Customizing Websites Through Automatic Web Search

Iñigo Aldalur[(✉)], Alain Perez[(✉)], and Felix Larrinaga[(✉)]

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

Abstract. The WorldWide Web has endured an incredible growth in the last decades. Nowadays, we can visit an unimaginable number of websites from different devices (laptops, mobiles, tablets...) in order to obtain immediate information. However, this information is separated in different resources and Web information search is unpleasant. The feeling of the users is frustrating when collecting information from different resources. Techniques such as Web personalization and Web customization have been an important research area during the last years. Web customization techniques have been widely used to develop website adaptation along the WorldWide Web. This customization is frequently performed by end-users that use the numerous tools available to carry out this assignment.

This article presents Excore, a Web customization tool that permits end-users to customize their websites with automatic Web searches. The article presents the benefits introduced by Excore as a response to the drawbacks end-users experience while they perform their Web activities. Evaluation of the Web customization tool is also addressed in the paper. Evaluation is performed with stakeholders by means of tests and surveys. Results show that testers overcome the detected drawbacks with the use of Excore.

Keywords: Human-Computer Interaction · End-User Development · Customization · Automation · Web search

1 Introduction

Customization is the mechanism that enables users to adapt content, design and functionality to their necessities. Customization might include redefining colors, relocating components, removing features, adding new elements... even adapting web content to people with dyslexia [37]. In order to achieve this goal, the web application is accommodated to the end-user requirements. The increasing volume of website content and actions available on the Web, triggers an

increasing yearning for controlling the Web experience. Often, with a view to carry out actions conducted through the Web, various websites are needed [18]. Generally those websites are completely independent from one another. As a result, end-users are forsaken on their own to obtain the information required by accomplishing their inter-site explorations. The problem is that users feel frustration when repetitive tasks are involved in extracting content from different websites. [2] studied that multiple windows and tabs have significant flaws that hinder users' performance. Here, the approach is to empower end-users to develop extracting content functionality by themselves avoiding this problem.

Lieberman et al. [24] define End-User Development as "a set of methods, techniques and tools that allow users of software systems, who are acting as nonprofessional software developers, at some point to create, modify or extend a software artifact". End-users are capable of commencing with basic adaptation mechanisms and step by step advance to more powerful adaptation mechanisms without facing insuperable barriers [39]. There is an ongoing shift to end-user centered technology, and even users with humble or no skill in Web-based programming languages might perceive the necessity of customizing Web applications according to their preferences [26]. End-user customization is a desired feature in products and in Information Technology there is no difference. Allowing end-users to accommodate content to their requirements enhances usability and might support the removal of accessibility barriers as well [37]. This work looks into Web Customization as a mechanism to adapt Web pages by end-users through a Chrome browser extension for automatic web search.

The remainder of the paper is organized as follows. Section 2 analyses the problem we want to solve, its causes and its consequences. Section 3 discusses related work in order to give the reader an idea of what has been done in this area. Sections 4 and 5 describe Excore (External Content Retriever), a browser extension that customizes web pages for repetitive web search tasks. Section 6 presents the evaluation and its results, Sect. 7 shows features we would like to enhance about Excore and Sect. 8 concludes the paper.

2   Problem Analysis

The Problem

Websites have evolved in the last decade but they still present shortcoming features in the form of repetitive assignments (opening new tabs, scrolling, dozens of clicks to finish with a task...). [20] has accomplished a study on user's feelings when performing web interactions and they have detected a feeling of frustration on users with repetitive tasks, such as gathering information from different resources. Furthermore, the study detected cognitive disorientation in user's daily web activities when looking for information in diverse sources.

Causes

Nowadays users open different tabs to search for information on the Web (multitasking) and they are continuously altering the view from one another (branching). For example, preparing a trip might require three web sessions: finding a

flight, booking a hotel and checking the weather. Multi-tasking is a common phenomenon at present time. [30] found that web users' multi-tasking takes up to 76% of the time of their common web activities (about 20% of the sessions had 5 or more tasks). Only 24% of the users use the web for an unique task. A multi-task session occurs when the Web navigation requires more than one web session to be completed and it has a definable point as to when the task is finalized or it is abandoned [27]. In the web literature, branching is defined as the act of initiating a new tab (or window), allowing people to pleasantly navigate multiple websites concurrently [16].

Consequences
When users are obliged to carry out cross-site activities in their daily routine, even when they have to do it occasionally, they lost the focus with their constant tab switching [9]. As a consequence, their tasks will prolong in time and can be inadequately completed.

Constant tab switching is one way of obtaining information from websites contained in different tabs. Nonetheless, users might switch through tabs to locate a previously opened tab or click on a tab by accident. In these cases, tab switching results in transient page views. Users do not actually aspire to gather information from these sites [43]. In fact, [17] found that users switch tabs at least 57.4% of the time, but user activity, measured in page views, is split among tabs rather than increasing overall activity. [32] claims that when multiple sessions are involved it becomes necessary to understand and represent the information contained in each session, which is essential to differentiate them.

Transferring data across applications is a common end-user task, and copying and pasting via the clipboard allows users to do so relatively easily. Using the clipboard, however, can also introduce inefficiencies and errors in user tasks [40]. Roberts et al. [34] researched on errors due to the copying and pasting data in the context of medical information. Should you add this drawback to tab-switching and loss of focus, end-users will have an unsatisfactory web experience.

[29] shows that interruptions can cause frustration, distract people, cause people to make mistakes, reduce their efficiency and increase the time that is needed for the primary task. And this is not all, [35] remarks that multitasking over different types of tasks can reduce productivity and [38] states that the ability of humans to accomplish simultaneous mental operations is limited by the ability of human brains. That is why users have desperate reasons to conclude a branching multi-task successfully. Our objective is to reduce these drawbacks and as a result, end-users will have a satisfactory web experience.

3   Related Work

An activity is defined as a coordinated set of actions by people towards a common objective, mediated by tools and subject to situational constraints [19]. It is exceedingly common to see how people are doing different activities at the same time or switching activities. In fact, people average about three minutes on

a pursuit before switching activity [15]. This instance is called task fragmentation which has been studied frequently; showing that it is very common in the workplace [4,15,36]. These studies have shown that work fragmentation is noxious to the actual work: after such a context switch, time is necessary for people to regain their bearings [36]. It must be taken into considerations that nowadays most people work with computers and that [4] research has shown that computers are notoriously bad at supporting the management of parallel activities and interruptions. Interruptions are a particularly detrimental kind of activity fragmentation in which an external signal forces a person to switch activity at an unplanned period and for an unknown duration [36]. This means that computer workers are constantly losing track of their activities. Some other works have tried to avoid this drawback, for instance Cowpath [9], WildThumb [25] and AwToolkit [12].

Cowpath [9] focuses on "Web trails", i.e. recurring navigation paths across distinct Web sites. Rather than switching between tabs and typing once and again the same URLs (Uniform Resource Locator), Cowpath augments the affected websites with additional hyper-links that "pave the way" of these Web trails. [9] mentions that if there is no need to go through the welcome page of a site, it saves clicks and facilitates focus and hence avoids task fragmentation. [42] proposes an algorithm that extracts information from a web search and avoids end-users repeating a secondary search. WildThumb [25] suggests a change in 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 by opening the correct one. Nonetheless, it does not avoid multi-tasking or losing track. If the user has dozens of tabs opened, the visual overview does not help to find the correct tab, the user must look for it and in this process, he would lose track of the active task. Another example is AwToolkit [12]. It 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 display, and then notifying users of these changes.

Wikipedia defines mashups as a web page or web application that uses content from more than one source to create a single new service displayed in a single graphical interface[1]. With Web users' search tasks becoming increasingly complex, a single information source cannot necessarily satisfy their information needs [23]. Joining content from different websites circumvents tab switching when locating information from diverse resources. FaceMashup [28] is an End-User Development environment that empowers social network users, supporting them in creating their own procedures for inspecting and controlling their data. MashupEditor [13] is a novel environment for End-User Development of Web mashups. MashupEditor adds 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. MAMS [41] is the first existing Mashup development process for Modeling and Simulation. Following a new Box/Wiring/Mashup method, users

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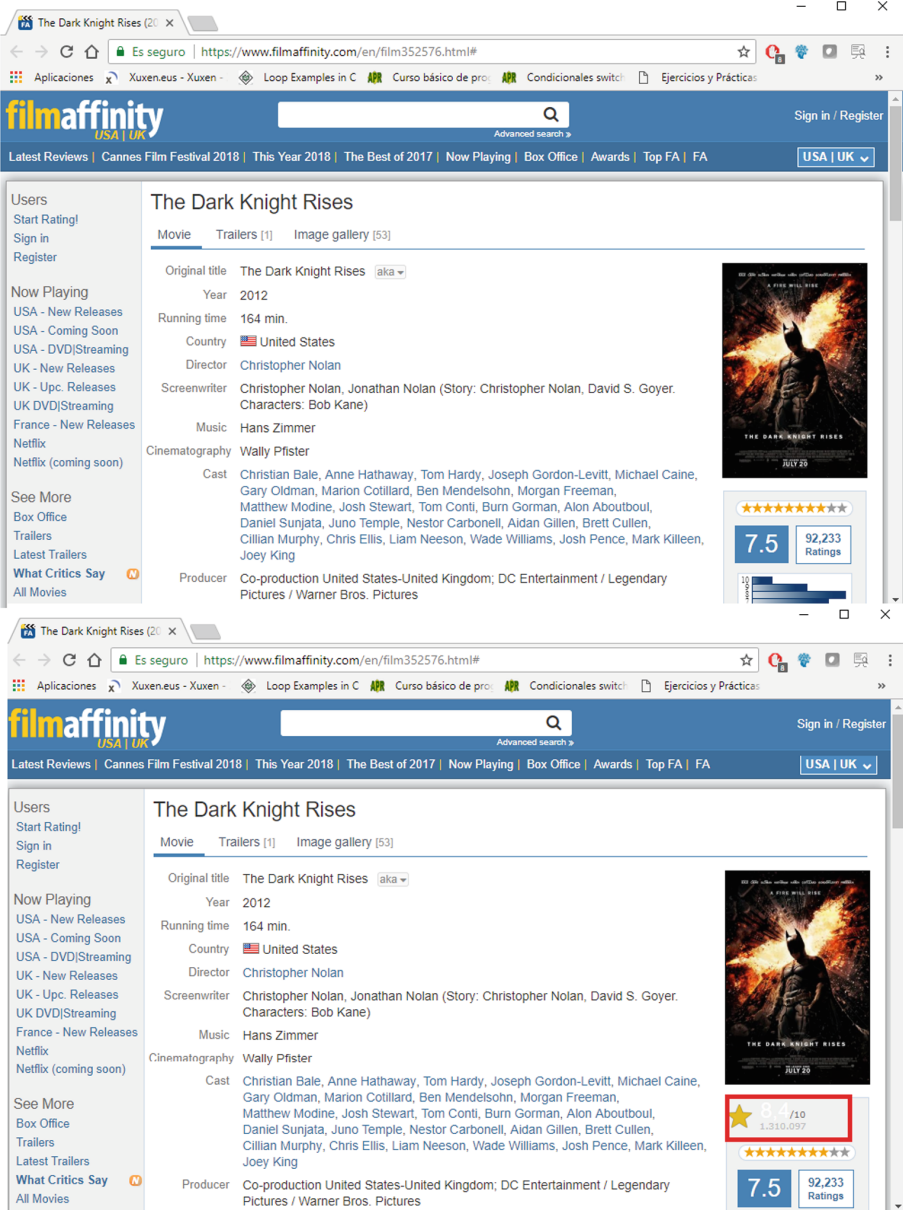
[1] https://en.wikipedia.org/wiki/Mashup.

Fig. 1. Filmaffinity website original website (top) and after customized with IMDB information (bottom)

can develop resources as mashup components, compose them as mashups and run these mashups in web browsers quickly.

Not only are mashups appropriate to join content from different websites in a single one but other methods are possible to be used. WebMakeup [1] is a Chrome browser extension that copies content from diverse web pages and end-users insert or paste this content in a single website. Moreover, it permits them to remove unnecessary web elements and move them if they wish through the website. With CrowdMock [11], users may define their own requirements (adding and removing content) and share them with the community, who can reproduce, edit and evolve them collaboratively. Bosetti [5] proposes a solution to reduce user interaction during web searches thanks to existing web engines. Chudnoskyy et al. [7] take a step forward to create mashups by assisting users with recommendations and automatic composition. [10] shows a tool that proposes a data acquisition system capable of capturing the user interaction in mashups interfaces. Subsequently, these interactions can be reproduced automatically without any human action, which can be a method to limit task fragmentation.

Web automation tools provide a system to increase human productivity by conducting repetitive tasks autonomously. We agree with [3] when they state that "any repetitious behavior should be a candidate for automation because automating things". TellMe [14] 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. SUGILITE [22] is a mobile system that enables the user to create an automation for different tasks across any or multiple smartphone apps and to execute automated tasks through a multi-modal interface.

Despite the existence of applications to automate and customize web interactions, these are insufficient to guarantee the accomplishment of user goals when changes in relevant context cannot be fully anticipated at design time [6]. The goal of this article is to provide end-users with a way to customize their own web searches in their favorite or most visited web pages, therefore reducing frustration in repetitive tasks to solve the problem analyzed in Sect. 2.

4 Automating Web Interactions

The following may be an example of someone's morning routine before going to work: Sit in front of the computer, open the browser, write an URL (e.g. New York Times), read the news headlines, click and open the most interesting ones in new tabs, copy the headlines and search on Google for more info about those news... How often do people do the same action on the net? Which actions do they repeat weekly or monthly? How many repetitive tasks do people do every day? Human beings are undoubtedly animals of habits and hence, we constantly repeat the same activities o a daily basis.

Why do not people develop a script to execute these routines automatically? Most of them are end-users and they do not have the skills to create these routines. What is more, even if they have the knowledge, they might not have the time to develop a script to automatize actions. Excore assists end-users to

create automatic web searches with simple interactions (clicks and copy and paste) in very few minutes. With the intention of illustrating the situation we have developed an example to compare film rating in two film websites: IMDB[2] and Filmaffinity[3]. Let's imagine someone is keen on films and he watches a film every night at home. To know if the film deserves being watched he compares the opinion of people about films in both websites. Why cannot we have both opinions in a single search? Fig. 1 shows the original website of Filmaffinity (top) and the same website including IMDB valuation of the same film (bottom).

4.1   Creation

Excore is a Chrome browser extension for end-users. Chrome is the most used browser around the world[4]. That is why, Excore has been developed as a Chrome extension. Once it is installed, the Excore icon can be seen on the top right corner of the browser. Two options will be shown by clicking on it: "New" and "Delete". New is the option to create an automatic task, the definition of the process that provides enough information to the browser extension to repeat the task itself. At the first stage, the background will change its color to certify where the new content is going to be inserted. After that, the user inserts the URL of the website the system has to extract the new content from, in this example, the IMDB website. Figure 2 illustrates the next step that consist of copying the element that the user wants to look for, such as the title of the film.

   Figure 3 shows the beginning of the next step of the creation process, the website previously inserted by the user. End-users must follow instructions that appear at the top of the website, i.e. paste and clicks. At the end of the process the users have to select the element the system will locate and copy it to be inserted in the initial website.

4.2   Execution

Now end-users are totally passive with regard to Excore because the system checks if Excore must be enacted based on the website address. The system inspects the URL allowing two possible scenarios.

1. **It is the same Web address.** The customization is always executed.
2. **Website in the same domain.** the customization is executed only when the system is able to locate the element to be searched. Otherwise, the system stops the execution until a new load event is triggered with the same domain.

Excore automation process repeats the interactions done by the user during the creation by locating the desired node, making a copy of it and inserting this node code and style into the customizable website. This process avoids completely the tab switching and copy and pasting actions on the part of the user. This action

---

[2] https://www.imdb.com/.
[3] https://www.filmaffinity.com.
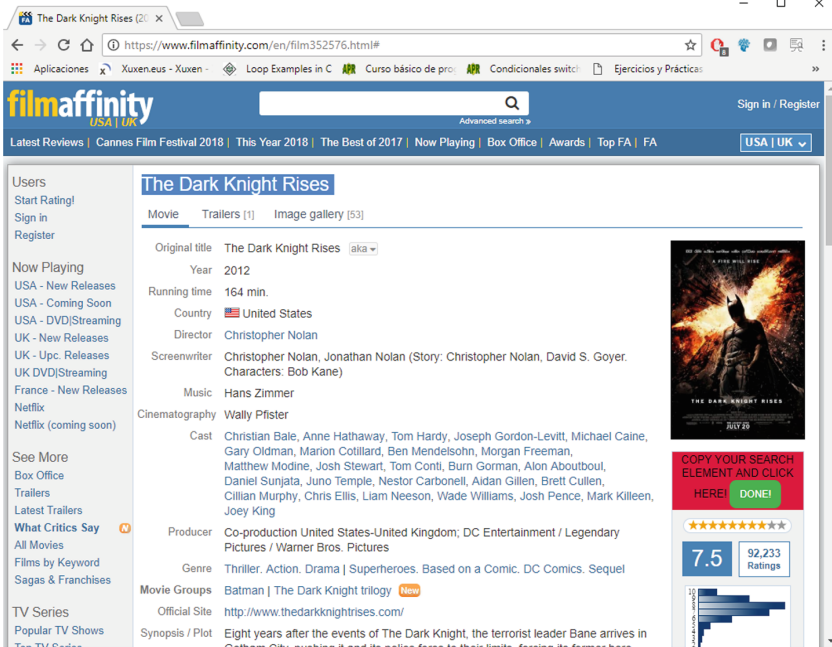[4] http://gs.statcounter.com.

Fig. 2. Copying yearning element in Filmaffinity
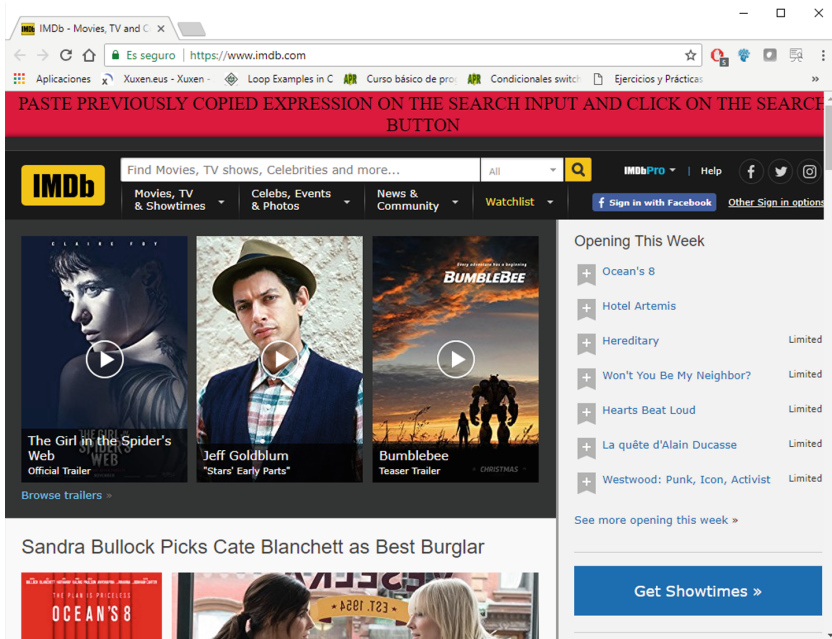

Fig. 3. Pasting process with Excore

saving should increase user focus in their task because he does not have to change the content on his screen. In order to confirm this idea, an evaluation has been accomplished and explained in Sect. 6 in depth.

Once the user is not interested in using the automated task further, the user is able to delete it by clicking on Excore extension icon and every single time he visits the website, the customization will not be enacted.
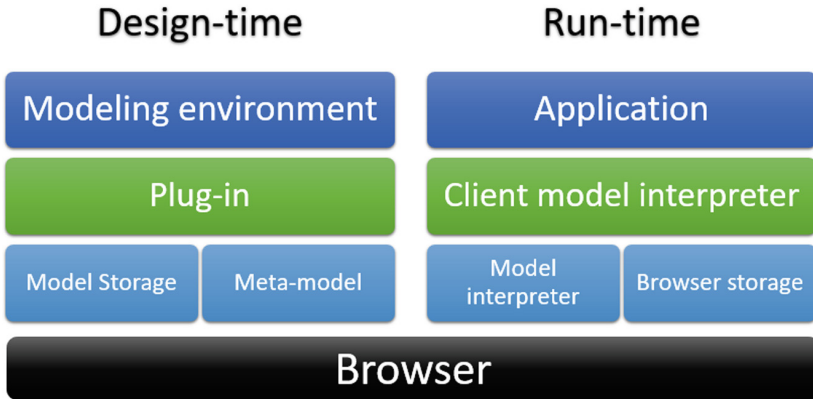
4.3 Architecture

Fig. 4. Excore architecture by layers

In this subsection, we present the architecture of Excore and how it fulfills the goals in the previous subsections. Figure 4 differentiates two sections: design-time or creation and run-time or execution.

When the user is at design-time, Excore web extension provides an environment in order to define a web search step by step. This web search is defined in the meta-model. The Excore extension provides the user with a modeling environment and guides them on the meta-model definition. The user will only read the necessary information for the next step and they will interact with the mouse while following these instructions. Instances are developed visually, no code implementation is needed. At the end, an instance is stored in the browser. Browsers have a space that allow extensions to store some information which is used by Excore to save the instance of the meta-model.

In run-time, the instance of the meta-model stored in the browser is analyzed or interpreted every time a load event is enacted. Once Excore detects that it must come into action, it interprets all events and actions previously defined by the user. These actions are executed in the background, allowing the user watch the original website without changes. Once the application has finished all actions in the background and it detects the final node, the original website will show a copy of the content trough the application, i.e. Excore.

5   Excore Internal Definition

End-User Development tools can be divided into 5 categories: Visual Programming, Spreadsheets, Programming By Demonstration, Domain Specific Languages and Model-Based [1]. 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 thanks to a simple and expressive domain-specific language. Domain-Specific Languages are considered as an approach to decrease complexity of software systems development. Accompanied by Domain Specific Language good practices [31] all requirements have been captured by our feature diagram (see Fig. 5). Next subsections will explain the feature diagram.

5.1   Hosting

Web customization is a set of changes performed by users in order to personalize their most visited or favorite websites. Customization is executed when the web page with a marked host is loaded and the load event is enacted (it has no sense to execute the customization before the event because some nodes might not be loaded and the customization would not take place). The host can be portrayed by a URL expression and when this expression matches the current address of the website, the customization is executed. The host and the current URL can be equal or they can have identical domain to enact web customization. Following the filmAffinity example illustrated by Fig. 1, the user is interested in comparing hundreds of films in the same domain. Otherwise Fig. 1 example would only work for "The dark knight rises" film and the user would have to repeat the automation task for every single film they want to compare. This would be nonsense.
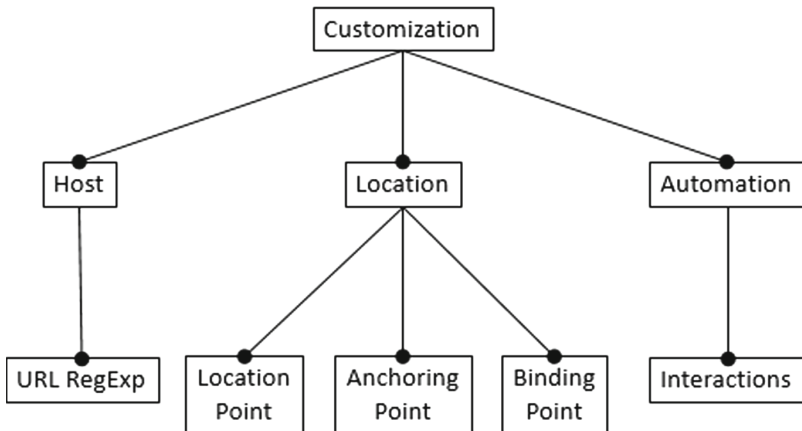

Fig. 5. Feature diagram for Web search customization

5.2   Location

As soon as the customization has started, three different elements must be located to perform the customization successfully. In order to obtain these elements three different locators must be defined. A Web locator can be defined as a mechanism for unique identification of an element in the Document Object Model (DOM) [33]. These locators are:

- **Location Point.** This locator points out from where the desired element is going to be extracted.
- **Anchoring Point.** This information indicates the position where the node extracted with the previous locator is going to be inserted in the target website. The element can be inserted before or after the node has been found with the anchoring locator.
- **Binding Point.** This designates the location of the parameter the system uses to do the automatic web search. In our example, the title of the film.

There are diverse types of locators: first generation, based on coordinates; second generation, based on the structure, (i.e. xpath) and node attributes and third generation, based on image [21]. The robustness of this type of locators is different. First generation locators are not being used nowadays 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 a pixel, first generation locator will probably not find the node. According to [21], locators based on node attributes are more robust than those based on structure and these are more robust in time than image based locator. Additionally, third generation locators are not appropriate to be used in Excore owing to the fact that this type of locator tries to find always exactly the same element (image). Each execution is different. If we had to use an image locator in the example of Fig. 1, it would only have worked for this film and not for the rest of films on the website. Hence, we have implemented a multiple locator algorithm. We use a Xpath to find the element and an attribute locator when possible. It is important to remark that it is impossible to use attribute locators all the time because some nodes do not have ID attributes or because their class or other attributes are not unique.

5.3   Automation

One of the aims of Excore is to do the search automatically. For that reason, the system must record all interactions users do to be able to repeat them automatically in execution time. First, the system saves the web page with which it will interact later. Once this website is ready (the load event has been enacted), Excore, that has saved the interaction type (i.e. click, copy, paste...) and the location of the element in which the system has to recreate that action, completes interactions one by one to the end. When the last interactions are completed, the location point acts by extracting the target element and the anchoring point to be inserted on the defined position of the customized website.

6   Evaluation

The main objective of Excore is to allow end-users to customize their favorite and most visited Web pages by including a mechanism to look for required content in diverse Web pages automatically. Allowing users to adapt content to their needs improves usability [37]. In order to validate if Excore accomplishes its purpose, we have evaluated Excore following the ISO 9241-11[5]. ISO definition of usability (ISO 9241-11, Guidance on Usability) comprises three quality characteristics: effectiveness (whether or not end-users are able to complete successfully the tasks), efficiency (if end-users conclude the assignment with Excore faster than manually) and satisfaction (the opinion of the end-user about Excore).

6.1   Research Method

Setting. The study was conducted in Mondragon University laboratory (Arrasate-Mondragon, Spain). All participants did not use computers with the same features but they had some minimum requirements(i.e., Intel Core i5, 4 GB RAM and Windows 10) and an installation of Chrome.

Procedure. Before participants started with the evaluation, they were informed of the purpose of the study and were given a brief description of it. Afterwards, an Excore instance was presented to exemplify the main functionality of the application. The example consisted of creating and automatic Web search to compare the opinion of people on some films in IMDB and Filmaffinity websites. Next, participants were given a leaf with the instructions to create a new automatic Web search. They had to do the evaluation with another example, a book price comparative. First they had to compare the price of ten different books manually from Powells[6] and Amazon[7]. They must find out which platform sells the book at a better price and after that, they had to compare another set of ten books by using Excore. Participants were asked to note down the time at which they started and finished both activities. In order to accurately measure evaluation time, there was a clock so that participants kept track of time. Finally, the participants were directed to a Google Forms on-line questionnaire.

Subjects. Twenty people participated in the study and 60% of the participants were female. As for their age, all of them were in the 20–39 age range. Participants were from Arrasate-Mondragon and the nearby towns. Most subjects were single and with a degree in Mondragon University and other universities but nobody had a technical degree, the aim of the evaluation was to test Excore with end-users. The day the evaluation was accomplished, all of them were working in different fields, such as financial, construction, teaching, agricultural or sports. 70% of participants have installed at least one plug-in in their browser with an average of 1.5. 70% of the participants spend more than 1 h on the Internet

---

[5] https://www.iso.org/standard/16883.html.

[6] http://www.powells.com/.

[7] https://www.amazon.com/.

Table 1. Satisfaction results from 1 (completely disagree) to 5 (completely agree)

Item	Mean	2-5 tabs	6-10 tabs	10+ tabs	St. Dev.
1–When I access the Web, I often have to change the tab ("tab switching")	3.5	2.75	3.5	4.5	1.24
2–Having to do "tab switching" is a nuisance/frustration	3.25	2.12	3.83	4.17	1.37
3–When I constantly change tabs I lose the focus/goal of what I am doing	2.65	1.75	2.67	3.83	1.35
4–I usually lose the focus/objective of what I am doing when I am on the Internet	2.55	1.87	2.33	3.66	1.28
5–I usually use the "copy and paste" options when I am browsing the Internet	4.05	4	3.83	4.33	0.99
6–Having to "copy and paste" is a nuisance/frustration	2.5	2.12	2.33	3.17	1.19
7–Using Excore helps me not to lose the focus/goal of the task I am doing	3.4	2.75	3.33	4.33	1.10
8–Diminish the possibility of losing focus on my task is useful	3.8	3.12	3.67	4.83	1.19
9–Reducing the need to navigate between different tabs is useful	3.85	3.62	3.83	4.17	1.09
10–Not having to "copy and paste" in web searches is useful	3.65	2.88	3.83	4.5	1.14
11–I think that being able to insert content from other web pages within a page is useful	4.05	4	3.83	4.33	0.94
12–I plan to use Excore in the future	3.2	3	3.17	3.5	0.77
13–I think Excore is interesting enough to recommend it to my friends	3.4	3	3.5	3.83	1.14
14–It has been easy to create a new Excore	3.4	3.12	3.5	3.67	1.05
15–It has been easy to remove an Excore	3.3	2.88	3	4.17	0.92
16–It has been easy to compare Powells and Amazon prices with Excore	4.1	3.38	4.17	5	1.12
17–During the use of Excore, I have known at all times how to do the things I had to do	3.65	3.12	4.17	3.83	1.09
18–In general, I am satisfied with what I have been able to do with Excore	3.6	3.62	3.17	4	0.82

everyday in their job and additionally, 55% of them connected for more than 1 h in their free time.

Instrument. A questionnaire was used to collect users' experience in the evaluation. The questionnaire was composed of four sections; background, effectiveness (true false to mark the completeness of the tasks), efficiency (time in minutes to complete tasks) and satisfaction. Satisfaction was measured using different questions with a 5-point Likert scale (1 = completely disagree, 5 = completely agree).

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

6.2 Results

The results[8] show the conclusion we can draw from the answers to the questionnaire. Some subjects have added some comments about Excore to the questionnaire:

- *Despite the fact that the state of development of Excore is not optimized, I think it is a very useful tool.*
- *It is a useful extension, the next thing would be to implement the redirection towards the website in which is cheaper to buy the book.*
- *I find Excore useful and easy to use.*

These comments convey the idea that the subjects in the evaluation have had a positive opinion about Excore and they may use it in the future.

In this section, we are going to analyze usability results in ISO 9241-11 considering its three characteristics:

Effectiveness. Both tasks were successfully completed by all subjects. The first task consisted of installing the extension on their browser. The second one consisted of creating an automatic web search and compare the prices of ten books in Powells and Amazon first manually and later automatically. No problems occurred during either process.

Efficiency. Efficiency is measured regarding the time required to conclude the manual search and the automatic search. The time needed in the manual search by all subjects on average has been 9 min and 6 s whereas the time needed in the automatic search has been 6 min and 54 s. This means that automatic web search has been an 24,2% swifter than the manual one. Due to this result, we might infer that the more automatic web search in parallel we have, the more efficient will be. All subjects needed less time to compare book prices with Excore than manually. The time difference needed to complete both task varies from one subject to another. This may be because during the evaluation, some of them were doing the manual search in parallel and some others did not.

We have compared the manual and the automatic searches to know if there are statistically significant differences between both. We have calculated that there is a statistical significant difference of 99,9% in the time needed to carry out each activity which decidedly supports the efficiency of Excore compared to manual web searches.

Satisfaction. Davis [8] said that satisfaction is determined by three dimensions: perceived usefulness, perceived ease of use and willingness to use it in the future. Perceived usefulness by subjects is answered through the first questions in Table 1 (items 1–11). In the middle of the table, willingness to use in the future questions

---

[8] https://tinyurl.com/y9bhva6v.
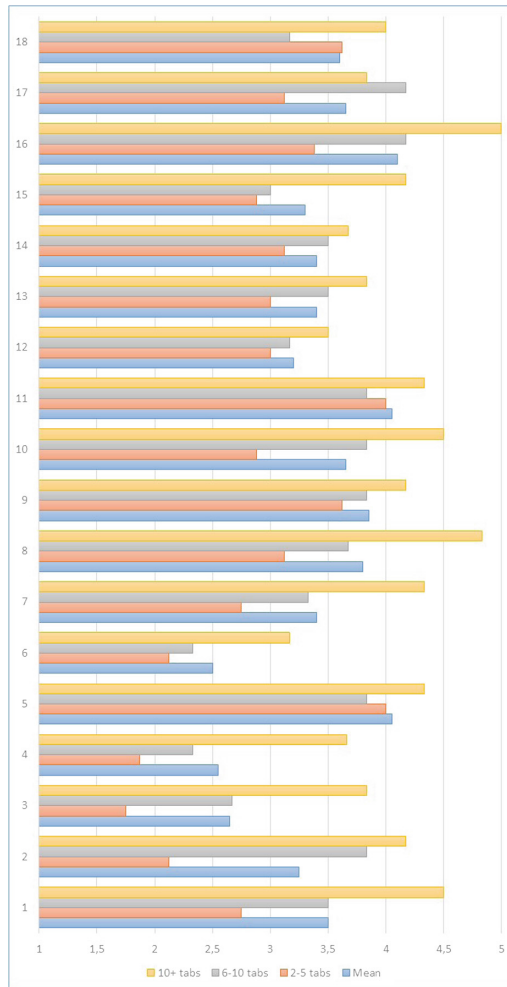
Fig. 6. Excore satisfaction results (Color figure online)

show subjects intention to do so (items 12–13). At the end of Table 1 it is shown what subjects answered whether Excore was easy to use for them (items 14–17). The last question reflects their general opinion about the tool and its utility.

Table 1 is divided in six columns. "Items" are the questions subjects answered after doing the evaluation, "Mean" is the average value from 1 (completely disagree) to 5 (completely agree) subjects gave to each item and "St. Dev." is the standard deviation in each question. The other three columns show the average value subjects gave to each question based on the average number of tabs they usually have opened in their browsers. At the beginning of the questionnaire subjects were asked about how many tabs they usually have opened during their web assignments. Eight subjects said that they have between 2 and 5 tabs opened

all the time, six subjects have between 6 and 10 tabs and 6 subjects more than 10 tabs. There is a relation between the amount of tabs the users open and the rating they give to Excore.

Items from 1 to 6 (yellow) are questions referring to the consequences cited in Sect. 2 (tab switching, lost of focus and copy & paste) about their feeling while browsing. Item 1 remarks that the more tabs a person has, the more tab switches he performs (Mean 2–5tabs = 2.75, Mean 10+tabs = 4.5) and the more frustration he feels (Mean 2–5tabs = 2.12, Mean 10+tabs = 4.17)(item 3). These results are logical because users with 3 tabs opened will do less tab switching than users with 11 tabs and people with a big number of tabs opened make more mistakes and as a consequence they feel frustration. Item 3 evaluates whether tab switching is the main cause of focus lost for users with more than 10 tabs (Mean 10+tabs = 3.83). Users also lost their focus when they were on the Internet and they had a large number of tabs opened (Mean 2–5tabs = 1.87, Mean 6–10tabs = 2.33, Mean 10+tabs = 3.66)(item 4). Copy & paste is a frequent action users do (Mean = 4.05) independently of the number of tabs(item 5). However, user fell frustration when they have a large number of tabs (Mean 2–5tabs = 2.12, Mean 10+tabs = 3.17) (item 6).

We have calculated statistically significant differences between groups of users. The users in those groups have been divided according to the amount of tabs they usually have opened. Users with 2–5 tabs and users with 6–10 tabs opened have statistically significant differences in items 2 and 3. Differences in item 2 validates our premise because the more tabs a user has, the more frustration they will feel at tab switching. It also makes sense that users with few tabs do not lose their focus as much as users with several tabs opened. There are statistically significant differences in items 1, 3 and 4 between users with 6–10 tabs opened per session and users with 10+tabs opened tabs per session. For item 1, it seems that users have to do more tab switching when they have a large number of tabs in their browser session. Items 3 and 4 are related to focus during their activities. This result validates our hypothesis due to the fact that the more tabs opened a user has, the easier it is to lose focus on the current activity. Finally, we analyzed the differences between 1–5tabs and 10+tabs users and the results show significant differences in all items except for item 5. Item 5 is about copy and paste activities and all type of users copy and paste during their web activities. The significant difference in items 1 to 4 is of 99%, items related to tab switching and the lose of focus. The significant difference of item 6 is 95%, showing that the bigger the number of opened tabs are, the bigger the nuisance of copying and pasting it is. The results on items 1 to 4 strongly validates the analysis previously done in the problem analysis section.

Items from 7 to 11 (blue) are questions about their experience with Excore during evaluation. Regarding item 7, users consider that the higher number of tabs the more Excore helps to maintain their focus on their activities (Mean 2–5tabs = 2.75, Mean 6–10tabs = 3.33, Mean 10+tabs = 4.33). Furthermore, users claim that it is useful to minimise the chance of losing focus (Mean 2–5tabs = 3.12, Mean 10+tabs = 4.83) (item 8), reducing tab switching (Mean

2–5tabs = 3.62, Mean 10+tabs = 4.17) (item 9) and copying and paste (Mean 2–5tabs = 2.88, Mean 10+tabs = 4.5) (item 10). It is remarkable that in this occasion subjects with more tabs consider Excore more useful regarding the three drawbacks: copy and paste, tab switching and lost of focus. Finally, subjects claim that inserting content from different websites in a single one is useful (Mean = 4.05) (item 11).

Regarding the statistically significant differences, the greater differences are in 10+tabs users. Items 7, 8 and 10 are affected by those differences. Items 7 and 8 are about the focus and users claim that Excore helps them to maintain the focus depending on the number of tabs opened. These results validate the fact that Excore helps to mitigate the consequences explained in the problem analysis section.

As for willingness to use in the future (orange), subjects are interested in using Excore in the future and recommending it to their acquaintances. Similar to usefulness, in this case people with more tabs opened are more interested in using and in recommending Excore.

Concerning subject's perception on Excore's easiness to use, they found easy to create a new automatic search and to remove it. Thanks to Excore, they thought it was really easy to compare prices from Powells and Amazon (Mean = 4.1) in particular people that have more than 10 tabs, because all of them value this question with 5. Lastly, subject did not have problems during the evaluation because they knew what they had to do (Mean = 3.65) and it is relevant to remark that subjects were satisfied when they finished their first experience with Excore (Mean = 3.6).

Figure 6 helps to illustrate these results visually. Numbers on the left indicate the items or question numbers, and bottom values are results users gave to each one. Yellow columns (results of people that have on average more than 10 tabs opened) are always higher than the other results. With the exception of item 5 and 18, the more tabs a user usually has opened, the more frustrated they feel when they are surfing the net. Excore contributed to mitigate this frustration. Figure 6 shows how people with an average of 6–10 tabs opened (gray column) are close to the mean (blue column) in all items. This might indicate that in general most people have between 6 to 10 tabs opened every session and that their opinion is reliable in web matters.

Figure 6 also gives visual evidence of subjects opinions on most significant consequences on multi-tasking, their opinion about web actions and if in their opinion Excore mitigates those problems or not.

7   Future Work

On the future, the first goal is to be capable of doing more than one web search in parallel. We have calculated that using the current version of Excore, the time reduction in repetitive tasks will be more than the 24.2%. Additionally, our purpose is to be able to include iterative web searches based on one result, i.e. when Excore inserts the new content, conduct a new automatic web search based

on the new content in another website. It is important to remark that some of this content could be dynamic and as a consequence, every time the dynamic content is updated, Excore should refresh all content based on the dynamic element.

In order to reduce user's frustration we must do more research in common user interactions like clicks, URL typing, scrolling, etcetera which can increase this feeling. Website customization is the precise technique to mitigate these problems.

Finally, we would like to enhance locator robustness in order to increase automatic search life expectancy. As mentioned in Sect. 5, we have implemented two different locator mechanisms owing to the fact that when one fails, the other could find the element. Excore requires several locators to successfully complete the web search and if one fails, the automation will come to nothing. This is why it is highly relevant to implement new locator mechanisms because if one fails another one might succeed. Furthermore, we are interested in improving our actual Xpath algorithm to adapt to web upgrades.

8   Conclusions

This article has presented Excore, a Chrome browser extension that enables end-users to customize their favorite websites with automatic web searches by using simple interactions such as clicks, copy and paste. The aim of this tool is to reduce the frustration users have when they carry out repetitive tasks by decreasing the consequences of this problem (tab switching, copy and paste actions and lost of focus in their daily activities). Evaluation results support the problem analysis. They reflect that people that on average have more opened tabs feel that tab switching, copy and paste are a problem and that they lose their focus more often than people who have few tabs opened. Results also show that Excore helps people who have a large number of tabs opened all the time to reduce these three shortcomings (copy and paste, tab switching and lost of focus). They also remark that it is easy to use. Moreover, Excore reduces the time needed to do frequent web searches around 25%.

References

1. Aldalur, I., Winckler, M., Díaz, O., Palanque, P.: Web augmentation as a promising technology for end user development. In: Paternò, F., Wulf, V. (eds.) New Perspectives in End-User Development, pp. 433–459. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60291-2_17
2. AlSada, M., Nakajima, T.: Parallel web browsing in tangible augmented reality environments. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, Seoul, CHI 2015 Extended Abstracts, Republic of Korea, 18–23 April 2015, pp. 953–958 (2015)

3. Amershi, S., Mahmud, J., Nichols, J., Lau, T., Ruiz, G.A.: LiveAction: automating web task model generation. TiiS **3**(3), 14:1–14:23 (2013)

4. Bardram, J.E., Bunde-Pedersen, J., Søgaard, M.: Support for activity-based computing in a personal computing operating system. In: Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, 22–27 April 2006, pp. 211–220 (2006)

5. Bosetti, G., Firmenich, S., Fernandez, A., Winckler, M., Rossi, G.: From search engines to augmented search services: an end-user development approach. In: Cabot, J., De Virgilio, R., Torlone, R. (eds.) ICWE 2017. LNCS, vol. 10360, pp. 115–133. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60131-1_7

6. Castaneda, L., Villegas, N.M., Müller, H.A.: Self-adaptive applications: on the development of personalized web-tasking systems. In: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-managing Systems, SEAMS 2014, Hyderabad, 2–3 June 2014, pp. 49–54 (2014)

7. Chudnovskyy, O., et al.: End-user-oriented telco mashups: the OMELETTE approach. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, 16–20 April 2012 (Companion Volume), pp. 235–238 (2012)

8. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. **13**(3), 319–340 (1989)

9. Díaz, O., Sosa, J.D., Trujillo, S.: Activity fragmentation in the web: empowering users to support their own webflows. In: 24th ACM Conference on Hypertext and Social Media (Part of ECRC), HT 2013, Paris, 02–04 May 2013, pp. 69–78 (2013)

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

11. Firmenich, D., Firmenich, S., Rivero, J.M., Antonelli, L., Rossi, G.: Crowdmock: an approach for defining and evolving web augmentation requirements. Requir. Eng. **23**(1), 33–61 (2018)

12. Garrido, J.E., Penichet, V.M.R., Lozano, M.D., Quigley, A.J., Kristensson, P.O.: Awtoolkit: attention-aware user interface widgets. In: International Working Conference on Advanced Visual Interfaces, AVI 2014, Como, 27–29 May 2014, pp. 9–16 (2014)

13. Ghiani, G., Paternò, F., Spano, L.D., Pintori, G.: An environment for end-user development of web mashups. Int. J. Hum.-Comput. Stud. **87**, 38–64 (2016)

14. Gil, Y., Ratnakar, V., Fritz, C.: TellMe: learning procedures from tutorial instruction. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI 2011, Palo Alto, 13–16 February 2011, pp. 227–236 (2011)

15. González, V.M., Mark, G.: "Constant, constant, multi-tasking craziness": managing multiple working spheres. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, 24–29 April 2004, pp. 113–120 (2004)

16. Huang, J., Lin, T., White, R.W.: No search result left behind: branching behavior with browser tabs. In: Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, 8–12 February 2012, pp. 203–212 (2012)

17. Huang, J., White, R.W.: Parallel browsing behavior on the web. In: Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, HT 2010, Toronto, 13–16 June 2010, pp. 13–18 (2010)

18. Karsai, G., Lang, A., Neema, S.: Design patterns for open tool integration. Softw. Syst. Model. **4**(2), 157–170 (2005)

19. Kuutti, K.: The concept of activity as a basic unit of analysis for CSCW research. In: Bannon, L., Robinson, M., Schmidt, K. (eds.) Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW 1991. Springer, Dordrecht (1991). https://doi.org/10.1007/978-94-011-3506-1_19

20. Lee, T.Y., Bederson, B.B.: Give the people what they want: studying end-user needs for enhancing the web. PeerJ Comput. Sci. **2**, e91 (2016). https://doi.org/10.7717/peerj-cs.91, https://doi.org/10.7717/peerj-cs.91

21. Leotta, M., Clerissi, D., Ricca, F., Tonella, P.: Capture-replay vs. programmable web testing: an empirical assessment during test case evolution. In: 20th Working Conference on Reverse Engineering, WCRE 2013, Koblenz, 14–17 October 2013, pp. 272–281 (2013)

22. Li, T.J., Azaria, A., Myers, B.A.: SUGILITE: creating multimodal smartphone automation by demonstration. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, 06–11 May 2017, pp. 6038–6049 (2017)

23. Li, X., Liu, Y., Cai, R., Ma, S.: Investigation of user search behavior while facing heterogeneous search services. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, 6–10 February 2017, pp. 161–170 (2017)

24. Lieberman, H., Paternò, F., Wulf, V. (eds.): End User Development. Human-Computer Interaction Series. Springer, Dordrecht (2006). https://doi.org/10.1007/1-4020-5386-X

25. Liu, S., Tajima, K.: Wildthumb: a web browser supporting efficient task management on wide displays. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI 2010, Hong Kong, 7–10 February 2010, pp. 159–168 (2010)

26. Macías, J.A., Paternò, F.: Customization of web applications through an intelligent environment exploiting logical interface descriptions. Interact. Comput. **20**(1), 29–47 (2008)

27. MacKay, B., Watters, C.R.: Exploring multi-session web tasks. In: Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, Florence, 5–10 April 2008, pp. 1187–1196 (2008)

28. Massa, D., Spano, L.D.: FaceMashup: an end-user development tool for social network data. Future Internet **8**(2), 10 (2016)

29. McFarlane, D.C., Latorella, K.A.: The scope and importance of human interruption in human-computer interaction design. Hum.-Comput. Interac. **17**(1), 1–61 (2002)

30. Mehrotra, R., Bhattacharya, P., Yilmaz, E.: Characterizing users' multi-tasking behavior in web search. In: Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval, CHIIR 2016, Carrboro, 13–17 March 2016, pp. 297–300 (2016)

31. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. **37**(4), 316–344 (2005)

32. Raghavan, S., Parampalli, U., Raghavan, S.V.: Re-engineering simultaneous internet sessions process-separated browsers. In: Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2017, Geelong, 31 January–3 February 2017, pp. 70:1–70:10 (2017)

33. Ricca, F., Leotta, M., Stocco, A., Clerissi, D., Tonella, P.: Web testware evolution. In: 15th IEEE International Symposium on Web Systems Evolution, WSE 2013, Eindhoven, 27 September 2013, pp. 39–44 (2013)

34. Roberts, K., Cahan, A., Demner-Fushman, D.: Error propagation in EHRs via copy/paste: an analysis of relative dates. In: American Medical Informatics Association Annual Symposium, AMIA 2014, Washington, DC, 15–19 November 2014

35. Rubinstein, J.S., Meyer, D.E., Evans, J.E.: Executive control of cognitive processes in task switching. Exp. Psychol. Hum. Percept. Perform. **27**(1), 763–797 (2001)

36. Sanchez, H., Robbes, R., González, V.M.: An empirical study of work fragmentation in software evolution tasks. In: 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering, SANER 2015, Montreal, 2–6 March 2015, pp. 251–260 (2015)

37. de Santana, V.F., de Oliveira, R., Almeida, L.D.A., Ito, M.: Firefixia: an accessibility web browser customization toolbar for people with dyslexia. In: International Cross-Disciplinary Conference on Web Accessibility, W4A 2013, Rio de Janeiro, 13–15 May 2013, pp. 16:1–16:4 (2013)

38. Schweickert, R., Boggs, G.J.: Models of central capacity and concurrency. Math. Psychol. **28**(3), 223–281 (1984)

39. Spahn, M., Dörner, C., Wulf, V.: End user development: approaches towards a flexible software design. In: 16th European Conference on Information Systems, ECIS 2008, pp. 303–314 (2008)

40. Stolee, K.T., Elbaum, S.G., Rothermel, G.: Revealing the copy and paste habits of end users. In: Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2009, Corvallis, 20–24 September 2009, pp. 59–66 (2009)

41. Wainer, G.A., Wang, S.: MAMS: mashup architecture with modeling and simulation as a service. J. Comput. Sci. **21**, 113–131 (2017)

42. Winckler, M., Cava, R., Barboni, E., Palanque, P., Freitas, C.: Usability aspects of the inside-in approach for ancillary search tasks on the web. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) INTERACT 2015, Part II. LNCS, vol. 9297, pp. 211–230. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22668-2_18

43. Zhang, H., Zhao, S.: Measuring web page revisitation in tabbed browsing. In: Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, 7–12 May 2011, pp. 1831–1834 (2011)