open different tabs (branching) to search for information on the Web (multi-tasking) and they continuously switch the view from one tab to another (tab-switching), which can be time consuming. For example, choosing which movie to go to the cinema may require 4 web sessions: looking at the movie listings, checking movie reviews, checking public transport to the cinema in question, and buying tickets.

Reference [20] conducted a study on users' perception when performing interactions on the web and detected a feeling of frustration in users caused by repetitive tasks, such as gathering information from different resources. In addition, the study detected cognitive disorientation in the user's daily activities on the web when searching for information from various sources.

B. CAUSES

Among other reasons, frustration is caused by repetitive tasks and the waste of time and effort they entail. The following question arises: what causes these repetitive tasks and wasted time?

- Multi-tasking: [21] found that multi-tasking occupies up to 76% of users' regular web activity time. While only 24% of web sessions are used for a single task, 20% of sessions have 5 or more tasks. A multi-tasking session occurs when Web navigation requires more than one web session to complete and has a definable point in terms of when the task is finalized or abandoned [22].
- Tab-switching: Constant tab-switching is a way to obtain information from websites in different tabs. Nonetheless, users may switch tabs to locate a previously opened

tab or click on a tab by accident. In these cases, tab switching results in transient page views. In reality, users do not aspire to gather information from these sites [23]. In fact, [24] found that users switch tabs at least 57.4% of the time, but user activity, measured in page views, is split between tabs rather than increasing overall activity. [25] claims that when there are multiple activities, it is necessary to understand and represent the information contained in each tab, which is essential to differentiate each activity.

- Branching: In the web literature, branching is defined as the act of initiating a new tab (or window), which allows people to pleasantly navigate multiple websites concurrently [26]. Branching is the step previous to multi-tasking and tab-switching and the problems they cause. For example, [27] studied that when the number of tabs opened increases, it becomes more challenging for users to find relevant information or identify which tabs to focus on. They also concluded that, on average, users have 8 tabs open and 66.9% of subjects have one or more issues each week due to the number of tabs open.
- URL typing: Errors are common during repetitive activities, as people's concentration decreases. Correcting these errors can lengthen the period of time a user needs to accomplish a task. Some studies have analyzed errors when typing a URL [28]. The difficulty of typing a specific domain name or the likelihood of making certain type of mistake is not purely random. These errors also affect user frustration.
- Clicks: A high number of clicks is also a sign of problems [29]. The more clicks the user makes, the worse the interaction is. Some studies show that the browsing tasks analysed result in a higher number of clicks than search tasks and this is due to the tendency of users to make unnecessary clicks [30]. Moreover, some users click back [31] to ensure that they have clicked on the correct link by re-reading its content. It has been documented as a revisiting strategy [32] and as a way to quickly preview a page [33]. Furthermore, in a real user session, partial movements are much more frequent than point-and-click movements. Some studies measured an average of 0.53 mouse clicks per minute, but 6.58 partial mouse movements per minute [34].
- Scrolling: According to chapter 8 of the usability.gov guidelines [35], users should be able to view the page without scrolling. However, some scrolling depth analysis suggest that in exploratory tasks users scroll significantly more than in lookup tasks [30]. The outcome is that people used the scroll bar on 76% of the pages, with 22% scrolling all the way to the end, regardless of the length of the page [30]. Previous work found that time spent on a page and the amount of scrolling on a page had a strong correlation with explicit interest, while scrolling and mouse-clicks were ineffective in predicting explicit interest [36]. Nonetheless, scrolling up and down a page without reading the content can also be a sign of frustration [29].
- Copy and paste: While the clipboard is a very useful tool for copying information between tabs, it can also introduce inefficiencies and errors during tasks [37]. For example, Roberts *et al.* [38] researched on errors caused by copying and pasting data in the context of medical information and their consequences.
- Interruptions: [39] shows that interruptions can also cause frustration, distract people, cause them to make mistakes, reduce their efficiency and increase the time needed for the main task. Moreover, this is not all, [40] remarks that multitasking on different types of tasks can reduce productivity and [41] states that the ability of humans to accomplish simultaneous mental operations is limited by the ability of human brain.
- Writing: Another cause of frustration is the continuation of writing over a long period of time. Typing for 1 hour induces muscle fatigue (60%-67% of the subjects) [42]. Therefore, reducing the time spent on repetitive tasks is extremely important.

C. CONSEQUENCES

Constant repetition of tasks has consequences beyond user frustration.

- Lost of focus: When users are interrupted or take their eyes off the system, they may lose important information about others' activities [43]. Furthermore, if users are forced to perform activities between sites on both a daily and occasional basis, they may lose concentration due to the constant switching of tabs [5]. As a result, their tasks will be prolonged in time and may be completed inadequately.
- Time consuming: Several studies have analyzed the time it takes to load a web page. [44] found that iterations of search loop, file writing, page load over HTTPS and source code diffing took between 7 and 13 seconds. [45] extracted that there is a relation between the server location and the time needed to load a website in a country. In their study for European countries, the average was 4.72 s and for the USA, it was 7.64 s. The results of their research showed that average page load time has a direct impact on the e-commerce conversion rate and customer satisfaction. Therefore, reducing the loading time of all web pages involved in the repetitive task could reduce frustration in repetitive tasks.

III. RELATED WORK

This section presents the literature survey on the technologies used to reduce the effects of repetitive tasks (multi-tasking, tab-switching, scrolling etc.) through the use of mashups, web augmentation and automation. First, the concept of task fragmentation and, more specifically, web fragmentation is introduced.

A. WEB FRAGMENTATION

An activity is defined as a coordinated set of actions performed by people towards the realization of a common objective, mediated by tools and subject to situational constraints [46]. It is common to see people performing different activities simultaneously or switching from one activity to another. In fact, people spend an average of three minutes on achieving an objective before switching to another activity [47]. This practice is called task fragmentation. Task fragmentation has been studied frequently, showing that it is very common in contemporary knowledge-oriented workplaces [47]–[49]. These studies show that work fragmentation is harmful to the actual work and that after such a change of context, it takes time for people to regain their orientation [49]. [50] observes that work fragmentation is correlated with lower productivity observed both at the macro level (for entire sessions) and at the micro level (around work fragmentation markers). Furthermore, longer activity shifts seem to worsen the effect. Nowadays, most people work with computers and consequently task fragmentation has a particular impact on work performance. [48] claims that computers are notoriously bad at supporting parallel activities and managing interruptions. Interruptions are a particularly detrimental kind of fragmentation activity in which an external signal (email, chat, phone call, or direct conversation) forces a person to switch activity in an unplanned period of time [49], [50]. Work fragmentation is an important phenomenon in the context of modern software development. The impact in this context may be even worse than in other domains, as developers build and maintain complex mental models of the software they are working on. These models can be more sensitive to interruptions and are costly to rebuild [50]. This means that computer workers constantly lose track of their activities and, consequently their work performance is negatively affected. This is especially relevant in the case of Web navigation or browsing. When performing a particular task (e.g., organizing a trip), the user often goes beyond the application's boundaries, visiting several (related) Web applications. In these cases, the user may feel a loss of context each time he/she navigates from one application to another, because the new application used has no way of tracking the user's previous navigation site [51].

[52] claims that in order to reduce performance losses caused by work fragmentation, proper activity management is necessary. The results of this study show that the wider the repertoire of task management practices applied in daily work, the greater the experience of task management effectiveness. Several studies have tried to avoid or repair the drawback caused by task or web fragmentation. For example, Cowpath [5] focuses on ''Web trails'', i.e. recurring navigation paths across different websites. Instead of switching tabs and typing the same URLs (Uniform Resource Locator) over and over again, Cowpath augments the affected websites with additional hyper-links that ''pave the way'' for these Web trails. [43] provides visualization techniques to help groupware users recover from short absences. First, they provide a simple attention monitor that informs groupware systems whether a user is looking at the screen and whether the workspace is visible. Secondly, they provide a framework of recovery techniques to visualise lost activity. [53] proposes an algorithm that extracts information from a web search and prevents end-users from repeating a secondary search. [51] proposes a solution to improve the user's browsing experience by addressing the lack of integration between visited pages and enabling customization. The study proposes to use activity diagrams, where each activity represents a relevant subtask in a more general navigation scenario. Thanks to their method, the user always stays on the same task, web navigation is not needed and task fragmentation is avoided. WildThumb [54] suggests a change to the web interface to support efficient task management in Web browsing. It provides the user with a visual overview of all tabs and reduces the error when opening the correct one. However, it does not prevent multi-tasking or loss of attention. If the user has dozens of tabs open, the visual overview does not help to find the correct tab, the user has to search for it and in this process, would lose track of the active task. Another example is AwToolkit [55]. This proposal consists of a set of user interface widgets that assist users in maintaining awareness of display changes. The main objective is to offer the ability to detect changes when users are not looking at a specific screen, and then notify users about these changes. [56] researches on the impact of web fragmentation on human attention when using multiple wearable and mobile devices. The paper identifies the importance of minimizing the mental effort caused when using mobiles. They propose Attelia II, a middleware that identifies breakpoints in the usage of those devices, and delivers notifications of those times. Attelia II works in real-time by considering the natural use of mobiles and wearables, without modifying applications and without any dedicated sensors. Awareness information in shared-workspace groupware is often ephemeral, and when users are interrupted or look away from the system, they can lose important information about others' activities. These attentional disconnections are brief (typically less than ten seconds), are common in many work environments, and can occur for many reasons. [5] mentions that if it is no necessary to go through the welcome page of a site, this should be avoided. The proposal saves clicks and facilitates focus and thus avoids task fragmentation.

B. MASHUPS

Another alternative to reduce the effects of tab switching, multi-tasking and repetitive activities is the use of mashups. Wikipedia defines a mashup as a web page or application that uses content from more than one source to create a new service that is displayed in a single graphical interface.[1] As Web users' search tasks becoming increasingly complex, a single source of information cannot necessarily satisfy their information needs [70]. Linking content from different

[1] https://en.wikipedia.org/wiki/Mashup

TABLE 1. Related work tool comparison.

	Year	Frag.	Mashup	WA	Autom.	Execution mode	End-user centered	Programing paradigm	Reference
Cowpath	2013	✓		✓		Client		Domain Specific Language	[5]
Gutwin et al.	2017	✓				Client			[43]
Winckler et al.	2015	✓				Client-Server	✓	Programming by Demonstration	[53]
Firmenich et al.	2014	✓		✓		Client		Programming by Demonstration	[51]
WildThumb	2010	✓	✓			Client	✓		[54]
AwToolkit	2014	✓				Server	✓		[55]
Attelia	2015	✓				Client-Server	✓		[56]
FaceMashup	2016		✓			Server	✓	Visual Programming	[2]
HyOASAM	2020		✓			Server		Domain Specific Language	[57]
EFESTO	2017		✓			Client	✓	Visual Programming	[1]
MashupEditor	2016		✓			Proxy	✓	Programming by Demonstration	[3]
ENIA	2019		✓			Client	✓	Visual Programming	[58]
MAMS	2017		✓			Server	✓	Visual Programming	[59]
Chudnoskyy et al	2012		✓			Client	✓	Visual Programming	[60]
WebMakeup	2017		✓	✓		Client	✓	Visual Programming	[11]
Bosetti et al.	2017			✓		Client	✓	Programming by Demonstration	[61]
Fernández-García et al.	2018			✓	✓	Client	✓	Visual Programming	[62]
OFIE	2020			✓	✓	Client	✓	Programming by Demonstration	[63]
CrowdMock	2018			✓		Client	✓	Visual Programming	[4]
EUCalipTool	2020			✓		Client	✓	Programming by Demonstration	[64]
SUGILITE	2017				✓	Client	✓	Programming by Demonstration	[65]
Rataplan	2020			✓	✓	Client	✓	Programming by Demonstration	[66]
TellMe	2011				✓	Client	✓	Programming by Demonstration	[67]
VASTA	2020				✓	Client	✓	Programming by Demonstration	[68]
PWT	2014				✓	Server	✓	Programming by Demonstration	[69]
EXCORE	2021	✓	✓	✓	✓	Client	✓	Visual Programming	

websites avoids tab switching when locating information from different resources. FaceMashup [2] is an End-User Development environment that empowers users of social networks by supporting them create their own procedures for inspecting and controlling their data. [57] proposes a Hybrid Open API Selection Approach for Mashup development (HyOASAM). By introducing user stories into Mashup development, Mashup developers can easily capture the role, aim, and motivation of a Mashup and then describe them with user stories. The open API discovery approach can be divided into three steps: extract three components from user stories, extract three corresponding elements from open API descriptions and calculate the similarity based on two sets of data. The main goal of EFESTO [1] is to highlight the features of frameworks that can lead to reduce end-users' efforts in developing interactive workspaces. The user interface layer provides and manages the visual language that allows end users to perform mashups without the need for technical knowledge. EFESTO allows users to visualize and manipulate data extracted from remote sources. The user interface runs in the user's web browser and communicates with the Logic and Data layer that runs on a remote Web server. MashupEditor [3] is a novel environment for End-User Development of Web mashups. MashupEditor aggregates content from different websites. The tool avoids tab switching during the composition process. End-users exploit an intuitive copy and paste metaphor, which provides component composition for existing Web applications. This means that MashupEditor eludes using copy and paste in repetitive activities. ENIA [58] provides a mashup with slightly more advanced features. The main parts of the ENIA mashup are: a services menu containing the list of services offered by the mashup, services describing the capacities provided by the mashup user interface, components corresponding to the services, which have been added to the workspace and instantiated for use, components containers, menus that provide capabilities to interact with the component container, a workspace which constitutes the work area where containers are deployed and users can interact with them and finally operations, which are formed by a subset of actions that can be performed on the mashup user interface. MAMS [59] is the first existing Mashup development process for Modeling and Simulation. Following a new Box/Wiring/Mashup method, users can develop resources as mashup components, compose them as mashups and run these mashups in web browsers quickly. Chudnoskyy *et al.* [60] take a step forward to create web compositions by assisting users with recommendations and automatic composition.

C. WEB AUGMENTATION

Mashups are not the only technology appropriate for merging content from different websites into one. Other methods can also be used. Web Augmentation is a combination of different techniques that improves the experience of users on existing web pages. There are a number of techniques applied in this field to reduce the effects identified in SectionII. WebMakeup [11] is a Chrome browser extension that copies content from diverse web pages and end-users insert or paste in a single website. In addition, it permits them to remove unnecessary web elements and move those elements to other positions on the website. Bosetti *et al.* [61] proposes a solution to reduce user interaction during web searches using existing web engines. It inserts new content into existing websites with the information provided by these search engines. [62] shows a tool that proposes a data acquisition system capable of capturing user interactions in web interfaces. Subsequently, these interactions can be automatically reproduced without any human action, which can be a method to limit task fragmentation. On-the-Fly Interaction Editor (OFIE) [63] relies on Programming By Demonstration to define the appropriate native input to be performed automatically when the rule is triggered. The approach enables the end-user to define actions by simply performing the required native input interaction on the Graphical User Interface of the application. The end-user does not require any necessary programming knowledge or to write/edit any code. With CrowdMock [4], users can define their own requirements (adding and removing content) and share them with the community, who can collaboratively reproduce, edit and evolve them. [64] investigates how social networks can be used to improve the composition of services by end users. To do so, they propose a graph-based definition of a social structure, and analyse how social connections can be exploited both to facilitate end users service discovery through the navigation of these connections, and to recommend services to end-users during the composition activity (EUCalipTool).

D. AUTOMATION

Web automation tools provide a system to increase human productivity by conducting repetitive tasks autonomously. It reduces the time needed on an activity and maintain users' focus on the activity. [71] states that "any repetitious behaviour should be a candidate for automation because automating things we have done before frees up time for us to do new things". SUGILITE [65] is a mobile system that enables users to create automation for different tasks through any or multiple smartphone apps and to execute automated tasks through a multi-modal interface. Rataplan [66] is a robust and resilient pixel-based approach for linking multi-modal proxies to automated sequences of actions in graphical user interfaces. With Rataplan, users define a sequence of actions and the system determines their desire for automation. After demonstrating a sequence, the user can link a proxy input control to the action, which can then be used as a shortcut to automate a sequence. TellMe [67] is an automation system that enables, via natural language instructions, to record Web based tasks and then replay them to automate those tasks in the future. VASTA [68] is a novel vision and language-assisted Programming By Demonstration system for task automation on smartphones. VASTA leverages computer vision techniques, such as object detection and optical character recognition, to accurately label interactions demonstrated by a user, without relying on the underlying user interface structures. It also takes advantage of advanced natural language understanding algorithms to analyze the user utterance to trigger the VASTA automation scripts, as well as to determine the automation parameters for generalization. PWT [69] provides end-users with a way to customize their own searches in their favourite or most visited web pages, thus reducing frustration in repetitive tasks. Despite the existence of applications to automate and customize web interactions, these are insufficient to ensure that user goals are met when changes in the relevant context cannot be fully anticipated at design time [69]. The aim of Excore is to contribute to the solution of repetitive task and the frustration caused in this context by reducing the causes and consequences of the problem analyzed in section II.

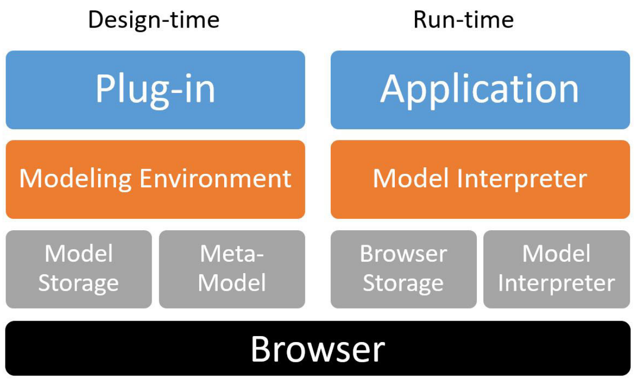


FIGURE 1. Excore architecture.

The table 1 summarizes all the tools explained in the related work. The first two columns show the name of the tool or the lead author and the year of the publication. The next four columns refer to the scope of the tools. Most of them are related to one category, but there are others that can be applied in two of them. The execution mode column refers to the machine on which the tool runs. Most of them run on the client-side as web extensions, mobile applications or desktop applications. In contrast, other tools run on the server-side. Winckler *et al.* and Attelia need client and server-side applications to be executed. FaceMashup runs on a proxy server. Most of the tools focus on end-users and are specially designed for them. Nonetheless, there are four tools that, even if designed for end-users, they cannot develop core functionality. These tools are designed for web fragmentation and end-users can only use them in their daily routine.

IV. AUTOMATING WEB AUGMENTATION PROCESS

The aim of this section is to show how Excore satisfies the objectives of the section II. The Excore architecture is presented in figure 1 and differentiates two sections: design-time or production and run-time or automation.

Design-time or production provides an environment, which defines the user interaction sequence by capturing all user actions. This interaction sequence is defined by the meta-model (see section V). The extension provides a modelling environment that guides the user in defining the meta-model. No code implementation is necessary in this process. At the end of this stage, the entire interaction sequence is stored in the browser. Browsers provide a space that allows extensions to store some information, which is used by Excore to save the interaction sequence of the meta-model.

In the run-time or automation stage, the interaction sequence stored in the browser is analyzed or interpreted every time a load event occurs. When Excore detects that it needs to take action, the automation process is started and executed. Once the execution is completed, the target website will display the new content extracted from different resources. An example will illustrate this process in an understandable way.

Everybody has more than one web activity, which is frequently accomplished. These actions can be executed a large number of times in a day (look for research articles), once a day (read newspapers), once a week (look for information about films on the cinema) or once a year (organize a trip). The task is one but it implies to search for information in different resources, which implies opening different tabs, copy a piece of text and paste it to look for information in another website, write content, click on different links…Moreover, the person can be interrupted by a phone call, a family member asking a question, the person remembers that he/she has a pizza in the oven, etc. Completing a task quickly, without mistakes or even interruptions is complicated.

People with programming skills could develop a script, which would be able to automate repetitive processes. However, end-users do not have the necessary knowledge to create these routines or do not even have enough time to develop them. Excore helps end-users to interactively create automated execution processes using the capture and replay method. Excore is able to capture every action performed by the end-user. Once the sequence of actions is completed, the tool performs the same sequence of actions in parallel mode. In order to illustrate a repetitive action sequence, an example of searching for books on the web has been developed. A person who likes to read frequently buys books through Amazon[2] which is a well-known book website. However, this is not the unique website for book shopping and it is advisable to compare prices on different resources such as Waterstones[3] and Bookshop.[4] Waterstones is a British website and the price is in pounds instead of dollars. This is why, information on the currency value is needed to correctly compare the prices of all books. Besides, this person desires ratings on books and he visits Goodreads[5] to obtain more information. Finally, it would be useful to obtain information about the author of the book and Wikipedia[6] can provide this data. The number of actions and the time needed to complete this search are significant. Web augmentation techniques offer the possibility to include the desired information on a single website by adding this desired content to the target website. If Amazon is the target website, figure 2 shows the initial and original "The Da Vinci Code" site and figure 3 shows the final site after adding content from the different sources. How can an end-user include this content from different websites into one? Excore permits this by separating the process into two procedures: production and automation.

A. PRODUCTION

Excore is a browser extension for end-users based on Chrome, which is the most widely used browser worldwide.[7] When installed from the Chrome store,[8]selecting the Excore icon (in the top right corner of the browser) enacts a menu with three options: New, Save and Delete. When using the plugging for the first time, the only possible option is "New" because the other two buttons are disabled. Once the "New" button is clicked, the production process starts changing the background colour when the mouse is over the web node. At this point, Excore records every single action the user performs in order to reproduce them during the execution process. Recorded actions are:

- Click: one of the most repeated actions. Excore saves the element in which the user has clicked. The user can click in an active or inactive element. Active elements are links, button or even inputs where the user will write or paste some text. Inactive elements can be paragraphs or images that can be clicked unintentionally and there is no consequence even if Excore repeats this action through the execution process.
- Copy: once a text fragment has been selected, the user can copy this fragment with the intention of pasting it into a search bar. Excore records this action if a text is selected.
- Paste: similarly, if the user is on a search bar after copying a text fragment and pastes this content, it will be recorded by the system.
- Keypress: if an input is selected and the keyboard is pressed, every single key is joined and saved as an input string. For example, this process permits saving the user name and password if the user must be authenticated before the content addition. Moreover, it permits adding content in repetitive searches such as user's city temperature.
- Double-clicking: defines the beginning of an automation process on a different website. This process starts with the addition of the URL, which will be visited to continue with the production process of the repetitive task (see figure 4). The process ends when the user double-clicks on the desired web node (see figure 8), closes the tab and inserts the content replacing the initial double-click node (see figure 3).

The following paragraphs illustrate the purchase of a book in Amazon using Excore. Imagine that the user has included information from the Goodreads website, the prices of the same book in Waterstones and Bookshop websites and the currency comparison between dollars and pounds. The last step is to include additional information about the book author. If the user double clicks on the author element, a node

[2]https://www.amazon.com/
[3]https://www.waterstones.com/
[4]https://bookshop.org/
[5]https://www.goodreads.com/
[6]https://www.wikipedia.org/
[7]http://gs.statcounter.com
[8]https://tinyurl.com/29cfkjav

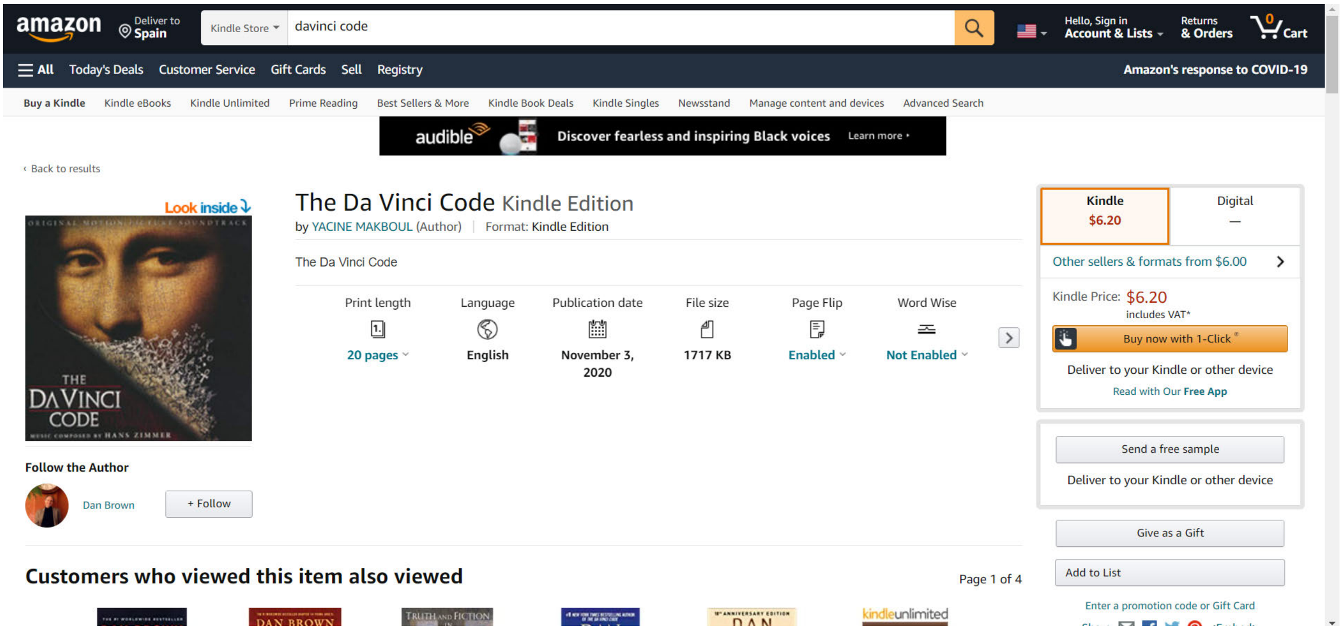


FIGURE 2. Original "The Da Vinci Code" book in Amazon website.

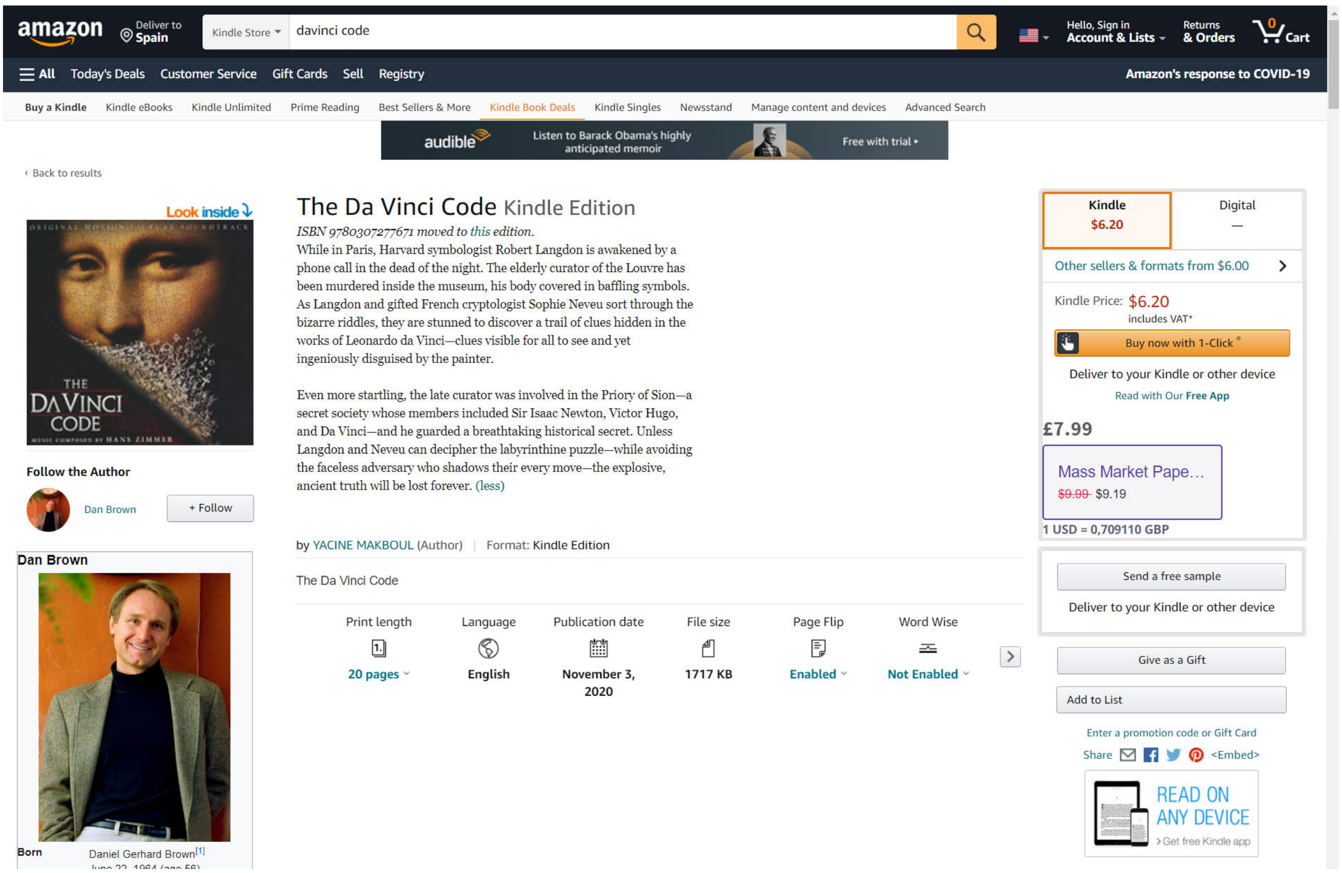


FIGURE 3. Augmented "The Da Vinci Code" book in Amazon website.

element is inserted (see figure 4). This element includes a message requesting from the user to insert the URL where he/she will continue with the task process. If necessary, the user will copy the desired test (author's name) and paste it into the new website (see figure 5). If the process does not need a copied element to continue with the process, Excore will still record all user interactions on the new website. When the user clicks on the "Done!" button, a new tab with

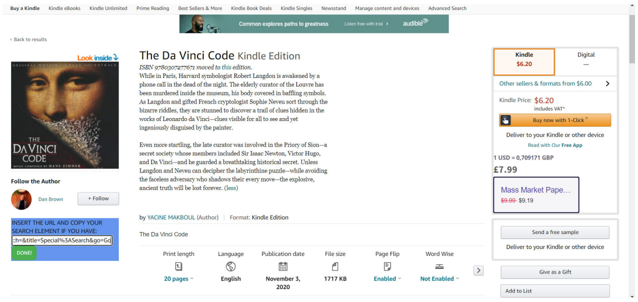


FIGURE 4. Request for the URL of the foreign website.

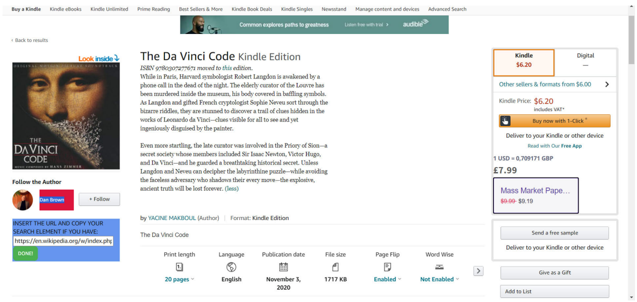


FIGURE 5. Selecting and copying the key word phrase to be used on the foreign website.

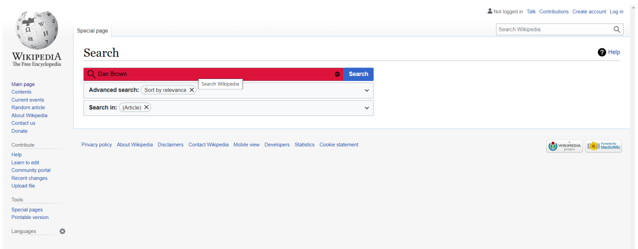


FIGURE 6. Pasting the previously copied key word phrase in the search bar.

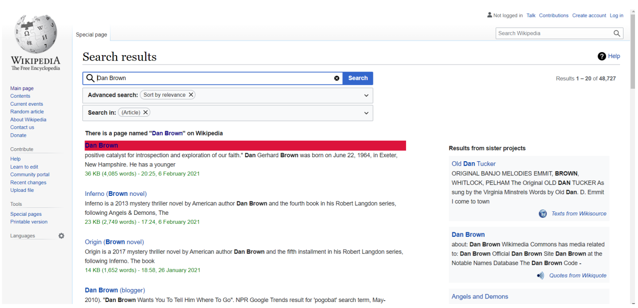


FIGURE 7. Clicking on the desired result item.

the inserted URL opens automatically. Now, the user has to do a common web search by pasting the author's name previously copied into the search bar (see figure 6) and click on the search button. Once the web page with the requested information is loaded, the user clicks on the desired element (see figure 7) or performs all the necessary interactions to obtain the desired element. Finally, the user must double-click on the desired element (see figure 8), the tab closes and the element is inserted into the Amazon website behind the author.

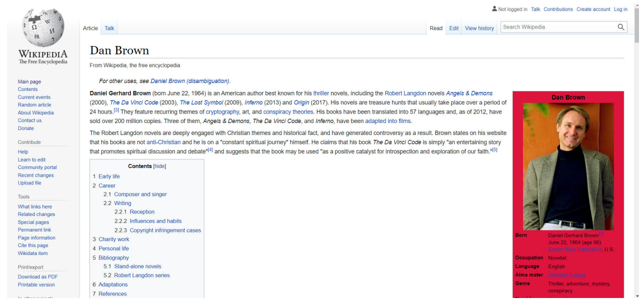


FIGURE 8. Selecting desired node.

The production process is not finished until the "Save" button is clicked on the Excore menu. At this moment, the website is reloaded and the executions process is enacted.

B. AUTOMATION

In this stage, the user is completely passive because Excore executes the process automatically. The system inspects the website address and the process starts in two different cases: 1) when the website address is the same, the automation process is always performed. 2) When the website address is in the same domain, Excore runs only when it locates the anchor points (the location where the new content will be inserted). Otherwise, Excore does not start the automation until a new load event is triggered in the same domain.

When Excore's automation process is initialized, it reproduces the interactions performed by the end user in the previous stage, the production stage. These interactions are not reproduced in the same order or sequence, but are reproduced concurrently. A new tab is opened for each automation process defined by a double-clicking by the user. In each tab, the interaction sequence is reproduced until the target web node is found. When this web element is found, a copy of the content and style is inserted into the augmented web page and the tab is automatically closed. At the end of the process, only the main web page tab will be opened and all desired web nodes of the different web pages will be inserted at their defined position. Figure 9 represents the automation process in which one of the elements, the author information from Wikipedia, has been inserted. This figure shows tabs for the other processes that have not already finished, Goodreads, Bookshop, Waterstone and currency values. This is why, the figure shows 4 tabs opened but none of them is the Wikipedia tab because it has finished that parallel process. Figure 10 shows that the process has not finished because the Goodreads process has not concluded but the other three web nodes have been inserted into position behind the closing tabs of the Amazon website pricing node whose interaction sequence has been completed. The automation process completely avoids tab switching, branching clicks, URL typing, writing and copy/paste actions by the user. Scrolling depends on the position of the new content into which the user has inserted it. This saving of actions should increase user focus and reduce the interruptions and multitasking. Running the automation in parallel should significantly reduce the

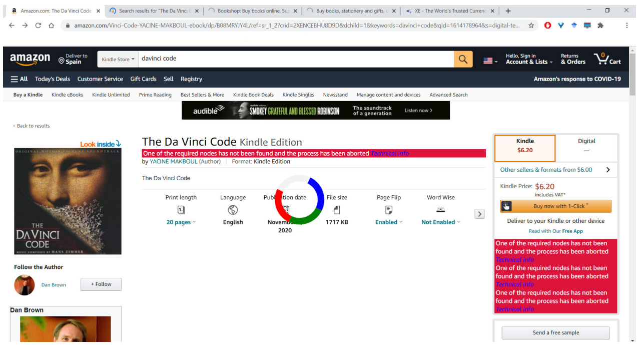


FIGURE 9. **Automation process with one inserted element.**

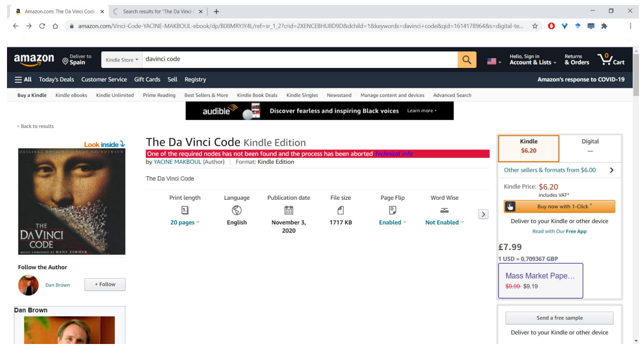


FIGURE 10. **Automation process with most elements inserted.**

execution process time and finish when the longer process finishes. This process will depend on the number of actions, connection speed, website server, etc. To confirm this suspicion, an evaluation has been performed and is explained in depth in section VI.

If the user is no longer interested in running the augmentation process, they can remove it by clicking on the Excore extension menu "Delete" button. When this option is selected, the website is reloaded, removing the previously inserted elements. As a result, the augmentation process will no longer be executed when visiting the website or domain.

V. EXCORE META-MODEL

There are different types of end-user tools depending on their characteristics. [11] divided end-user Development tools into 5 categories: Visual Programming, Spreadsheets, Programming By Demonstration, Domain Specific Languages and Model-Based. Visual Programming tools include visual symbols and graphical notations which are used by end-users as if they were small boxes in which users interact with those components to create their own executable programs. These executable programs must be interpreted by the system using a simple and expressive Domain-Specific Language (DSL). DSLs are considered as an approach to decrease complexity of software systems development. Accompanied by DSL good practices [72] all requirements have been captured by our abstract syntax diagram (see figure 11).

Web augmentation is a set of changes made by users to add content from different resources in their most visited or favourite websites. The augmentation is executed when the load event is enacted. It is pointless to run the augmentation before the event due to the fact that certain web elements might not be loaded yet and consequently, the augmentation process would malfunction. In figure 11, the Excore elements represent the start of the augmentation process, which is described by a unique identifier and an URL expression. If this URL expression matches with the current website address, the customization is executed. If it does not match but it is in the same domain, the augmentation is enacted if any widget anchoring point is identified. Following the Amazon example, not only is the augmentation enacted with the book "The Da Vinci Code" but any book will trigger the augmentation process.

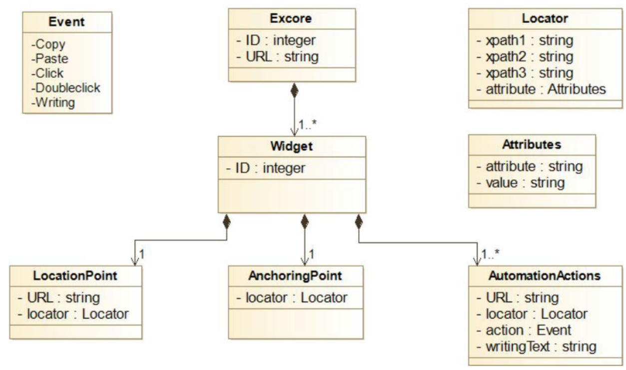


FIGURE 11. **Excore DSL: abstract syntax diagram.**

Widgets are the new inserted nodes extracted from the different resources. Each augmentation have one or more widgets and they are identified by an ID. This ID helps the system to identify the anchor point of each widget. The anchor point indicates the position where the node extracted from a different web page will be inserted in the target website. Each widget will have only one anchor point, which is defined by the locator system. A web locator can be defined as a mechanism for unique identification of an element in the Document Object Model (DOM) [73].

In addition to the anchor, eacg widget needs a location point, which provides information about where the desired element is to be extracted from. To complete this function, the location point needs the URL of the web page from which the web node will be extracted and its locator.

Finally, widgets need a list to store all user interactions during the production process, the automation actions. The system must know in which domain the action is performed and therefore URL is needed. Moreover, it needs the event type (copy, paste, click, double-click or writing) and the text introduced by the user when the writing event is performed. Finally, the locator system is needed to identify on which web node the action was performed.

With regard to the locator system, there are several types of locators: first generation, coordinate-based; second generation, structure-based, (i. e. xpath) and node attributes and third generation, image-based [74]. The robustness of these types of locators is different. Robustness is defined as the ability of a computer system to cope with errors during execution [74].

First generation locators are not used today due to their lack of robustness, they are extremely sensitive to modest changes in the DOM structure and web page layout. If the position of the nodes changes by one pixel, the first generation locator will probably not find the node. According to [75], locators based on node attributes are more robust than those based on structure and these are more robust over time than the image-based locator. In addition, third generation locators are not suitable foruse in Excore because this type of locator tries to always find exactly the same element (image). Each execution is different. If we were to use an image locator in the example of figures 2 and 3, it would have only worked for the book ''The Da Vici Code''. Thus, we have implemented a multiple xpath algorithms. Based on the robustness of xpath, the first xpath algorithms create xpaths based on one attribute starting for IDs, following by class attributes and ending with the rest of attributes. When this is not possible, the criteria of each algorithm are different but some prioritize combination of attributes in the same DOM level and others the combination of attributes at different levels. Finally, taking into account that website updates may cause these three locators to fail, a contingency data is stored. Excore stores all attribute types and their values of their node and all its ancestors. The goal is to regenerate an xpath that uniquely identifies the desired node as shown in previous works: [76]–[78] and [79].

Excore's goal is to mitigate the causes and consequences of the problem described in section II. Automation is the process that helps this mitigation. Therefore, Excore records all interactions that users make so that they can be automatically repeated. Parallelization is the other process. Each widget definition provides enough information to be executed in parallel. Because of this, a new tab is opened and the system reproduces user interactions in parallel reducing the execution time drastically. The results of this last statement are illustrated in the next section.

VI. EVALUATION

Following the Design Science guidelines [19], in this evaluation we will try to validate Excore by testing whether it reduces or eliminates the causes of the problem or its consequences. To achieve this objective, we conducted two user tests for the evaluation of Excore with two different groups of subjects. In the first one, we collected quantitative data comparing the actions of some users in a real task of their daily life and the actions they perform doing the same activity with Excore. This type of evaluation is used to compare exactly how Excore reduces the number of interactions, interruptions and time. The actions that are measured are those reported in causes and consequences described in section II. In the second evaluation, subjects performed a guided activity using Excore and we collected qualitative data. We used some standard questionnaires to collect this data (NASA-TLX (Task Load Index), SUS (System Usability Scale) questionnaires, ASQ (After-Scenario Questionnaire)) and general questions based on the causes and consequences explained in section II. In this evaluation the aim is to check the user's feeling about Excore in terms of number of interactions, interruptions and time. Furthermore, the aim is to check if Excore is usable and if the evaluation activity is adequate (the workload) for the end users and, consequently, the evaluation results are realistic. For this reason, the NASA-TLX, SUS and ASQ questionnaires have been used. 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'' [80]. SUS is used to quickly assess how well people understand the usability of a software application they are working on [81]. ASQ [82] assess participants' satisfaction after completing a task.

A. QUANTITATIVE EVALUATION

1) RESEARCH METHOD

Settings. The study was conducted in Mondragon University (Arrasate-Mondragon, Spain). All participants used their own computers on which Google Chrome was installed.

Procedure. At the beginning, subjects were evaluated one by one and knew nothing about the purpose of the activity. They only know that they were going to be monitored in a repetitive task they performed frequently and was chosen by them on the spot. The next day, they were asked to use Excore to complete the same task and were again monitored. On this day, it was explained to each subject how to use Excore. The second part was performed the following day so that enough time would pass to not remember previous results.

Subjects. Four use cases were performed by end users, without programming skills, working at the university in this evaluation. Each subject works in a different area of the administration and research groups.

Instrument. Each subject completed the activity on their own while a reviewer captured each action performed in the activity. The actions measured are those introduced in section II.

2) RESULTS

The evaluation results are summarized in Tables 2 and 3. The first nine columns of both tables refer to causes described in section II and last two columns refer to the consequences of the same section. During the evaluation, a reviewer wrote down every single action accomplished by the subjects with and without Excore. Table 2 shows the number of each action performed by subjects in their activity without using Excore. In contrast, table 3 shows the actions performed by the subjects in the same activity using Excore. The first use case corresponds to users who work in the library and search for information about books on the web on a regular basis. The second group of users work in the administration and they search for prices for booking flights, hotels, taxis, etc. in order to book the best option for business trips or conferences. The third group of subjects are researchers and frequently, they search for information on journal or conference articles, author information, conference and journal rankings etc. Finally, the fourth group of participants is very active updating and including new material for the subjects they teach. This is why they search for new material, practices and information on the web.

During the activity, the reviewer noted each of the actions performed by the subjects. *Multitasking* was counted when the user started with another action, which was not related with the activity itself. *Interruptions* were counted when the user stopped the activity someone or something disrupts the activity (a phone call or message, a colleague entering to the office…). *Focus* was counted when the user lost his concentration on the activity. Those issues are very related one to each other because in all of them the users breaks the activity for a while but for different reasons. *Branching* was opening a new tab and *URL typing* when the user introduced a URL in the tab. These actions are closely related because when the user opens a new tab they usually introduce a new URL. In some occasions, the user introduces the URL in an opened tab but in most occasions, a new tab is opened. *Tab-switching* was taken into account when the user changed the tab, *Clicks* when they clicked with the mouse, *Scrolling* when they scrolled to find information that was not at the initial position of the website, *Writing* when the user introduced some text such as the nickname and the password (not big text inputs have been included) and *Copy&Paste* when the user copied a piece of text and pasted it in a text input. Finally, *Time* measures the time needed to complete the activity in seconds.

Comparing tables 2 and 3, it is easy to see that the number of actions in each column has decreased significantly for all use cases when using Excore. The number of actions for the first use case were 93 without Excore and 3 with Excore (UC1 in tables 2 and 3). This is a 96.77% reduction in the number of actions. The time required to complete the actions was 1125 seconds without Excore and 248 with Excore, a reduction of 77.96%. The second use case (UC2 in tables 2 and 3) reduces the number of actions from 219 to 11 (a reduction of 94.98%) and from 2156 seconds to 465 (a reduction of 78.43%). The third use case (UC3 in tables 2 and 3) goes from 137 to 8 actions (94.16%) and from 1698 seconds to 259 (84.75%). Finally, the fourth use case (UC4 in tables 2 and 3) executed 128 actions without Excore and 10 with Excore (92.19%) and from 1588 seconds to 292 with the tool (81.61% of reduction). In summary, the number of actions has been reduced by 94.45% (from 577 to 32) and an 80.75% reduction in time (from 6567 to 1264 seconds). The quantitative evaluation shows that Excore significantly reduces the effects of the causes and the consequences of the problem explained in section II.

B. QUALITATIVE EVALUATION

1) RESEARCH METHOD

Settings. The study was conducted in Mondragon University (Arrasate - Mondragon, Spain). All participants used their own laptops, due to Covid19 measurements, in which they had installed Google Chrome.

Procedure. At the very beginning of the evaluation, participants were informed of the purpose of the study and were given a brief description of it. Next, an Excore instance was presented to exemplify the main functionality of the application. The example consisted of adding content to IMDB website from Filmaffinity, Rottentomatoes and a TV guide website. Then, participants were asked to adapt the Amazon website in a defined way and another website they visit frequently to perform repetitive tasks. Finally, the participants were directed to an online Google questionnaire.[9]

Subjects. Thirty one people took part in the evaluation and 61.29% of the participants were men. Participants came from Arrasate-Mondragon and nearby towns. No one had a technical knowledge, the aim of the evaluation was to test Excore with end users. Most of the subjects were working in different fields at the time of the evaluation. These fields were financial, construction, teaching, agricultural or sports. 56.25% of the subjects have never used an editing program such as Photoshop. 96.8% of participants have installed at least one plug-in in their computer's browser. 58.1% of the participants visit more than 10 web pages every day and 67.8% of them spend more than 60 minutes on the Internet every day in their job and 51.6% of them are connected more than 60 minutes in their free time. 71% of the subjects have more than 6 tabs opened all the time.

Instrument. A questionnaire was used to collect the user's experience in the evaluation. The questionnaire was composed of five sections; background, their perceived workload (NASA-TLX questionnaire), usability (SUS questionnaire), satisfaction (ASQ questionnaire) and general questions related with the causes and consequences of the section II. The general questions were measured using different questions with a 7-point Likert scale (1=completely disagree, 7=completely agree).

Data Analysis. Descriptive statistics were used to characterize the sample and to evaluate the participants' experience with Excore.

2) NASA-TLX

NASA-TLX results are reported in Table 4 and summarized in the figure 12. 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 sensations of the subjects during the exercise. We wanted to be sure that the evaluation was balanced in order to obtain objective results about MAWA. For the answers, a Likert scale between 0 and 10 was used.

- Mental Demand: used to determine the subject mental and perceptual activity during the task. The results indicate that the mental demand was in the middle. The average valuation is 4.58 and the median is 5. This might be because it was the first time subjects used Excore. This means that most users were close to the median even though the maximum vote was 8 and the minimum 1. The first box in Figure 12 shows how the opinion of most of the subjects was between 3 and 6.

[9] tinyurl.com/w6dmxqcv

TABLE 2. Number of actions completed by users in a repetitive task without Excore.

	Without Excore										
	Multitasking	Branching	Interruptions	Tab-switching	Copy&Paste	Clicks	Scrolling	URL typing	Writing	Focus	Time
UC1	3	4	4	16	6	38	8	4	4	6	1125
UC2	7	11	9	43	22	63	26	12	15	11	2156
UC3	4	8	7	22	10	45	17	8	9	7	1698
UC4	5	4	7	15	9	39	25	5	10	9	1588

TABLE 3. Number of actions completed by users in a repetitive task with Excore.

	With Excore										
	Multitasking	Branching	Interruptions	Tab-switching	Copy&Paste	Clicks	Scrolling	URL typing	Writing	Focus	Time
UC1	1	0	1	0	0	0	0	0	0	1	248
UC2	0	0	1	0	0	1	3	0	4	2	465
UC3	0	0	0	0	0	2	3	0	2	1	259
UC4	1	0	2	0	0	0	2	0	2	3	292

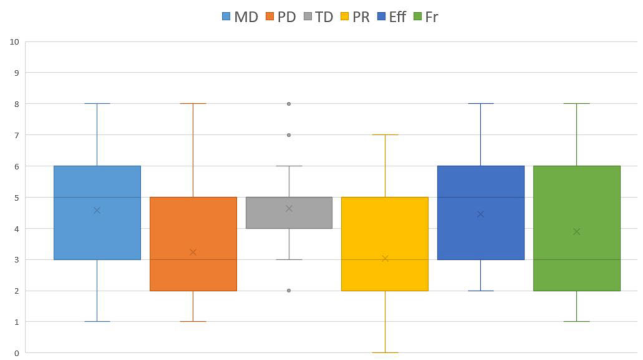


FIGURE 12. NASA-TLX scores.

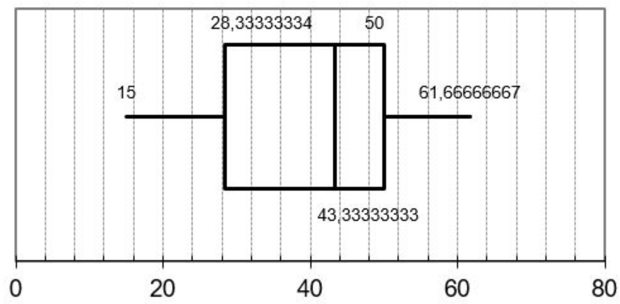


FIGURE 13. Disaggregated NASA-TLX question scores.

TABLE 4. NASA-TLX results.

Feature	AVG	Med	SD	MAX	MIN
Mental demand	4.58	5	1.74	8	1
Physical demand	3.22	3	1.97	8	1
Temporal demand	4.64	5	1.30	8	2
Performance	3.03	3	2.16	7	0
Effort	4.45	5	1.74	8	2
Frustration	3.9	3	2.02	8	1

- Physical Demand: this is closely aligned with the mental demand. This question is related to the physical activity required during the task. The results are lower than the mental demand. Subjects voted in average 3.22 with a median of 3. In this case, most subjects voted below 5, which is the middle value, although the maximum value is 8. We can conclude that mentally and physically, the activity was not demanding for the end users. The second box in Figure 12 shows that most subjects voted between 2 and 5 confirming that the activity was undemanding.
- Temporal Demand: measures the pressure felt by the subject with the tasks accomplished in the evaluation. Most voted are concentrated in the middle of the scale, between 4 and 5. The average is 4.64 and the median 5. This data affirms that the evaluation was not demanding and the subjects do not feel pressure while they were doing the activity. Figure 12 third box shows that most users voted between 4 and 5. Some individual subjects voted more than 6 confirming that pressure was not high but enough not to be basic.
- Performance: is used to know how successful subjects are about the accomplishment of the task. Most of subjects are proud of the activity they completed in the evaluation. The average opinion is 3.03 with a median of 3. The maximum value is 7 and the minimum 0 in the NASA-TLX questionnaire. This means that the activity was adequate for end-users. Figure 12 fourth box shows that most opinion were between 2 and 5. Additionally than more than a 75% of them voted less than 5 hence, subjects feel they carried out the activity successfully.
- Effort: evaluates how demanding the activity was mentally and physically. The effort values are very similar to the mental demand. The average opinion is 4.45 and the median 5. Subjects do not claim that the evaluation exercise was exhausting because results are low. Figure 12 fifth box shows most votes were between 3 and 6. This confirms that the activity was not undemanding and it was very demanding, this is adequate.
- Frustration: measures how irritated, stressed and annoyed the subject feel during the activity. Most subject do not feel frustration during the activity because their global opinion is below 4 (average 3.9). The frustration during the activity is below the mental demand and effort. This indicates that the activity was appropriate

to be completed with the knowledge subjects have. Figure 12 last box shows that most voted were between 2 and 6. Frustration was not high because more than 75% of them voted less than 6.

Standard deviation measures the amount of variation or dispersion of a set of values. The lower the value, the more values are near the average value. Comparing the standard deviations of all questionnaires conducted in this evaluation, NASA-TLX questionnaire has the highest results. This may be the consequences of having the widest Likert scale and the subjects' opinion being dispersed. Analyzing the standard deviation of NASA-TLX, temporal demand is the issue with the lowest value. This is because most of the values are concentrated between 4 and 5. The other topics have a similar standard deviation and most subjects' opinion are concentrated within 3 points.

Figure 13 shows in more detail the disaggregated scores for each questionnaire item. Based on [83], the value must be higher than 60 in order not to consider any issue in NASA-TLX. In our particular case, 39.73 is the value obtained, so we can consider that our results are satisfactory.

3) SUS

SUS results are reported in Table 5 and summarized in the figure 14. SUS is used to measure usability with 10 questions. This usability scale was used in this evaluation because we wanted to know the overall assessment of usability from the user's perspective on Excore. For the answers, SUS uses a Likert scale from 1 to 5.

The interpretation of the score can be complex but in this scale, odd-numbered questions should be higher than 3 and even questions should be below 3 to affirm that the usability of the tool is valid. The evaluation results confirm that Excore is usable. The first question related to their desire to use Excore in the future frequently, most people will use it (average 3.64, median 4). Questions 2 and 3 are closely related because the first question asks whether the tool is unnecessarily complex and the second one whether it is easy to use. Subjects claim that Excore is undemanding for end users (Q2 AVG 2.19, median 2, Q3 AVG 3.58, median 4). Question 4 refers to the need for technical support and the users do not feel the need for help. This question had the lowest value in this questionnaire with a mean of 1.9 and a median of 1. Question 5 refers to the integration of the tool and the participants are satisfied with the result. The integration of content from different websites can move some web elements making the appearance of the website sloppy. Nonetheless, users consider the result to be adequate (AVG 3.58, median 4). Questions 6 refers to the inconsistencies of the tool and the opinion is adequate due to the fact that participants rated this question with a 2.16 and a median of 2. Question 7 asks whether most people would learn to use this system very quickly. This is an important questions considering that Excore is designed for end users. The result fits this objective (AVG 3.74, median 4). Question 8 inquires whether the user found the system very cumbersome to use. The result states that is easy to use. Question 9 asks whether the user felt very confident using the system. The result is lower than expected although the mean is higher than 3. This might be due to the fact that it was the first time the subjects used Excore. The last question inquires about the need to learn many things before using Excore and the participants state that it is not necessary. The mean score is 2.1 and the median 2.

The standard deviation in this questionnaire is very similar. All results are surrounding the 1. This means that, in general, the opinion of the subjects is similar and there are no differences in any of the topics.

Figure 15 shows in more detail the disaggregated scores for each item of the questionnaire. [84] estimates that 65 is the minimum value for which the tool is considered to have no usability problems. The result obtained in the Exocre SUS evaluation is 66.77, so users consider the overall usability of the tool acceptable.

4) ASQ

The results of ASQ are reported in Table 6 and summarized in the figure 16. The ASQ questionnaire is a three-item questionnaire used to assess participant's satisfaction after the completion of an assignment. A Likert scale from 1 to 7 is used in this questionnaire. The questions address three important aspects of user satisfaction with the system: the ease of completing the task, the time to complete the task, and the adequacy of supporting information. The results are very similar to each other (ASQ1 AVG 3.06, ASQ2 AVG 3.16, ASQ3 AVG 3.06). The average confirms that the evaluation tasks were adequate in difficulty and time and the information provided before starting the process was appropriate. The minimum value in all questions is 1 (strongly agree) even the maximum is 5 in the first question and 6 for the other two. The standard deviation is close to one, which confirms that the opinion of most of the subjects is similar. This questionnaire confirms that the evaluation process was correctly introduced to end users who had not used Excore before.

5) GENERAL QUESTIONS

The results of the general question are reported in Table 7 and summarized in figure 17. A Likert scale from 1 to 7 is used in this questoinnaire. These questions evaluate subjects' feeling about Excore's influence in mitigating or eliminating the causes and consequences of the problem described in section II. The more causes and consequences that are solved, the greater Excore's influence on the resolution of the problem. Questions 1 to 9 refer to the causes and questions 10 and 11 refer to the consequences described in section II.

- Q1 multitasking: the subjects state that Excore avoids multitasking. The average opinion is 6.16 and the median is 6. Only one person voted 4 and the others more than 5 (see table 7). This result could be highlighted because Web navigation requires more than one web session to complete and Excore reduces it to one. Additionally, tasks performed via the web are faster with

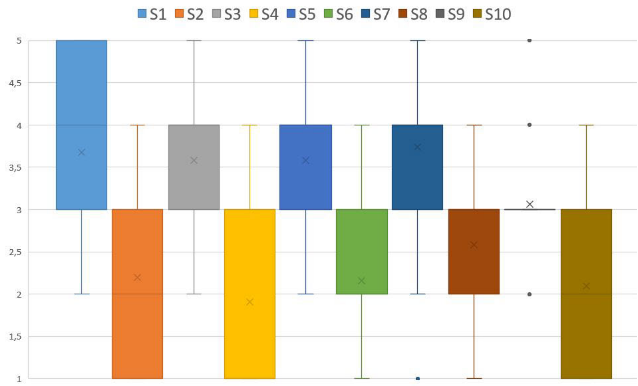


FIGURE 14. SUS scores.

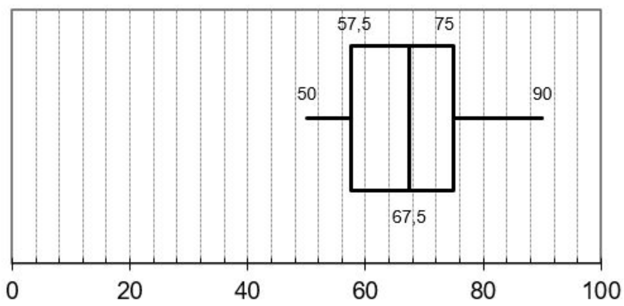


FIGURE 15. Disaggregated NASA-TLX question scores.

TABLE 5. SUS results.

Items	AVG	Med	SD	MAX	MIN
S1	3.67	4	0.98	5	2
S2	2.19	2	0.94	4	1
S3	3.58	4	0.96	5	2
S4	1.9	1	1.11	4	1
S5	3.58	4	0.72	5	2
S6	2.16	2	0.82	4	1
S7	3.74	4	1.06	5	1
S8	2.58	3	0.99	4	1
S9	3.06	3	0.73	5	2
S10	2.1	2	0.98	4	1

Excore and so are less likely to suffer an interruption before completion.

- Q2 branching: subjects consider that branching is avoided by using Excore. The average opinion is 6.22 and the median is 6. In this issue, only one user voted 4 and the others more than 5. Branching is closely related to the multitasking because new tabs are opened when a new task is started. Excore makes unnecessary for the user to open new tabs because they open and close automatically in the background. By having all the desired information on the same website, branching is not essential.
- Q3 interruptions: are frequent in the daily routine in any job, but are unlikely in activities that are completed quickly. Subjects think Excore helps reduce interruptions (AVG 6.19, median 7). Most subjects voted more than 5, with the exception of two participants, and half voted the maximum (thus the median is 7). Interruptions

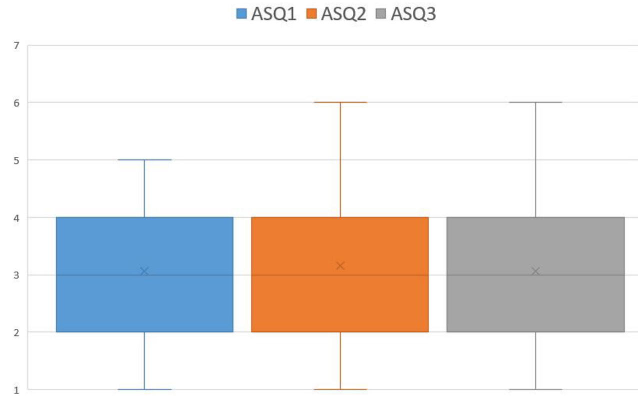


FIGURE 16. ASQ scores.

TABLE 6. ASQ results.

Items	AVG	Med	SD	MAX	MIN
ASQ1	3.06	3	1.21	5	1
ASQ2	3.16	3	1.37	6	1
ASQ3	3.06	3	1.12	6	1

make a task take longer and decreasing the number of pauses in a task is critical.

- Q4 tab-switching: how many tabs do you usually have open in your browser? 71% of participants have more than 6 open. It is not uncommon to see people with dozens where the title of the website is unreadable. Subjects report that Excore reduces a large number of tab-switches in their web activities. Excore is exceptionally well rated by users (AVG 6.38, median 7). Moreover, the minimum vote is 5 and more than half of the subjects voted 7, the maximum score. There is no doubt that Excore drastically reduces tab-switching.
- Q5 copy and paste: this is a frequent action performed by users in their web activity. If the user does not perform this action, the task will be completed faster. Additionally, some errors can be avoided such as not copying the complete text, pasting the text in the wrong field, copying text from one website and having problems to detect the website in which the user wants to past the text, etc. Subjects consider that Excore helps to avoid copying and pasting (AVG 5.55, median 6). Although the result is lower than the previous topics, it is an excellent result (maximum vote 7, minimum vote 2).
- Q6 clicks: this can be considered the most frequent action in the web. This action is risky because the user can click in dangerous elements when browsing some websites. [85] shows that these dangerous situations are extremely common in all types of domains, which makes a large number of users vulnerable to different possible attacks. Furthermore, in section II click related problems have been explained. Reducing the number of clicks is essential. Subjects claim that Excore reduces the number of clicks (AVG 5.68, median 6). Most votes are between 4 and 7, with the exception of 1 vote. Considering that

Excore automatically processes all user actions, clicks are avoided by user actions and the risk of clicking dangerous elements is eliminated.

- Q7 scrolling: is a common action in web browsing. Users scroll to find information within a web page. Depending on the location of the information being sought, more or less scrolling is required. The subjects consider that Excore does not help to solve this cause as much the causes shown above. However, the score is excellent (AVG 5.03, median 5) and confirms that Excore reduces scrolling. The maximum value was 6 and the minimum 2. Some subjects disagreed with this statement, but most of the votes were above 4.
- Q8 URL typing: Excore automates the entire process and user does not have to open a new tab and introduce any URLs because they are included in the creation process. Hence, URL typing is completely unnecessary for the user. The opinion of the participants in this question corroborates the statement because the result is outstanding (AVG 6.19, median 6). The minimum vote was 5 confirming that all users agree that Excore eliminates the URL typing action in the web while users act on it.
- Q9 writing: this is not the most common web action. Inputs and text areas are common elements in web forms where a user must type some data required by the developer to perform out a certain action. Excore can automatize some form processes and some repetitive inputs but it is not possible when new content must be written. This could be the reason why this issue has the lowest rating in the questionnaire (AVG 4.39, median 4) although the result is not unsatisfactory because it is more than the middle value. This means that, from subjects' point of view, the writing issue is solved by Excore but not as much the other ones.
- Q10 focus: maintaining the focus on your web actions is crucial, otherwise, a basic action may be prolonged in time. The faster a web page loads, the better it is to maintain focus on the task. Subjects agree with this statement and claim that Excore is useful to maintain the focus on a task (AVG 6.26, median 6). The maximum value is 7 and the minimum 4, which means that all participant approved Excore for this statement.
- Q11 time: saving time is closely related to the focus, as explained above. Besides, users prefer to perform a task as fast as possible and if an action needs visiting a large number of websites, the loading velocity is fundamental. Excore loads all the necessary information on a web page but fetches all the information in parallel. This means that the maximum time it needs to obtain all the information is the time needed to load the slowest page or the process with more steps. In the opinion of participants, this issue is the best rated with an average rating of 6.45 and a median of 7. This is because more than half of them voted 7. Additionally, the minimum vote was 5, which means that the subjects ensure the effectiveness of Excore by reducing the time needed to obtain the desired information.

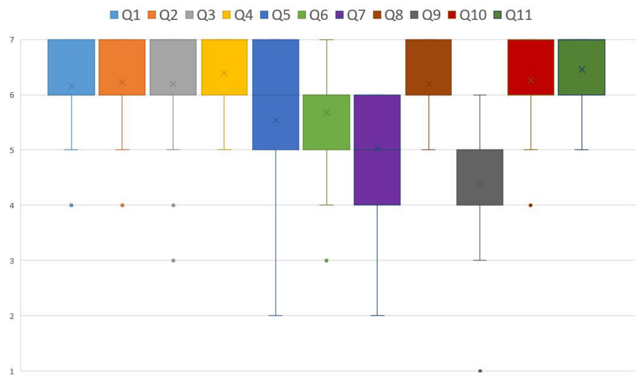


FIGURE 17. Question scores.

TABLE 7. Question results.

Items	AVG	Med	SD	MAX	MIN
Q1	6.16	6	0.86	7	4
Q2	6.22	6	0.88	7	4
Q3	6.19	7	1.04	7	3
Q4	6.38	7	0.71	7	5
Q5	5.55	6	1.26	7	2
Q6	5.68	6	1.01	7	3
Q7	5.03	5	1.14	6	2
Q8	6.19	6	0.75	7	5
Q9	4.39	4	1.14	6	1
Q10	6.26	6	0.86	7	4
Q11	6.45	7	0.72	7	5

The standard deviation in these questions is around 1. This means that the opinions of the subjects are similar to each other and that there is no subject who thinks Excore is useless to solve these 11 drawbacks. The standard deviation is higher in the question which maximum and minimum value are more different. This is understandable and means that the results are adequate.

To sum up, qualitative evaluation confirms that Excore mitigates all the causes and consequences of the problem it aims to solve. With the exception of writing issue, subjects claim that Excore is an excellent tool that reduces the drawbacks motivated by repetitive tasks in the users' web experience.

VII. CONCLUSION AND FUTURE WORK

With the unstoppable growth of the web, end users have been forced to consume information on different resources. This information consumption activity is ofter repetitive (performed frequently). Repetitive activity on the web includes branching, tab switching, clicking, copy and pasting, scrolling, URL typing and writing content on web inputs. However, this is not all, in the course of the activity users endure interruptions caused by other people, phone calls, emails…In addition, people also tend to start another task before finishing the previous web activity (multitasking). The concentration required to perform these repetitive tasks and the relative ease with which end users make mistakes leads to frustration and disorientation. The consequence is the loss of focus on the activity and that the activity is prolonged in time. With the intention of solving these drawbacks, we have proposed WA as a technique to reduce the number of repetitive tasks performed and the time consumed during this operations. To this end, in this paper we have presented Excore, a Chrome browser extension for Web Augmentation. We have shown how it works by illustrating it with an example.

Excore is a visual programming tool that captures all user interactions during their activity in the production process. The users recreate their web interactions as normal with the exception of having to indicate the insertion location of the new element. This location will be used to insert information from an external website. When the current web address is appropriate, it automatically reproduces in parallel all previous user interactions in the automation process. The parallel processes are based on new elements to be inserted by the system. The more elements, the more parallel processes. In this way, we drastically reduce the time required to obtain the desired information. At all times, the user's focus remains on the augmented page and the process is executed in the background. The information needed to reproduce the user's interactions is stored in the browser storage following the architecture designed for this purpose.

Two user studies have been conducted: a quantitative and a qualitative evaluation. The quantitative evaluation has been performed with four use cases analyzing a repetitive task. All causes and consequences of the problem of repetitive web browser interactions in the same activity with and without Excore have been measured. The total number of actions have been reduced by 94.45% and the time required to complete the activity has been reduced by 80.75% with the use of Excore. As for the qualitative evaluation, four different questionnaires have been carried out, NASA-TLX (perceived workload), SUS (usability), ASQ (satisfaction) and general question related to the causes and consequences of the problem of repetitive tasks. The NASA-TLX questionnaire certifies that the evaluation has been appropriate and the ASQ questionnaire ensures that it has been successfully completed. The SUS questionnaire verifies that Excore is usable for end users. Overall, the results are positive. All cause and consequence issues have been rated positively by the evaluation subjects, although the writing issue has been borderline. Despite the fact that the evaluation subjects had no experience in the use of Excore or WA tools, they were able to achieve the proposed goals.

Future developments will include support to facilitate debugging of the resulting applications by end users. If Excore fails to retrieve content from different resources, the system will help end users detect the reason why this has happened. Furthermore, Excore will help end users to repair the failure or update the transition to the new requirements. For example, web updates may cause users to have to perform additional interaction. Rather than creating a new augmentation, the aim of this debugging feature is to help end users update existing ones.

Initially, Excore has not been designed to share augmentations with other users. Nonetheless, this feature could be crucial for the use of the extension to increase. Clicking on a new button in the Excore menu would download a file with the basic information to reproduce the enhancement in another browser. This file could be then shared among end users. To enable further sharing of augmentations, an Excore community could be created. In this web community, all users could upload their own creation to be downloaded and used by any community member. The debugging feature will help to adjust other members' augmentations to suit the user's needs. Lastly, the community will make possible for Excore developments to be done collaboratively. This could boost Excore to be utilized worldwide.

REFERENCES

[1] G. Desolda, C. Ardito, M. F. Costabile, and M. Matera, "End-user composition of interactive applications through actionable UI components," *J. Vis. Lang. Comput.*, vol. 42, pp. 46–59, Oct. 2017.

[2] D. Massa and L. Spano, "FaceMashup: An end-user development tool for social network data," *Future Internet*, vol. 8, no. 4, p. 10, Mar. 2016.

[3] G. Ghiani, F. Paternò, L. D. Spano, and G. Pintori, "An environment for end-user development of web mashups," *Int. J. Hum.-Comput. Stud.*, vol. 87, pp. 38–64, Mar. 2016.

[4] D. Firmenich, S. Firmenich, J. M. Rivero, L. Antonelli, and G. Rossi, "CrowdMock: An approach for defining and evolving web augmentation requirements," *Requirements Eng.*, vol. 23, no. 1, pp. 33–61, Mar. 2018.

[5] O. Díaz, J. D. Sosa, and S. Trujillo, "Activity fragmentation in the web: Empowering users to support their own webflows," in *Proc. 24th ACM Conf. Hypertext Social Media ECRC (HT)*, Paris, France, May 2013, pp. 69–78.

[6] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-user development: An emerging paradigm," in *End User Development*. Dordrecht, The Netherlands: Springer, 2006, pp. 1–8.

[7] A. Cypher, M. Dontcheva, T. Lau, and J. Nichols, *No Code Required: Giving Users Tools to Transform the Web*. San Mateo, CA, USA: Morgan Kaufmann, 2010.

[8] S. Barman, S. Chasins, R. Bodik, and S. Gulwani, "Ringer: Web automation by demonstration," in *Proc. ACM SIGPLAN Int. Conf. Object-Oriented Program., Syst., Lang., Appl. (OOPSLA)*, Amsterdam, The Netherlands, Oct. 2016, pp. 748–764.

[9] M. Spahn, C. Dörner, and V. Wulf, "End user development: Approaches towards a flexible software design," in *Proc. 16th Eur. Conf. Inf. Syst. (ECIS)*, Galway, Republic of Ireland, 2008, pp. 303–314.

[10] J. A. Macías and F. Paternò, "Customization of web applications through an intelligent environment exploiting logical interface descriptions," *Interacting With Comput.*, vol. 20, no. 1, pp. 29–47, Jan. 2008.

[11] I. Aldalur, M. Winckler, O. Díaz, and P. Palanque, "Web augmentation as a promising technology for end user development," in *New Perspectives in End-User Development*. Cham, Switzerland: Springer, 2017, pp. 433–459.

[12] G. Karsai, A. Lang, and S. Neema, "Design patterns for open tool integration," *Softw. Syst. Model.*, vol. 4, no. 2, pp. 157–170, May 2005.

[13] M. AlSada and T. Nakajima, "Parallel web browsing in tangible augmented reality environments," in *Proc. 33rd Annu. ACM Conf. Extended Abstr. Hum. Factors Comput. Syst. (CHI)*, Seoul, South Korea, Apr. 2015, pp. 953–958.

[14] V. F. de Santana, R. de Oliveira, L. D. A. Almeida, and M. Ito, "Firefixia: An accessibility web browser customization toolbar for people with dyslexia," in *Proc. Int. Cross-Disciplinary Conf. Web Accessibility (WA)*, Rio de Janeiro, Brazil, May 2013, pp. 16:1–16:4.

[15] N. O. Bouvin, "Unifying strategies for web augmentation," in *Proc. 10th ACM Conf. Hypertext Hypermedia, Returning Our Diverse Roots (HYPERTEXT)*, Darmstadt, Germany, Feb. 1999, pp. 91–100.

[16] O. Díaz, C. Arellano, and M. Azanza, "A language for end-user web augmentation: Caring for producers and consumers alike," *ACM Trans. Web*, vol. 7, no. 2, pp. 9:1–9:51, May 2013.

[17] M. Bolin, M. Webber, P. Rha, T. Wilson, and R. C. Miller, "Automation and customization of rendered web pages," in *Proc. 18th Annu. ACM Symp. User Interface Softw. Technol.*, Seattle, WA, USA, Oct. 2005, pp. 163–172.

[18] G. Leshed, E. M. Haber, T. Matthews, and T. A. Lau, "CoScripter: Automating & sharing how-to knowledge in the enterprise," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, Florence, Italy, Apr. 2008, pp. 1719–1728.

[19] M. L. Markus, A. Majchrzak, and L. Gasser, "A design theory for systems that support emergent knowledge processes," *Mis Quart.*, vol. 26, no. 3, pp. 179–212, 2002.

[20] T. Y. Lee and B. B. Bederson, "Give the people what they want: Studying end-user needs for enhancing the web," *PeerJ Comput. Sci.*, vol. 2, p. e91, Nov. 2016, doi: 10.7717/peerj-cs.91.

[21] R. Mehrotra, P. Bhattacharya, and E. Yilmaz, "Characterizing users' multi-tasking behavior in web search," in *Proc. ACM Conf. Hum. Inf. Interact. Retr. (CHIIR)*, Carrboro, NC, USA, Mar. 2016, pp. 297–300.

[22] B. M. Kay and C. Watters, "Exploring multi-session web tasks," in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, Florence, Italy, Apr. 2008, pp. 1187–1196.

[23] H. Zhang and S. Zhao, "Measuring web page revisitation in tabbed browsing," in *Proc. Int. Conf. Hum. Factors Comput. Syst. (CHI)*, Vancouver, BC, Canada, May 2011, pp. 1831–1834.

[24] J. Huang and R. W. White, "Parallel browsing behavior on the web," in *Proc. 21st ACM Conf. Hypertext Hypermedia (HT)*, Toronto, ON, Canada, Jun. 2010, pp. 13–18.

[25] S. Raghavan, U. Parampalli, and S. V. Raghavan, "Re-engineering simultaneous internet sessions process-separated browsers," in *Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)*, Geelong, VIC, Australia, Jan./Feb. 2017, pp. 70:1–70:10.

[26] J. Huang, T. Lin, and R. W. White, "No search result left behind: Branching behavior with browser tabs," in *Proc. 5th Int. Conf. Web Search Web Data Mining (WSDM)*, Seattle, WA, USA, Feb. 2012, pp. 203–212.

[27] J. C. Chang, N. Hahn, Y. Kim, J. Coupland, B. Breneisen, H. S. Kim, J. Hwong, and A. Kittur, "When the tab comes due: Challenges in the cost structure of browser tab usage," in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, 2021, pp. 1–15.

[28] R. Tahir, A. Raza, F. Ahmad, J. Kazi, F. Zaffar, C. Kanich, and M. Caesar, "It's all in the name: Why some URLs are more vulnerable to typosquatting," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 2618–2626.

[29] A. Aula, R. M. Khan, and Z. Guan, "How does search behavior change as search becomes more difficult?" in *Proc. 28th Int. Conf. Hum. Factors Comput. Syst. (CHI)*, Atlanta, GA, USA, Apr. 2010, pp. 35–44.

[30] K. Athukorala, D. Głowacka, G. Jacucci, A. Oulasvirta, and J. Vreeken, "Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 11, pp. 2635–2651, Nov. 2016.

[31] A. T. Scaria, R. M. Philip, R. West, and J. Leskovec, "The last click: Why users give up information network navigation," in *Proc. 7th ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, Feb. 2014, pp. 213–222.

[32] E. Adar, J. Teevan, and S. T. Dumais, "Large scale analysis of web revisitation patterns," in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, Florence, Italy, Apr. 2008, pp. 1197–1206.

[33] B. Kules and B. Shneiderman, "Users can change their web search tactics: Design guidelines for categorized overviews," *Inf. Process. Manage.*, vol. 44, no. 2, pp. 463–484, Mar. 2008.

[34] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, Oct. 2011, pp. 139–150.

[35] *The Research-Based Web Design & Usability Guidelines, Enlarged/Expanded Edition*, United States Dept. Health Hum. Services, U.S. Government Printing Office, Essentials Res. Methods Psychol., Washington, DC, USA, 2006.

[36] M. Vigo and S. Harper, "Real-time detection of navigation problems on the world 'wild' web," *Int. J. Hum.-Comput. Stud.*, vol. 101, pp. 1–9, May 2017.

[37] K. T. Stolee, S. Elbaum, and G. Rothermel, "Revealing the copy and paste habits of end users," in *Proc. IEEE Symp. Vis. Lang. Hum.-Centric Comput. (VL/HCC)*, Corvallis, OR, USA, Sep. 2009, pp. 59–66.

[38] K. Roberts, A. Cahan, and D. Demner-Fushman, "Error propagation in EHRs via copy/paste: An analysis of relative dates," in *Proc. Amer. Med. Inform. Assoc. Annu. Symp. (AMIA)*, Washington, DC, USA, Nov. 2014, p. 1.

[39] D. C. McFarlane and K. A. Latorella, "The scope and importance of human interruption in human-computer interaction design," *Hum.-Comput. Interact.*, vol. 17, no. 1, pp. 1–61, Mar. 2002.

[40] J. S. Rubinstein, D. E. Meyer, and J. E. Evans, "Executive control of cognitive processes in task switching," *Exp. Psychol., Hum. Perception Perform.*, vol. 27, no. 1, pp. 763–797, 2001.

[41] R. Schweickert and G. J. Boggs, "Models of central capacity and concurrency," *J. Math. Psychol.*, vol. 28, no. 3, pp. 223–281, Sep. 1984.

[42] B. Callegari, M. M. de Resende, and M. da Silva Filho, "Hand rest and wrist support are effective in preventing fatigue during prolonged typing," *J. Hand Therapy*, vol. 31, no. 1, pp. 42–51, Jan. 2018.

[43] C. Gutwin, S. Bateman, G. Arora, and A. Coveney, "Looking away and catching up: Dealing with brief attentional disconnection in synchronous groupware," in *Proc. ACM Conf. Comput. Supported Cooperat. Work Social Comput. (CSCW)*, Portland, OR, USA, Feb./Mar. 2017, pp. 2221–2235.

[44] B. Cody-Kenny, U. Manganiello, J. Farrelly, A. Ronayne, E. Considine, T. McGuire, and M. O'Neill, "Investigating the evolvability of web page load time," in *Proc. 21st Int. Conf. Appl. Evol. Comput.*, in Lecture Notes in Computer Science, Parma, Italy, vol. 10784. Cham, Switzerland: Springer, Apr. 2018, pp. 769–777.

[45] W. Stadnik and Z. Nowak, "The impact of web pages' load time on the conversion rate of an e-commerce platform," in *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology—ISAT 2017* (Advances in Intelligent Systems and Computing), Szklarska Poręba, Poland, vol. 655. Cham, Switzerland: Springer, Sep. 2017, pp. 336–345.

[46] K. Kuutti, "The concept of activity as a basic unit of analysis for CSCW research," in *Proc. 2nd Eur. Conf. Comput. Supported Cooperat. Work*, Amsterdam, The Netherlands, Sep. 1991, pp. 249–264.

[47] V. M. González and G. Mark, "'Constant, constant, multi-tasking craziness' managing multiple working spheres," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, Vienna, Austria, Apr.2004, pp. 113–120.

[48] J. E. Bardram, J. Bunde-Pedersen, and M. Søgaard, "Support for activity-based computing in a personal computing operating system," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, Montréal, QC, Canada, Apr. 2006, pp. 211–220.

[49] H. Sanchez, R. Robbes, and V. M. González, "An empirical study of work fragmentation in software evolution tasks," in *Proc. IEEE 22nd Int. Conf. Softw. Anal., Evol., Reeng. (SANER)*, Montreal, QC, Canada, Mar. 2015, pp. 251–260.

[50] L. C. Cruz, H. Sanchez, V. M. González, and R. Robbes, "Work fragmentation in developer interaction data," *J. Softw., Evol. Process*, vol. 29, no. 3, p. e1839, Mar. 2017.

[51] S. Firmenich, I. Garrigós, and M. Wimmer, "(De-)composing web augmenters," in *Proc. 14th Int. Conf. Web Eng. (ICWE)*, in Lecture Notes in Computer Science, Toulouse, France, vol. 8541. Cham, Switzerland: Springer, Jul. 2014, pp. 359–369.

[52] H. Franssila, "Work fragmentation, task management practices and productivity in individual knowledge work," in *Proc. 16th Int. Conf. Eng. Psychol. Cognit. Ergonom. (EPCE), 21st HCI Int. Conf. HCII*, in Lecture Notes in Computer Science, Orlando, FL, USA, vol. 11571. Cham, Switzerland: Springer, Jul. 2019, pp. 29–38.

[53] M. Winckler, R. A. Cava, E. Barboni, P. A. Palanque, and C. Freitas, "Usability aspects of the inside-in approach for ancillary search tasks on the web," in *Proc. 15th IFIP TC Int. Conf. Hum.-Comput. Interact. (INTERACT)*, Bamberg, Germany, Sep. 2015, pp. 211–230.

[54] S. Liu and K. Tajima, "WildThumb: A web browser supporting efficient task management on wide displays," in *Proc. 15th Int. Conf. Intell. User Interfaces (IUI)*, Hong Kong, Feb. 2010, pp. 159–168.

[55] J. E. Garrido, V. M. R. Penichet, M. D. Lozano, A. J. Quigley, and P. O. Kristensson, "AwToolkit: Attention-aware user interface widgets," in *Proc. Int. Work. Conf. Adv. Vis. Interfaces (AVI)*, Como, Italy, May 2014, pp. 9–16.

[56] T. Okoshi, J. Ramos, H. Nozaki, J. Nakazawa, A. K. Dey, and H. Tokuda, "Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. (UbiComp)*, Osaka, Japan, Sep. 2015, pp. 475–486.

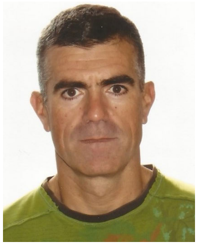
[57] B. Jiang, P. Liu, Y. Wang, and Y. Chen, "HyOASAM: A hybrid open API selection approach for mashup development," *Math. Problems Eng.*, vol. 2020, pp. 1–16, Apr. 2020.

[58] A. J. Fernández-García, L. Iribarne, A. Corral, J. Criado, and J. Z. Wang, "A microservice-based architecture for enhancing the user experience in cross-device distributed mashup UIs with multiple forms of interaction," *Universal Access Inf. Soc.*, vol. 18, no. 4, pp. 747–770, Nov. 2019.

[59] G. Wainer and S. Wang, "MAMS: Mashup architecture with modeling and simulation as a service," *J. Comput. Sci.*, vol. 21, pp. 113–131, Jul. 2017.
[60] O. Chudnovskyy, T. Nestler, M. Gaedke, F. Daniel, J. I. Fernández-Villamor, V. I. Chepegin, J. A. Fornas, S. Wilson, C. Kögler, and H. Chang, "End-user-oriented telco mashups: The OMELETTE approach," in *Proc. 21st World Wide Web Conf. (WWW)*, Lyon, France, Apr. 2012, pp. 235–238.
[61] G. Bosetti, S. Firmenich, A. Fernández, M. Winckler, and G. Rossi, "From search engines to augmented search services: An end-user development approach," in *Proc. 17th Int. Conf. Web Eng. (ICWE)*, Rome, Italy, Jun. 2017, pp. 115–133.
[62] A. J. Fernández-García, L. Iribarne, A. Corral, J. Criado, and J. Z. Wang, "A flexible data acquisition system for storing the interactions on mashup user interfaces," *Comput. Standards Interfaces*, vol. 59, pp. 10–34, Aug. 2018.
[63] Z. Bellal, N. Elouali, S. M. Benslimane, and C. Acarturk, "Integrating mobile multimodal interactions based on programming by demonstration," *Int. J. Hum.-Comput. Interact.*, vol. 37, no. 5, pp. 1–16, 2020.
[64] P. Valderas, V. Torres, and V. Pelechano, "A social network for supporting end users in the composition of services: Definition and proof of concept," *Computing*, vol. 102, no. 8, pp. 1909–1940, Aug. 2020.
[65] T. J. Li, A. Azaria, and B. A. Myers, "SUGILITE: Creating multimodal smartphone automation by demonstration," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Denver, CO, USA, May 2017, pp. 6038–6049.
[66] T. Veuskens, K. Luyten, and R. Ramakers, "Rataplan: Resilient automation of user interface actions with multi-modal proxies," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 2, pp. 60:1–60:23, 2020.
[67] Y. Gil, V. Ratnakar, and C. Fritz, "TellMe: Learning procedures from tutorial instruction," in *Proc. 16th Int. Conf. Intell. User Interfaces (IUI)*, Palo Alto, CA, USA, Feb. 2011, pp. 227–236.
[68] A. R. Sereshkeh, G. Leung, K. Perumal, C. Phillips, M. Zhang, A. Fazly, and I. Mohomed, "VASTA: A vision and language-assisted smartphone task automation system," in *Proc. 25th Int. Conf. Intell. User Interfaces (IUI)*, Cagliari, Italy, Mar. 2020, pp. 22–32.
[69] L. Castaneda, N. M. Villegas, and H. A. Müller, "Self-adaptive applications: On the development of personalized web-tasking systems," in *Proc. 9th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, Hyderabad, India, Jun. 2014, pp. 49–54.
[70] X. Li, Y. Liu, R. Cai, and S. Ma, "Investigation of user search behavior while facing heterogeneous search services," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., Feb. 2017, pp. 161–170.
[71] S. Amershi, J. Mahmud, J. Nichols, T. Lau, and G. A. Ruiz, "LiveAction: Automating web task model generation," *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 3, pp. 14:1–14:23, 2013.
[72] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, 2005.
[73] F. Ricca, M. Leotta, A. Stocco, D. Clerissi, and P. Tonella, "Web testware evolution," in *Proc. 15th IEEE Int. Symp. Web Syst. Evol. (WSE)*, Eindhoven, The Netherlands, Sep. 2013, pp. 39–44.
[74] I. Aldalur, F. Larrinaga, and A. Perez, "ABLA: An algorithm for repairing structure-based locators through attribute annotations," in *Proc. 21st Int. Conf. Web Inf. Syst. Eng. (WISE)*, Amsterdam, The Netherlands, vol. 12343. Cham, Switzerland: Springer, Oct. 2020, pp. 101–113.
[75] M. Leotta, D. Clerissi, F. Ricca, and P. Tonella, "Capture-replay vs. programmable web testing: An empirical assessment during test case evolution," in *Proc. 20th Work. Conf. Reverse Eng. (WCRE)*, Koblenz, Germany, Oct. 2013, pp. 272–281.
[76] H. Kirinuki, H. Tanno, and K. Natsukawa, "COLOR: Correct locator recommender for broken test scripts using various clues in web application," in *Proc. IEEE 26th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Hangzhou, China, Feb. 2019, pp. 310–320.
[77] A. Stocco, R. Yandrapally, and A. Mesbah, "Visual web test repair," in *Proc. ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng. (ESEC/SIGSOFT FSE)*, Lake Buena Vista, FL, USA, Nov. 2018, pp. 503–514.
[78] S. R. Choudhary, D. Zhao, H. Versee, and A. Orso, "Water: Web application test repair," in *Proc. 1st Int. Workshop End-End Test Script Eng. (ETSE)*, New York, NY, USA, 2011, pp. 24–29.
[79] I. Aldalur and O. Díaz, "Addressing web locator fragility: A case for browser extensions," in *Proc. ACM SIGCHI Symp. Eng. Interact. Comput. Syst. (EICS)*, Lisbon, Portugal, Jun. 2017, pp. 45–50.
[80] S. G. Hart, "Nasa-task load index (NASA-TLX); 20 years later," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*. Los Angeles, CA, USA: SAGE Publications, 2006, pp. 904–908.
[81] J. Brooke, "SUS: A retrospective," *J. Usability Stud.*, vol. 8, no. 2, pp. 29–40, 2013.
[82] J. R. Lewis, "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use," *Int. J. Hum.-Comput. Interact.*, vol. 7, no. 1, pp. 57–78, Jan. 1995.
[83] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," *Adv. Psychol.*, vol. 52, pp. 139–183, Apr. 1988.
[84] J. Brooke, "SUS: A 'quick and dirty' usability scale," in *Usability Evaluation in Industry*, vol. 189. 1996.
[85] I. Sánchez-Rola, D. Balzarotti, C. Kruegel, G. Vigna, and I. Santos, "Dirty clicks: A study of the usability and security implications of click-related behaviors on the web," in *Proc. Web Conf. WWW*, Taipei, Taiwan, Apr. 2020, pp. 395–406.

IÑIGO ALDALUR received the B.Sc. degree in computer science from the University of the Basque Country, Spain, in 2009, and the Ph.D. degree from the Department of Computer Languages and Systems, University of the Basque Country, in 2017. He is currently a member of the Information Systems Research Group, Department of Electronics and Computer Science. His research interests include software engineering, web engineering, and human–computer interaction.

ALAIN PEREZ received the Ph.D. degree in computer science with the thesis "Semantic Web and Semantic Technologies to Enhance Innovation and Technology Watch Process," also by MGEP-MU, in 2016. He is a Computer Science Engineer in 2010. He is currently a Lecturer and a Researcher with the IT Department, MU-EPS. Since 2009, he has been working on research projects related to interoperability, the IoT, smart cities and semantic web, in which stand out: Cityfied, Rennovates, and SmartEnCity.



FELIX LARRINAGA received the B.Sc. degree (Hons.) in industrial electronics from Mondragon University, in 1995, the Ph.D. degree in mobile communications from Staffordshire University, U.K., in 1999, and the M.Sc. degree in computing engineering, in 2010. Since 2000, he combines lecturing and research with Mondragon University with an extensive experience in industrial work as a project leader. He is currently a Lecturer and a Researcher with Mondragon University, Basque Country, Spain. He has actively participated in the creation of the new research team on web engineering with the Faculty of Engineering. His research activities of the group are in the areas of interoperability, web architecture, microservices, and semantic web technologies. The team works on multidisciplinary projects with other universities, technological centers and enterprises in areas such as energy, health or industry. He is working on several projects related to manufacturing, smart cities and energy (QU4LITY, Productive 4.0, MANTIS, ARROWHEAD, CITYFIED, RENNOVATES, and SMARTENCITY). He is expertise in web engineering, interoperability, web architecture, and semantic web.