Advanced web analytics tool for mouse tracking and real-time data processing

Lukáš Čegan[1], Petr Filip[2]
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
[1]lukas.cegan@upce.cz
[2]petr.filip1@upce.cz

***Abstract*—Web analytic tools offer important support for better recognition of the web user's behavior, identification of bottlenecks and errors in user interface design, performance measurement of web environment, monitoring of website availability or recommendation of appropriate website content. These tools are based on tracking techniques and sophisticated algorithms that process and evaluate large volumes of captured data. In this paper, we propose a new solution to capture mouse movements of web users, to identify their area of interest. This solution is based on real-time data transformation, which converts discrete position data with high sample period to predefined functions. This transformation has a high degree of accuracy, which is exemplified by case scenarios. The result of this solution is a significant saving in data, transmitted from the client to the server, which leads to significant savings in system resources on the server side.**

Keywords—mouse tracking; real-time data processing; user behavior

I. Introduction

The World Wide Web is definitely the most widespread service of the current Internet with more than 3.5 billion users. Most of the population of the developed world uses this service daily for shopping, communication, information search, company presentation, advertising, etc. Many companies are trying to leverage this fact and try to commercialize their products and services in this large market. However, this creates considerable competition among companies, which is reflected in the high demands on the quality of web applications. For these reasons, increasing emphasis is placed on the quality of user experience design in the process of developing web applications. This better user experience design leads to greater user satisfaction with the web application and hence greater revenues. The quality of user experience design can be measured and evaluated using appropriate tools to track the activities of each web user. Fortunately, today there are enough tracking tools on the market. The range of tracking functions of these tools is different, as well as the method of data communication between the web client and the tracking server. In many cases, the volume of data transferred is considerable, which has a negative impact on the quality of the service provided. One of the important tracking functions for evaluating user experience is to track mouse movement. Unfortunately, this feature is one of those which is based on logging a large amount of data that is necessary to send to the server for evaluation. This problem can be addressed by the new solution that is presented in this paper. This solution is based on tracking mouse movement and subsequent vectoring of the mouse path on the client side, resulting in significant reduction of transmitted data while maintaining accuracy of measurement. The quality of the vectoring function is judged in the case study that is part of this paper.

The structure of this paper presents as follows. After introducing the objective of this paper, the related work is presented in Section II. Next, in Section III, a proposed solution for tracking mouse movements of web users is described. The concept of mouse movement tracks vectorization is given in Section IV. The Section V described the performance evaluation of proposed solution. In Section VI is mentioned experimental results and discussion covering results of the presented work. Finally, the last Section VII gives conclusions and future research opportunities.

II. Related Work

Mouse tracking is a basic technique which produces a large amount of user behavioral data that can be used to evaluate and improve a wide range of areas such as user experience design, prefetching and recommendation systems, and psychological studies. A number of research papers dealing with the monitoring of activities and tracking mouse movements of web users has been published by various authors in recent years. Many of these research studies focus on evaluating web user behavior for recommendation and prediction modeling in the e-commerce area [1], [2], [3]. Additional published papers are devoted to discovering the relationship between the users' eye movement and mouse movement [4], [5]. The area of neural and psychological studies cannot be omitted either [6], [7], [8]. Unfortunately, less scientific work is devoted to the motion capture tool itself [9], [10]. In addition to solutions published in scientific work, commercial tools are available on the market. These tools usually offer a comprehensive set of features to monitor web users. They also have analytic capabilities to evaluate captured data whose outputs provide valuable information to business analysts about website performance.

The main tools are Crazy Egg, Clicktale, Hotjar, Mouseflow, SessionCam, LuckyOrange, UsabilityTools, Inspectlet and Hoverowl. The operation of commercial and academic web tracking tools is based on the same principle, namely capturing defined events by JavaScript API that occur during a web user session. This captured data is periodically asynchronously sent to the logging server, where they are processed and stored for future use. However, this communication is very demanding in terms of the amount of data that needs to be transferred from the client side to the server side. The proposed mouse pathway vectorization, as described in this paper, largely addresses this problem.

III. Proposal Solution

A proposed solution is based on a new advanced web analytics platform, which provides a range of functions through which it is possible to tracking and analyze mouse movements in real-time with a small amount of data transmitted to the server. The architecture of the proposed platform is shown in Fig. 1.

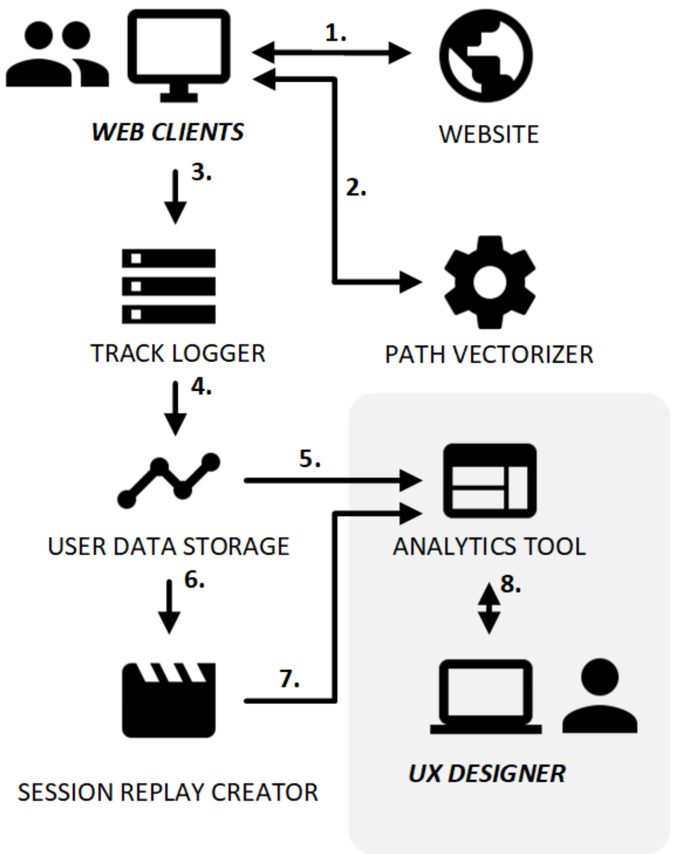

Fig. 1. Architecture of tracking and analytics platform. (1) Web client requests website from web server and (2) vectorizer library from path vectorizer (also web server). User activity is tracked and data is sent asynchronously to Track logger. (4) processed data is stored in User data storrage. (6) Seesion replay creator use data for video. Analytical tool use data and video for user report. (8) UX designer use analytical reports.

The proposed platform consists of five parts: tracking logger, track vectorizer, user sessions storage, session replay creator and analytics tool.

A. Track logger

The track logger is a component responsible for data transfer from client to server. The component is implemented as a JavaScript library that sends data to the server using asynchronous POST HTTP requests. Before sending an HTTP request, the data is processed by a vectorization component. To minimize the transmitted data, they are also compressed using the LZ77 algorithm (Lempel-Ziv, 1977). This is used by the LZ-UTF8 library [11].

B. Path vectorizer

Path vectorizer is a component responsible for the mouse movement path vectorization. This vectorization minimizes the data that is essential for describing the mouse trajectory. A description of the internal functioning of this component is described in detail in Section IV.

C. User data storage

User data storage is a component, providing storage of measured user session data. As a data repository, relational databases Oracle is used. The stored data is used by components: Session replay creator and Analytics tool.

D. Session replay creator

Session replay creator is a component that creates a preview of each user session. The preview displays mouse movement and each visitor activity in a time. At this time, it is not possible to create a video in the web environment that would record the user's work in the client's web browser. For this reason, creating a preview is based on capturing all HTTP communication between the client and the server. These web resources are retrieved and stored by the logging server. From these sources, a user session preview is created on request and played in the built-in iFrame, where the user's activities are simulated. All sources of each HTML page must be archived so that the user session can be previewed, even after the source page is updated. Unfortunately, it is not possible to create a preview of a session when a third-party site redirection occurs, such as a logging process with Google's oAuth service.

E. Analytics tool

Analytics tool manages measured user sessions. Each session contains metadata that allows users to search and filter sessions. The user sessions can be played on the built-in player. For more convenient analysts' work, a preview can be played back quickly. At this point, no other logic is implemented in the tool that could segment users by other criteria such as a conversion ratio.

The next Figure 2. illustrates a sequence diagram that describes the integration of each component of the proposed solution, from time perspective. The communication starts, when the Web browser sends a HTTP request to the web server. The web server processes a request and then sends a response (web page). Next, the web browser parses the source code of the web page, and it starts loading the tracking and vectorization library. After the web browser receives the library, the monitoring and evaluation of user activity is initiated. The catchEvent feature ensures that both the mouse movement data, and the page scrolling and resizing data, are captured and stored in the FIFO queue. From this queue, the data is sequentially removed and processed using a vectoring tool. Then, the processed data is sent asynchronously to the track logger, which uploads it in the storage.

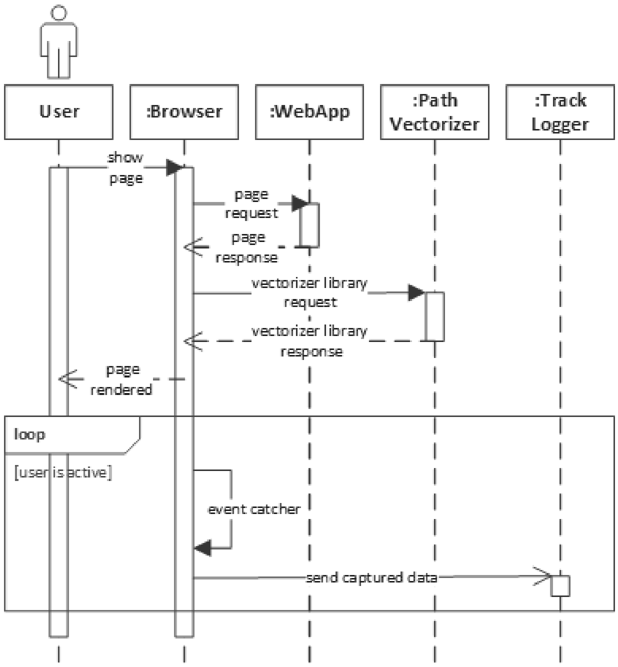

Fig. 2. UML sequence diagram of the mouse tracking implementation

IV. Path Vectorizer

The path vectorizer is a core component of a proposal analytical tool. The component is responsible for evaluating the captured position data and their subsequent vectorization. Evaluation is done in real time. The vectorization process is illustrated in the following Figure 3.

The vectorization process starts with an event, when a web page is rendered in the client browser. Subsequently, the tracking module is initialized, which waits until the occurrence of the Web API event “mouseMove”. This event fires a function that saves data to a local storage (FIFO queue). Furthermore, the distance between the last and the penultimate point of the queue, and their time difference, is calculated. In the case of a small positional distance or a large time difference, the motion of the mouse is evaluated as interrupted, and therefore the timer is not reset. In other cases, the timer is reset. The timer is an element that stores the time stamp of the last known mouse movement activity. When the timer is inactive for three seconds, an event occurs that initiates the vectorization process of the stored data. First, all the data is removed from the queue. These data then enter into the vectoring calculation. The result of the calculation is the curve expressing the mouse path and a time curve expressing the movement of the cursor over the curve, over time. Once the boundary volume of data is collected, these curves are sent to the server.

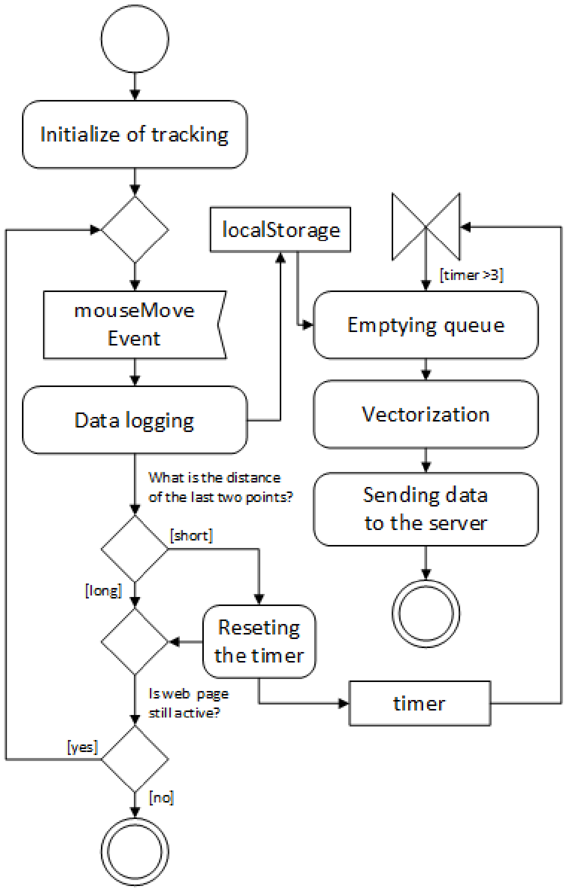

Fig. 3. UML activity diagram of JavaScritp tracker

The Vectorization activity, shown in the diagram above, consists of a sequence of atomic tasks:

- finding boundary points,
- and replacing points with a curve.

Boundary points are defined as points of the dataset where the curve type is changed or where the cursor movement speed is changed. The proposed solution distinguishes only two types of curves: line and b-spline. Finding boundary points is based on analyzing the sequence of data points and finding a breakpoint.

V. Performance Evaluation

The effectiveness of the proposed solution has been investigated in an experimental test. For the needs of the experiment, a special test website was created. The test website has been deployed to a testbed platform. The platform consists of the physical machine Dell E6440, Intel i5 (4310M), 2.70 GHz, 8GB RAM, Windows 10 64 bit. and the virtualization platform Oracle VirtualBox 5.1.26. The virtual machine host represents the desktop client with web browser Chrome 60. The guest represents the server side with operation system Debian 9.1 and the web server Apache 2.4.25.

The impact of the vectorization algorithm was conducted on three test scenarios. The test scenarios included captured mouse movement data:

A. without vectorization,
B. with vectorization and without using compression,
C. with vectorization and with compression.

In order to compare the values of each scenario, all three scenarios were executed at the same time.

The following figure demonstrates the user's mouse movements on one page (Fig. 4). All the points that were captured using the web and the "mousemove" event are visible on the chart – Scenario A.

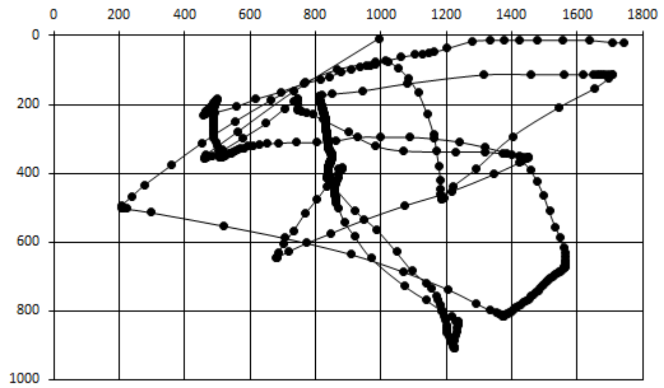

Fig. 4. Graph of mouse movement

Figure 5. shows the same mouse movements as in the previous figure, but points were processed using a vectoring tool. As can be seen in the figure, the number of points expressing the mouse's path has significantly decreased with preserved position accuracy – Scenario B.

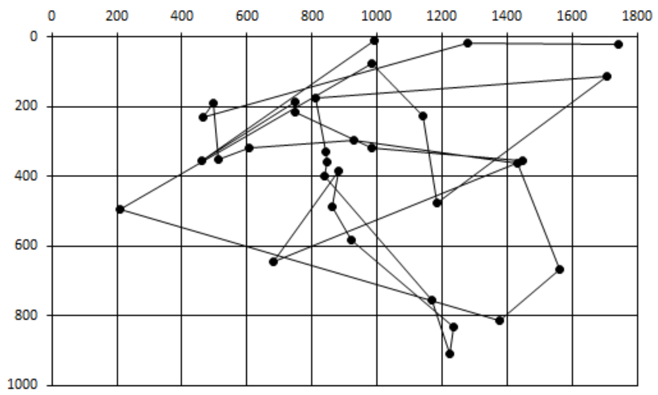

Fig. 5. Graph of mouse movement

VI. Experimental Results

Table I. showed cumulative values of HTTP communication. For each scenario, the value of the number of HTTP requests is expressed. Only requests that sent the captured data to the server, are counted. The last column of the table shows the total volume of data transferred for all requests.

TABLE I. CUMULATIVE COMMUNICATION VALUES

Scenario	Requests	Data size [kB]
A	72	465
B	11	118
C	8	73

Table II. shows a result of the vectorization process. In the first column, the type of curve is listed. For each curve, the number of points that have been replaced, their total length, the total cursor time on the curve, and the accuracy of the substitution are calculated. Calculation of replacement accuracy is based on the residual sum of squares.

TABLE II. VECTORIZATION

Curve	Points	Length [px]	Time [s]	Accuracy[RSS]
Line	714	731617	139	0,91
B-spline	78	34351	14	0,73

VII. Conclusion

This paper presents a new solution for tracking web user activity that allows understanding of how a user interacts with websites. This interaction includes path movement, scrolling, and clicks. The main part of the proposed solution is an advanced component for vectorization of mouse movement trajectory that dramatically reduces the amount of data that needs to be transferred to the server. The degree of vectorization accuracy has been investigated and evaluated by the performance experiment. As can be seen from the results, the vectoring of the captured points, results in considerable savings of transmitted data while maintaining a high degree of accuracy.

As future work, we are planning to implement the new module to a session replay creator component which allows creating of a full-featured video file for more accurate reconstruction and playback of visitor activities. Combining this tool with other analytic tools, such as Google Analytics, provides a full picture of web user behavior.

Acknowledgment

This work is published thanks to the financial support Faculty of Electrical Engineering and Informatics, University of Pardubice under grant SGS_2017_025 "Active monitoring of web users' behavior".

References

[1] Md. Tanjim-Al-Akib, Lutfullahil Kabir Ashik, Hosne-Al-Walid, Krishanu Chowdhury, "User-modeling and recommendation based on mouse-tracking for e-commerce websites," in *Computer and Information Technology (ICCIT)*, 2016, pp. 517-523.

[2] Song HengJie, Liao Ruoxue, Zhang Xiangliang, Miao Chunyan, Yang Qiang, "A mouse-trajectory based model for predicting query-url relevance," in *Proceedings of the National Conference on Artificial Intelligence*, 2012, pp. 143-149.

[3] Arapakis Ioannis, Leiva A. Luis, "Predicting User Engagement with Direct Displays Using Mouse Cursor Information," in *Proceeding of the 39th International ACM SIGIR conference*, 2016, pp. 1-10.

[4] K. Rodden, X. Fu, A. Aula, I. Spiro, "Eye-mouse coordination patterns on web search results pages", in *CHI O8 extended abstracts on Human factors in computing systems*, 2008, pp. 2997-3002

[5] Q. Guo, E. Agichtein, "Towards predicting web searcher gaze position from mouse movements" in *Proceeding of 28th international conference on Human factors in computing systems*, 2010, pp. 3601-3606

[6] Eric Hehman, Ryan M. Stolier, Jonathan B. Freeman, "Advanced mouse-tracking analytic techniques for enhancing psychological science" in *Group Processes & Intergroup Relations*, Vol 18, Issue 3, 2015, pp. 384 – 401.

[7] Thomas A. Farmer, Sarah A. Cargill, Nicholas C. Hindy, Rick Dale, Michael J. Spivey, "Tracking the Continuity of Language Comprehension: Computer Mouse Trajectories Suggest Parallel Syntactic Processing" in *Cognitive science*,Volume 31, Issue 5, 2007, pp. 889–909

[8] Sara Incera, Theresa A. Markis, Conor T. McLennan, "Mouse-Tracking Reveals When the Stroop Effect Happens." in *The Ohio Psychologist*, 2013, pp. 32-34.

[9] L. A. L. Torres and R. V. Hernando, "(smt) real time mouse tracking registration and vizualization tool for usability evaluation on websites" in Proceedings of the IADIS International Conference on WWW/Internet, 2007, pp. 187-192

[10] Clements Schefels, Sven Eschenberg, Christian Schöneberger, "Behavioral Analysis of Registered Web Site Visitors with Help of Mouse Tracking" in IEEE 14th International Conference on Commerce and Enterprise Computing, 2012, pp.33-40

[11] GitHub (2017, Sep. 2) A high-performance Javascript string compression library. [Online]. Available: https://github.com/rotemdan/-lzutf8.js