Master's Thesis

August 10, 2021

Taboor: An interactive approach to semi-automatically support task-based workflows in web browsers

Dimitri Kohler

of Austria, born in Locarno, Switzerland (12-612-826)

supervised by

Prof. Dr. Thomas Fritz

University of Zurich[UZH]

s.e.a.l.

software evolution & architecture lab

Master's Thesis

Taboor: An interactive approach to semi-automatically support task-based workflows in web browsers

Dimitri Kohler

University of Zurich[UZH]

s.e.a.l.
software evolution & architecture lab

Master's Thesis

Author: Dimitri Kohler, dimitri.kohler@outlook.com

Project period: 10.02.2021 - 10.08.2021

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

Acknowledgements

I would like to thank my supervisor Prof. Dr. Thomas Fritz for all the insight and guidance given to me before and during the work. A big thank you goes to Roy Rutishauser and Prof. Dr. Reid Holmes for their inputs in the design of Taboor and the study. For the study I also want to thank all participants, without whom this work would not have been possible. Finally, I want to thank my two proofreaders for going through my work again and again, my girlfriend for listening to my every thought while being stuck at home during the pandemic and , last but no least, a thank you to my parents for all the support through out the years.

Abstract

With the world becoming more and more digital, managing multiple tasks and their artifacts at once has become a central part of knowledge work and requires cognitive energy that could be invested in the task it-self. Most Web browsers and many web extension try to provide features that can help to manage browser artifacts by allowing to save, group or mark them. However, most of these features do not manage to fully support task-based work and were no designed to help with clutter on screen.

We propose an approach that allows users to temporarily mark and group artifacts with colors, gives them a quick way to interact with the groups of artifacts and tries to help them re-find artifacts they have previously visited and might be of interest. To evaluate the usefulness of this bundle of features, we conducted a study with six participants over two weeks followed by an interview. We found that Taboor was liked by our participants and encourages task-based work and thinking, however participants with already established and clear strategies struggled to integrate Taboor in their existing workflows and found Taboor to be less useful. From our participants' feedback and usage logs collected by Taboor we formulated five design implications to further help improve future artifact management tools.

Zusammenfassung

Da die Welt immer digitaler wird, ist das gleichzeitige Bearbeiten von mehreren Aufgaben und ihrer Artefakte zu einem zentralen Bestandteil von Knowledge Work geworden und erfordert kognitive Energie, die in die eigentliche Aufgabe investiert werden könnte. Die meisten Web-Browser und viele Web-Erweiterungen bieten Funktionen, die bei der Verwaltung von Browser-Artefakten helfen können, indem sie es ermöglichen, diese zu speichern, zu gruppieren oder zu markieren. Die meisten dieser Funktionen sind jedoch nicht in der Lage, die aufgabenbasierte Arbeit vollständig zu unterstützen und wurden nicht dafür entwickelt, die Unordnung auf dem Bildschirm zu beseitigen.

In dieser Arbeit verfolgen einen neuen Ansatz, der es den Nutzern erlaubt, Artefakte vorübergehend farblich zu markieren und zu gruppieren, der ihnen eine schnelle Möglichkeit bietet, mit den Artefaktgruppen zu interagieren, und der ihnen hilft, Artefakte, die sie bereits besucht haben und die von Interesse sein könnten, wiederzufinden. Um die Nützlichkeit dieses Bündels von Funktionen zu bewerten, haben wir eine Studie mit sechs Teilnehmern über zwei Wochen durchgeführt, gefolgt von einem Interview. Wir fanden heraus, dass Taboor bei unseren Teilnehmern gut ankam und aufgabenbasiertes Arbeiten und Denken fördert. Teilnehmer mit bereits etablierten und klaren Strategien hatten jedoch Schwierigkeiten, Taboor in ihre bestehenden Arbeitsabläufe zu integrieren und empfanden Taboor als weniger nützlich. Aus dem Feedback unserer Teilnehmer und den von Taboor gesammelten Nutzungsprotokollen haben wir fünf Design-Implikationen formuliert, um zukünftige Artefaktmanagement-Tools weiter zu verbessern.

Contents

List of Figures

List of Tables

List of Acronyms

URL Unified Resource Locator

RQ Research Question(s)

UI User Interface

IDE Integrated Development Environment

Chapter 1

Introduction

Knowledge workers encounter a multitude of different tasks throughout the day and regularly work on multiple tasks in parallel on their computer (Zhang et al. 2015). This, combined with regular interruptions prompting workers to often switch between tasks (González and Mark 2004), can lead to a lot of clutter on the screen and the user losing track of their open web resources. As a consequence a lot of mental and cognitive effort is needed just to keep track of the different web resources and their associated tasks open on screen (Pilzer et al. 2020). Especially while multitasking, resuming a task after an interruption can be very costly, since a lot of effort is required to rediscover all the relevant web resources.

Most browsers come with features trying to ease the cognitive load for knowledge workers, like bookmarks or the browser history. However, most of these features do not manage to fully support task-based work and were not designed to help with the clutter on screen. There is research trying to design tools to solve these problems. However, most approaches are not applicable to web browsers specifically. Therefore, we propose Taboor, a novel, interactive approach that focuses on supporting knowledge workers in managing medium-term tasks in their web-browser. The design builds upon the two concepts of fuzziness of artifact groups and medium-term artifact management with recency, frequency & duration as the main indicators for task association of browser artifacts. These concepts can be found in the three core features of Taboor, marking pages with colors to group them, a quick access bar to interact with the created groups and related highlights to (re-)find related web pages and quickly create groups.

To verify our approach we have formulated 3 Research Question(s) (RQ). RQ1 is meant to explore the different strategies employed by knowledge workers to manage their browser artifacts. This is important for future development of similar tools and to see where our approach might be incompatible with existing strategies. With RQ2 we seek to find out if Taboor has an effect on the attention residue (i.e. clutter) in the browser based on the data gathered by Taboor, such as the number of open tabs or the number of page switches made through our a day. Finally, with RQ3 we want to determine the ease of integration of Taboor and how the approach was perceived based on user feedback.

RQ1: How do knowledge workers manage browser artifacts?

a How do knowledge workers organize their browser tabs and web pages?

b How do knowledge workers re-find information they had closed before?

RQ2: Can Taboor help to reduce the attention residue from the amount of information available in the browser?

a Can Taboor actively support knowledge workers to reduce the number of open tabs?

b Does Taboor have a significant impact on the visit duration of web pages?

c Does Taboor have a significant impact on the number of switches between pages?

RQ3: How do knowledge workers integrate the extension in their workflow?

a How do knowledge workers utilise medium-term marked pages?

b How do knowledge workers utilise the quick access for groups?

c How do knowledge workers utilise related highlight pages?

Taboor was implemented as a prototype browser extension compatible with all Chromium-based browsers as well as Firefox. The design and implementation are described in more detail in Chapter 3. To evaluate the prototype, we conducted a two-week user study with six knowledge workers, where they were able to use the prototype throughout their work days. During the study, Taboor gathered data on the browsing behaviour we used to evaluate the effects of the design. After the study, we conducted a follow-up interview and used the answers in combination with the gathered data to answer our research questions. The study design is discussed in Chapter 4, followed by the results for the research questions in Chapter 5.

From the results of the study we derive five design implications we believe to be important when designing tools or features to support task-based work. With the design implications we also make specific suggestion on how to improve Taboor in the future. These design implications are introduced and discussed in Chapter 6.

Furthermore, we discuss threats to the validity of our results in Chapter 7, followed by the final conclusions and possible future work in Chapter 8.

Chapter 2

Related Work

Computer-based knowledge work is characterized by a high degree of multitasking (Zhang et al. 2015). In the digital age people are more connected than ever and this can lead to interruptions and task switches throughout the day which in turn lead to more multitasking (González and Mark 2004). These factors are one of the reasons why knowledge work on the computer has a higher cognitive cost than traditional analog knowledge work (Pilzer et al. 2020). In the following sections, we take a look at why computer-based multitasking takes so much effort (Section 2.1) and how this challenge has been approached by researchers, open-source projects and commercial tools (Section 2.2). Finally, in Section 2.3 we discuss how the grouping of artifacts can be implemented by studying some of the approaches from Section 2.2.

2.1 Multitasking in Computer-Based Knowledge Work

Computer-Based Knowledge Work has been researched extensively. It enables a degree of multitasking and task switches not possible with paper-based knowledge work (Zhang et al. 2015). Naturally, operating systems as well as programs running on these operating systems have evolved to support more and more multitasking over the years. A lot of information can be displayed on a screen at once and this can become very taxing for the user to maintain. The clutter on-screen increases the cognitive load on the user (Pilzer et al. 2020). Thus, it often becomes a trade-off between putting effort into managing all the artifacts on screen or to just deal with the clutter by spending time and energy searching for a specific artifact. Digital well-being research tries to find solutions to these time and energy consuming tasks.

There have been multiple studies focusing on how users interact with digital artifacts in their normal computer usage. Zhang et al. (2015) looked at datasets from 3000 subjects containing 15 Million logged interactions. They found that task-switching often occurs in a “star structure” as they call it. The datasets also showed Power-law distributions, which indicate that people tend to come back to certain hubs, like the internet browser or text editing, while working on their tasks.

Huang and White (2010) have done a similar study where they analyzed internet browsing logs collected via a browser extension. Their observations match with the findings of Zhang et al. (2015).

They “found that the frequency of out-clicks and tab switches follow the power law, while the number of page views per tab follow a log-normal distribution” (Huang and White 2010). This means that a few tabs are used often and act as a hub, while other tabs only exist for a single or very few visits and get closed at the end. While many tabs are never revisited, they often get used to look at multiple websites, indicated by the log-normal distribution of the page views per tab. From their results, Huang and White (2010) deduced implications for the design of browsers and websites. For example, they suggest that websites should stop branching users to new tabs and that the functionality to open a new tab by clicking a link, might not be in a users interest.

2.2 Digital Well-Being Tools

2.2.1 Research Tools

Since multitasking needs a lot of cognitive energy and is such a central part of working in today’s agile work spaces, there has been a lot of research around digital well-being. Here we look specifically at tools and solutions that try to help users cope with the cognitive load of multitasking. Generally these tools try to help users with what seem like the two main sources of the cognitive load during multitasking. Firstly, supporting a user to navigate the big amount of resources available to solve a task (Sahm and Maalej 2010; Pilzer et al. 2020) or secondly enabling a user to get back into a task after a task-switch or an interruption (Leiva 2011).

Sahm and Maalej (2010) have created a "context aware artifact recommendation and switching tool". This tool tries to find the next useful artifact based on the context of the current task, the history of how a user interacted with different artifacts before and a community profile describing what other users have done before in similar contexts. The results of their simulations and short usage sessions show the potential of such a tool. Users had a lower cognitive load when searching for resources on screen and they were faster in finding needed information.

Pilzer et al. (2020) had similar results with their approach to supporting software developers. By tracking users eyes, they were able to create a relevance model for the open windows on a computer. Based on this model, less relevant windows were dimmed which made the relevant windows easier to find. According to their study, this reduced clutter on the desktop and the participants needed less switching between windows to find relevant artifacts and reported to be more focused.

MouseHints (Leiva 2011) tries to ease the cognitive load when resuming tasks. This approach is based on a tool that records mouse movement and interaction with elements on web pages. When a user returns to a previously visited web page, this information is shown to them. The user visually sees where the mouse moved and what was clicked. Leiva (2011) found that such visualization allows users to resume their tasks three times faster and helps them complete tasks in half the time.

2.2.2 Commercial Tools

There are also many commercially available tools trying to solve similar problems.

Tabby is a browser extension that tries to help avoid clutter by automatically closing unused tabs (Tabby 2020). Closed tabs are saved and can be reopened at any time so that no information is ever really lost. To determine if a tab is stale and can be closed, time spent on a tab, time since the last visit of a tab and the frequency of visits are considered. Users can also choose the maximum number of tabs that should be allowed to be open at the same time. According to their FAQ their goal is to seamlessly fit into a users workflow without them having to adapt their behaviour and to "help users focus on the most valuable tasks longer, and be more productive" (Tabby 2020).

There are also tools like Inmotion (Motion 2020) or Workona (Workona 2020) that have similar goals as Tabby but come in a package with even more functionality. Inmotion in particular helps to schedule meetings, organize documents and manage browser windows and tabs. As a digital wellbeing application it claims to increase productivity and help save up to two hours of work per day. But for this work the tab management functionality of both tools is especially interesting. Inmotion as well as Workona allow to create fixed work spaces for tabs and to open and close work spaces or single tabs. In Workona these workspaces can even be shared across teams. These functionalities are intended to help users and teams to manage their browser artifacts and thus save them time and energy when working. The workspaces have to be maintained manually, even though Workona provides basic templates that can be modified to fit a user's or team's needs.

A slightly different idea was at the start of Sessionbuddy (2017). This browser extension was initially meant to recover browser sessions in case of a computer crash. With version four it has grown to become a tab and bookmarks management tool. It allows to create collections of web pages and tabs, to merge groups and open them while still providing the session recovery it started off with. An interesting functionality for grouping is the ability to add tags to groups, which enables the user to tag similar groups with the same tags and to find them quickly via the search function. SessionBuddy uses a mix of manual grouping and automated grouping. It regularly captures the session, for example when a window or the browser is closed, and saves all the open tabs as a group. The user can then modify, tag or delete the groups.

Without installing any extension, all commonly used browsers (e.g. Google Chrome, Microsoft Edge and Firefox) provide functionality to save and manage pages as bookmarks. However there are differences in functionality for the short-term grouping and management of tabs and pages from browser to browser.

Most widely used Chromium-based browsers (e.g. Google Chrome and Microsoft Edge) provide a way to manually group tabs with the goal of providing the user with a tool to get a better overview over their open tabs. Chrome (Google 2020) as well as Edge (Microsoft 2020b) (in beta) allow for coloring and naming of tabs while they are open, by default groups are not collapsed but all tabs stay visible. If a tab is closed, the assigned color and its open page are lost. Similarly, Vivaldi allows for "stacking" tabs into groups (Vivaldi 2017) which reduces the clutter on screen when there are many tabs open at once. This allows to keep tabs open while still being accessible. In contrast, Opera provides the user with the ability to create multiple named "workspaces" of which only one can be displayed at once (Opera 2020). This allows the user to be more focused on a single task and allows for quick task switches. The tabs in inactive "workspaces" are offloaded to save resources. Apart from coloring tabs, Edge also provides a grouping feature called "Collections" (Microsoft 2020a). Collections are groups of web pages, notes and files that can be managed from a sidebar in the browser. Firefox does not provide any such functionality by default in it's current version. However it did have a feature for tab grouping up until 2016, when it was removed with version 45.0 (Mozilla 2016a). Firefox recommended to use Bookmarks instead (Mozilla 2016b).

2.3 Artifact Grouping

Interaction with groups of artifacts and the detection thereof is solved in different ways. The simplest solution is to allow the user to manually group artifacts. This requires an easy to use User Interface (UI) design. In today's browsers this basic functionality is supported with bookmarks and folders. On the other end of the spectrum of artifact grouping, fully automated systems try to recommend useful groups of artifacts to users.

2.3.1 Manual Grouping

As already mentioned, the most prominent system to use manual grouping in a browser are the bookmarks. However there are tools that try to improve upon this concept by providing functionalities like closing multiple artifacts at once or grouping pages into workspaces through a simple UI (Motion 2020; Workona 2020; Sessionbuddy 2017; Microsoft 2020a).

Research in this area often looks at all the digital artifacts on a computer, rather than just the browser artifacts. Smith et al. (2003) with their "GroupBar" project, tried to integrate manual grouping into the Windows taskbar. The UI makes functionality like opening or closing a group of windows at once easily accessible and provides the user with a way to focus on a specific group, where all the other windows are stored and can be reopened from the application. The results from their study indicate that with GroupBar users were able to solve tasks quicker and satisfaction was higher than with the standard Windows TaskBar. However, many users were missing a way to label or color-code the groups they created.

Bragdon et al. (2010) developed CodeBubbles, a manual grouping tool for the Eclipse Integrated Development Environment (IDE). CodeBubbles helps programmers focus on a task by providing them with a UI that only shows relevant functions, opposed to a full file, and builds a flow based on a user's navigation through the codebase. They spent a lot of effort to create a dynamic UI that positions bubbles of code snippets in an intuitive way and allows the user to group up bubbles by dragging them together. In a small study Bragdon et al. (2010) showed that with CodeBubbles, Computer Science students were able to solve 2 tasks faster and were more successful in completing the tasks.

2.3.2 Automated Grouping

There are, broadly speaking, three ways to automatically group artifacts. One way is to look at the context and semantics of an artifact. To do this, some kind of natural language processing is required, often in combination with machine learning (Maalej, Ellmann, and Robbes 2017).

TaskPredictor.WDS and TaskPredictor.email Shen et al. (2006) used the metadata of windows and email respectively to group artifacts by task. TaskPredictor.WDS uses an artifact's name and its path or Unified Resource Locator (URL), segments them into single words and uses them to get a prediction from a pre-trained machine learning model. Similarly, Taskpredictor.email uses the from, to, cc, bcc and the subject fields to predict which task an incoming email belongs to. Even with a lot of noise in the training data, Shen et al. (2006) were able to achieve an accuracy of 80% and 90% for TaskPredictor.WDS and TaskPredictor.email respectively in a multi-month study with 9 participants.

Another way is to look at how a user interacts with artifacts. Zhang et al. (2015) for example recorded the switches between artifacts. They specifically looked at the frequency of switches and their recency, this means that a more recently and/or frequently visited artifact is likely more related to the current artifact, than one that has not been visited in a long time and/or not very often. Azimpour-Kivi and Azmi (2011) suggest a similar approach. To group pages they formulated a similarity calculation based on the frequency and duration of visits with the page length being used to normalize the visit durations. They found that their approach was more effective than other available methods. There have been many approaches based on the interaction with artifacts, Bernstein, Shrager, and Winograd (2008) used the data on switches between windows to determine the association of windows and to visually move associated windows closer together.

The third way to determine relatedness of artifacts, is to employ a combination of the two previous approaches (Pilzer et al. 2020). Which is meant to increase the accuracy of the groups and has performed well (> 70% accuracy) compared to previous approaches in associating windows to tasks (Oliver et al. 2006) as well as in finding the most relevant open windows on a desktop (Pilzer et al. 2020). In 2009, Shen et al. (2009) created TaskPredictor2. This version of TaskPredictor utilizes a hybrid approach it uses the context of a window and considers frequency and recency similarly to the previous two approaches. Shen et al. (2009) also switched their machine learning approach to an online algorithm to be able to use feedback from users in future predictions of TaskPredictor2. To validate their approach Shen et al. (2009) had one user work with the tool installed for 4 months. Out of the 96 proposed task switches the user interacted with, they rejected 36, accepted 37 and modified (time of the switch and/or category of the task) and then accepted 23.

Chapter 3

Approach

In this section we further analyze and compare artifact management features of existing tools. Based on this analysis and previous research we introduce central concepts to our design, namely medium-term management of artifacts and fuzzy sets. In the final part we showcase and discuss the final design of Taboor and its core features.

3.1 Existing Features

To be able to better compare features that facilitate organizing and re-finding browser artifacts, we categorized the most common features from the most commonly used browsers (Chrome, Edge, Firefox and Safari) as well as two Browser Extensions, Workona and Session Buddy. We categorized them by the artifact they help to organize or re-find. Web page artifacts are organized by their unique URL while tab artifacts can contain any web page and only the tab itself is relevant to the functionality. "Persistent" indicates if the artifact is still accessible / stored somewhere after it has been closed. "Access" describes where the feature can be found in the browser. Some features can be accessed through multiple ways or the way to access them can slightly differ from browser to browser.

3.1.1 Saving Pages

From our evaluation (Table 3.1) it seems that most of these features are either for long-term usage and storage of artifacts, like bookmarks or collections, or they support short-term usage of artifacts (tab marking, workspaces). We believe that there is a need for a feature or a bundle of features that supports something in between, a medium-term solution. The functionality must be easily accessible, must integrate seamlessly with knowledge workers existing strategies and needs to keep active maintenance to a minimum. We would expect this to encourage users to close tabs more often and make task switching easier since pages can be quickly saved as groups, found and opened and don't need to be actively managed.

3.1.2 Re-finding pages

To further encourage the closing of tabs, the fear of losing information needs to be addressed as well. We believe closing a tab is more likely to happen if the closed page can quickly be found again. Even if the page has not been saved, we need to give the users an easy way to re-find previously visited web pages. Similarly to the tab organization there are already a few features available for users.

Feature	Artifact	Persistent	Access	Description
Bookmarks (all Browsers)	Web page	Yes	Bar at the top, Sidebar / extra Window	Strong Tool to save and organize URLs and to open single pages
Collections / Tab Groups (Edge, Vivaldi, Workona, others)	Web page	Yes	Often Sidebar, extra Window or Dropdown	Strong tool to group up pages and open them together
Tags (Session Buddy)	Web page	Yes	Dropdown	Makes pages easily searchable
Tab Marking (Chrome, Edge)	Tab	No	Contextmenu of Tab	Allows to organize open tabs with colors
Workspaces (Opera)	Tab	No	Sidebar	Allows to easily switch between open tabs

Table 3.1: **Evaluation of existing artifact management features** *Sources: (Microsoft 2020a, 2020b; Vivaldi 2017; Opera 2020; Google 2020; Sessionbuddy 2017; Opera 2020)*

As shown in Table 3.2, there are only a few features that help with re-finding information. But they are very integrated and powerful tools. Especially the integration of the browser history in the URL Bar enables users to find pages they want to revisit easily. A problem we see with these features is that they only provide functionality to re-find single pages one at a time. When switching a task, multiple pages might need to be closed at once to minimize the attention residue of the old task(s). However, exactly those pages might be reused at a later point when a task is resumed. We believe a system that helps users with finding groups of relevant pages might make it more likely for the users to close tabs, especially in a task-based workflow, and therefore reduce artifact clutter on screen and the related cognitive load for the user.

Feature	Artifact	Persistent	Access	Description
Browser History / Recent History (all Browsers)	Web page	Yes (settings)	Sidebar or extra Window	Stores all visited pages and includes a search bar
URL Bar (all Browsers)	Web page & Tabs	Yes, keywords	Top of the Window	Shows current URL and allows to search history, open Tabs, remembers keywords from searches.
Recently Closed (all Browsers)	Tab	No	Dropdown	Shows a certain number of recently closed tabs

Table 3.2: **Evaluation of existing artifact re-finding features** *Source: Own depiction*

3.1.3 Conclusion

We believe that by creating a bundle of features that provide improved accessibility to recently closed groups of web pages through medium-term storage of such groups, as well as providing ways to re-finding and re-opening them, we can help knowledge workers with their task switching process and help them focus on single (medium-term) tasks.

3.2 Fuzzy Sets

A core concept to our design decisions are fuzzy sets. An earlier study by Schlaefli (2020) found that users often associate pages with multiple groups or with a single group but only for a limited amount of time. Bernstein, Shrager, and Winograd (2008) also suggested that association of artifacts is more natural to task-based work than categorization. This leads us to believe that task-based sets are not clearly defined groups but rather fuzzy sets. This concept influenced the design of core features such as marking pages and the highlights.

We believe that fuzzy set theory can be applied not just in the UI design but also in the design of algorithms to recommend groups of pages. Fuzzy set theory has been initially formulated by Zadeh (1965). Since then it has found applications in psychology (Kochen 1975), engineering (Balaman 2019) and many other areas. Fuzzy sets excel in areas where uncertainty exists or in other words where there is vague data (Balaman 2019). We think that this uncertainty is given here because not every page in a group of web pages might be as important or relevant to a task as other pages in the group and additionally we believe this relevancy to be highly subjective for every user.

3.3 User Driven Design

As suggested by Lowdermilk (2013) and Knight (2018), we used created and used three design personas (Appendix A.1) to design and validate the functionalities of our feature bundle and further improved it over multiple iterations to better fit the user profiles and how they could interact with the features. As a first inspiration for the design we also used properties from our initial analysis of existing tools, like the colors used in Google Chrome to group tab or the sidebar used by Microsoft Edge to access the collections. To make this bundle of features widely available, we decided to create a prototype web extension for Firefox as well as all Chromium-based browsers (i.e. Chrome, Edge and Vivaldi). Safari is not supported due to its incompatibility with the google extension API, which the prototype builds upon (Google 2016).

Throughout the creation of the web extension we regularly had two users test the application in short 30 minute sessions and give feedback on the features, concepts and ease of use of the web extension. We went through multiple iterations of the UI and the functionalities. In the following sections we describe the three core features and their UI elements in their final form used in the user study.

3.4 Core Features

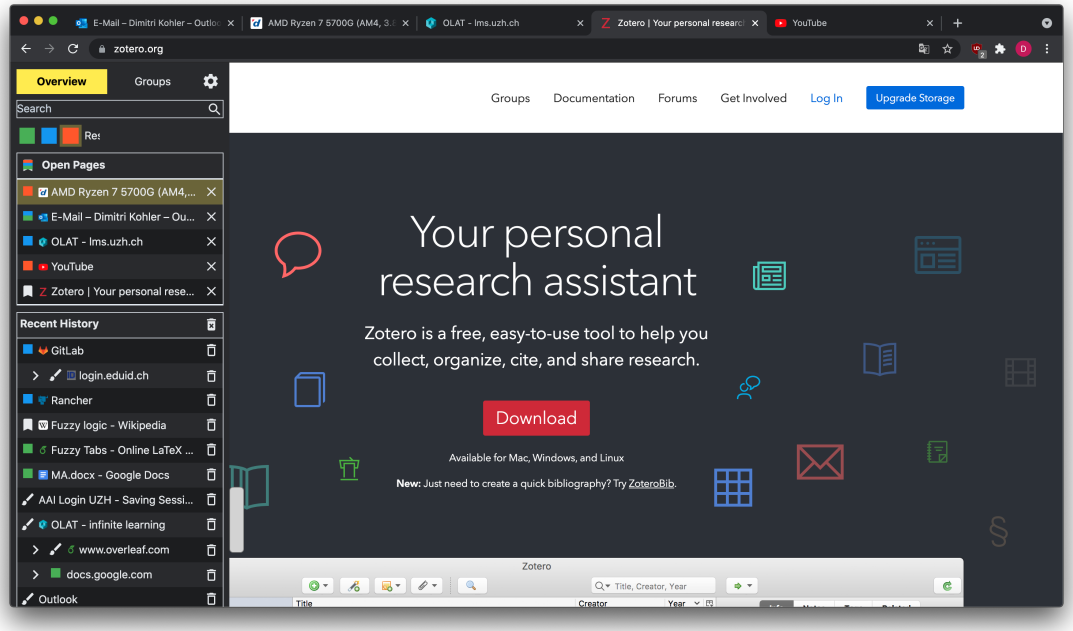


Figure 3.1: **Screenshot of Google Chrome with Taboor open** The screenshot shows Google Chrome with Taboor open in the sidebar. *Source: Own depiction*

All the core features were combined in a single browser extension. The browser extension renders a hideable sidebar on the left of a currently open web page (Figure 3.1). The core functionality is integrated with different views that show all the currently open pages, a recent history of visited pages and an overview of the different marked as well as named groups of pages. A detailed description of all the functionalities, can be found in Appendix A.4.

3.4.1 Marking Pages

The main feature provides an easy way to save, re-find and then reuse pages a user has already visited before. Marking pages is meant to be a way to quickly group and store web pages needed in the near future. After a few days of no usage (depending on the settings), marks are reset and all unused marked pages are automatically marked with a black color. This group of black pages is a way to clean up unused pages while minimizing the danger of losing information.

Colors are not applied on a tab level, like with the tab marking feature in Google Chrome, but on a per page level. A user can mark a page, navigate away in the browser and still find the marked web page in the extension UI. To make marking pages as easy as possible, Taboor also allows to mark pages from a "Recent History" (Figure 3.6 a) which shows the last 100 visited web pages in chronological order. To make the marking functionality as accessible as possible users can hover over the paint brush icon next to the name of a page and select one of five colors (Figure 3.2). Marking pages with colors is an integral part of our new workflow to store and re-find pages

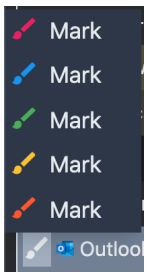


Figure 3.2: **Dropdown to mark a page** This figure shows how a web page can be marked in Taboor and the five colors available. *Source: Own depiction*

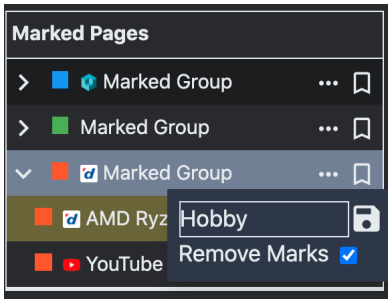


Figure 3.3: **Dropdown to add a page or group to a named group** This figure shows the dropdown to set a name for a named group and shows the option to remove the marking of the pages at the same time. *Source: Own depiction*

with Taboor. The concept behind the workflow is that UI elements showing currently open and recently visited pages provide a short-term solution to storing and re-finding information. From these two UI elements, pages can be marked and are stored for up to two weeks (depending on the settings) as a medium-term group. After up to two week of no usage of a marked page the marking decays and the page is marked with a black color. All decayed pages can be found in the black group as long as the user does not delete them manually. If the user decides that they want to store a page or a group long-term, marked pages can be converted to a named group (Figure 3.3. Named groups are persistent and not subject to the predefined decay. Based on the concept of fuzzy sets, another principle in the design of marked groups is that a page can have multiple colors and appear in multiple named groups at the same time. This gives the users more freedom and allows for flexible group building. This property is also indicated by the icons we designed. When a page is marked, the brush icon becomes a square with the assigned color(s) and becomes a ribbon if the page is member of a named group. Figure 3.4 shows all the possible constellations.

This workflow from short- to medium- to long-term storage is meant to provide a novel way to create groups of pages and to minimize the effort to manage groups and marked pages while keeping clutter in the groups to a minimum. A user can interact with these groups either trough the group view (Figure 3.6 b) or through the quick access bar.

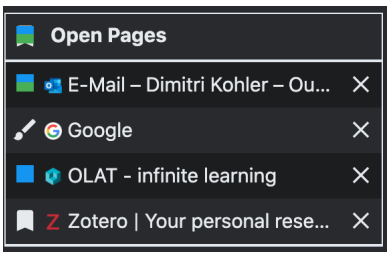


Figure 3.4: **All possible indicators for markings of pages** This figure shows the marking indicator in different states, from top to bottom: multiple colors and in named group, multiple colors, unmarked, single color, in named group. *Source: Own depiction*

3.4.2 Quick Access Bar

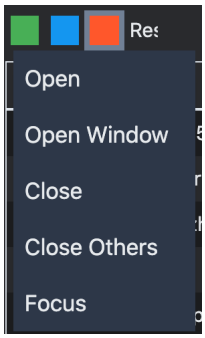


Figure 3.5: **The quick action bar and it's dropdown menu** This figure shows quick action bar with three marked and one named group. It also shows how a user can interact with the groups through the quick access bar. *Source: Own depiction*

The Quick Access Bar is a UI element that provides users with quick access to marked and named groups of pages. It is placed at the top of the sidebar and shows boxes of recently used colors and the first three letters of recently used named groups (Figure 3.6). By clicking on its respective square or text, a user can interact with that group of web pages (Figure **??**). The following interactions are possible:

- Open - Opens all pages of the group or color in the current window.
- Open Window - Opens all pages of the group or color in a new window.
- Close - Closes all pages of the group or color.
- Close Others - Closes all pages not in the group or color.
- Focus - Opens all pages of the group or color and closes all others.

The goal of the Quick Access Bar is to create additional value to marking pages and reduce the number of clicks necessary to access stored web pages. These functionalities provided by the Quick Access Bar aim to enable users to switch between groups of pages, close multiple pages at once and focus on single groups more efficiently. The squares in the bar are sorted by recency of use of a color or group and up to six groups can be displayed at once.

3.4.3 Related Highlights

With a single click on a webpage in the extension, users are presented with suggestions for related highlights (Figure 3.6 c). The related highlights allow the user to find related pages to a certain web pages, which they had visited before based on how often they visited another page, how long they visited this page and how similar the page title is to the initial page. From this view it is possible to mark multiple related pages at once and it enables re-finding and adding useful pages a user might have dismissed to existing groups.

To calculate the frequency (Equation 3.1) and time spent on a page (Equation 3.2), the extension tracks all switches between pages or tabs and builds a switch graph containing information on the visited web pages and when a switch happened. From the graph starting from the inital page, the highlighting algorithm creates a subgraph with all pages within a range of 5 steps (P). This subgraph is then analyzed based on a simplified version of the calculation of Interest (Equation 3.3) of web pages proposed by Azimpour-Kivi and Azmi (2011). Our simplification is that we do not consider the length of a page when calculating the time spent on a page (P_i). This simplification is due to efficiency concerns when collecting this data in real-time but it might bias the algorithm towards longer pages since more time could be spent there.

$$F(P_i) = \frac{Frequency(P_i)}{\sum_{P=pages\_in\_subgraph} Frequency(P)} \tag{3.1}$$

$$TS(P_i) = \frac{TimeSpent(P_i)}{\sum_{P=pages\_in\_subgraph} TimeSpent(P)} \tag{3.2}$$

$$Interest(P_i) = \frac{2}{\frac{1}{F(P_i)} + \frac{1}{TS(P_i)}} \tag{3.3}$$

The resulting percentage of interest is used to find the pages with the highest frequency and time spent compared to all other pages in the subgraph. Only focusing on a subgraph reduces the probability to find pages from tasks that have been completed or are no longer relevant. This also introduces recency as another (soft-)factor for the highlights algorithm.

To further increase the accuracy of the related highlights, a string similarity calculation was used on the tab titles of the webpages. For this we used an npm package called “string-similarity” by Kurdekar (2021). The package is based on Dice’s Coefficient (Dice 1945) to determine the similarity of two strings. We ended up using this package after testing two of our own implementations of a string similarity algorithm, which we found to be less efficient and less accurate than the “string-similarity” package.

This hybrid approach is able to find the "most interesting" used pages that have at least some similarity between their titles. This approach can not produce perfectly accurate highlights, however due to the fuzzy nature of related pages we believe that this accuracy suffices. To help

manage the fuzziness of the results, we introduced different highlighting policies which users can change on the fly, that directly determine the thresholds for interest and title similarity. Users have three highlighting policies to choose from:

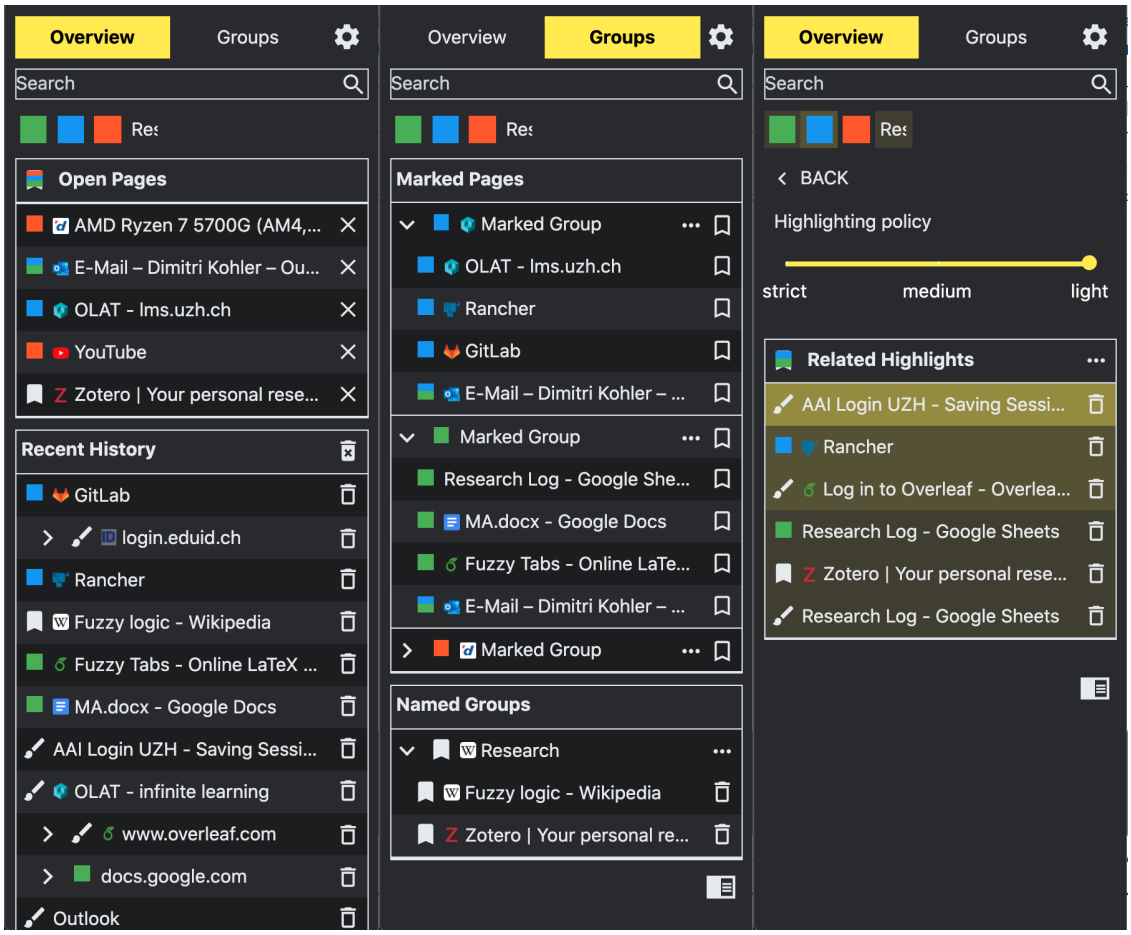
- Strict - Only pages with a very high interest ($>= 1/numberOfPages$) and high title similarity ($>= 0.18$)
- Medium - All strict pages and pages with a medium interest ($>= 0.5/numberOfPages$) and medium title similarity ($>= 0.13$)
- Light - All medium pages and pages with a low interest ($>= 0.1/numberOfPages$) and low title similarity ($>= 0.08$)

We refined these thresholds with the feedback from our two test users, and found these thresholds to deliver sensible results. However, depending on the browsing behaviour of a user (i.e. how much time is spent on single pages, or how often they switch between pages) results might vary.

Pages that have been determined as related highlights, are highlighted across all UI elements (Open Tabs, Recent History and all Groups). The highlighting is indicated by a specific background color users can choose in the settings. The background color's alpha changes according to its highlighting policy (Figure 3.6 c). Pages that are part of the strict subset have a high alpha background, while pages in the light subset have a low alpha background. We believe this design helps to further emphasize and visualize the fuzziness of groups

3.5 Architecture

Taboor is based on a fork of Tabai originally created by Schlaefli (2020) for their Master's Thesis. The frontend as well as the backend were heavily modified but there is an overlap in the used packages and APIs. Like the previous work, the extension was programmed using Typescript and React. How to access the source code from our private repository can be found in Appendix A.2. Our goal was to create an extension that only needs a single installation in the browser, with no additional local installations needed. This meant that all algorithms had to be as efficient as possible to minimize resource usage in the browser and the memory footprint of storing every page switch had to be considered. We managed this by limiting the amount of web pages that could be displayed at once to 100 and by having a clean up algorithm that would delete all data older than 7 days.



(a) Main View (b) Group View (c) Highlights

Figure 3.6: **Screenshots of the three main views of Taboor.** From left to right: main view (a) with the list of open pages and the recent history, group view (b) showing all marked pages grouped by color and all named groups, highlights (c) view with the highlights. *Source: Own depiction*

Chapter 4

Study Method

4.1 Study Design

4.1.1 Stage 0: Pilot Study

We presented our draft introduction documents to two pilot participants. The feedback from these two participants helped us to improve and refine the introduction of the extension and we hope that this enabled participants of the main study to give better feedback, since they could understand the functionalities of the extension better. The final version of the introduction documents can be found in Appendix A.3, A.4 & A.5.

4.1.2 Stage 1: Baseline

In the first week of the main study, participants installed the web extension in their browser and worked normally while the extension collected censored data on their browsing behavior. During this time the extension UI was disabled. At the end of the first week (5 work days) the participants were asked to enable the UI and download the log file of the first week containing the censored data. This log file is then sent to the researchers.

4.1.3 Stage 2: Working with the Extension

In the second week of the main study, participants were introduced to the web extension with the introduction documents. Additionally a short video call was held with each participant to give them a personal introduction to the study as well as the extension and for them to be able to ask questions. Participants were encouraged to try and integrate the extension's functionalities into their daily workflow. After two days they were reminded to use the extension via email, to try and motivate them to keep on using the extension. After using the extension for another week (5 work days), the participants were again asked to download the log file containing a censored data.

4.1.4 Stage 3: Semi-structured Interview

At the end of the study participants were invited to a 20 minute semi-structured interview. In the interview participants were asked questions about their strategies for managing tabs and web pages during their browsing with and without the extension and participants gave insights into how features of the extension worked for them and how they affected them. The interview outline can be found in Appendix A.6.

4.2 Participants

Our two participants in the pilot study (one female, one male) worked in product management and IT Consulting and are between 25 and 30 years old. For the main study we were able to work with six participants (two female, four male). Two participants worked in product marketing and four work in IT. Our participants for the main study were 33.5 years old on average. When asked about their specific role in the field, two participants mentioned being founders of their own company and two participants described their role as being Senior Software Engineers. One participant described their role as Key Account Manager and another participant said they work as an IT Project Manager. Of the six participants in the main study, three used Google Chrome, two used Firefox and one participant used Vivaldi as their main browsers.

One participant (P6) was not able to provide us with their log files since their browser set up with multiple profiles and browsers did not allow us to gather sufficient data. However they participated in the final interview and were able to give us valuable feedback on the extension. Another participant (P1) could not provide us with a full data set at the end of the baseline week due to a Chromium update temporarily breaking parts of our tracking code in all Chromium-based browsers.

Our participants spent a average of 1170 minutes (median 1260 minutes) in the browser in the first week. In the second week participants' browser time went up by 15% to an average of 1320 minutes (median 1453 minutes), possibly due the low number of data points and the addition of P1's data who had an average browser time of 1724 in the second week. Without the data from P1 we found the hours of browsing to be consistent across both weeks of the study. On an average work day our participants spent 4 hours working in the browser.

4.3 Data Collection

The data collected after the first week represents a big graph containing all the visited pages as well as all the switches between pages and tabs. While the application itself stores URLs in the browser, the logs used for the study do not contain any URLs and do not contain any identifying data. Every visited web page here referred to as "nodes" gets a random numeric id. Node id 0 is reserved for any time the browser loses focus. The switches between the nodes, here called "edges", contain two ids (from, to) of the two nodes involved in the switch and a list of timestamps when the switch(es) occurred. In addition to the switches between visited pages, we also collect data on the number of open tabs. Everytime a tab is opened or closed, the extension logs the time of the event and the number of tabs open after the event.

The data from the second week of the study extends the gathered data with usage data of core features, namely marked pages and highlights. Specifically we collect data on when a web page is marked with a color, added to a named group or opened after it was suggested as a highlight. This additional data is stored per feature and consists of an id for a web page (same as in the graph), the type of the event (open, close, add or remove) and a timestamp for when the event occurred.

The interviews were conducted in German or Swiss-German via Zoom. With the consent of the participants we recorded the audio of the calls, transcribed their answers and then translated the answers to English. After the transcription was completed the audio files were deleted.

4.4 Data Analysis

4.4.1 Quantitative

The quantitative data was analyzed with a small web tool we developed to analyze the log files generated by the extension. It's source code is available on GitHub (Appendix A.2). The tool calculated the following values:

- Sum of all web page visit durations
- Number of Pages vs Number of Visits (Absolute and Percentage)
- Average day for the Number of open Tabs (15 min interval moving average)
- Time spent on Pages vs Number of Visits
- Average time per Visits vs Number of Visits
- Average day for the Number of Switches (15 min interval moving average)
- Average Time between Visits

For week 2 additionally:

- Number of Marks added
- Number of Marks removed
- Number of Marks opened
- Number of Pages added to named groups
- Number of Pages removed from named groups
- Number of Pages opened from named groups

The data was analyzed per week and participant. To further analyze and compare the data we transferred the calculated values from our web tool to Excel. Since the data only covered two weeks with five days of work each from five participants we decided to do a qualitative analysis of the quantitative data.

4.4.2 Qualitative

The quantitative data was visualized and further analyzed to see if we could find interesting patterns and trends in the data especially in combination with the interview answers. For example we found that participants saying they very rarely close tabs had an upward trend for the number of open tabs on an average day while participants saying they are actively trying to close tabs had a downward trend throughout the day. Even if the average number of tabs was similar.

To analyze the answers from the semi-structured interviews, we employed an open coding approach. From the 6 interviews we extracted 200 statements, which we iteratively categorized by their content to get a better grasp of the different strategies used by people to organize their tabs and web pages as well as their impressions of the extension. We ended up with 19 categories. Where sensible, we used the sentiment of the statement (good / bad, useful / not useful) as a subcategory, so we could better understand the trends for every category.

Chapter 5

Results

From the collected data and the interviews we tried to learn more about the strategies knowledge workers use to manage and re-find browser artifacts (RQ 1), what effects Taboor has on the participants browsing behavior (RQ 2) and how Taboor's core features integrate with their existing workflows (RQ 3).

5.1 RQ1: How do knowledge workers manage browser artifacts?

The participants gave us insight into their daily workflows and we saw different approaches to how artifacts are organized. In contrast we only saw one dominant strategy on how to re-find artifacts that revolved around the URL bar in the browser.

5.1.1 (a) How do knowledge workers organize their browser tabs and web pages?

Participants reported a wide range of tab and web pages organization strategies. The main differences in strategy could be seen in the usage of bookmarks and how quickly users are to close or recycle tabs.

Bookmarks usage ranged from "not at all" (P1) to having close to one hundred bookmarks organized in extensive folder structures (P4, P6). P6 even goes as far as to have multiple profiles with independent bookmark structures and individual browser settings and extensions per profile. For different tasks and projects they use different profiles. P2, P3 & P5 reported that they have between 5-10 bookmarks that they use for pages they visit infrequently and can't simply remember off the top of their head and for pages they use on a daily basis for quick access. All participants reported to have some pages they can easily remember and thus don't see a need to bookmark them. The participants that worked with bookmarks, mentioned that they rarely spend time to clean up their bookmarks. P4 and P6 tend to add a lot of pages to their bookmarks and only remove them when they find a web page to not be reachable anymore. The other participants said that they normally don't change their bookmarks at all. None of the participants reported to be using any other grouping tool provided by the browser or a browser extension at the time of the study.

All of the participants mentioned the natural order tabs are opened in and thus often sorted by recency from left to right. However some of the participants had stricter strategies with the order of their open tabs. P3 for example sorts their tabs by task and importance. They work through

their tasks from left to right and close all the tabs of a task after it is finished. They also have their browser set up to open with the same tabs (and order) it has been closed with.

All participants mentioned that the importance of a tab generally decreases the further right in the tab bar it is. One participant (P1) normally groups tabs belonging to the same task but said that it is often too time consuming to constantly drag all the tabs to their group. For them most browsers don't support this strategy properly because just having the tabs grouped next to each other does not reduce the clutter on screen.

3 participants reported having 3-5 fixed tabs that are used regularly and kept open on the very left of the tab bar. P4 even set up their browser so that their 3 standard tabs always open on browser start up.

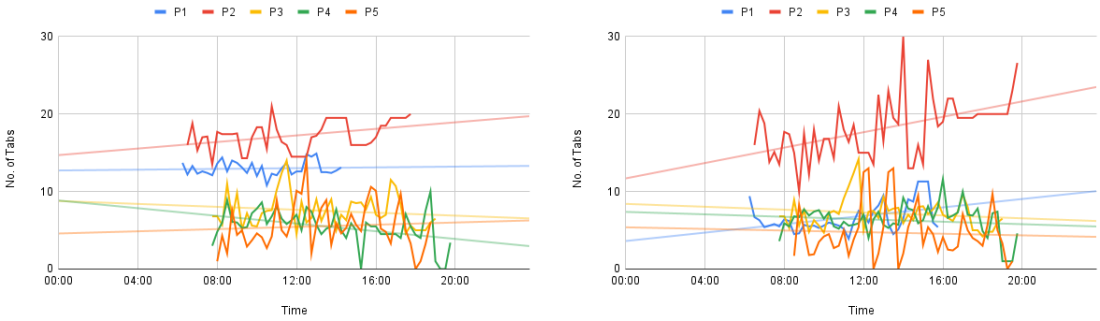


Figure 5.1: **Average number of tabs throughout an average work day** These two charts show the open tabs on an average day of week 1 without the extension (left) and week 2 with the extension (right) *Source: Own depiction, from the collected study data*

Closing and recycling tabs was mentioned by multiple participants as an important part of their strategy. In the number of open tabs over an average day and the trend throughout the day, we can see two groups of strategies forming, which we named "Tab Killer"- and "Tab Hoarder"-Strategies. Participants mainly employing Tab Killer-Strategies (P3 and P4) reported that they very strictly close tabs and always try to have no more than 5 tabs open. This can also be seen in the number of tabs recorded by the extension, the number of open tabs of P3 and P4 is consistently at or below 10 and even decreasing throughout the day (Figure 5.1). Talking about closing tabs, P4 said "I'm very quick when it comes to closing tabs. I rarely have more than five or six tabs open at once and I'm pretty strict with that.". P3 simply doesn't see a good reason to have more than 6 or 7 tabs, they reported "I have three fixed tabs and two for specific tasks I'm working on. As soon as I finish a task I close all its tabs. I'm a tab killer."

In contrast to this strategy, participants employing Tab Hoarder-Strategies (P1, P2), reported to tend to keep tabs open for longer periods of times which leads to their average number of open tabs being higher and this number even tends to grow throughout the day (Figure 5.1). For P2 the main reasons to keep tabs open was to not lose information and to easily find it again. They said "I just don't really see a reason to close most of the tabs, it's really convenient to find information again. I'd call it controlled chaos". P1 similarly mentioned that they find it very convenient to have most of the important tabs available and that the URL bar helps them to find the appropriate tab again.

P5 is an interesting case, on the topic of closing tabs they said: "I don't really close too many tabs, only if I lose the overview. Then I just close the browser and start from scratch." In their data we can see this behavior (Figure 5.1), the tab number grows over a few hours and then suddenly

drops only to grow again. This mix of strategies leads to a lower average number of open tabs but the upwards trend in the number of tabs throughout a day is still present.

5.1.2 (b) How do knowledge workers re-find information they had closed before?

We observed similarities in how participants go about re-finding web pages. Apart from the different usage of bookmarks, participants tend to be able to remember most frequently used pages and don't save them anywhere. If they can't remember a web page and don't have them saved in their bookmarks, participants normally try to find the page they are looking for in the URL bar. P1 describes their strategy as follows: "My main tool to find pages I closed before is the URL bar at the top of the browser. There I try to enter keywords and hope that I can find the needed page in the resulting list." If the URL bar does not yield any results, all but one participant mentioned using the browser history. Only P6 has their browser set up to be in incognito mode at all times, where no history is recorded, and thus relies on their extensive bookmarks structure.

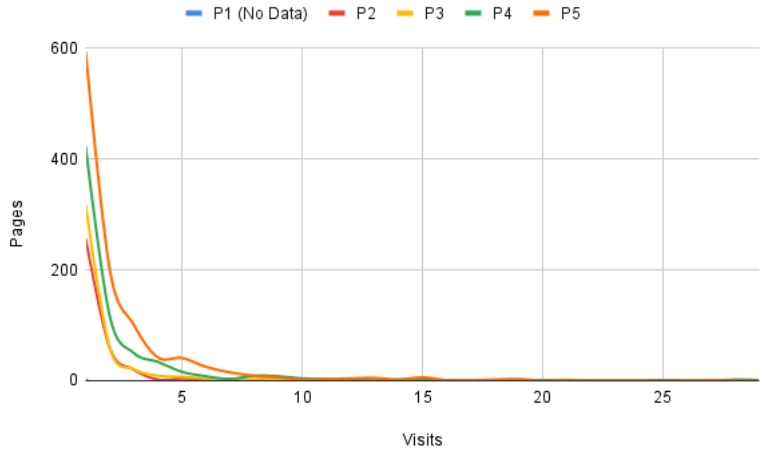


Figure 5.2: **Number of pages vs number of views to a page** This chart shows the power-law distribution of the number of pages for the view counts of a week. *Source: Own depiction, from the collected study data*

Our gathered data shows that on average over 65% of all unique web pages are only visited once and over 85% of all unique web pages are visited no more than 3 times (Figure 5.2). While these pages make up a big amount of the visited pages, we found that they only make up 32% of all unique visits to pages while the top 4 most visited pages make up 16.5% of all visits. (Figure 5.2) also clearly shows the power-law distribution of the visits that was also observed by Zhang et al. 2015 and Huang and White 2010 in their works.

With more visits to a page the time between visits also changes. On average a site with two visits has around 4 hours in between the visits. While a web page with 80 visits has around 30 minutes in between visits. (Figure 5.3) shows the data gathered in the two 5 day weeks (Monday to Friday) of the study. We can see that the average times between visits stay similar. It is important to note that the data includes the time not spent in the browser, like working in a different program, during breaks or at night.

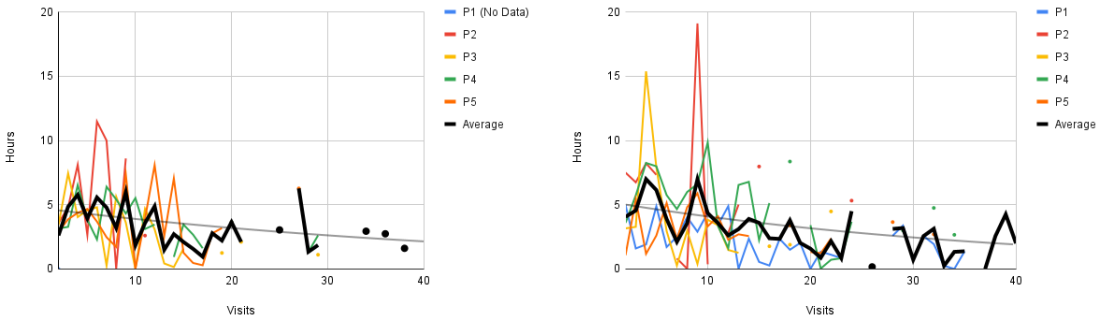


Figure 5.3: Average time between visits by number of visits Comparison of the average time between visits between week 1 without the extension (left) and week 2 with the extension (right) *Source: Own depiction, from the collected study data*

5.2 RQ2: Can this browser extension reduce the attention residue from the amount of information available in the browser?

From three participants we heard that the extension helped them to better focus on single tasks and they felt like the extension helped them to have a better overview over their open tabs. Four of the participants also said that due to the extension's functionalities and views it was more likely for them to close tabs with the extension. However, as shown in the subquestions, we can not see these effects in the analysis of the collected usage data.

5.2.1 (a) Can Taboor actively support knowledge workers to reduce the number of open tabs?

The number of open tabs on an average day is similar for all participants between week 1 and week 2 (Figure 5.1). P1 had a significant change in the number of open tabs. However P1 explained, "I only finished up three tasks [in the second] week. Since it's my last week in this job I did not start anything new really." Another observation is that the trend for P5 changed from upwards in week one to downward in week two, aligning them more with the participants with Tab Killer-Strategies. P5 also mentioned that they feel like they are closing tabs more often than before, because the extension makes it convenient to open them again later. All other participants reported that they do not believe that the extension had an influence on their number of open tabs.

We can see a change in the variance of the number of open tabs, especially for the Tab Hoarders. P2 started the study with a variance of 3.4 in the first week, which was increased to 14.6 in the second week. Even for P1 with fewer tabs open in the second week the variance changed from 0.8 to 3.5 from week 1 to week 2. P5 showed a similar change in variance from 8.39 in the first week to 11.18 in the second week while using the extension. In the group of Tab Killers the variance decreased slightly but not significantly. P3 as well as P4 had a variance of 4.6 and 4.5 respectively in the first week and a variance of 3.8 in the second week.

5.2.2 (b) Does Taboor have a significant impact on the average visit duration of web pages?

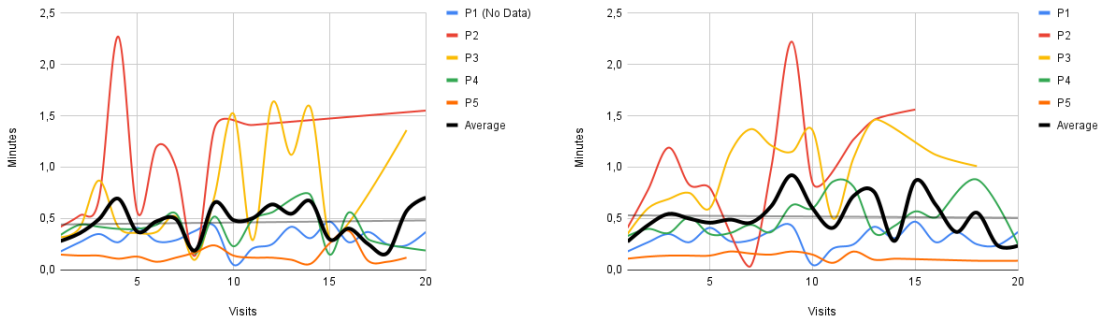


Figure 5.4: **Time spent on a page per visit vs number of visits to a page** Comparison of time spent on a page per visit between week 1 without the extension (left) and week 2 with the extension (right) *Source: Own depiction, from the collected study data*

Between week one and week two of the study we did not see a significant change in the visit duration per visit (Figure 5.4). We also did not see a change in the "fail" visits (shorter than 1 second). It is important to note that we filtered out any visits taking longer than 30 minutes, since we assume such a visit either happened during a break or at night with the browser open. Any time the browser lost focus or gained focus it was counted as the end or the beginning of a new visit.

While the extension does not seem to have any impact on the visit duration, we can see that visit duration is similar from week one to week two for every individual participant. P5 tends to spend much less time per visit than P2 or P3.

5.2.3 (c) Does Taboor have a significant impact on the number of switches between pages?

Similarly to 2b) we can see no significant change with the number of switches between web pages. We can also not see any correlation between the number of switches and the number of open tasks, since Tab Hoarders seem on par with Tab Killers in this category.

What we can see is that in week 1 as well as week 2 (Figure 5.5) P1 - P4 had more switches in the morning and fewer switches in the afternoon. Only P5 had a higher number of switches in the afternoon than in the morning. This leads us to believe that a higher number of switches does not correlate with a higher number of tabs.

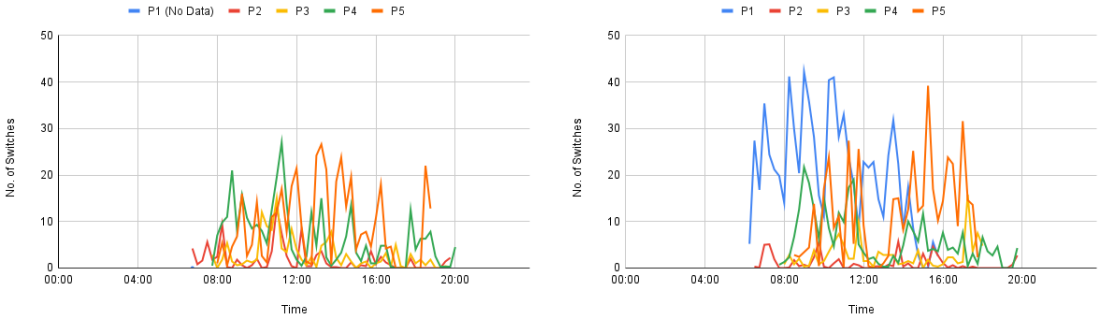


Figure 5.5: **Switches between web pages throughout an average work day** These two charts show the switches between pages (and tabs) on an average day of week 1 without the extension (left) and week 2 with the extension (right) *Source: Own depiction, from the collected study data*

5.3 RQ3: How do knowledge workers integrate the extension in their workflow?

We found that participants that tend to have a higher number of tabs open at once found Taboor to be easy to integrate in their workflows. Participants with a lower average number of tabs throughout the day and stricter rules on how to manage them struggled to integrate Taboor in their workflow. The three core features are discussed in the subquestions.

5.3.1 (a) How do knowledge workers utilise medium-term marked pages?

Using the implementation

The 5 participants created 16 marked groups in total throughout the second week of the study with an average size of 3.7 pages per group. The most used colors were red and blue, which are also the first two colors in the list of colors to choose from.

Of all the marked pages the majority was visited between 2 and 14 times (Figure 5.6). The pages that were only visited a single time were marked and never opened again. We assume that these pages were either a test at the start of the study, a late addition at the end of the study or a replacement for a bookmark that was never used.

Three participants also used named groups, they created a total of 8 named groups with an average size of 3.1 pages per group. P3 removed all marked pages after creating the named groups and exclusively worked with the named groups. P3 mentioned that "I quickly realized that some of them easily belong together so I combined them in a named group and ended up working with those." Similarly P1 was quick to create named groups and stop working with colors. They said "I normally have a group name in mind and don't really think of colors. I can't remember which color belongs to which task, so I would rather name them right away."

Participants with more of a Tab Killer-Strategy reported to have struggled to integrate the extension into their workflow. For P3 for example marking pages was just an additional step to get to named groups. Tab Killers often tried to use marking pages as a replacement for their bookmarks and found it to not be as useful. Generally the Tab Hoarders saw more value in the extension because they found it to be helpful to get a better overview and to easily close and open

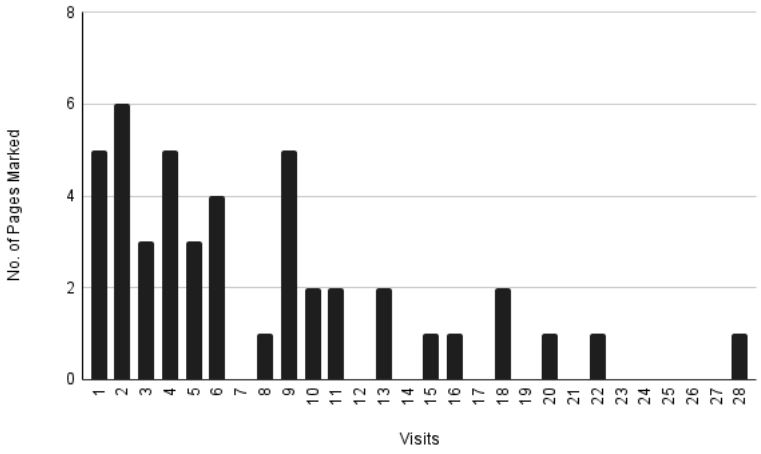


Figure 5.6: **Number of marked pages vs number of visits to those pages** This chart shows how often marked pages have been visited after they have been marked. *Source: Own depiction, from the collected study data*

multiple tabs. P5 noted that "If the marked pages were as easily accessible as my bookmarks bar, I would prefer to work with the colors."

Instead of recreating their bookmarks, most participants ended up using the colors to group pages they needed regularly for a certain task because they found it convenient to open them all at once and close them again. P4 & P6 who already have an extensive bookmarks library struggled to recreate their bookmarks strategy with the extension. Since they found their bookmarks easier to comb through than the marked groups and named groups. They also mentioned that they rarely open multiple pages at once so they did not see a big value in opening and closing multiple pages at once.

Using the concept

Despite their problems to integrate the extension into their workflow, 5 out of 6 participants liked working with the colors and found it to be an intuitive way to group pages. P2 said "The way I work, think and structure my things the colors fit perfectly."

P4 & P5 mentioned that the concept helped them to organize their tabs in a more task-based manner, which worked well with how they actually work day to day. P4 especially mentioned that the extension helped them to see more pages they use together a lot. For certain tasks they opened the same pages one after another regularly but with the task-based thinking that marking pages motivates, they intuitively started to mark all the pages and were able to open them all at once.

All in all the concept of coloring pages was liked by all participants but one, however the concept of the markings decaying, was not found to be very useful by the participants. The default decay time of 3 days was set too short for P5, saying that the decay should be at least 8 days, so users don't lose marked pages after a long weekend or a week away. They also said that the action of marking a page already shows that the user finds a page to be important and in relation to others of the same color so it should never be removed. As an improvement they suggested creating a special archive where pages previously marked with the same color would be stored and could stay grouped up while cleaning up the marked pages views. For P6 the

design of the decay itself is "trying to be smarter than the user" which they dislike. P6 also would have liked some indication in the UI how long it will be until a mark is decayed and an indication that it is decaying at all.

5.3.2 (b) How do knowledge workers utilise the quick access for groups?

The quick access of groups was the most liked feature across all participants. However, there is only incomplete tracking data on the usage of the quick bar. We know that participants opened marked pages 21 times in total and named groups a total of 10 times via the quick access bar. Unfortunately we can not say how often pages were closed via the quick access bar, since this data also contains events where a window with multiple tabs was closed and we can not differentiate between these events.

From the 5 different actions of opening or closing tabs, the participants mostly used the open group and close group functionality. Those with experience in development liked and used the "close others" action since it is very common in IDEs and they were already familiar with such a functionality. The "open in window" action was not used consistently, but two participants mentioned using it in rare situations where they wanted to compare information between two pages. The "focus" action was not used by any participants.

For the two participants that did not use the "close others" actions, the big problem was that they would have to reopen their standard tabs, like their email web tool. Their only work around for this problem was to add the standard pages to every single group so they would not get closed. This was inconvenient and the participants ended up not using the "close others" action.

Another problem for the participants was the access to the quick access bar. To use it participants would need to first open the sidebar, wait for the UI to load, then click on the color or name of the group they want to interact with and then choose the action from the drop down. P5 argued that compared to their bookmarks this was one or two clicks more to open pages and the loading time of the UI was too long to quickly and regularly use the quick access bar.

5.3.3 (c) How do knowledge workers utilise related highlight pages?

Highlights were not actively used by any participants. All participants mentioned stumbling upon them by accident but they were not able to gain any value out of the recommended pages. Participants did try the highlights during the interviews. However, they often were missing pages they would have expected to find as related pages or were overwhelmed with the amount of pages that was suggested.

During the interview P2 mentioned that the feature might be useful when doing research into a topic to re-find information but overall the highlighted pages were not found to be useful in any daily work situation the participants encountered during the study.

Participants liked the concept but in it's current implementation they did not find a use for it. Instead of highlights related to a single page, P4 suggested to once a week recommend groups of pages that are often open at the same time or often opened one after another, so the user could find browsing patterns in their daily work and optimize them by grouping the pages together.

Chapter 6

Discussion

Based on our findings, we formulated five design implications that should be used for future development of this or similar approaches. After describing and discussing the design implication, we also make suggestions how and where these design implications could improve Taboor.

The design implications are mainly based on the six interviews conducted with the participants, since the quantitative data we collected did not show any significant effects. The habits and strategies the participants used were very strong and all participants reported to have struggled to use Taboor consistently for a week without falling back into their old habits. However, we feel like their interview answers already give us insight into where Taboor could be improved to increase the ease of use and the ease of adaptation for our participants.

6.1 Design Implication 1: Supporting the Spectrum of Existing Strategies

In the study we have seen a wide range of strategies for how the participants manage their tabs and bookmarks. Some participants are very strict with the amount of tabs they want to have open and others like the convenience of having everything open and easily accessible. Our categorization into Tab Killers and Tab Hoarders is not meant to be a binary categorization of our participants but rather a spectrum to place their strategies on. It also does not mean that one single strategy is better than the others. The overall goal with this design implication should be to create a tool that supports the whole spectrum of existing strategies rather than just becoming another alternative strategy.

Obviously, personal preferences and the nature of tasks users encounter in their work influence how they organize their tabs and which strategies are viable. In our study participants that had many independent tasks would close tabs more frequently than participants like P2 who have many different tasks in the same topic that might be interconnected. We can see how in this case it is harder for a knowledge worker to close tabs even after finishing one specific task of a topic, since some pages might be useful for other tasks. However, not all participants in such a situation resorted to leaving tabs open, but rather created extensive bookmarks libraries in order to not lose information and still feel comfortable to close tabs frequently.

We hope this spectrum shown in Table 6.1 can help to shape future versions of Taboor, by helping future research to create features that are designed to support these very different strategies.

All participants acknowledge that the value of the extension grows with the number of open tabs. In that way the extension reached its goal and helped to better organize tabs. We think that having only a small number of tabs open at any time requires a lot of organization already, so Tab Killers naturally see less value in a tool to manage browser artifacts than Tab Hoarders

Tab Killer		Tab Hoarder
• Strict Tab Limit • Quickly Close Tabs • Aware of which pages are open • Extensive Bookmarks	• Order Tabs • Standard Tabs • Grouping Tabs • Few to no Bookmarks	• Leave Tabs open ("Controlled Chaos") • Only close tabs when clutter gets too much • URL bar to find open pages

Table 6.1: **Overview of browser artifact management strategies** This table shows the spectrum of different strategies and where we place them on the spectrum. All participants used a mix of these strategies. *Source: Own depiction*

who have to organize a much bigger amount of tabs and don't have an organization strategy already. We believe that the extension in its current form offers too little value for Tab Killers and does not integrate into their existing strategies easily enough. One possible solution could be to integrate the functionalities with the existing bookmarks to make the integration into users existing strategies more natural. This change is also discussed with the design implication 3 (section 6.3).

Finally, we believe that features should focus on benefiting Tab Hoarders while learning from Tab Killer-Strategies like having a strict maximum number of tabs. This could help Tab Hoarders to find new strategies to minimize the amount of tabs they build up and encourage them to close tabs more quickly. This would lead to less clutter on screen and possibly to more focused and efficient work. With this approach Tab Killers could benefit because the extension builds upon their strategies and thus might support them in optimizing their existing strategies.

6.1.1 Tool Improvement

Taboor itself was designed with some strategies from both extremes in mind. Our Design Personas were able to cover a majority of the strategies (Appendix A.1). However, we feel like there was too little deep understanding of Tab Killer-Strategies and the extension turned out to create more value for Tab Hoarders. Tab Hoarders reported the extension to be helpful to be better organized and to save time. While Tab Killers struggled more to incorporate the extension into their workflows and saw less value in the extension.

On one hand we believe this is due to the design simply not being able to support Tab Killer-Strategies fully. On the other hand it might also be harder to create as much value for Tab Killers as for Tab Hoarders since they already have strict policies in place to stay organized and do not profit as much from easier management of web pages and tabs Taboor provides.

As a concrete example of an improvement based on this Design Implication we suggest to consider the tab order in the design of the features. Marked or grouped pages need to be sortable. These sorted groups of web pages could then be opened in tabs in the order of the groups via the quick access bar. We believe that this improvement would save time for users and might make tab organization more efficient.

6.2 Design Implication 2: Focusing on Task-based vs Theme-based Grouping

Participants with extensive bookmark libraries used a theme- or topic-based strategy to group their saved pages. In comparison, we found the marking pages feature to successfully encourage task based grouping and task-thinking. In our opinion these two concepts do not compete with each other but rather complement each other. All features of a grouping management tool need to be designed to be designed to support one of the grouping strategies and as a bundle they need to be designed to consistently support one of the grouping strategies. In Taboor not all features turned out to support the same grouping strategy.

While marking pages and the quick access bar supported task-based grouping, we believe that the design and algorithm behind the highlights ended up supporting theme-based grouping. The algorithm mainly looked at which pages were frequently visited and had a similar title to the initial page. We believe that this approach does not work with task-based grouping and much rather support theme-based grouping. This decreased the usefulness of Taboor as a whole, since one of the features was designed for a different strategy and was not useful even though it was integrated with other features of Taboor.

We see a big value in supporting task-based grouping of web pages, since it is currently not properly supported by current browsers and task-based grouping might help to decrease the cognitive load of multitasking by enabling more efficient task-switching and reducing the clutter on screen.

6.2.1 Tool Improvement

Named groups as well as highlights need to be improved to better support task-based grouping. Named groups were used by participants to replace their bookmarks but they found them to be lacking in terms of structure and accessibility. One way to improve this would be to allow sub-groups inside named groups and to make them manageable via a dedicated page instead of the sidebar. This would allow users to extensively manage the named groups and have a better overview over all the saved web pages. A different way to improve the tool would be to remove named groups entirely and focus on task-based grouping in marked groups instead of trying to emulate and improve the theme-based grouping provided by bookmarks. To not lose naming, we suggest allowing naming colors and to increase the number of available numbers.

To improve the highlights we believe a complete redesign is required. The new design needs to employ a new algorithm that is better at suggesting task-based groups. This could be based on which pages are often open at the same time and which pages often have been opened one after another, as well as the order of the tabs. It would have to be seen if the fuzzy set principle we followed for this theme-based version of the highlights still holds for task-based groups or if they are more clear cut or too short lived to even benefit from a fuzzy design. Apart from the algorithm the way to access highlights needs to be reworked as well. Participants were rarely using highlights due to them being hidden most of the time and needing user action to be seen. They need to be integrated more prominently into the tool. To achieve this, participants suggested creating frequent recommendations for task-based groups a user might find useful. We will discuss this further in design implication 5 (Section 6.5).

6.3 Design Implication 3: Improving Accessibility through Integration

All participants mentioned the integration of the tool with browser native features as a point of improvement. The bookmarks and the bookmarks bar especially were used often as a benchmark for the accessibility of the features. The bookmarks bar makes bookmarks easily accessible and they can be opened with a single click. Groups and single pages should be made accessible through native browser UIs. One option would be to integrate groups and pages with bookmarks. This could make it possible to manage bookmarks via the tool and vice versa. Which would mean that a new tool could build on the accessibility of the bookmarks and possibly even increase the usefulness of bookmarks similar to what Sessionbuddy (2017) does.

The prime example for accessibility through integration is the widely used URL bar. All strategies described by the participants to re-find web pages are based on the functionalities the URL bar provides. It is a universal tool that integrates different natives features, like the open tabs, browser history, bookmarks and even the search history and makes them easily accessible through a search bar at the top of the browser.

All in all, we think if a feature can not provide more value and be as accessible as the bookmarks in the bookmarks bar or the open pages in the tab bar it will not be used in an already established workflow. By integrating a feature into native functionality, there can be a symbiotic effect, where the new feature profits from the accessibility of the native feature and the usefulness of the native feature increases by providing the new functionality.

6.3.1 Tool Improvement

There are three integrations that could help to increase the accessibility by integration. Firstly named groups as well as marked pages should be integrated with bookmarks. This would allow participants to access their existing bookmarks and newly saved pages in the extension as well as in the bookmarks bar. Marked pages would be more accessible. In their current form four to five clicks are needed to open a marked page. With an integration into the bookmarks bar the number of clicks could be reduced to two.

Secondly, to also improve the accessibility to groups of pages and not just single pages, the quick access bar should also be integrated into browser native functionalities. We believe this would make opening and closing multiple pages at once quicker and easier to integrate into existing workflows. Which would mean we could better support task-switching. There are three options we could see to integrate the quick access bar:

- In the top right of the browser, integrated with the extension icon.
- At the top of the browser, integrated in the bookmarks bar.
- In the top of the browser, integrated via keywords in the URL bar.

The first solution would mean that the quick bar is not always visible but can be accessed via a dedicated button in the top right. With this solution the quick access bar would again be hidden and would require one additional click to access compared to the bookmarks bar. The second solution is possibly the most complex to implement. To be as accessible as the bookmarks bar, we suggest integrating the quick access bar directly into the bookmarks bar. The third option would mean that the already widely used URL bar would be enhanced even further and groups would become accessible through keywords with suggestions and auto complete.

Finally, participants that regularly closed their browser struggled with the accessibility of the tool as a whole. Since the sidebar is only accessible from a website, they needed to open a random website to access the functionalities of the tool. The functionalities need to also be accessible at the start up of an empty browser or tab. This can be achieved by integrating the tool into the empty default tab pages and making the web pages and groups accessible through there.

6.4 Design Implication 4: Promoting Tab Closing Behaviour

If presented with the option, participants used functionalities to close multiple tabs and found them to be useful to manage their open tabs, to switch tasks and to reduce clutter on screen. This helped participants to better focus on their task at hand. Current browsers have a lot of functionality to open specific web pages but very little to close them again. Especially not as groups. This could be achieved by firstly adding functionality to close certain bookmarks in the browser and secondly providing functionality that allows closing tabs that are not in a certain group or saved as bookmarks.

Another way to promote closing tabs could be to detect stale tabs, based on the time since the last visit and the frequency of visits to a tab. This could be implemented as a recommendation to the user and could provide a direct way to bookmark pages before closing them.

6.4.1 Tool Improvement

The extension already offers functionalities to close groups of tabs. The functionalities to "close others" and to "focus" on one group of marked pages were generally found to be useful by the participants. However, they were not used by many participants because they did not want to close their standard tabs. We believe a feature to "pin" certain tabs or web pages would enhance these two options, remove the problem of closing the standard tabs and make it therefore more likely to be used.

Another improvement would be to also bring functionalities to close pages to bookmarks and the bookmarks bar. As described with design implication 3 (Section 6.3), the quick access bar should be integrated into browser native functionalities. But its functionality could also be extended to also allow closing (and opening) bookmarks.

6.5 Design Implication 5: Letting the User Decide

We think it is important that users have the final say if a saved page is deleted, in order to reduce the risk of information loss and to allow users to keep an overview. We saw this with the decay built into the marked pages. Participants did not like it when the extension would decide for them which pages should be removed and then automatically remove them. We believe that a recommendation system for the removal of artifacts solves this problem, since the users keep full control but are supported in maintaining their marked pages. This system could also be extended to existing bookmarks.

We have seen that bookmarks are a versatile and widely used tool to manage web pages. But we found that participants working with bookmarks rarely maintained them. Participants with more extensive bookmarks would add bookmarks more regularly than the others but the process of removing bookmarks was described similarly by all participants. They all do not actively remove bookmarks. Only when they stumble upon bookmarks that no longer work or they

know for certain they no longer need them, do they actively remove them. We believe supporting knowledge workers in cleaning up their bookmarks would help them work more efficiently. Bookmarks that are no longer needed could be detected by checking if a saved page can still be accessed or how long a page has been saved without being used.

We suggest detecting such unused pages and to then frequently present the user with a list of possible removals together with the reason for the recommendation e.g. "dead link" or "unused for x days". The same thing could be done when detecting stale tabs mentioned in design implication 4 (Section 6.4).

We think that recommendations are vital when information is removed. But recommendations should also be used to suggest groups of related pages. This has been done in many other tools before and give the user control over the groups created.

6.5.1 Tool Improvement

The built in decay of marked pages of up to two weeks was not seen as useful by our participants. We believe there are two main issues that lead to the dislike of this feature. On the one hand, the visual design of this function was not clear enough. There was no indication in the UI how long a page would remain marked and the markings were suddenly changed to another color. On the other hand the decay being automatic did not reduce information loss as much as we expected. As this way pages lost their colored marking and the marking was already considered important information by the participants. To address both of these issues there needs to be a clearer indication of the decay and the decision to let a page finally decay has to be put into the users' hands. Users should get recommendations for pages that could be removed as they have not been used for a defined amount of time. The extension should not automatically and not only based on time alone change or remove markings. This way users are more aware of the decay and can intervene if they want to keep the recommended markings. Another solution would be to improve the UI and clearly show the users when the marking will be removed. This could be done with a clock symbol showing the time left for every page in the grouping overview or the pages soon to be removed could be shown as a list in the extension so users could clearly see the changes that are about to happen.

As mentioned in the previous section, recommendations could also be used to recommend useful groups of pages to the user. This would enable users to find patterns they normally don't recognize and could help them create task-based groups quickly. There are two parts to this, first the algorithm needs to be adapted (as described in Design Implication 2 / Section 6.2) and then the recognized groups need to be presented as recommendations for the user to decide if they want to save them or not.

Chapter 7

Threats to Validity

We see three main threads to our work. Firstly, due to the nature of knowledge work and due to the goal of determining the real world usability of Taboor, there are many factors that might have influenced the browsing behaviour of our participants. This means that comparing data from two different weeks might show differences that were due to other factors than the features of Taboor.

Secondly, the study was not long enough. A longer study would have made outside factors less impactful and the participants might have been able to better adapt to the features Taboor provides. The results would be more generalizable and we could potentially see a change in the quantitative data over time. The existing strategies are habits that have formed over years and users can only change them over a longer period of time.

Finally, due to the small number of participants and the big variety of strategies we also believe that we were not able to get a full picture of all the strategies knowledge workers employ to organize their browser artifacts. The study was conducted with knowledge workers from only two industries, IT and Online Marketing, so the strategies described in this work might be specific to these two industries. A more diverse group of participants is needed to help to verify the results.

Chapter 8

Conclusions & Further Work

In our work, we presented Taboor, a novel, interactive approach that focuses on supporting knowledge workers in managing medium-term tasks in their web-browser. Taboor was designed to support knowledge workers in their existing strategies to manage their browser artifact. Its three core features are marking pages with colors to group them, a quick access bar to interact with the created groups and related highlights to (re-)find related web pages and quickly create groups.

The feedback from a two week study was mostly positive and Taboor was perceived as a useful tool to manage browser artifacts. However, some participants struggled to consistently integrate the tool into their strategies in the short duration of the study. As one reason for this participants reported to have built strong habit over the years and integrating the tool meant a big change from normal workflow. Another reason is that Taboor was not integrated enough into the native browser functionalities and therefore was not accessible quickly enough. A third possible reason is that the bundle of features did not uniformly support task-based grouping.

The core features of marking pages with colors and the quick access bar, were very well liked and were found to promote task-thinking and task-based grouping. The quick access bar especially was seen as a useful tool to close and open groups of tabs and to switch-tasks efficiently. The third core feature, the related highlights, was barely used by the participants. Neither the re-finding of pages nor the grouping of pages were seen as useful. We believe this is not only due to the accuracy of the highlights but also due to the theme-based nature of the recommendations generated by our algorithm, while all the other features of the extension supported task-based grouping.

Based on these results, we formulated and discussed five design implications that should be considered in future designs of browser artifact management tools:

1. Supporting the spectrum of existing strategies
2. Focusing on task-based vs theme-based grouping
3. Improving accessibility through integration
4. Promoting tab closing behaviour
5. Letting the user decide

We believe that the goal of future research should be to support Tab Hoarder-Strategies to reduce clutter on screen and make managing artifacts as accessible as possible. While Tab Killers-Strategies could be used as an inspiration for the design of new features. This would also mean that Tab Killers could then also profit from those features and make their existing workflows more efficient.

Further work needs to be done to improve Taboor and make it more accessible as well as easier to integrate in to existing strategies. Ways to improve Taboor were suggested in the Discussion (Chapter 6) with every design implication.

Apart from improvement to Taboor, we believe that further research needs to happen in the area of existing strategies. Due to the wide range of strategy reported from our six participants, we believe that existing strategies need to explored and analyzed in more detail to be able to design features that integrate seamlessly. We believe that a feature and tool will only be used if it can support existing habits instead of providing new strategies.

Bibliography

Azimpour-Kivi, Mozhgan, and Reza Azmi. 2011. "A webpage similarity measure for web sessions clustering using sequence alignment." In *2011 International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 20–24. June. https://doi.org/10.1109/AISP.2011.5960993.

Balaman, Şebnem Yılmaz. 2019. "Chapter 3 - Biomass-Based Supply Chains and Logistics Networks" [in en]. In *Decision-Making for Biomass-Based Production Chains*, edited by Şebnem Yılmaz Balaman, 55–75. Academic Press, January. ISBN: 9780128142783, accessed August 10, 2021. https://doi.org/10.1016/B978-0-12-814278-3.00003-0. https://www.sciencedirect.com/science/article/pii/B9780128142783000030.

Bernstein, Michael S., Jeff Shrager, and Terry Winograd. 2008. "Taskposé: exploring fluid boundaries in an associative window visualization" [in en]. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*, 231. Monterey, CA, USA: ACM Press. ISBN: 9781595939753, accessed May 12, 2021. https://doi.org/10.1145/1449715.1449753. http://portal.acm.org/citation.cfm?doid=1449715.1449753.

Bragdon, Andrew, Robert Zeleznik, Steven P. Reiss, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Joseph J. LaViola. 2010. "Code bubbles: a working set-based interface for code understanding and maintenance." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2503–2512. New York, NY, USA: Association for Computing Machinery, April. ISBN: 9781605589299, accessed August 9, 2021. https://doi.org/10.1145/1753326.1753706.

Dice, Lee R. 1945. "Measures of the Amount of Ecologic Association Between Species" [in en]. *Ecology* 26 (3): 297–302. ISSN: 1939-9170, accessed August 9, 2021. https://doi.org/10.2307/1932409. https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409.

González, Victor M., and Gloria Mark. 2004. ""Constant, constant, multi-tasking craziness": managing multiple working spheres." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 113–120. New York, NY, USA: Association for Computing Machinery, April. ISBN: 9781581137026, accessed August 10, 2021. https://doi.org/10.1145/985692.985707.

Google. 2016. *API Reference* [in en]. Accessed August 9, 2021. https://developer.chrome.com/docs/extensions/reference/.

———. 2020. *Keep tabs on your tabs in Google Chrome* [in en-us], May. Accessed August 9, 2021. https://blog.google/products/chrome/manage-tabs-with-google-chrome/.

Huang, Jeff, and Ryen W. White. 2010. "Parallel browsing behavior on the web." In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, 13–18. HT '10. New York, NY, USA: Association for Computing Machinery, June. ISBN: 9781450300414, accessed May 18, 2021. https://doi.org/10.1145/1810617.1810622. https://doi.org/10.1145/1810617.1810622.

Knight, Westley. 2018. *UX for Developers: How to Integrate User-Centered Design Principles Into Your Day-to-Day Development Work* [in en]. Google-Books-ID: yLd9DwAAQBAJ. Apress, December. ISBN: 9781484242278.

Kochen, Manfred. 1975. "APPLICATIONS OF FUZZY SETS IN PSYCHOLOGY" [in en]. In *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, edited by Lotfi A. Zadeh, King-Sun Fu, Kokichi Tanaka, and Masamichi Shimura, 395–408. Academic Press, January. ISBN: 9780127752600, accessed August 10, 2021. https://doi.org/10.1016/B978-0-12-775260-0.50021-9. https://www.sciencedirect.com/science/article/pii/B9780127752600500219.

Kurdekar, Akash. 2021. *string-similarity* [in en]. Accessed August 9, 2021. https://www.npmjs.com/package/string-similarity.

Leiva, Luis A. 2011. "MouseHints: easing task switching in parallel browsing." In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, 1957–1962. CHI EA '11. New York, NY, USA: Association for Computing Machinery, May. ISBN: 9781450302685, accessed March 11, 2021. https://doi.org/10.1145/1979742.1979861. https://doi.org/10.1145/1979742.1979861.

Lowdermilk, Travis. 2013. *User-Centered Design: A Developer's Guide to Building User-Friendly Applications* [in en]. Google-Books-ID: XiX5bNJjW0kC. "O'Reilly Media, Inc.", May. ISBN: 9781449359805.

Maalej, Walid, Mathias Ellmann, and Romain Robbes. 2017. "Using contexts similarity to predict relationships between tasks" [in en]. *Journal of Systems and Software* 128 (June): 267–284. ISSN: 0164-1212, accessed March 5, 2021. https://doi.org/10.1016/j.jss.2016.11.033. https://www.sciencedirect.com/science/article/pii/S0164121216302357.

Microsoft. 2020a. *Organize your ideas with Collections in Microsoft Edge.* Accessed August 9, 2021. https://support.microsoft.com/en-us/microsoft-edge/organize-your-ideas-with-collections-in-microsoft-edge-60fd7bba-6cfd-00b9-3787-b197231b507e.

———. 2020b. *Tab Grouping feature in Microsoft Edge* [in en-US]. Accessed August 9, 2021. https://answers.microsoft.com/en-us/microsoftedge/forum/all/tab-grouping-feature-in-microsoft-edge/646e0028-dd52-42e8-9579-8874e3b002cc.

Motion. 2020. *Motion.* Accessed March 11, 2021. https://www.inmotion.app/.

Mozilla. 2016a. *Firefox 45.0, See All New Features, Updates and Fixes* [in en]. Accessed August 10, 2021. https://www.mozilla.org/en-US/firefox/45.0/releasenotes/.

———. 2016b. *Tab Groups feature discontinued* [in en]. Accessed August 10, 2021. https://support.mozilla.org/en-US/kb/tab-groups-removal.

Oliver, Nuria, Greg Smith, Chintan Thakkar, and Arun C. Surendran. 2006. "SWISH: semantic analysis of window titles and switching history" [in en]. In *Proceedings of the 11th international conference on Intelligent user interfaces - IUI '06*, 194. Sydney, Australia: ACM Press. ISBN: 9781595932877, accessed May 12, 2021. https://doi.org/10.1145/1111449.1111492. http://portal.acm.org/citation.cfm?doid=1111449.1111492.

Opera. 2020. *Workspaces | Tidy your browser with Opera | Opera* [in en]. Accessed August 9, 2021. https://www.opera.com/features/workspaces.

Pilzer, Jan, Raphael Rosenast, André N. Meyer, Elaine M. Huang, and Thomas Fritz. 2020. "Supporting Software Developers' Focused Work on Window-Based Desktops." In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. CHI '20. New York, NY, USA: Association for Computing Machinery, April. ISBN: 9781450367080, accessed March 5, 2021. https://doi.org/10.1145/3313831.3376285. https://doi.org/10.1145/3313831.3376285.

Sahm, Alexander, and Walid Maalej. 2010. *Switch! Recommending artifacts needed next based on personal and shared context* [in en]. Gesellschaft für Informatik e.V. ISBN: 9783885792543, accessed March 23, 2021. http://dl.gi.de/handle/20.500.12116/20185.

Schlaefli, Roland. 2020. "Automated Tab Organization" [in English]. Master's thesis, University of Zurich, August.

Sessionbuddy. 2017. *Session Buddy* [in en-US]. Accessed August 10, 2021. https://sessionbuddy.com/dev-status/.

Shen, Jianqiang, Jed Irvine, Xinlong Bao, Michael Goodman, Stephen Kolibaba, Anh Tran, Fredric Carl, Brenton Kirschner, Simone Stumpf, and Thomas G. Dietterich. 2009. "Detecting and correcting user activity switches: algorithms and interfaces" [in en]. In *Proceedings of the 14th international conference on Intelligent user interfaces*, 117–126. Sanibel Island Florida USA: ACM, February. ISBN: 9781605581682, accessed May 12, 2021. https://doi.org/10.1145/1502650.1502670. https://dl.acm.org/doi/10.1145/1502650.1502670.

Shen, Jianqiang, Lida Li, Thomas G. Dietterich, and Jonathan L. Herlocker. 2006. "A hybrid learning system for recognizing user tasks from desktop activities and email messages." In *Proceedings of the 11th international conference on Intelligent user interfaces*, 86–92. IUI '06. New York, NY, USA: Association for Computing Machinery, January. ISBN: 9781595932877, accessed August 9, 2021. https://doi.org/10.1145/1111449.1111473. https://doi.org/10.1145/1111449.1111473.

Smith, Greg, Patrick Baudisch, George Robertson, Mary Czerwinski, Brian Meyers, Dan Robbins, Eric Horvitz, and Donna Andrews. 2003. "GroupBar: The TaskBar Evolved." In *Proceedings of OZCHI 2003*, 34–43.

Tabby. 2020. *Browse confidently | answers related to tabby* [in en]. Accessed March 23, 2021. https://www.tabby.us/faq.

Vivaldi. 2017. *Group tabs into Tab Stacks* [in en-US], January. Accessed August 9, 2021. https://help.vivaldi.com/desktop/tabs/tab-stacks/.

Workona. 2020. *Workona Tab Manager | Best Chrome, Edge, & Firefox Extension.* Accessed April 5, 2021. https://workona.com/tab-manager/.

Zadeh, L. A. 1965. "Fuzzy sets" [in en]. *Information and Control* 8, no. 3 (June): 338–353. ISSN: 0019-9958, accessed August 10, 2021. https://doi.org/10.1016/S0019-9958(65)90241-X. https://www.sciencedirect.com/science/article/pii/S001999586590241X.

Zhang, Tongda, Xiao Sun, Yueting Chai, and Hamid Aghajan. 2015. "A look at task-switching and multi-tasking behaviors: From the perspective of the computer usage among a large number of people." *Computers in Human Behavior* 49 (August). https://doi.org/10.1016/j.chb.2015.03.012.

Appendix

A.1 Design Personas

A.1.1 Persona 1: Hans Muller, Software Engineer

Name: Hans Muller
Occupation: Software Engineer
Demographic: 30 years old; Lives in Zurich, Switzerland; Mainly uses their computer through out the work day; Rarely closes tabs
Goals and Needs:
Hans uses the internet a lot in his job. Many tools needed for his job are web tools and more often than not he looks for more elegant solutions to his problems on the internet. Naturally he has to manage all the different tabs he opens and closes. For his everyday web tools he has set up bookmarks in the web browser so he can navigate to those sites quickly. Often he gets overwhelmed and just closes all the tabs at once at the end of the day and starts from scratch the next morning. Other times he spends a lot of time looking for a tab he had found the day before in his open tabs and in his browser history.
Pain Points:
The main pain point for Hans is that he cannot properly control all the open tabs without investing a lot of time he would rather invest in coding and fixing bugs. Generally he would like to spend less time in the browser searching for things and more time in his coding environment.
Relevant Patterns of Behavior:
Hans mainly works on their computer. About 50% of the time spent on the computer is currently spent in the web browser, checking the ticketing system, build-status of the software he maintains, searching for more elegant solutions to the problems he tries to solve and looking at funny videos of cats. He knows his way around the browser and has personalized the bookmarks and the start up pages.
Personality:
“I love coding and writing good code. But search for better solutions than for all the problems I solve throughout a day can get very time consuming and I end up browsing for hours. I would love to focus more on coding.”

A.1.2 Persona 2: Maria Gruber, Product Manager

Name: Maria Gruber
Occupation: Product Manager
Demographic: 27 years old; Lives in Vienna, Austria; Gets a lot of calls and has a lot of meetings Often get interrupted and has to solve a problem quickly
Goals and Needs:

Maria is the central contact for anything concerning the product she manages. She often gets interrupted when working on a task and has to be able to switch to a new problem quickly. Resuming a task can cost a lot of time and cognitive energy. For the majority of her tasks she uses Google to find phone numbers, regulatory information for different countries or other information relevant for her task at hand.
She is very structured in her work and she manages her tabs in the browser very carefully. She uses different windows for different tasks and closes them rather quickly after a task is completed.
Pain Points:
Even though Maria manages her tabs she still sometimes needs to spend a lot of time to get back into a task, because she has to a) search for the window related to the task and b) then has to find the relevant tabs in the window. Sometimes it also happens that she closes a window too early and she has to either go through her browsing history of the last few days or start her search all over to find the relevant information again.
Relevant Patterns of Behavior:
Maria uses her web browser a lot to find information relevant to her tasks. However she often has to work on multiple tasks at once. Most of the tasks do not have much in common so switching between them comes with a significant cognitive load. She knows how to search for things and the basics of a web browser, however she is not too familiar with all the functionality available.
Personality:
"I love to keep my brain occupied and get things done throughout the day. I try to work very precisely and try to solve every single task with a lot of care. For this I need to have the full picture of a problem and all the information possible."

A.1.3 Persona 3: Felix Jung, Student

Name: Felix Jung
Occupation: Student, Biology
Demographic: 19 years old; Lives in Munich, Germany; Very tech savvy; Loves to learn new things
Goals and Needs:
Felix recently started their studies and most of the lectures happen online. He get introduced to many new topics on different e-learning platform and every lecture provides their materials in a different form. To keep track of all the different places and topics in the web browser, Felix sets up bookmarks and structures them in folders to be able to quickly find all the resources again. This works well for him, however managing all these bookmarks becomes tedious over the semester. Resources relevant at the beginning of the semester become less relevant and become mixed with more relevant resources in his bookmark folder structure.
Pain Points:
A main pain point for Felix is the management of his bookmarks. Just from the name of a bookmark he is not able to determine its relevance and he would have to read through all the bookmarked pages again to filter them.
Relevant Patterns of Behavior:
Felix organizes his web resources very well, however over time his groups / folders grow to a size where he himself cannot tell relevant from irrelevant resources anymore.
Personality:
"I love to learn and I do this in a very structured way. But often there is so much to learn that I lose track of some things."

A.2 Source Code & Repositories

The source code for Taboor is currently stored in a private repository on GitHub and can be made available up on request. Please contact us at: dimitri.kohler@outlook.com

The source code for the study website and visualizations can be found here: https://github.com/dikohl/taboor_vis

A.3 Study Introduction

What is taboor:

Taboor is meant to help you organize your tabs simply via a sidebar in your browser.

Through out the work day, many different tasks can happen at the same time and for every task you may need to visit multiple sites to complete it. Often people just keep tabs from many different tasks open for later use and to not lose any information. This makes it harder to keep an overview and requires a lot of energy to manage.

Taboor tries to provide you with functionality to make organizing and re-finding tabs easier as well as to help you minimize the risk of you losing information.

Study Design:

This study takes two weeks.

Week 1: Extension installed, but NOT used.

Week 2: Extension enabled, try to use the extension in your daily workflow.

At the end of every week you are asked to download a log file from the settings page and submit it to the researchers. This file contains anonymized usage data on your browsing behaviour and how you used the extension (more below).

After the full two weeks there will be a short (15 min) interview where you are asked to give feedback and your thoughts on the extension.

Usage Log Data:

nodes: A list of random ids representing all the visited pages
i.e. "[id], 1, 2, 3, 4".

edges: A list of switches between visited pages / nodes (source and target) and when they happened
i.e. "[source, target, time], (1, 2, 25.6.2021 12:00:01), (2, 4, 25.6.2021 12:05:33)".

numOpenPages: A list of how many tabs were open after a tab got closed or opened
i.e. "[number, time], (7, 25.6.2021 13:10:01), (8, 25.6.2021 13:11:10)".

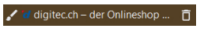
stashUsage: A list of when a marked page was added, removed or opened
i.e. "[id, color/name, time, type], (7, #ffffff, 25.6.2021 13:10:01, added), (8, #000000, 25.6.2021 13:11:10, opened)".

highlightUsage: A list of whenever highlight was opened (with it's related parent page)
i.e. "[parentId, id, time], (2, 7, 25.6.2021 13:10:01), (8, 5, 25.6.2021 13:11:10)".

searchUsage: A list of when the search was used
i.e. "[0, time], (0, 25.6.2021 13:10:01), (0, 25.6.2021 13:11:10)".

A.4 Extension Tutorial

Taboor is a Tab Manager for your web browser trying to help you to better organize all the web pages you visit through out your day.
The extension adds a sidebar to websites that can be opened by clicking on the grey bar on the bottom left on the window.
It's core features are:

Feature	Icon / Button	Actions
Mark pages (or groups) with a color quickly, to open and work on them again in the near future.		Hover
Create **named groups** of pages you use frequently.		Left Click & Enter
Highlight useful **pages** from your history that might be useful for the task you're working on.		Left Click: Show Highlights **Double Click:** Open Page

Marked Pages

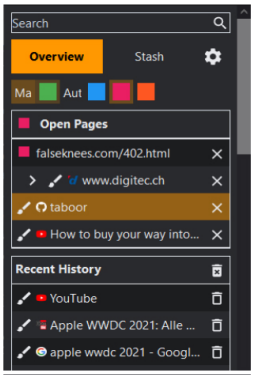
Marked pages are meant to be an easy way to remeber pages for use later in the day / week, without having to keep them open in an extra tab. You can mark single pages, mulitple pages in dropdowns or all the open pages at once by hovering over the brush icon / the colored square. After 3 days of not being used (depending on your settings) marked pages are recolored and marked with black. This way marked pages are cleaned up automatically, without getting lost.

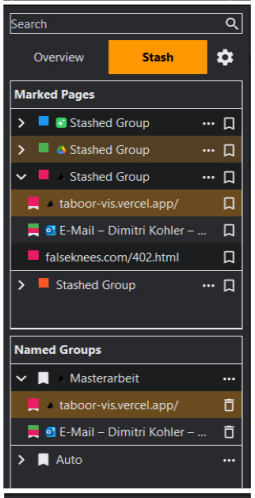
After you have marked pages you can interact with the different groups in three ways:
1. In the Overview, through the Quick Bar. Here you can open or close marked groups.
2. In the Overview, by right-clicking on a marked (or named) page.
3. In the Stash, where you see all the marked pages grouped by color.

The right-click menu generally provides functionality to open or close tabs:

Open	Open the page(s) in the current window as a new tab.
Open Window	Open the page(s) in a new window.
Close	Close the page(s) in any window.
Close Others	Close all pages not from this group.
Focus	Open all pages of a color and close all others.

In the Stash, marked pages or groups can be converted in to a named group by pressing the empty ribbon.



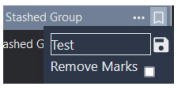


Named Groups

Named groups can only be created from colored pages or groups in the stash. named groups are a more long-term and uniquely named version of groups. Pages in a group are marked with a white ribbon below the colored square.

After naming a group, the functionality remains similar to marked groups.
These groups are less prone to change and more unique compared to marked groups.

As mentioned earlier the naming input can be opened by clicking on the empty ribbon on the right of a page or group in stash.
There is also an option to remove marks from pages that are added to a named group.



To add a page or marked group to an existing named group, you can just enter the same name as the existing group in the dialog and press save.

Highlights

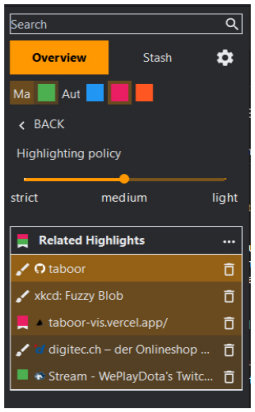
Highlights are meant to help to create useful groups and to re-find information related to a page.
These highlights are accessed by a single left-click on any page in the Overview.

Since some pages are more relevant than others, pages are presented with background of different shades.
The amount of pages shown depends on the highlighting policy that can be changed on the fly.

The Highlighting Policies work as follows:
- Strict: Show only related pages
- Medium: Show likely related pages
- Light: Show possibly related pages

Related pages can also be temporarily removed to create a more useful group of pages by clicking on the bin icon.

After creating a useful group, all the pages can be marked at once via the marking functionality!
If only a single page fits in to a color, pages can also be marked individually.



A.5 Extension Example Scenarios

To guide you through all the tools available to you, we created two scenarios showcasing a switch between two tasks and creating a group of pages to re-find relevant pages.

Feel free to come back to these scenarios while working with the extension.

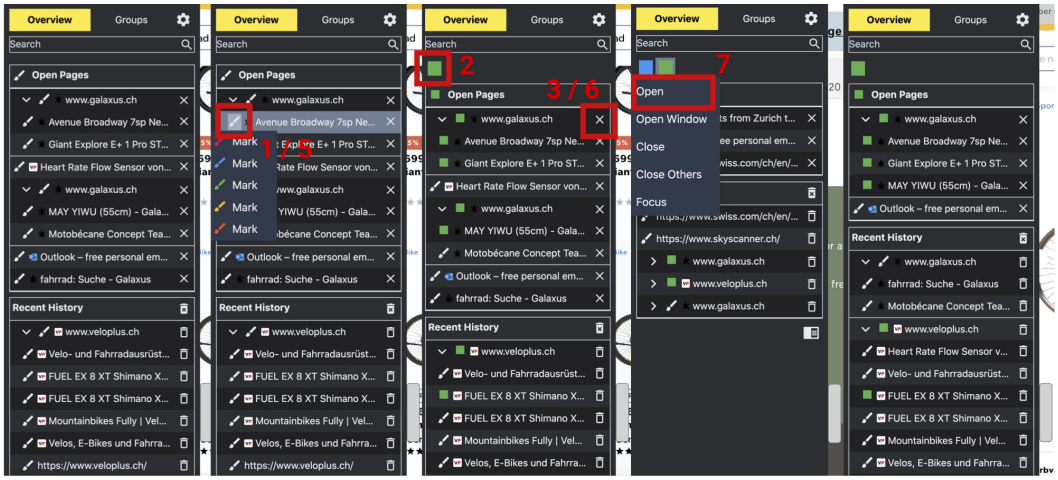
Scenario 1: Switching Tasks

You are looking up product information on different bicycles, when you get an urgent call requiring you to find information on possible flights to London for tomorrow. You don't want to lose the research done on bicylces but also want to clean up your open tabs to not lose the overview.

1. **Mark all open pages** containing information about bicycles you want to keep with GREEN.
2. Click the **GREEN square** at the top and click "**Close**".
3. Any other pages you don't need anymore can be closed manually.
4. Now you can work on finding the flight you need as usual.
5. You can mark interesting or important pages with BLUE, to find them again later.
6. After you are done with looking at flights you can **close all tabs related to the flight**.
7. To resume your inital task, you can again click on the **GREEN square** and click "**Open**" or "Open Window" to open all the pages about bycicles in a separate new window.

P.S. If you want to fully focus on GREEN you can also choose "Focus" to open all GREEN pages and close **all** other pages.

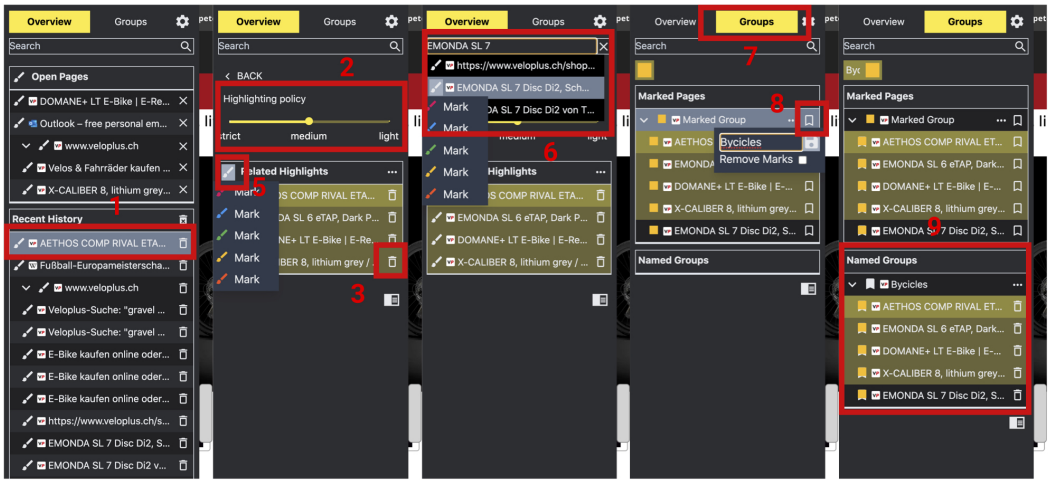
If you do not remove the BLUE marks in the groups-view, they will be cleared after 3 days (depending on your settings) and all the pages will move to a BLACK group.



Scenario 2: Refind important pages and persisting them

You want to create a group of pages that contains all relevant pages on bycicles and don't want to look through the whole history.

1. **Left-Click a page** about bycicles in the open pages or in the history to see related pages
2. Try different highlight policies and see which one contains the pages you want
3. **Remove pages** from the group that don't fit
4. **Double-Click** one of the pages if you want to **visit** the page.
5. **Mark** the whole group with YELLOW.
6. If you know specific pages that are missing, you can **search** for them in the search and mark them as YELLOW there.
7. Go to "**Groups**"
8. Click the **empty ribbon** of the yellow group, enter a **name and save**.
9. Now you have persisted a group of pages that will not be removed after 3 days.



A.6 Interview Questions

Generally about their strategies:

1. What are your strategies to manage and keep track of your websites?
    - Usage of Bookmarks?
    - Usage of multiple Windows?
    - Usage of Extensions?
2. Do you actively organize your open tabs? If yes, how much time do you think you spend per day to do so?
3. What do you do if you need a website again that you had previously closed? Where do you look?
4. (If not specific enough) More specifically, do you save websites which were helpful to solve a task somewhere? Why or why not?
    - Where? (comments, bookmarks, documentation file, notebook)
    - Do you think that it could be useful to persist/store them for future reference?
    - What benefits of saving links can you foresee?
    - How would it be best to do so?

About the extension:

1. Was the extension easy to use and intuitive? Did you have any problems?
2. Could you please describe to me how you managed your websites with the extension? Compared to before.

Impressions on the Views - data shown, not functionalities (show the extension):

1. What do you think of the use of a sidebar in the browser?
    - Would you prefer a different position on screen? Where?
    - Do you think it should be bigger / smaller?
2. What do you think of the Views provided by the Extension?
    - Open Tabs
    - Recent History
    - Highlights
    - Marked Pages
    - Named Groups
3. Can you think of another view that would be useful to have?
4. (optional if not specific) How did you use the different views? Were there any specific problems where a view was especially helpful?
    - Were you able to quickly re-find pages you might have closed before?

- How many minutes / hours / days back do you normally have to go to re-find a page? (very recent or multiple days back?)

Impressions on the Functionality:

1. What do you think of the Functionality provided by the Extension? Were any especially useful?
   - Marking Webpages
   - Highlights
   - Search
   - Quick Access Bar
   - Named Groups
   - Recent History
2. Were there any problems where the functionalities were especially helpful?
   - Did you need to adapt your workflow a lot to incorporate them?
   - Did using the extension distract you from your task at hand?
3. Was there any situation where a functionality was not working as you would have expected it?
4. Was there anything missing that would have helped you to manage your tabs better?
5. (Specific question depending on the usage data gathered)
   - You used X a lot, how did you use it, when was it especially useful?
   - You stopped using X after a few tries, what was the problem?
   - ...

More specific question about certain functionality.
Marked Pages:

1. How did you integrate marked pages in your workflow? Was it a big change?
2. What was your strategy for marking pages?
   - Did you assign a certain meaning to certain colors?
   - Have you encountered situations where having more than one color for a page was useful?
3. At what point did you choose to add a marked page to a named group?
   - How did you named groups come to be? Every page One-by-one or all pages at once?
4. What did you do when you didn't need a marked page anymore? (task completed or page not useful)
   - Did you leave pages marked until they were automatically cleaned up or did you manually remove markings?
   - How do you like the functionality that unused marked pages lose their marking after a certain amount of time?

5. What do you think of the concept of having a way to temporarily mark / save some pages?

Highlights:

1. Have you encountered a situation where highlights / related pages were useful?
    - Did the highlights help you to re-find important pages you might have missed?
    - Did the highlights help you to create useful groups of marked pages quickly?
2. Have you tried the different highlighting policies?
    - How would you rate their accuracy from 1 to 7 (strict / medium / light)?
    - Did you find one of them to be especially useful?
3. Do you think that there are any criteria that are especially important for good highlights (so far, we used a frequency, duration and title similarity approach)?

Quick Access Bar:

1. How did you integrate the functionality in your workflow? Was it a big change?
2. Were all the options to open / close colors and groups useful?
    - Which one did you use the most?
    - What other functionality would you expect?
    - Would you remove any functionality?
3. Do you think the quick bar helps you to organize you tabs more easily?

In general, Ease of Use and Ease of Adaption:

1. Do you think you were more likely to close a tab because it would be easy to re-find and easily accessible?
2. Do you feel like you were able to save time with the extension?
3. Do you feel like it took more or less effort to manage your tabs with the extension?
4. Is there anything else that you would improve or change?

Future use:

1. Could you imagine using this tool (or an improved version of it) in your daily work to manage your tabs? - Why or why not?
2. Where do you see the highest value for an approach similar to ours?

Demographic Questions:

1. What is your job? What is your job role? (Individual Contributor, Lead, Other)
2. What browser are you using?
3. How many hours per day do you spend working in the browser (at work)?
4. What gender do you identify with? (Female, Male, Other, Prefer not to Say)
5. May I ask what your age is?

Wrap up:

1. Is there something else that you want to mention about the extension or the topic in general?
2. Do you have any feedback with regards to this study?