DOBBS: Towards a Comprehensive Dataset to Study the Browsing Behavior of Online Users

Christian von der Weth, Manfred Hauswirth
Digital Enterprise Research Institute (DERI), National University of Ireland, Galway (NUIG)
Email: {christian.vonderweth|manfred.hauswirth}@deri.org

***Abstract*—The investigation of the browsing behavior of users has been a topic of active research since the Web started. However, new online services changed the meaning behind "browsing the Web" and require a fresh look at the problem. Platforms such as YOUTUBE or LAST.FM have started to replace the traditional media channels (cinema, television, radio) and media distribution formats (CD, DVD, Blu-ray). Particularly social networks (e.g., FACEBOOK) attracted whole new, particularly less tech-savvy audiences. Advances in mobile technologies made browsing "on-the-move" the norm and changed the user behavior, often being influenced by the user's location and context in the physical world. Commonly used datasets, such as web server access logs or search engines transaction logs, are inherently not capable of capturing the browsing behavior of users in all these facets. DOBBS (DERI Online Behavior Study) is an effort to create such a dataset in a non-intrusive, completely anonymous and privacy-preserving way. DOBBS provides a browser add-on which keeps track of users' browsing behavior. In this paper, we outline the motivation behind DOBBS, describe the add-on and dataset, and present some first results to highlight the strengths of DOBBS.**

I. INTRODUCTION

Since the advent of the World Wide Web getting deeper insights into the browsing behavior of users has been a major research field. While such information enables assessing the popularity of websites, they also provide useful information for improving the design and usability of websites, designing and implementing of new functionalities for web browsers, or advancing the ranking algorithms of search engines. A lot of work has been published on user browsing behavior. However, recent changes in the landscape render some of the assumptions used in these studies no longer valid and the changed setup requires a fresh look on the problem:

(1) Passive browsing. Originally, browsing the Web was mainly the active task of searching for information. With today's bandwidth resources and new kinds of online platforms, this has changed significantly. Platforms like YOUTUBE, NETFLIX or LAST.FM allow users to watch video clips or movies or listen to online radio. It has been shown that more and more users prefer these new sources over traditional ones such as television or radio, or traditional media formats such as CD, DVD, or Blu-ray. Given these trends, browsing the Web has become more and more a passive activity where users visit a website but not necessarily interact with that site. Users might even leave their computer to do completely different tasks, e.g., cooking or cleaning while continuing to listen to online radio.

(2) New web technologies. Technologies like Ajax [8] or WebSockets [4] provide methods for updating content on a page without reloading that page by requesting only small bits of information from a server. The involved benefits – particularly the reduced bandwidth consumption and making web application behave more like desktop applications – spurred the adoption of such technologies by many popular websites. Updates might be explicitly invoked by users or done automatically. A typical application is a stock ticker on a website showing the latest stock prices and updating all prices once a minute. Again, many websites make users digest information in a rather passive fashion.

(3) New browsing technologies. While a lot has been done "under the hood" – for example, the support of web standards, media formats, etc., or performance optimization techniques like caching of web page content – web browsers also have improved with respect to their usability. A very popular feature is tabbed browsing, i.e., the support of multiple tabs within a browser window. Together with multiple browser windows, tabbed browsing allows users to arbitrarily parallelize their browsing activity. This is particularly common when combining active and passive browsing activities. For example, a user can search for information while listing to online radio in a background tab and occasionally check the latest, automatically updated stock data in a second browser window.

(4) Evolving Web demographics. Social network and social media sites, micro-blogging sites such as TWITTER, the omnipresence of online shops, online browser games, etc. contribute to different emerging trends: Firstly, they attract new groups of users including less tech-savvy of users that previously were less inclined to frequently use the Web, if at all. Secondly, as studies show, particularly social network sites have strong sociological impact indicated by the increasing time users spend online and how they arrange their social life accordingly. Hence, for many users, browsing the Web means socializing with other people over the Internet.

(5) Browsing while mobile. With the advances in mobile technologies people are able to browse the Web almost everywhere. We argue that mobile browsing typically differs from browsing the Web at home or at work. Firstly, while being mobile, e.g., as a traveler or commuter, browsing is typically only a sideline activity, resulting in short online sessions (e.g., getting the latest news or writing a tweet). Secondly, and more interestingly, the actual browsing task is often being influenced by the user's location and context in the physical world. For example, a passenger on a train might want check the schedule of follow-up connections, or a traveler might want to retrieve online information about the sights she or he is visiting.

This research has been supported by the European Commission under Grant No. FP7-287305 (OpenIoT) and Grant No. FP7-287661 (GAMBAS) and by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-II) and Grant No. SFI/12/RC/2289 (INSIGHT).

978-1-4799-2902-3/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.8

These new characteristics are not well understood so far. This is largely due to the lack of meaningful datasets capturing user behavior. DOBBS (DERI Online Browsing Behavior Study) is an effort to create such a comprehensive and representative dataset in a *non-intrusive*, *completely anonymous* and *privacy-preserving* way. The heart of DOBBS is a browser add-on that users can install, and that logs all major events in the context of browsing the Web, such as the opening of new tabs, the loading of new pages, the clicking on bookmarks, and many more. All data is sent to a central repository, and the resulting dataset is made public. Logging user behavior always raises privacy concerns, and DOBBS addresses these concerns in a very pragmatic manner to protect the privacy of users. The add-on does not capture, store or send any personal data, such as e-mail or IP addresses, and encrypts all sensitive data before they are sent to the server.

Having deeper insights into users' browsing behavior allows scientists from a large variety of disciplines to derive both fundamental and applied knowledge from different perspectives; most importantly:

(1) The information how long and how active users visit a website potentially helps to improve ranking algorithms of Web search engines. For example, a site that typically resides in a background tab might be considered differently than sites that typically involve more active user interaction.

(2) Knowledge about browsing behavior enables the development of new features that improve the online experience of users. Examples are the hiding of tabs containing passively used web pages including a quick access to them, or the automatic rearranging of tabs according to their usage.

(3) From a technical perspective, new optimization approaches are conceivable. This might include special "idle modes" for browsers or individual tabs where dynamic pages are not automatically updated if, e.g., the tab is in the background or the user recognized as inactive.

(4) Understanding how and when users browse the Web facilitates deriving statements on *why* they use it (e.g., for information seeking, entertainment, or socializing). This in turn provides novel insights into the sociological impact of the Web – that is, how the Web and "being online" shapes peoples' life.

II. Related Work

Server-side studies. One type of data source are server access logs [6]. The results [13] show that users often exhibit different behavior patterns, rather than a single one, when browsing for information. [10] collected the server accesses on a university router, confirming a long-tailed distribution in site traffic. A second type of server-side data sources are search engine transaction logs. [2] investigated how user behavior, derived from click streams recorded within the MSN search engine, can be used as implicit feedback to improve the ranking of query results. The authors of [3] analyzed an AOL query log and found that certain topical categories can exhibit both short-term and long-term query trends. Using query logs of the Yahoo! search engine, [11] found that after a few hundred queries a user's topical interest distribution converges and becomes distinct from the overall population.

Client-side studies. Collecting data on the client side, in general, requires users to install browser extensions (or use special browsers) that log all user actions. [5] focused on demographic factors, i.e., how age, sex, race, education, and income affects how long and which sites are visited – with respect to the five most visited categories: social media, e-mail, games, portals, search. [9] investigated how the browsing behavior of users depends on the current task they are performing (e.g., fact finding, information gathering, etc.). In [1] the authors identify twelve different types of revisitation behavior, based on which they outline recommendations towards web browser, search engines, and web design. [12] highlights the typical use of parallel browser windows or tabs as means to navigate between pages and that different users typically show very characteristic behavior. [14] investigated tabbed browsing on users' browsing behavior.

Summing up, server-side collected data typically suffer from insufficient information when it comes to investigating the online browsing behavior. Controlled/supervised studies conducted in a lab under time constraints are limited to investigate user behavior while solving a specific task but typically do not elicit the normal behavior. Closest to our approach is the Web History Repository Project [7] which also features a browser add-on[1] to capture browsing events.

III. The DOBBS System

The DOBBS system uses a browser add-on[2] that captures browsing events and sends them to a central server. Browsing events comprise, for example, adding/closing of tabs, loading of web pages, window status changes, and user activity changes. The sending of logged data is done via HTTP POST requests to the DOBBS backend server. The sole functionality of these scripts is to write the logging data into a database. In the following we describe in detail the basic design decisions and the browsing events the add-on tracks, and discuss the applied means to preserve user privacy and the limitations of DOBBS.

A. Basic Design Decisions

The main goal regarding the implementation was to make the add-on an "install-and-forget" application, i.e., once being installed the add-on runs silently in the background. The rationale is that users are not constantly reminded of the add-on, which might affect browsing behavior and their willingness to share data. Users might get second thoughts on contributing if they reconsider which websites they visited, despite all efforts to preserve users' privacy. To make the add-on as unobtrusive as possible, we deploy the following two concepts:

Immediate logging. The add-on sends each recorded event immediately to the server. Alternative solutions would involve storing information about events (temporarily) on the client side, and sending the set of event as a bulk to the server. Pursuing similar goals as DOBBS, the Web History Repository Project [7] accumulates all newly recorded data until the user explicitly sends the data via clicking a button. While this gives users full control whether data is sent or not, it also kind

[1] http://webhistoryproject.blogspot.ie

[2] The add-on is available on the project website (http://dobbs.deri.ie).

of interferes with users' normal browsing routine, regularly reminding him/her on the running logging process.

Best-effort logging. The add-on does not feature a specific error handling, and as such does not show any warnings or error messages. Due to its simplicity, the only relevant error that can occur refers to the unsuccessful sending of logging data to the server in case of network connection problems. In this case, the add-on simply ignores this unsuccessful attempt, discards the data, and continues trying to send subsequent events as if nothing happened. With this approach we addressed the trade-off between handling all exception and the degree of complexity of the add-on in favor of the latter. We deem this design decision reasonable since users, if they lose their Internet connection, they are unlikely to continue browsing.

B. Privacy Preservation

Naturally, logging user behavior in such a detailed fashion raises privacy concerns, potentially discouraging users to contribute. We therefore have addressed this issue with great care to preserve the privacy of participants:

(1) Anonymization. Participants are only identified by a randomly generated integer value, without any connection to their real-world identities, during the installation of the add-on. No information that may point to the real-world identities of participants, such as IP addresses or explicitly requested email addresses, are ever collected or transferred to the server.

(2) Encryption. All sensitive data – that is, the URLs (and its components; see Section III-C) of the web pages the participants were browsing on – are first encrypted on the user side and then sent to the server. The DOBBS add-on applies the SHA-1 algorithm for encrypting the data.

(3) Manual control. A participant can manually stop the logging process at any time. The add-on adds a menu entry in the "Tools" menu of Firefox suspend and resume the logging. Additionally, the add-on respects Firefox's Private Browsing[3] mode, i.e., the logging is suspended during private browsing and is resumed after leaving that mode again.

(4) No keystrokes are logged. No explicit user input is in the scope of the logging process of DOBBS. This includes additional browser input fields, e.g., the optional search field in the toolbar, but particularly any kind of form fields embedded in web pages, e.g., for user names or passwords.

(5) Anonymized contact. Users can provide comments or feedback or ask questions. For this, the project website not only lists a contact email address, but also provides a dedicated contact section. There, users can leave comments or question without providing a (real) name and email address.

C. Recorded Events

The main unit of information within DOBBS is an event. The dataset distinguishes between *window events*, *session events*, and *browsing events*, which we describe in the following. Beside event-specific attributes, all event types share a set of common attributes (see Table I). The complete list of all recorded events and their representations within the dataset can be found on the project website (http://dobbs.der.ie).

Attribute	Description
`time`	time on client side when an event has occurred
`tz_offset`	difference between UTC time and client time, in minutes (e.g., for GMT+2, `tz_offset` = -120)
`user_id`	unique numeric identifier of a user, randomly generated during the installation of the add-on
`window_id`	unique numeric identifier of a browser window, randomly generated at the time of opening
`session_id`	unique numeric identifier of a logging session, randomly generated at session start
`tab_id`	numeric identifier of an open browser tab, unique within each browser window

TABLE I. CORE ATTRIBUTES THAT ARE LOGGED FOR ALL EVENTS

Window events. Window events encompass all events that are associated with interacting with a browser window, e.g., the opening and closing of a browser window or individual tabs within a window. Both windows and tabs feature a unique identifier. The add-on captures any change in the state of a window (maximized, minimized, normal, full screen). The add-on also keeps track if a browser window lost focus, i.e., became a background window on the user's desktop, or regained the focus again.

Session events. A session denotes the interval in which all occurring events are recorded and sent to the server. Users can also end a session explicitly or implicitly by entering the Private Browsing mode. Analogously, switching the logging process back on or leaving the Private Browsing mode initiates a new session. Each session features a unique identifier. We also consider the activity state of a user (*active* or *inactive*) as a session event. As a design decision, we consider a user as inactive if the user was not active for at least one minute.

Browsing events. Browsing events are associated with navigating between web pages. This includes when a new page has been loaded in the active or a background tab, but also the state of visibility of a web page. A page becomes visible after it has been loaded in the active tab, or the background tab containing the page becomes the active one. Analogously, a page becomes invisible before it is unloaded in the active tab, or the tab containing the page is moved in background. The add-on also captures the cause of the events (e.g., click on link or bookmark). Browsing events carry the information about the visited pages. However, to encrypt the full URL would significantly reduce the possibilities regarding the logging data analysis. For example, it would be impossible to evaluate how long a user visited a specific domain. To address the trade-off between privacy preservation and the possible insights into browsing behavior, we distinguish four different "levels" for each URL: the domain, the domain and

Level	Example
domain	`example.org`
(sub-)domain	`topic.example.org`
full path	`topic.example.org/dir/index.php`
full URL	`topic.example.org/dir/index.php?id=42`

TABLE II. CONSIDERED COMPONENTS OF A URL

[3] While in the Private Browsing mode, no sensitive browsing information are stored, including visited pages, entries in text boxes or search bars, new passwords, cookies, etc.

all subdomains, the full path, and the full URL itself. Table II shows an example. The add-on encrypts each component individually before sending the complete record to the server.

D. Technical Limitations

Getting the full picture of users' browsing behavior requires the correct capturing of all events and their successful transfer to the server. Different situations, however, can occur that involve a loss of data but are beyond our means to avoid them.

Network problems. The logging process requires that recorded events are sent to the DOBBS backend. In the case of connection problems the sending fails. As already motivated, we consciously refrain from specific error handling but send logging data in a best-effort manner. We argue, however, that the effect of a loss of recorded events due to connection problems on the logging data is rather low. Firstly, network infrastructure is in most areas quite reliable, thus significantly lowering the probability of unexpected loss of connection. And secondly, if users are disconnected from the Internet they are not likely to continue browsing anyway.

Browser errors. Obviously, the functionality of the add-on depends on the functionalities of the browser, which is out of the scope of our efforts. In rare and (for us) undetermined situations, Firefox might crash completely which essentially translates into its unexpected termination with the effects on the logging process as described in the next paragraph. However, we also encountered some more subtle errors – only indicated by error messages on the terminal – that do not lead to a complete crash but affect the functionality of the add-on. For example, we observed that after Firefox throwing a specific error, the events referring to the switching between tabs were temporarily not fired. These occurrences are, however, extremely rare and have, to the best of our knowledge, only a very limited effect on the logging.

Unexpected termination of a browser window. The event that a user closed the browser window only fires correctly if a user explicitly closed Firefox. A failure also affects all derived events: the unloading of pages, the closing of open tabs, and the ending of the current session. Besides the situation that Firefox crashes, users can manually shutdown Firefox so that the window closing event is being fired. A user can terminate the running process of Firefox (e.g., sending the SIGTERM or SIGKILL signal from a terminal on UNIX/Linux-based systems). Although we cannot avoid this behavior we deem it very unlikely. A much more common behavior, however, is that users shutdown their computer without closing Firefox beforehand. Since a shutdown involves sending the SIGTERM signal to all still running application, Firefox again closes without the window closing event being fired.

We deem the loss of window closing events as most critical since we expect it to be the most common error case. To deal with this problem two basic approaches are conceivable. One solution is to filter out all sessions for which no session ending event has been recorded, before proceeding with the data analysis. While our current observations show that this is only the case for a rather small number of sessions on the whole, it potentially excludes data from participants that mostly or always shutdown the computer without closing Firefox. An alternative solution is to estimate the time a window has been closed. The most straightforward way to estimate the time is to use the time last recorded event associated to a session, e.g., the time of the last page load, optionally with some offset.

IV. Preliminary Results

DOBBS is a rather recent initiative and is still building momentum in terms of getting participants to contribute. Thus, significant results based on a large user basis are not available yet. We therefore aim to highlight the strengths and potential benefits of DOBBS compared to traditional available datasets.

Browing in parallel. All modern browsers allow multiple browser windows and support tabbed browsing. The fundamental difference is that multiple browser windows facilitate viewing of different web pages at the same time. Figure 1 shows the usage of multiple windows for the five longest participating users, Figure 2 for the usage of multiple tabs. In both figures the number on the x-axes refer to the same user in both figures (i.e., the bars for each user are directly above each other). Figure 1 shows that User 1 is browsing most of the time with one browser window. Only ∼5% of the time s/he is using at least two windows in parallel (and almost never three or more). Regarding User 1's usage of multiple tabs, ∼78% of the time s/he has at least two parallel open tabs, ∼40% of the time at least four, ∼18% of the time at least eight, and ∼1% of the time at least 16 parallel open tabs. The results show, even given the small sample size, that the behavior of users with respect to parallel browsing can vary significantly.

Browsing the Web is not everything. The DOBBS add-on leverages two event mechanisms of Firefox to explicitly quantify the time users do not actively browse the Web: a user's explicit time of inactivity, and the time a browser window is in the background. Furthermore, the data also allow us identifying phases of inactivity implicitly by the prolonged absence of any new event. To illustrate this, Figure 3 shows the average ratios for (a) the explicitly observed inactive time of users, (b) the implicitly calculated inactive time of users, and (c) the explicitly observed background time of browser windows. It seems that idling during an on-going browsing session is very common, indicated by three measures. However, the measures are not necessary closely related. For example, the case that the average explicit idle time exceeds the average background time indicates that users stop browsing but do not switch to another application, but, e.g., just sitting back to watch a video clip. Which measure to use for quantifying the activity of users depends on the research question behind an analysis.

What users are (really) interested in. DOBBS provides detailed information about (a) the *loaded time*, i.e., the overall time that pages of a domain were loaded in the browser either in an active or in an background tab, (b) the *display time*, i.e., the time pages of a domain were actually visible because they were in the active tab of a browser window that had the focus, and (c) the *viewing time*, i.e., the part of the display time during the user was considered active. The viewing time may comprise multiple individual views, e.g., switches between tabs. To illustrate this, Figure 4 shows the distribution of the three measures for a browsing session of a user grouped by the domain of the visited web pages. The number above each bar represents the number of page loads with the same domain. It is easy to see that, loaded time, display time, viewing time, as

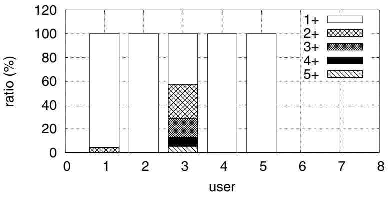


Fig. 1. Usage of parallel open browser windows for different users.

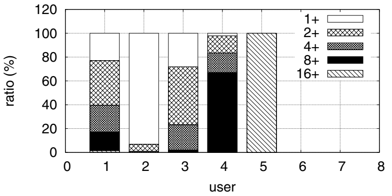


Fig. 2. Usage of parallel open tabs within individual browser windows for different users.

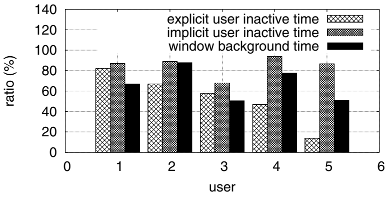


Fig. 3. Explicit and implicit inactive time, and the window background time for different users.

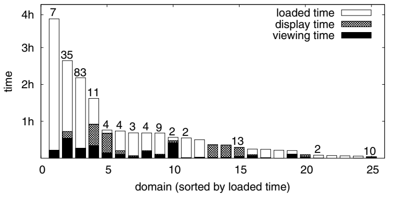


Fig. 4. Distribution of times a domain has been loaded, displayed and views for a single session.

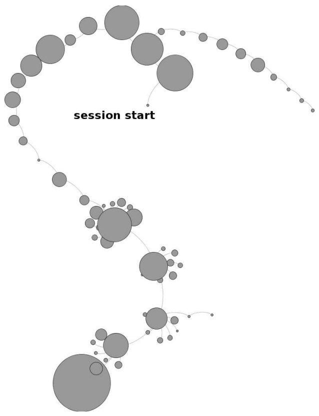


Fig. 5. Graph representation of a browsing session. The size of a node reflects the duration the user spent on the corresponding web page.

well as the number of visits typically induce different rankings. Furthermore, one can consider combining individual measures to derive new ones. For example, the ratio between viewing time and displaying time might represent a good indicator how "absorbing" a website or a web page is – again, as Figure 4 also indicates, this ratio would induce a different ranking. Having these different measures to quantify the popularity of a web page actually broadens the notion of popularity. Again, which measure to apply eventually will depend on the kind of research questions motivating an analysis of the dataset. For example, while advertisers might mainly be interested in the viewing time, the frequency of visits is interesting for web server administrators from a performance point of view.

Following the footsteps of users. The granularity of the DOBBS dataset allows retracing each navigation step users during browsing sessions. This particularly refers to the usage of multiple tabs with one browser window. With page loads and tabs as the two dimensions to specify the browser usage, Figure 6 gives examples for the four different cases derived according to these two dimensions. Data points represent page loads. Points on a horizontal line indicate new page loads in the same tab; diagonal lines represent new page loads in a new browser tab originating from the currently displayed tab. Figure 6(a) shows the two cases where users do not use tabbed browsing: navigating from one page to another using the same tab, or simply open the browser for a single page load. In Figure 6(b), a user used individual tabs for each page load (here, four tabs directly after opening the browser window). Finally, Figure 6(c) shows a session of a user who regularly opened new pages new tabs mostly (but not always) originating from the first tab.

Alternative graph representations are also conceivable. To give an example, the graph in Figure 5 shows the same browsing session using a "traditional" representation, where the size of the nodes reflect the loaded times of the different pages (note that the representation can reflect various characteristics describing a session). Such visualizations may provide the basis for sophisticated graphical user interfaces with which users can more easily and more intuitively overview and navigate through their past browsing history. Using a graph-based representation, the browsing behavior can be depicted as a directed tree with the root being the startup of the browser. Analyzing these trees to, e.g., categorize different types of parallel browsing, requires appropriate measures. These can be straightforward measures such as the average number of page loads per tab or more sophisticated approaches applying graph-based measures such as the outdegree of nodes, the depth of the tree, the average shortest path from the root, etc. Table III lists the values as calculated for the same session shown in Figure 6(c) for various measures. Again, the choice of the measure(s) will largely depend on the specific set of research questions to be answered through an analysis of the dataset.

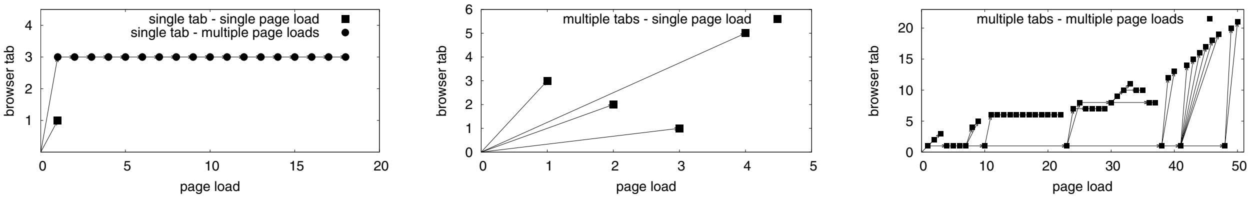
Measure	Value
number of opened and used tabs	21
number of page loads	50
(number of tabs) / (number of page loads)	0.42
number of focus changes	77
diameter of graph	18
average path length	5.8
maximum outdegree	7
modularity	0.727

TABLE III. DIFFERENT GRAP-BASED MEASURES FOR CHARATERIZING A BROWSING SESSION.

V. LESSONS LEARNED

To the best of our knowledge, DOBBS represents a rather unique effort towards investigating the online browsing behavior of Web users. Besides the presented results, we gained further interested insights during its realization.

Expected long-term benefits. As our preliminary result already show, the granularity of the collected data goes far beyond the possibilities of conventional sources such as web server access logs or search engine transactions logs. Thus, DOBBS allows investigating new research questions in a much broader context of online browsing behavior that cannot be answered based on commonly available datasets.



(a) Single tab.

(b) Multiple tabs - single page load.

(c) Multiple table - multiple page loads.

Fig. 6. Examples for the four basic types of browsing behavior regarding the usage of tabbed browsing.

Unsupervised experiments. DOBBS is a designed as a field study to elicit the normal browsing behavior of users. Once a user has installed the add-on, there is no interference from any controlling entity. This includes, in principle, that users can consciously manipulate the resulting logging data by behaving in a specific manner. Thus, any analysis of the DOBBS dataset must be performed with a careful interpretation of the results. This particularly holds true when it comes to the identification of "outliers", i.e., browsing behavior that significantly differs from the average.

Incomplete logging data. In Section III-D, we outlined reasons that may cause incomplete logs. If an analysis of the dataset does not (heavily) depend on these types of missing logging data their absence can be ignored. Alternatively, we proposed approaches for filtering out affected data or extrapolating the missing information using the available data. The granularity of the DOBBS dataset makes both approaches valid and applicable for most evaluations. Still, any alteration of the dataset needs to be done carefully to ensure meaningful results.

Dependencies between measure parameters. DOBBS collects a large variety of information describing users' online browsing behavior. Despite the comprehensive set of measured parameters, the add-on cannot completely capture the exact behavior of users. This often leaves room for different interpretations of the logging data. For example, one can argue about whether a page should be considered to be viewed by a user if a browser window did not have the focus and/or the user was inactive during that time. Thus, any evaluation should be preceded by an analysis of the alternative interpretations, and result should be accompanied by the assumptions made.

Spreading the word. Motivating user to contribute to the DOBBS dataset is challenging. Firstly, the add-on does not provide an added value to users; contributing to DOBBS is an act of goodwill. And secondly, despite the anonymisation and application of encryption techniques, user might perceive privacy risks. To address this, our approach is to be as open and responsive as possible. To this end, DOBBS features its dedicated project website providing all relevant information and a download link to dataset. Furthermore, the add-on is available as open source under the very open BSD license.[4]

VI. Conclusions

In this paper, we introduced DOBBS, our approach towards creating a comprehensive dataset capturing browsing behavior of online users. DOBBS provides a browser add-on that keeps track of the most relevant events. The logging is done in a completely privacy-preserving manner. We also presented results based on the current dataset showcasing the potential benefits of the DOBBS dataset which go far beyond the capabilities of traditional sources such as web server access logs or search engine transaction logs. DOBBS is a long-term effort. The collected data will be provided as a public dataset for research purposes on the project website (http://dobbs.deri.ie). Naturally, the value of this dataset increases with the number of participants and the length of their participation. We therefore would like to encourage every interested Internet user to download and install the browser add-on, thus contributing to DOBBS. For anyone interested in updates, the results, and the latest version of the dataset, we refer to the DOBBS project website. Besides providing the dataset and all project-relevant information, the sites also features a contact section allowing participants or interested users to leave comments or feedback, as well as to ask questions in an anonymous manner, i.e., without revealing any personal data.

References

[1] E. Adar, J. Teevan, and S. T. Dumais. Large Scale Analysis of Web Revisitation Patterns. In *CHI'08*. ACM, 2008.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving Web Search Ranking by Incorporating User Behavior Information. In *SIGIR'06*. ACM, 2006.

[3] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal Analysis of a Very Large Topically Categorized Web Query Log. *J. of the Am. Soc. for Inf. Sci. and Technol.*, 58(2), 2007.

[4] I. Fette and A. Melnikov. The WebSocket Protocol. RFC 6455, Internet Engineering Task Force, http://www.ietf.org/rfc/rfc6455.txt, 2011.

[5] S. Goel, J. M. Hofman, and M. I. Sirer. Who Does What on the Web: A Large-scale Study of Browsing Behavior. In *ICWSM'12*. AAAI, 2012.

[6] L. K. J. Grace, V. Maheswari, and D. Nagamalai. Web Log Data Analysis and Mining. In *Advanced Computing*, volume 133 of *Communications in Computer and Information Science*. Springer, 2011.

[7] E. Herder, R. Kawase, and G. Papadakis. Experiences in Building the Public Web History Repository. In *Proc. of Datatel Workshop*, 2011.

[8] A. T. Holdener, III. *Ajax: The Definitive Guide.* O'Reilly, 2008.

[9] M. Kellar, C. Watters, and M. Shepherd. The Impact of Task on the Usage of Web Browser Navigation Mechanisms. In *GI'06*. Canadian Information Processing Society, 2006.

[10] M. Meiss, J. Duncan, B. Gonçalves, J. J. Ramasco, and F. Menczer. What's in a Session: Tracking Individual Behavior on the Web. In *HT'09*. ACM, 2009.

[11] S. Wedig and O. Madani. A Large-scale Analysis of Query Logs for Assessing Personalization Opportunities. In *SIGKDD'06*. ACM, 2006.

[12] H. Weinreich, H. Obendorf, and E. Herder. Not Quite the Average: An Empirical Study of Web Use. *ACM Trans. on the Web*, 2(1), 2008.

[13] L. Xue, M. Chen, Y. Xiong, and Y. Zhu. User Navigation Behavior Mining Using Multiple Domain Description. In *WI-IAT'10*. IEEE, 2010.

[14] H. Zhang and S. Zhao. Measuring Web Page Revisitation in Tabbed Browsing. In *CHI '11*. ACM, 2011.

[4] http://code.google.com/p/deri-dobbs/