User Test Logger: An Open Source Browser Plugin for Logging and Reporting Local User Studies

Vagner Figueredo de Santana(✉) and Felipe Eduardo Ferreira Silva

IBM Research, São Paulo, SP 04007-900, Brazil
vagsant@br.ibm.com, felipe.ferreiral@ibm.com

Abstract. Tooling user studies tasks is fundamental to reduce the burden of evaluators, practitioners, facilitators, and observers during user interface (UI) evaluations. However, available tools for collecting data beyond clicks are usually paid, count on limited vocabulary of UI events, have complex setup, or do not provide easy download of captured data. In this context, we present a general-purpose data logger for local user studies able to capture up to 40 different types of UI events. It provides detailed data logging, downloading, and reporting features. The evaluation of the tool was performed in a focus group with 4 Human-Computer Interaction (HCI) specialists. The focus group resulted in new features and changes/corrections in existing features. The redesigned tool is open source and freely available to the HCI community. The tool can be used by HCI specialists or data scientists as a core tool for empirical evaluations or an additional data source in mixed methods assessments, offering greater control on types of events captured and counting on multiple types of report.

Keywords: Data logger · Interaction log analysis · User study · User test · Usability · Accessibility · Evaluation

1 Introduction

The study of log files dates back to the first Operating Systems, supporting debugging and auditing processes. Nowadays, interaction logs are used in multiple systems, since interaction patterns represent part of human behavior while interacting with computing systems. Such systems are present in multiple contexts, for instance, e-commerce, education, e-government, business intelligence, analytics, among others. In the context of Human-Computer Interaction (HCI), the possibility of logging detailed interaction data paved the path for multiple approaches based on task grammars/models, e.g., GOMS (Goals, Operators, Methods, and Selection Rules) [11] and Concur Task Trees (CTT) [15], and data-oriented approaches, e.g., as Web Event-logging Tool (WET) [8], MouseTrack [2], and Web Event Logger and Flow Identification Tool (WELFIT) [19], just to name a few.

In this work we present User Test Logger, a general-purpose web browser logging and reporting tool that can be used in a variety of HCI studies. It can be configured to

M. Antona and C. Stephanidis (Eds.): HCII 2019, LNCS 11572, pp. 229–243, 2019.
https://doi.org/10.1007/978-3-030-23560-4_17

record any type of JavaScript or jQuery [19] event and also provides reporting and downloading features to connect with statistical and graph analysis tools.

This work is structured as follows: the next section presents related tools; Sect. 3 presents a description of tool's features; Sect. 4 details the focus group performed to evaluate the tool; Sect. 5 describes the redesign of the tool based on focus group results; Sect. 6 discusses outcomes and possibilities of using the proposed tool.

2 Related Works

The task of comparing tools requires a common ground for attributes, techniques, and data sources considered. In this realm, Santana and Baranauskas proposed a taxonomy for website evaluation tools considering 4 main dimensions [18]:

1. **Participant-evaluator interaction** – refers to the interaction between evaluators and participants during an evaluation. It can be divided into:
   a. *Localization*: Remote or non-remote;
   b. *Time*: Synchronous or asynchronous;
   c. *Use*: Formal or informal;
2. **Effort** – refers to the effort required from the evaluator and from the participant to setup an evaluation scenario. It can be divided into:
   a. *Evaluator (HCI practitioner, facilitator)*: Model/grammar maintenance, environment configuration/setup, or no action;
   b. *Participant*: Actions at the beginning, during the test, or no action;
3. **Automation type** – refers to the automation characteristics of the tool. It can be divided into:
   a. *Capture*: User expressions, physiological signals, ambience, browser events, or customized events;
   b. *Analysis*: Visual reports or statistical reports;
   c. *Critique*: content, structure, or layout;
   d. *Adjustment*. Content, structure, or layout;
4. **Data source** – refers to the data source considered in the evaluation. It can be divided into:
   a. *User Interface*: Structure or content;
   b. *User data*: Interaction data or questionnaire data;

This taxonomy highlights 4 dimensions ranging from data source to automation provided by the tool. Table 1 compares tools based on the presented taxonomy. It is possible to verify that that tools that focus on formal studies are related to local user tests. In addition, critique is offered by few tools. And the types of events captured are usually restricted and evaluators cannot choose which events to capture. When considering studies involving accessibility, most tools are restricted to mouse interaction. From the Universal Design (UD) perspective, the vocabulary of events captured by the tools should cope with the whole diversity of users' interaction strategies and users' devices.

Table 1. Tools for logging and reporting user interaction at the Web.

Tool	Participant–evaluator interaction	Effort	Automation type	Data source
Click heat[a]	Remote, asynchronous, informal	Tool setup	Capture (click), analysis (Heatmap)	Interaction data
Click tale[b]	Remote, asynchronous, informal	Tool setup	Capture (click, mousemove), Analysis (heatmap, mouse path)	Interaction data
Crazy egg[c]	Remote, asynchronous, informal	Tool setup	Capture (click, mousemove), Analysis (heatmap, playback)	Interaction data
Heatmap.com[d]	Remote, (A)synchronous, informal	Tool setup	Capture (click), Analysis (Heatmap)	Interaction data
Morae[e]	Local, synchronous, formal	Tool setup and run the study	Capture (audio, click, keypress, mousemove, screen recording), Analysis (playback)	Interaction data
Mimic [3]	Remote, asynchronous, formal	Tool setup	Capture (blur, click, dblclick, focusin, focusout, keypress, mousemove, resize, keypress, scroll), Analysis (heatmap, timeline, playback)	Interaction data
User test logger[f]	Local, synchronous, (In)formal	Tool setup and run the study	Capture (40 events vocabulary), Analysis (usage graph, heatmap, and mouse path.) Critique (accessibility/usability issues)	Interaction data

[a] www.dugwood.com/clickheat/

[b] www.clicktale.com

[c] www.crazyegg.com

[d] www.heatmap.com

[e] www.techsmith.com/morae.html

[f] https://github.com/IBM/user-test-logger/

UD can be defined as the design of products and environments that are usable by everyone, in the widest possible extension, without the need for adaptation or specialized design [6]. In this direction, User Test Logger allows evaluators to select types of events to be captured during the session, beyond mouse interaction.

When considering privacy, laws as General Data Protection Regulation (GDPR) [10] and *Lei Sobre Proteção de Dados Pessoais* [14] highlight the right of privacy, capturing only the needed data, previously stating goals and ways of using the captured data. In this direction, User Test Logger focuses on local studies so that logged data is not transmitted to an external server and participants are properly informed by practitioners/facilitators about types of data being captured in such a way that even users can easily see what was captured if it is the case.

Finally, existing tools lack easy setup and easy access to raw interaction data, key features for Data Scientists aiming at performing further analysis in external software/libraries as R, Python, or SPSS. The proposed tool aims at addressing existing gaps in the connection between HCI and Data Science, supporting field studies, applying interaction log analysis in usability tests, A/B tests, accessibility evaluations, or any type of *in situ* user studies.

3 Proposed Tool

The proposed tool was developed as an add-on for Firefox web browser and is available at Github.com.[1] The rationale of offering it as a browser add-on was to make it a tool easy to setup and use, allowing HCI practitioners to install, configure, and run the tool via web browser UI. The main pillars guiding its development are:

- Inclusive view of data captured;
- Value privacy;
- Prevent disturbing participants during the session;
- Provide easy access to captured data;
- Inclusive way of reporting captured data.

According to the taxonomy for user evaluation tools presented in [18], the User Test Logger is a tool for *capture, analysis and critique for (in)formal local user studies*. This means that the tool counts on features for capturing, analyzing, and pointing out issues related to local user studies considering predefined tasks (i.e., formal) or exploratory studies (i.e., informal). The next subsections present an overview of the initial version of the tool, including its architecture, setup example, capture, and analysis.

3.1 Architecture

The tool was developed inspired by the classic Model-View-Controller (MVC) architectural design pattern [5] applied to the Firefox web extension structure [1].

[1] https://github.com/IBM/user-test-logger/.

It considers the Model as being the *background* component, View as *popup* component, and Controller as the *content scripts* (Fig. 1).

- **Model** component (background) is responsible for establishing the communication with the content scripts and storing the data.
- **View** (popup) component is responsible for providing controls, feedback, and reporting features for the user.
- **Controller** (content script) is responsible for capturing and formatting UI events.

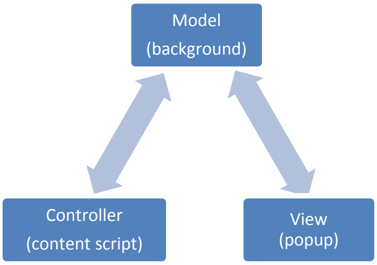


Fig. 1. User test logger architecture overview.

The User Test Logger works as follows: As soon as the plugin is loaded, the background component is loaded in the browser. Once the browser is running, every time the user opens a new web page, a content script is loaded specifically for logging that page, and a connection is established with the background component. All the events logged will be sent by the content scripts to the background, where all logged data is stored. The popup can be displayed by clicking on the "L" button (Fig. 2). This component exposes features to the user and sends the triggered actions to the background. Then, if the settings are changed, the background updates the content scripts.

Regarding components and libraries, the add-on uses jQuery [12] to cope with compatibility issues and to ease the manipulation of events. In order to save the log file on the client-side, the FileSaver [9] is used. Finally, User Test Logger uses D3 [4] drawing library for creating all visualizations.

3.2 Setup

One of the goals of the proposed tool is to ease the setup process of data logging. The easiest way of trying the tool is by loading it as a "Temporary Add-on". The add-on zip file is available at the plugin's github.com page.[2] Once the zip file is downloaded and decompressed at the users' device, the setup can be done by typing *about:debugging* in the Firefox address bar, clicking the button "Load temporary add-on", and selecting the *manifest.json* file downloaded.

[2] https://github.com/IBM/user-test-logger/archive/master.zip.

Figure 2 shows the first version of the tool menu, under the “L” button. In this menu, the item “record” starts the data logging, the menu item “report” contains the visualizations and reports for the logged data, the item “dump raw data” allows the evaluator to download logged data and clear the browser’s memory, and under the item “events” it is possible to define which events are going to be logged, also called events of interest (Fig. 3).

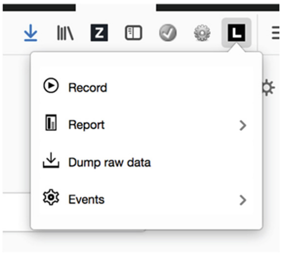


Fig. 2. User test logger main menu.

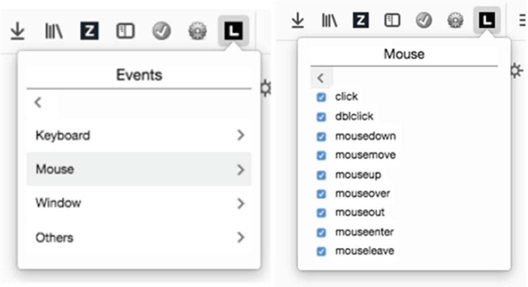


Fig. 3. Selection of events of interest prior to logging.

3.3 Data Capture

The tool supports capturing all standard JavaScript events, jQuery events, touch events, geolocation events, and device orientation events (Fig. 3). The whole set of events captured by the tool (i.e., event vocabulary) includes 40 events. Once the configuration of events of interest is done, the HCI practitioner can click on record to start the capture. After the user test session, the practitioner can pause data capture and explore reporting features or dump the interaction log for analyzing it in an external software. In addition, the tool supports analysis of interaction occurred with UI elements without id attribute and coming from multiple browser tabs. To do so it uses DOM (Document Object Model) tree path for identifying uniquely all UI target elements.

The tool can be used to capture logs from sessions separately or to log a set of participants’ sessions, depending only on the evaluation experiment design. Finally, given that the tool supports logging highly detailed data, the tool favors participants’ privacy over transmitting logged data to an external server or software component.

3.4 Analysis

The first version of the tool counted on three types of report, namely: usage graph (Fig. 4), mouse fixations heatmap (Fig. 5(a)), and mouse plot (Fig. 5(b)).

Usage Graph. The usage graph is a directed graph used to represent user interaction, event by event, based on the algorithm detailed in [19]. It can be seen as the combination of walks (non-empty alternating sequence of nodes and edges) representing what, where, and when users performed actions. In the usage graph, a node is identified by its label, which is the concatenation of the event name and an identifier of the UI element where the event occurred, e.g., "*mouseover@logo*", "*click@logo*", "*focus@/html/document/body*". Moreover, each node counts on information regarding the total of sessions they occurred, mean distance from the root node (session start), mean timestamp, among others. In addition, all these information supports identifying patterns and usability problems/accessibility barriers candidates.

These candidates are identified via a heuristic that aims at pointing out cyclic actions and deviations, comparing nodes in the usage graph, comparing paths that are far from the starting of the session due to attempts or deviations from the task at hand. For more details, please refer to [19].

The rationale of choosing the usage graph is that it is not restricted to mouse events and allows the identification of repeated sequences of events in one or more sessions. Figure 4 exemplifies how nodes (representing events in UI elements) are distributed and how nodes that are part of cyclic actions are highlighted.

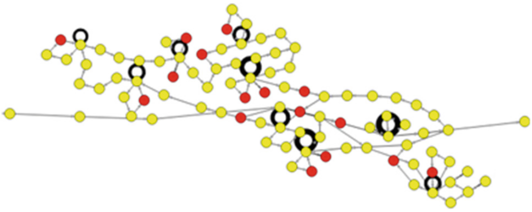


Fig. 4. Usage graph sample report.

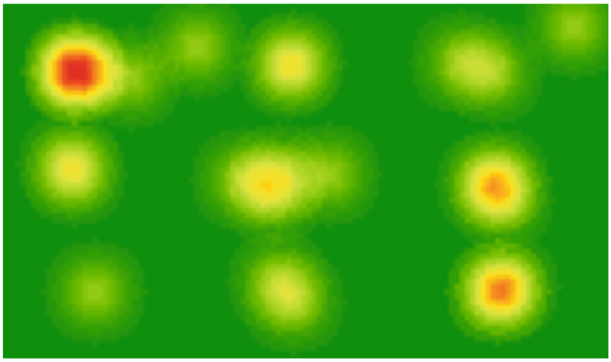


Fig. 5. Heatmap showing interaction data with a 9-points calibration page.

Heatmap. The heatmap provided by the User Test Logger is generated by detecting mouse fixations, as an analogy to eye fixations, by using the dispersion algorithm presented in [17]. In sum, it considers a mouse fixation when mouse movements are close to another and for longer durations in comparison with other mouse movements (i.e., mouse saccades).

In this version the heatmaps count on solid background instead of an overlayer. The rationale for it resides on the fact that reports aim at tracking multiple pages and multiple browser tabs. Hence, the heatmap of the first version of the tool shows mouse fixations for the whole session, which may involve multiple screens and tabs, as a way of summarizing the whole session.

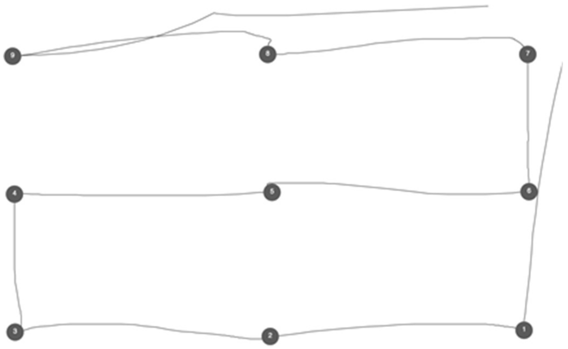


Fig. 6. Mouse plot showing interaction data with a 9-point calibration page.

Mouse Plot. The mouse plot shows mouse movements, clicks, and double clicks performed by participants (Fig. 6). It can be useful for comparing task performance and showing multiple ways the participants executed tasks. The mouse plot report of the first version of the tool also shows mouse movements and clicks for the whole session.

Finally, reports showing mouse movements are grounded on results from [7] showing the high correlation between eye and mouse movements in web browsing tasks. Besides logging and reporting features, the tool provides multiple download formats (e.g., DOT, PNG, and CSV) allowing HCI practitioners and Data Scientists to use the logged data in different analysis tools.

4 Focus Group

A focus group is defined by Krueger [13] as a *"carefully planned discussion designed to obtain perceptions in a defined area of interest in a permissive, non-threatening environment"*. According to Rubin [16], focus group is a valuable technique at early stages of a project to evaluate concepts and get feedback, judgment, feelings, exploring how representative users think and feel towards a product or service. In addition, Rubin discusses that these concepts can be presented in low-fidelity or high-fidelity prototypes. In our case, the first version of the tool was used in the presentation to our representative users.

The goal of the focus group performed in this research was to show the first version of the logger to potential users, HCI specialists with experience on performing user tests, and gather feedback on existing and desired features. The invited specialists act as researchers and possess MSc or PhD titles as highest education degree. The 6 invited participants have background on Design or Computer Science, all of them acting in the realm of HCI. The invitation was performed by email; 5 of them accepted; 4 showed up in the scheduled meeting, performed via video conference. In summary, participants characteristics are:

- **Sex**: 2 men and 2 women;
- **Background**: 2 in Design and 2 in Computer Science;
- **Highest education degree**: 2 MSc title and 2 PhD.

The materials used in the focus group involved the first version of the tool, recording the whole session in video and audio, and the video conference software used for screen sharing. The recording was performed only after specialists agreed in having the meeting recorded.

The focus group took approximately 1 h 30 min. In the first 30 min the facilitator walked through the tool's features then a round of discussion took place. An observer also participated taking notes about specialists' feedback. The round of discussions was driven by the following open-ended questions made to each of the specialists:

1. What do you think about available features?
2. What do you think that is missing?
3. What do you think that must be improved?

4.1 Results

The analysis on the collected was performed according to impact/outcome reported by each of the specialists about available/missing features. For instance, in different occasions specialists exemplified situations based on previous experiences on performing user tests. According to this rationale, the following lists summarize the results obtained in terms of available/missing features and on suggestions on how to improve existing features:

Available Features

- Heatmap and mouse plot should be done for each page, showing all the visualization as an over layer of the actual web page;
- Change the current format for raw data from JSON to CSV, easing the process of analyzing the logs in external software, given that much of existing software have capabilities for importing CSV, not all of them allow importing JSON format;
- Provide multiple ways of downloading/visualizing the available reports;

- Display the number of logged events, showing that the tool is properly capturing and providing feedback to users about the status of the tool;
- Show hints about the impact when capturing certain types of events;
  - This suggestion was made by one of the participants that tried the tool during the focus group. He mentioned that tested it in a current project having a UI with lots of asynchronous components, which results in *DOMInsertedNode* events. Hence, he mentioned that some decisions on what data to record could be informed so that only relevant data would be recorded.
- All the reports should be done for each user separately.

Missing Features

- Upload a setup file for the study, containing the types of event to capture and any additional configuration;
- Upload data captured back to the tool to analyze already downloaded data;
- Display, highlight, or differentiate in any way what are the tabs being logged;
- Provide an easier way to analyze the common patterns performed by the users;
  - Although the usage graph has all the information about the user interaction, they suggested that a simpler and faster way to analyze the common patterns and issues might of interest of specialists performing data analysis.

5 Redesign of the Tool

The redesign resulted in a new version of the tool, combining improvements on available features and implementation of some of the missing features, detailed next. Figure 7 shows the new main menu, now displaying the number of logged events, providing guidance and feedback to the user about the current status of the tool.

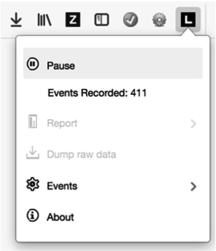


Fig. 7. User test logger main menu after redesign.

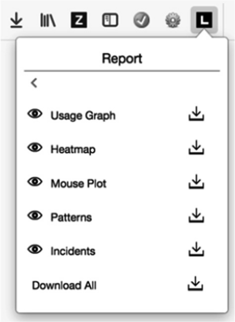
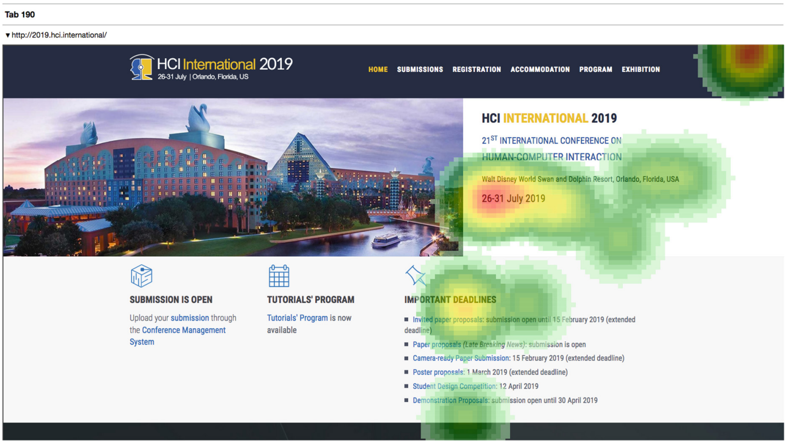


Fig. 8. User test logger reports menu after redesign.

The raw data format was changed from JSON to CSV as one of the designers suggested, in order to facilitate the specialists' task of importing it in external software in cleaning and analysis stages. Moreover, more control was provided to the specialists, allowing them to see or download individual reports (Fig. 8).

In the report section, improvements were made in the heatmap and mouse plot, considering the suggestion for separating the visualization for each page. After the changes, the visualizations are computed and displayed for each browser tab and web page (Figs. 9 and 10), as an over layer to the Web page. One report page now counts on one section per tab and, inside this section, a report for each URL used in that tab.


Tab 190

▾ http://2019.hci.international/

HCI INTERNATIONAL 2019

21ST INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION

Walt Disney World Swan and Dolphin Resort, Orlando, Florida, USA

26-31 July 2019

SUBMISSION IS OPEN

Upload your submission through the Conference Management System

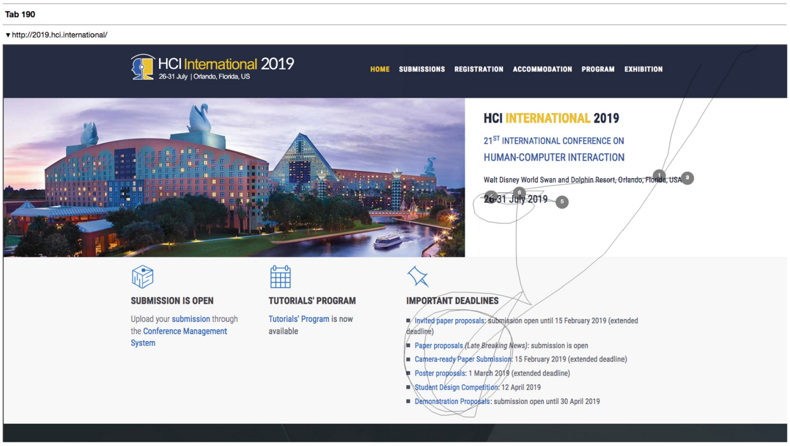
TUTORIALS' PROGRAM

Tutorials' Program is now available

IMPORTANT DEADLINES

- Invited paper proposals: submission open until 15 February 2019 (extended deadline)
- Paper proposals (Late Breaking News): submission is open
- Camera-ready Paper Submission: 15 February 2019 (extended deadline)
- Poster proposals: 1 March 2019 (extended deadline)
- Student Design Competition: 12 April 2019
- Demonstration Proposals: submission open until 30 April 2019

Fig. 9. Heatmap report after redesign.



HCI INTERNATIONAL 2019

21ST INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION

Walt Disney World Swan and Dolphin Resort, Orlando, Florida, USA

26-31 July 2019

SUBMISSION IS OPEN

Upload your submission through the Conference Management System

TUTORIALS' PROGRAM

Tutorials' Program is now available

IMPORTANT DEADLINES

- Invited paper proposals: submission open until 15 February 2019 (extended deadline)
- Paper proposals (*Late Breaking News*): submission is open
- Camera-ready Paper Submission: 15 February 2019 (extended deadline)
- Poster proposals: 1 March 2019 (extended deadline)
- Student Design Competition: 12 April 2019
- Demonstration Proposals: submission open until 30 April 2019

Fig. 10. Mouse plot after redesign.

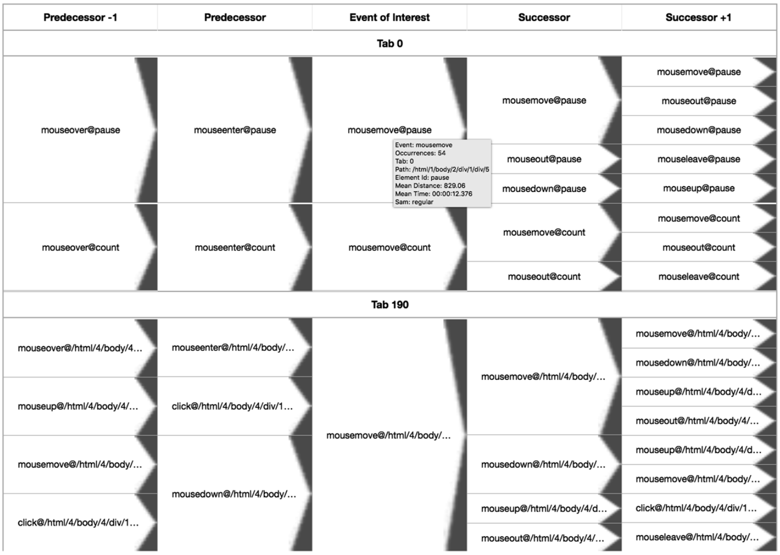
The new features implemented after the focus group were done considering the need for reports that present relevant information about the user interaction in an easier and more accessible way than the usage graph (Fig. 11). Two reports were developed, namely:

1. **Patterns**: An HTML accessible table that shows the usage graph's nodes whose occurrences are above the 80th percentile, representing nodes representing repeated actions, its source node(s), and its outgoing node(s). This allows specialists to identify UI events repeated during the interaction, their source and they subsequent action;
2. **Incidents**: An HTML accessible table that shows the usage graph's nodes defined as incidents by the usage graph and its SAM-based heuristic (for more details on the SAM-based heuristic, please refer to [19]). The rationale of the incidents report is the same of the patterns report, showing key events in the central column and the corresponding originated and subsequent events in the outer columns.

Finally, regarding features that were not implemented we have the following:

1. Report features for each user – Since the User Test Logger is a web extension, the idea is to keep it as simple as possible. It is not desirable that the proposed tool become too much complex nor demanding extra CPU or RAM usage, which could impact negatively the participants experience the UI in evaluation;
2. Upload setup file for the study – Given that the only setup performed by the specialist is to select the events of interest and that these values persist, the easiest way to select the events to be recorded is by selecting them in the events menu item, as shown in the Fig. 3;

3. Upload data captured back to the tool – Given the goal of the tool of supporting a rich and detailed data capture, allowing specialists to analyze in depth in additional graph or statistical software, the goal was to allow specialists to record and download all reports for further analysis;
4. Indicate what are the tabs being logged – Given that all tabs are logged and that the tool also aims at not disturbing the end user participating in the user test, the "L" button was designed to be as simple and as subtle as possible. In the design phase of the tool, for example, it was considered to have a red circle over the menu to show that it was in use. However, it is known that these indicators end up calling users' attention that, for a logger tool in a local setup, is not desirable.



Predecessor -1	Predecessor	Event of Interest	Successor	Successor +1
		Tab 0		
mouseover@pause	mouseenter@pause	mousemove@pause	mousemove@pause	mousemove@pause
				mouseout@pause
				mousedown@pause
			mouseout@pause	mouseleave@pause
			mousedown@pause	mouseup@pause
mouseover@count	mouseenter@count	mousemove@count	mousemove@count	mousemove@count
				mouseout@count
			mouseout@count	mouseleave@count
		Tab 190		
mouseover@/html/4/body/4...	mouseenter@/html/4/body/...	mousemove@/html/4/body/...	mousemove@/html/4/body/...	mousemove@/html/4/body/...
				mousedown@/html/4/body/...
mouseup@/html/4/body/4/...	click@/html/4/body/4/div/1...			mouseup@/html/4/body/4/d...
				mouseout@/html/4/body/4/...
mousemove@/html/4/body/...	mousedown@/html/4/body/...		mousedown@/html/4/body/...	mouseup@/html/4/body/4/d...
				mousemove@/html/4/body/...
click@/html/4/body/4/div/1...			mouseup@/html/4/body/4/d...	click@/html/4/body/4/div/1...
			mouseout@/html/4/body/4/...	mouseleave@/html/4/body/...

Event: mousemove
Occurrences: 54
Tab: 0
Path: /html/1/body/2/div/1/div/5
Element Id: pause
Mean Distance: 829.06
Mean Time: 00:00:12.376
Sam: regular

Fig. 11. Patterns/incidents report in the second version of the tool.

5. The tool tip shows additional information for each node contained in the usage graph.

6 Discussion

Although there are multiple tools for interaction logging, the literature still lacks a general-purpose logging tool supporting an easy way for HCI practitioners and Data Scientists to capture detailed interaction data, covering more than clicks and mouse movements. The proposed tool addresses this gap and is available at Github (https://github.com/IBM/user-test-logger). The tool counts on documentation and on videos describing how to install, capture data, and start analyzing highly detailed interaction data.

The participants from the focus group provided valuable feedback and insights to our team. Although the User Test Logger was, by design, developed to fill a gap existing in evaluation tools, the participants needed reports similar to existing ones in similar tools, as the heatmap overlayer that was added after the focus group.

Bearing in mind limitations of the focus group, the number of participants is the main issue, given that it was planned to be run with 6 participants.

The impacts aimed with this tool involve technological aspects, supporting the capture of detailed data, and social aspects, given that it goes beyond of click streams, data usually used in web studies that take from granted sight and use of the mouse. This is one of the key aspects of the tool, considering that one of its pillars is to be inclusive, considering the whole diversity of types of events that can be captured and, hence, representing multiple ways of interacting with the Web.

To the best of our knowledge, there is no open source web evaluation tool providing easy access to logged data and supporting the analysis of interaction patterns in an accessible way, allowing specialists (with or without disability) to analyze reports containing highly detailed events (or micro-interactions, as defined in [3]).

Future works involve including multiple events in the heatmap and mouse plot and porting the plugin to different popular web browsers as Chrome and Safari.

Finally, next steps in this research include publicize the tool to HCI practitioners and get feedback so that it can be a valuable tool for logging interaction data, especially on contexts involving accessibility and analysis of interaction data beyond those coming from mouse and pointing devices.

Acknowledgements. We would like to express our appreciation to all specialists that participated in the focus group and provided valuable feedback to our team.

References

1. Anatomy of an Extension: Mozilla (2018). https://developer.mozilla.org/en-US/docs/Mozilla/Addons/WebExtensions/Anatomy_of_a_WebExtension
2. Arroyo, E., Selker, T., Wei, W.: Usability tool for analysis of web designs using mouse tracks. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems (CHI EA 2006) (2006)
3. Breslav, S., Khan, A., Hornbæk, K.: Mimic: visual analytics of online micro-interactions. In: Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI 2014), pp. 245–252. ACM, New York (2014). https://doi.org/10.1145/2598153.2598168
4. Bostock, M.: Data-Driven Documents (D3) (2017). https://d3js.org/
5. Buschmann, F., Henney, K., Schimdt, D.: Pattern-Oriented Software Architecture: On Patterns and Pattern Language. Wiley, New York (2007)
6. Centre for Excellence in Universal Design.: What is Universal Design? (2012). http://universaldesign.ie/What-isUniversal-Design/
7. Chen, M.C., Anderson, J.R., Sohn, M.H.: What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing. In: CHI 2001 Extended Abstracts on Human Factors in Computing Systems (CHI EA 2001) (2001)

8. Etgen, M., Cantor, J.: What does getting WET (web event-logging tool) mean for web usability? In: 5th Conference on Human Factors and the Web (1999)
9. FileSaver JavaScript Library.: (2019). https://github.com/eligrey/FileSaver.js
10. General Data Protection Regulation (GDPR) (2016). https://eur-lex.europa.eu/legalcontent/EN/TXT/?uri=CELEX:32016R0679
11. John, B.E., Kieras, D.E.: The GOMS family of user interface analysis techniques: comparison and contrast. In: ACM Transactions on Computer-Human Interaction 3, 4 Dec 1996, pp. 320–351 (1996)
12. jQuery.: (2019). http://jquery.com/
13. Krueger, R.A., Casey, M.A.: Focus Groups: A Practical Guide for Applied Research. Sage Publications, London (2014)
14. Lei de Proteção de Dados Pessoais.: (2018). www.planalto.gov.br/ccivil_03/_ato20152018/2018/lei/L13709.htm
15. Paternò, F.: Model-Based Design and Evaluation of Interactive Applications. Springer, Heidelberg (2012)
16. Rubin, J.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley, Hoboken (1994)
17. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 Symposium on Eye Tracking Research and Applications (ETRA 2000) (2000)
18. de Santana, V.F., Baranauskas, M.C.C.: A taxonomy for website evaluation tools grounded on semiotic framework. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2018. LNCS, vol. 10907, pp. 38–49. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92049-8_3
19. Santana, V.F., Baranauskas, M.C.C.: WELFIT: a remote evaluation tool for identifying Web usage patterns through client-side logging. Int. J. Hum. Comput. Stud. **76**(2015), 40–49 (2015)