An Evaluation with Web Developers of Capturing User Interaction with Rich Internet Applications for Usability Evaluation

Luciano T. E. Pansanato¹, Adriano Rivolli², Douglas F. Pereira³

Department of Computing, Federal University of Technology – Paraná, Av. Alberto Carrazzi, 1640, 86300 000, Cornélio Procópio, PR, Brazil

¹luciano@utfpr.edu.br; ²rivolli@utfpr.edu.br; ³doug_felpe@hotmail.com

Abstract

The user interaction logging with a Rich Internet Applications can provide valuable information for usability evaluation and user behavior study. However, current research has focused only on issues of implementation and validation of tools for capturing user interaction instead tool usage by web developers. This work presents an evaluation through user testing of a tool called WAUTT for capturing user interaction with Rich Internet Applications. The main goal of this evaluation is to investigate the use of a tool for capturing user interaction with web developers. The results indicate that the participants have good understanding of the tool, found easiness to operate the tool for capturing and analyzing the user interaction, consider its future re-use, and they felt satisfied with the experience. The results also assisted for identifying the user interaction data considered more useful to web developers.

Keywords
Usability Evaluation; User Interaction Logging; Log File Analysis; Rich Internet Application; User Testing; Web Developer Usage

Introduction

Web applications have evolved from static pages or pages with content generated dynamically to applications that provide sophisticated interfaces. The term Rich Internet Application (RIA) refers to a family of heterogeneous solutions with the objective to add new capabilities to the conventional Web [1]. RIA technologies have been increasingly adopted by user-centric web applications that demand advanced presentation and interactivity resources.

The popularity of RIAs comes from their capability to promote presentation and interaction of quality. A traditional web application interface consists of multiple pages, refreshed at each user interaction. In RIAs, the interface can be a single page comprising subpages that manages all the users’ interactions, like in a desktop application. This paradigm avoids full-page refreshes at each interaction and allows applications to independently load, display and update individual page elements [1].

RIAs pose a problem whenever it is necessary to get information on how users actually use them. Data on user interaction that can be stored in the server are not enough to extract detailed information on the application’s real use. Regarding usability evaluation, the installation of additional software in the client’s machine may be necessary to capture the entire user interaction. However, a user may not be willing to install software of this nature in his/her equipment for safety or performance reasons.

The Web Application Usage Tracking Tool, WAUTT [2], is a tool used to capture detailed information at the level of the page elements and events associated to user interaction with a RIA. The idea underlying WAUTT is to use a JavaScript code capable of capturing user interaction in the client and registering the obtained data in a server for further analysis and interpretation.

Data from user interaction with a RIA provide valuable information for usability evaluation and user behavior study. However, substantial web developers can have some issues in using a tool to capture the interaction as well...
as the captured information. Current research has focused only on issues of implementation and validation of tools for capturing user interaction with web applications. This work presents an evaluation of WAUTT with web developers to investigate the use of a tool for capturing user interaction for getting knowledge about tool usage. The knowledge obtained will address future work to improve tools for capturing user interaction with web applications and web development practices.

This work is organized as follows. Next section describes related works according to different approaches adopted to capture user interaction and it highlights at the end the lack of studies with web developers on the use of tools for capturing user interaction. Subsequently, WAUTT is presented focusing on its capture approach and data that are captured. Thereafter, the evaluation proposed is presented which is based on user testing with web developers. The results of this evaluation are presented subsequently. The final section presents the conclusions and future works.

**Related Work**

The capturing of user interaction is highly used as a way to get data for usability evaluation and to study user behavior on the Web [3] [4] [5] [6]. Data on user interaction can also be used to justify the realization of some maintenance and/or alteration in the web application. For instance, a careful analysis of the data can identify less and more visited areas and most frequently used functionalities.

The capturing of user interaction can be done via web server. Generally, the scope of the interaction captured through the web server is limited since only visited pages are registered instead of all users’ interactions with a RIA. For instance, the ErgoMonitor environment [7] carries out an analysis using selective data collected from the server’s access log related to the interactions realized by the user and the web interface. However, this analysis of the data stored in the server access log alone is not sufficient for applications which have great parts of the interaction realized in the client.

The AWUSA [8] also uses data from the server but it takes into consideration pages and links categories determined from an analysis of the application’s HTML code. These categories are used to increase the level of detail about the user interaction information. In addition, to meet its objective, this tool also used data from the application log (transaction data) and other additional log files. Thus, the need for more information is clear, even more when a RIA is considered.

An alternative to capturing user interaction is to use a proxy server between the client and the web server. This alternative, however, requires that the user makes a configuration in his/her browser to use the proxy and later to undo that configuration. The use of a transparent proxy could avoid this configuration, but this type of proxy is limited to the traffic of one network only. The approach adopted by WebQuilt [9] differs from the traditional HTTP proxy, since it is based on the URL which redirected all the links in a way that the URLs themselves point to the proxy and the intended destination is coded inside the URL string. Thus, the user does not have to do a configuration in his/her browser to send all requisitions to the proxy. However, the WebQuilt does not capture the interaction in the client-side; only information about the visited pages is available to the proxy. Especially, the focus of WebQuilt is information visualization technique applied to the data.

The approach adopted by UsaProxy [10], a HTTP proxy, consists of intercepting the traffic and registering data on any request sent to the web server as well as the answer that the web server sends back to the client. UsaProxy adds a JavaScript code to each requested page before delivering it to the client. This code captures the user interaction with the page such as mouse move and click, page scroll, window resize, focus change, and key press. Data are sent periodically to the proxy to be stored in a log file. As the inclusion of the JavaScript in the page only takes place when the page content is a HTML text (text/html), the UsaProxy approach does not work with the HyperText Transfer Protocol Secure (HTTPS), since the content is encrypted.

WAUTER [11], WebHint [12], and Web Usability Probe [13] are similar to UsaProxy. Actually, the WebHint uses the UsaProxy to capture user interaction. WAUTER also intercepts the HTML code sent to the client and inserts a JavaScript to capture user interaction. The difference is that the proxy is installed in the client’s machine to overcome the practical question of web server access. Thus, questions about the need to install software in the user’s equipment and the working with HTTPS still remain unanswered. In addition, WALTER and WebHint do
not capture data related to the attributes of HTML elements and style sheets (CSS).

The capturing of the user interaction can also be done through the client. For instance, the WebTracker [14] is used to capture user interaction with the browser. This tool captures the selections made in the menus and toolbars (buttons), and the actions realized by the keyboard. These interactions are associated with the open page in the browser plus date and time. The scope of the interaction captured by the WebTracker is limited to the browser’s functionalities and visited pages; mouse clicks are only captured when a link is selected on the open page in the browser. Another tool, called Wrapper [15], registers the user interactions with the browser and several other actions such as adding to favorites, copy, print, save and scroll bar use.

The problem with these tools (that capture through the client) lies in the granularity level of the obtained information which is too high for RIAs, considering that the interest is to get detailed information at the level of the page elements and events associated with user interaction. Even when the user interaction is obtained through a program installed in the user’s equipment, such as the WebLogger [16], which captures page scroll and mouse move, it is necessary to process the existence of a click in the coordinates associated with the desired element to obtain data on the page elements that were used. Ultimately, the need to install these applications in the client’s equipment turns into a highly intrusive process for the user.

WebRemUSINE [17], WELFIT [18] and USABILICS [19] can also capture user interaction through the client. These tools prevent the installation of software in the user’s machine since they use a JavaScript code inserted in the pages by the application developer himself. In WebRemUSINE, the script does the capture and sends the data to an applet Java responsible for forwarding to the web server at the end of the session (indicated explicitly by the user). In WELFIT and USABILICS, data are sent by the JavaScript code itself.

Information captured by the WebRemUSINE, WELFIT and USABILICS provides lots of details about the user interaction with the application. In general, the JavaScript code redefines the event handlers to capture the user interaction. An event handler is a part of a code associated with the interaction object. Captured data are related to interaction objects that, when selected or manipulated by the user, triggering events corresponding to the actions. WebRemUSINE, WELFIT and USABILICS are limited to standard event handlers of the JavaScript language and do not capture events defined by the user (application developer). Moreover, the WELFIT does not capture data related to the style sheets attributes (CSS).

This work uses the version of the WAUTT which also captures user interaction through the client. As WebRemUSINE, WELFIT and USABILICS, the capture occurs through a JavaScript code inserted in the RIA’s pages. However, besides capturing interaction related to standard events, WAUTT also register other events to be captured and that were created by the developer to deal with specific interactions or are defined by some JavaScript library used in the RIA.

Despite advances of the tools for capturing user interaction with rich internet applications, studies on the use of such tools by web developers are not available. An evaluation with web developers must consider an investigation about tool usage and the information is considered useful for usability and/or behavior analyses.

**Capturing User Interaction with RIA Using WAUTT**

The use of server log files to support the evaluation of web systems is a convenient and reliable strategy [5]. In general, a server log file entry contains client IP address, request date/time, requested page (URL), HTTP code, requisition protocol, and size of the requested file. The server log files have a limitation because they contain information about the pages (URLs) accessed by the user and they do not offer detailed data about the real user interaction with a RIA. Data on user interaction that can be stored in the web server are not enough to extract detailed information on the RIA’s real use.

The WAUTT [2] provides the “wautt.js” script for capturing user interaction. The technology adopted for the RIA is transparent to WAUTT, since the developer inserts a HTML code in the pages, which includes the remote file “wautt.js” available in the WAUTT server. Therefore, all logic that involves the capture of the interaction and data forwarding to the WAUTT server is centralized in a single place and only referenced in the RIA. This approach is
similar to that adopted by Google Analytics (www.google.com/analytics/) and other tools that capture user interaction through the client [17] [18] [19].

The information captured by WAUTT provides lots of details about the user interaction with a RIA. The collected data are stored in a relational database and can be accessed via the administrative interface, as shown in FIGURE 1. Each user interaction registered can be associated with the following data groups:

-Session data. This data group includes the session identifier, browser and operational system (name and version), and the IP address. A session involves a sequence of actions of the same user during the use of the monitored application. A session is created when a user opens the browser and loads the application, and is concluded when the browser is closed.

- Browsing data. This data group includes the page title, URL, parameters, request method (GET or POST), and page size. Browsing involves the sequence of the pages loaded during the use of the application. While in some applications the user interacts with a single page without refreshing it completely, in other applications several pages are loaded.

- Event data. This data group includes specific details of the action performed by the user in the interface and varies according to the event type. These data allow identifying which (event type and details) and when (time) the interaction took place.

- DOM data. This data group includes the name of the page element and its attributes. For instance, the attributes are common to all elements such as id, class, title, and specific attributes (according to element type) such as type, value and name of an element of the type input.

- Task data. This data group refers to the active task, if the resource of distinct tasks is being used. This information allows the realization of the analysis of the captured data according to the task.

An advantage of capturing user interaction is the possibility to collect large volumes of data automatically. However, adequate tools are needed to help in the quantitative analysis and interpretation of the collected data. After using WAUTT to capture user interaction, the collected data are made available in the WAUTT server. WAUTT summarizes some basic information related to the captured data: session details, number of open pages by session, graph of data by period of time, total of data by event, page element, task and session. In addition, WAUTT also allows exporting data in three different formats (JSON, XML e CSV) to be submitted to analyses by specific tools.

**Evaluation with Web Developers**

User testing is a type of applied experimentation in which developers check that the system being developed is usable by the intended user population for their tasks [5]. In the case of WAUTT, its intended users are web developers that work with development of RIAs.
The evaluation with web developers was carried out in agreement with the recommendations of ISO/IEC standards for software product evaluation, which divides the process into four stages: specification, planning, pilot evaluation, and evaluation. Next subsections describe this evaluation with web developers. The results are presented in the next section.

**Specification**

In specification stage, the scope and objective of the evaluation were defined. The scope of the evaluation was the WAUTT interface and its functionalities which implement the capture and analysis of user interaction with RIA. The goals were to identify preferences in using WAUTT and whether web developers could have difficulties to use the captured information for usability evaluation purposes. In addition, an underlying objective was to identify other user interaction data that tools could collect to support RIA development and evaluation.

**Planning**

In planning stage, an initial plan was developed to carry out the evaluation. The work to produce and execute the user testing was estimated from this initial plan which includes the following:

- Definition of the target group of users (participants);
- Set of tasks to be completed by participants;
- Description of the material to be produced;
- Guide for conducting the evaluation.

The target group of users considered to participate of the evaluation is composed by developers with experience in development of RIAs and academic background in computing (graduate or undergraduate). In addition, a test monitor was involved to conduct the evaluation. The participants were recruited from software development companies using unintentional sampling. In summary, the process of selecting participants was non-probability sampling for convenience.

The set of tasks to be completed by participants comprises four tasks regarding using the tool for capturing user interaction with a RIA:

- Task 1: make the registration in WAUTT server of the target application (free choice) and download the script for capturing user interaction. The participant was informed that the interface of the target application must be rich in terms of interaction elements and events that can be captured.
- Task 2: include the script downloaded in the pages of the target application.
- Task 3: browse the target application to collect data concerning user interaction.
- Task 4: use the functionalities provided by WAUTT to view and analyze the details of the collected data through data filtering and reporting.

Tasks 1 and 2 refer to the preparation of a RIA to have the user interaction captured by WAUTT. These tasks are important to investigate the understanding of web developers (participants) concerning the tool principle for capturing user interaction. In short, the preparation consisted of the registration of the target application in the WAUTT server and the inclusion of instructions in the HTML code of the pages is similar to the following:

```html
<script src="http://lab.utfpr.edu.br/wautt/js/start.php?appid=2a7725e1a20f84b538e6b8a1e238b95c"></script>
</script>
```

Task 3 is useful for web developers to understand what types of interaction can be captured. Participants followed the generation of log file entries concomitantly with its interaction with the RIA (through functionality provided by WAUTT). Task 4 refers to the understanding in using collected data to get some useful information about the RIA usage. Web developers should be able to relate collected data with some evidence of problems or behaviors (desired or not).

The tasks were presented to the participants in agreement with the concept of simulated work task situation [20]. A simulated work task situation is an open description of the context/scenario of a given situation. It then works as the trigger of the participant's goal and the base for judgment. The objective is to ensure the largest possible
realism by the involvement of web developers who, based on the simulated work task situation, develop individual and subjective interpretations of the user tasks.

The material prepared to be distributed for participants includes: consent form to participate in the test; description of the WAUTT; survey questionnaire; description of user tasks (scenarios); and post-test questionnaire.

The primary purpose of the consent form is to let the participants know the goals of the test and to explain what will happen to the findings as well as guarantee privacy and confidentiality of information. In addition, this consent form also works as an agreement between evaluator and participants to confirm the professional commitment of serious participation in the evaluation [5].

The descriptive material relating to WAUTT enabled participants familiar with this tool and its purpose. A brief presentation was also offered to the participants for introducing an overview of the proposed user testing.

The survey questionnaire was designed for characterizing the participants. The questions was used to collect basic demographic information (for instance, gender and age) and details about participant’s experience regarding web development, integrated development environments, database management systems, and web analytics techniques.

The post-test questionnaire was designed for collecting subjective judgments on WAUTT usage and it has five questions in Likert scale, one multiple choice question, and two open questions. Likert is a bipolar scaling method, measuring either positive or negative response to a statement. A seven-point scale was used in the post-test questionnaire (1: very strongly disagree; 2: strongly disagree; 3: disagree; 4: neutral; 5: agree; 6: strongly agree; 7: very strongly agree).

A guide with precise instructions for all procedures needed to user testing was also designed. This guide aimed to minimize the influence on task performance of different procedures in conducting several sessions of user testing.

**Pilot Evaluation**

In pilot evaluation stage, a complete user testing with two users was carried out to identify problems with the planning [5]. During this stage, the following criteria were analyzed: (a) coverage and quality of the user tasks; and (b) sufficiency and quality of the descriptive material and presentation. At the end of pilot evaluation, only minor problems concerning ambiguity in user task descriptions were encountered.

**Evaluation**

The user testing was carried out with 30 participants randomly selected in agreement with the target group of users defined in the planning stage. The participants made the tasks in laboratory conditions and in a controlled way (with a test monitor). However, there was no supervision in the sense of aiding the participants to complete the tasks. The chosen environment was a usability laboratory and its choice aimed to minimize the influence of the environment on task performance.

**Results and Discussion**

This section presents initially the characterization of the participants as potential users of a tool for capturing user interaction. Following, the data collected are analyzed and summarized using descriptive statistics. The results concerning the tasks are first presented, and then the results concerning the post-test questionnaire are presented.

**Participants**

The survey questionnaire applied at the beginning of the user testing shows that the selected participants have an average of 2.25 years of experience in web developing. The participants (86.7%) have advanced knowledge of HTML and CSS as well as programming languages for web development (73.3%), most of them (79.9%) prefers JSP and ASP. The participants also have intermediate knowledge of integrated development environments (IDE) and database management systems (DBMS) (70% and 63.3% respectively). All the participants have used some methods of web analytics, server log file analysis (80%), page tagging (56.7%), other (16.7%). Therefore, the participants can be considered potential users of tools for capturing user interaction.
Tasks

The analysis of the time to completion tasks may indicate if participants have encountered difficulties for completing the tasks. Time to completion is the calculated amount of time required for any particular task to be completed. Table 1 presents the results concerning each task. Analysis of data dispersion revealed a significant standard deviation (SD) and wide total range − minimum (Min) and maximum (Max) values. The wide variability found can be explained, among other factors, by the use of the concept of simulated work task situation in the task descriptions, which allow the participants to take different paths to complete the task. However, this concept was used to ensure realism and commitment through the participant’s involvement in the task.

The results for Task 2, Task 3 and Task 4 present high variability for time to completion (observed by standard deviation in Table 1). However, the lower variability for Task 1 can be due to characteristics of the task: make the registration and download the script. The participants’ behavior tends to be more homogeneous because Task 1 is simple. Dispersion in the values was specially expected in Task 3 and Task 4 because some participants tend to explore all options while others have focus on completing the task. Therefore, the variability for Task 3 and Task 4 is mainly influenced by its exploratory nature: browsing a web application; and using the functionalities provided by a tool.

An outlier participant was identified for Task 1 and Task 2 who also presented high time to completion for Task 3 and Task 4. If the outlier is due to a strange or rare event that never will happen again, the data from that participant could be excluded (for example, due to wrong data). However, because the case of a participant takes more time for completing the tasks which is an event that may occur again in user testing, it is not advisable to exclude the data from the analysis.

The description of the tasks could be adjusted to avoid that high variability through increasing the level of detail of the actions was to be accomplished by the participant. A more homogeneous behavior of the participants is expected when they follow detailed instructions on what to do in the tasks. However, the freedom of exploration was important to obtain subjective judgments of the participant regarding his/her experience. The subjective judgments were obtained from the post-test questionnaire whose results are presented in the sequence.

Post-test Questionnaire

At the end of user testing, a set of questions was presented to the participants to rate their experience. This post-test questionnaire was used to obtain different categories of information: understandability, operability, data analysis functionality, future re-use, and satisfaction. Table 2 presents a summary of the post-test questions and results.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1. I found the tool for capturing user interaction easy to understand</td>
<td>3.3%</td>
</tr>
<tr>
<td>Q2. I found the tool for capturing user interaction ease of use (operate)</td>
<td>3.3%</td>
</tr>
<tr>
<td>Q3. I found the various functionalities in the tool were useful for data analysis</td>
<td>10.0%</td>
</tr>
<tr>
<td>Q4. I would like to use the tool for capturing user interaction in the future</td>
<td>3.3%</td>
</tr>
<tr>
<td>Q5. I felt very satisfied using the tool for capturing user interaction</td>
<td>3.3%</td>
</tr>
</tbody>
</table>

TABLE 1 TIME TO COMPLETION FOR EACH TASK; VALUES IN MINUTES AND SECONDS

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2min 05s</td>
<td>0min 58s</td>
<td>1min 12s</td>
<td>4min 03s</td>
</tr>
<tr>
<td>2</td>
<td>2min 25s</td>
<td>1min 16s</td>
<td>1min 31s</td>
<td>6min 10s</td>
</tr>
<tr>
<td>3</td>
<td>5min 49s</td>
<td>2min 27s</td>
<td>1min 54s</td>
<td>9min 42s</td>
</tr>
<tr>
<td>4</td>
<td>3min 52s</td>
<td>1min 26s</td>
<td>2min 27s</td>
<td>6min 55s</td>
</tr>
</tbody>
</table>

TABLE 2 SUMMARY OF THE POST-TEST QUESTIONS AND RESULTS
The most of participants expressed agree (5), strongly agree (6), or very strongly agree (7) for the five questions (96.7%, 93.3%, 90%, 70%, and 83.3% of participants, respectively for questions Q1, Q2, Q3, Q4 and Q5). These results indicate that the participants have good understanding of the tool, found easiness to operate the tool for capturing and analyzing the user interaction, consider its future re-use, and they felt satisfied with the experience. The median for the five questions (5.5, 6.0, 6.0, 5.0, and 5.0, respectively for questions Q1, Q2, Q3, Q4 and Q5) and mode (5 for all questions) also provides support for these results. However, at last one participant (3.3%) did not consider the future re-use of the tool (Q4) and he/she was not satisfied with the experience (Q5).

The multiple choice question of the post-test questionnaire was used for investigating what user interaction data are considered useful for identifying usability problems in rich internet applications. The question allowed the participants to choose several among the following options:

- Page load (and URL);
- Window resize;
- Page scroll;
- Mouse move (and x-y coordinates);
- Mouse click (and x-y coordinates);
- Mouse click in page element (and attributes);
- Key press;
- Focus change in page element (and attributes).

The most chosen options by the participants were the focus change, mouse click in page element, and page load. Those preferences are directly related with the knowledge of the path taken by the user in the interaction with a web application to accomplish a task. That knowledge is important, for instance, to verify if the user uses the sequence of interaction considered ideal by the web developer for using determined functionality. Additionally, a discerning analysis of those data can identify which areas are more or less visited or still which functionalities are used with greater or smaller frequency.

An open question was used to identify which other information the participants considers important to be collected through a tool for capturing user interaction. The information more mentioned is related with the time. For instance, page load time (or page speed), time elapsed until the first interaction, permanence time in the page, time between two events, feedback time, response time of the client-server communication, among other measures of time related to rich internet applications.

Participants’ concern with time measures is comprehensible, once time is a critical factor for web applications. For example, the users’ satisfaction increases according to response time decreases, and small variations are acceptable.
In contrast, great variations can affect the user’s behavior.

Another open question approaches which other resources can be used for analysis and interpretation of the data captured from user interaction, besides those provided by the tool used in the evaluation. The main resources mentioned by participants are summarized by the following:

- Identify user interaction sequences which are repeated;
- Provide more graphical resources;
- Search by a user interaction sequence informed;
- Provide customized reports and graphics.

Ultimately, the suggestions presented by participants indicate the application of pattern recognition and information visualization techniques.

Conclusions

The results of this work indicate that the participants have good understanding of the tool, found easiness to operate the tool for capturing and analyzing the user interaction, consider its future re-use, and they felt satisfied with the experience. The results also assisted for identifying the user interaction data considered more useful to web developers. In addition, the knowledge obtained from this evaluation allows addressing future works for the improvement of WAUTT in the sense of supplying time measures related to user interaction with rich internet applications and new resources to aid the analysis of the data. Other future works include the planning and realization of more tests with users to investigate in more depth the user acceptance regarding the use of tools for capturing and analyzing user interaction data in the level of page elements and events.

REFERENCES


