

Analysis and Visualization of Interactions with Mobile Web Applications

Paolo Burzacca, Fabio Paternò

► **To cite this version:**

Paolo Burzacca, Fabio Paternò. Analysis and Visualization of Interactions with Mobile Web Applications. David Hutchison; Takeo Kanade; Madhu Sudan; Demetri Terzopoulos; Doug Tygar; Moshe Y. Vardi; Gerhard Weikum; Paula Kotzé; Gary Marsden; Gitte Lindgaard; Janet Wesson; Marco Winckler; Josef Kittler; Jon M. Kleinberg; Friedemann Mattern; John C. Mitchell; Moni Naor; Oscar Nierstrasz; C. Pandu Rangan; Bernhard Steffen. 14th International Conference on Human-Computer Interaction (INTERACT), Sep 2013, Cape Town, South Africa. Springer, Lecture Notes in Computer Science, LNCS-8120 (Part IV), pp.515-522, 2013, Human-Computer Interaction – INTERACT 2013. <10.1007/978-3-642-40498-6_40>. <hal-01510563>

HAL Id: hal-01510563

<https://hal.inria.fr/hal-01510563>

Submitted on 19 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Analysis and Visualization of Interactions with Mobile Web Applications

Paolo Burzacca, Fabio Paternò

CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy

{paolo.burzacca, fabio.paterno}@isti.cnr.it

Abstract. We present a novel solution for intelligent analysis and visualization of user interactions with Web applications through mobile devices in order to help identify usability issues. The proposed tool is also able to support comparison of optimal use with actual user interactions. We also report on an example application of our tool to the evaluation of a real mobile Web site.

Keywords: Tools for Remote Evaluation, Web Applications, Mobile HCI

1 Introduction

Logging tools for supporting usability evaluation have long been investigated in the context of desktop access. However, such tools cannot be simply re-proposed as they are for mobile devices, given their specific characteristics in terms of relevant events and usability issues. Indeed, mobile devices are becoming ever richer in terms of sensors, such as accelerometers, GPS, and so on. In addition, because of the limited screen size mobile devices have specific usability problems, such as tedious activities in zooming in and out for viewing the desired piece of information or touch-based interactions that select the wrong elements [3].

Remote evaluation supports the analysis of the user behavior in real contexts of use, and this is important with mobile applications that can be accessed in varying contexts. Various tools for capturing logs of user interactions have been proposed but they were limited in various aspects. Thus, there is a need for improvements in tools able to support remote usability evaluation, in particular when mobile devices are considered. For example, WebQuilt [6] only performed proxy-based logging and, thus, it was not able to gather detailed information regarding user interactions and the surrounding environment. Other approaches exploiting client-based logging [1][5] provided reports of the data collected not easy to interpret. The evaluation of the interactions with mobile application by comparing task models with the logs associated with real use was investigated in [5]. In the tool that we present here we propose a novel approach in which the logs of the actual use are compared with an optimal log created by a user interface expert.

2 The Environment for Remote Evaluation

The proposed environment aims to support remote evaluation of Web applications accessed through any type of device. In this paper we focus on the specific aspects related to mobile access. In order to perform a usability evaluation there are three main phases: *preparation*, during which the evaluators define the tasks that should be performed during the test and provide examples of optimal performance of such tasks; *test*, during which the users access the application from any device through a proxy server that includes the logging scripts and stores the logs in the backend; and the final *usability analysis*, which is based on the information collected beforehand and exploits visualizations that facilitate the identification of potential usability problems.

The usability server includes a proxy through which users should access the Web applications. Such logging can detect any type of event, even events defined by the evaluators. The backend part supports functionalities able to provide useful visualizations of logs through interactive, configurable timelines, and intelligent comparison of actual behaviour with optimal interaction sequences.

One of the activities to carry out during the preparation phase of a user test by the evaluators is the creation of the task list. For each task the evaluator has to provide the name, description, if it has to be started from a specific page then its URL should be specified, whether it can be skipped, whether its performance depends on some other task. For the interaction with end users the choice was to interfere as less as possible with the access to the applications. Thus, the only interaction is given by the task panel, which provides some basic instructions at the beginning of the session, and is used to indicate the start and the end of a task performance, and the next task to accomplish. Thus, the user starts the test through the public user interface, and the proxy includes the control panel, which contains instructions regarding the task, and it allows users to start/finish/jump the task, with minimal intrusiveness.

In our environment the logging is performed through an infrastructure that has been designed for this purpose. We use a log model in which each event is defined by `<timestamp, element, id, event, which, extra>`, *timestamp* indicates when the event occurred; *element* is the DOM element associated to the event; *id* is the unique identifier of the element; *event* indicates the event type (e.g. click, touchstart, scroll,...); *which* indicates the value of the event object (e.g. what character has been entered when pressing a key or what mouse button has been pressed); *extra* is used to provide additional information, when necessary, such as URL page and window size for the on load page event, GPS coordinates for GPS events, screen coordinates for mousemove or click events. The events recorded include: all the standard events, touch events, events related to accelerometer and GPS, DOM Mutation events, basic semantic events (pageview, starttask, closetask, ...). We also manage semantic events that are used in our environment in order to provide more information that can be useful for the evaluators. Thus, we have a set of basic semantic events mainly associated with the task panel access or the task performance. In addition, it is possible to define custom semantic events, which are composition of basic events or standard events on specific parameters with specific event names.

3 The Timelines and Other Visual Information

Timelines are used to represent information gathered from the logs. Previous solutions [1] have provided rather static and limited representations of such timelines, we have thus designed new solutions to represent and manipulate them.

The tool allows evaluators to access the list of available timelines from a given user test ordered by time of the corresponding log files. The first timeline is that generated by the evaluator to store an optimal performance of the given task (Figure 1 shows an example), and it is used as a comparison reference for assessing the performance of the various users, and identifying possible usability problems. Each timeline expands temporally from left to right. For each timeline some useful data are shown: an identifier, an incremental number associated with the corresponding task, when the corresponding log was detected, and the overall duration of the user session. In addition, some information regarding the device used is provided. For this purpose, during the user test the environment takes the user agent and uses it to retrieve more detailed information from a Device Description Repository (we have used WURFL¹).

Our environment provides the evaluator with various interactive functionalities useful to ease the comparison of logs and their analysis. The timeline visualization is completely dynamic and interactive. All timelines and events can be selected in order to apply specific operations to them. The initial representation of the timeline shows all the events in the log. This can be confusing and complicate the analysis because not all the events may be relevant. Thus, the tool provides evaluators with the possibility of interactively filtering the events through a panel with one check-box for each event category (e.g. form events, keyboard events, touch events, ..).

The events shown are represented through a label and a coloured small circle. The timelines also include some vertical black lines among the events indicating when a page change has occurred. When an event is selected then the corresponding information (e.g. when it occurred, in which element it occurred, ..) is shown. It is also possible to search for a given event, in this case only the occurrences of such event are shown. In case multiple timelines are shown then they are lined according to the first occurrence of the searched event. In order to facilitate the comparison of various logs the tool allows the evaluator to line up the timelines according to when users started to navigate on a given page as well (Figure 1 shows an example). This facilitates the comparison of how different users interacted within the same page. The tool has a zoom feature for adjusting the time scale, thereby allowing the evaluator to expand or contract the events in the timelines. Indeed, it often happens to find many events grouped in very small time intervals, and thus they have overlapping, confusing representations, which can be improved through this feature.

While the timelines provide an effective interactive dynamic representations of the users' sessions, we also introduced an event analyzer with the goal to provide an overall summarizing representation of what happened during the test. The evaluator can select an event category and the tool shows a list of bar charts showing the number of occurrences of such events for each session.

¹ <http://wurfl.sourceforge.net/>

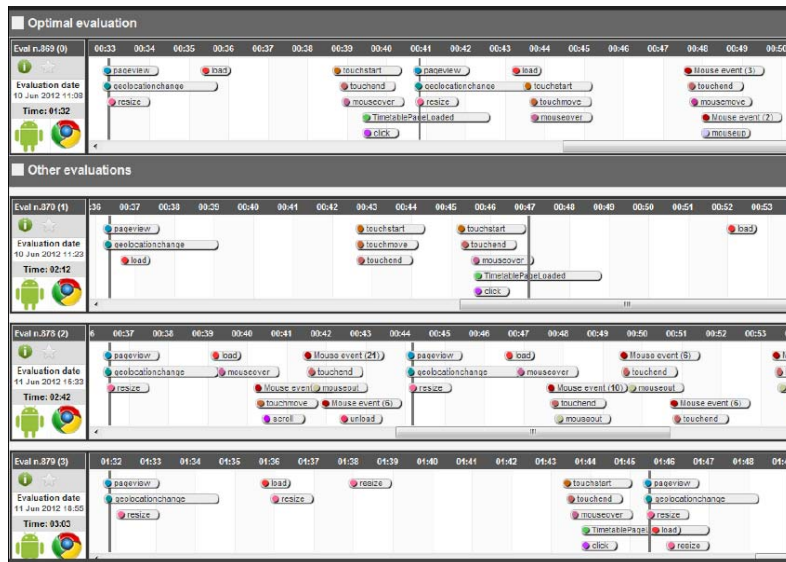


Fig. 1. An example of a set of timelines aligned according to one page access.

The tool also provides the possibility of more explicit comparisons among various sessions. For example, it is possible to compare through bar charts the navigation time across the various pages that have been accessed by the considered sessions.

In addition to time-related information, another useful piece of information is the path followed by the user during the navigation. It is not immediate and intuitive to extract such information from the timelines. Thus, our environment also provides the possibility to show visual storyboards, which also easily highlight possible cycles during the navigation. As you can see in Figure 2, each page is represented by a rectangle with an identifier, the visit time, and an arrow indicating the next page accessed.

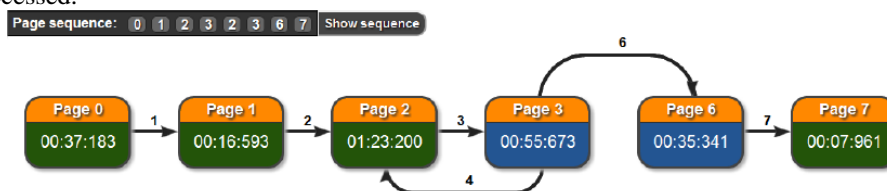


Fig. 2. An example representation provided by the visual storyboard

If a page is visited multiple times then the time is the sum of all the visit times. The arrows are annotated with numbers indicating their temporal order. The rectangles indicating the pages navigated have different colours depending whether they have been accessed in all the considered sessions or not. For each page that can be analysed it is also possible to show its screen dump captured during the user session, graphically annotated in order to indicate where the user interactions occurred.

4 Automatic Log Analysis

In our environment we have also investigated the possibility of providing more automatic support in the log analysis, still through comparison of the user logs with the optimal log initially provided by the evaluator. For this purpose, we have applied the Sequence Alignment Method (SAM) to the comparison of logs of user interactions. Previously, it has been applied to identify navigation patterns in Web applications [2]. We have investigated how to exploit it in order to help identifying potential usability problems. In general, the SAM approach aims to calculate the differences among two sequences of elements. The difference between sequences is represented by the number of operations necessary to make them equal, where each operation has a different weight, which depends on its importance. The operations available for equalizing two sequences are reordering, insertion, and deletion.

In our logs we have information related to both pages accessed and events generated. Thus, when we apply the SAM analysis we distinguish the distance in terms of both types of elements. Actually, in the events we also distinguish between custom semantic events and the others. In the end, the distance between two logs is calculated by summing the distance of these three sequences, where the composing distances have different weights to reflect their different importance for the usability analysis. Indeed, the differences in pages accessed is more important, since if the users have visited pages different from those accessed in the optimal session, then they have probably not accomplished their task correctly. For the custom semantic events, most analysis is limited to check whether they occur in the sequence considered. For the other events they are considered the same if they occur in the same page and on the same user interface element. The weights associated with the elements compared and the type of SAM operation play an important role in determining the final result. Since it is difficult, and probably not possible, to find general weights that always provide the most meaningful results, we decided to allow evaluators to customize them. For example, in the page sequences comparison an evaluator should consider the application type considered and its navigation model: if the task can be accomplished through different paths within the Web application then the weights for the page sequences differences should be low because some differences are probably not indicating any particular problem. The tool also provides some useful information regarding the differences among the sequences. For example, in the case of the page sequences the tool also indicates the pages that differ in the two sessions. As Figure 3 shows, for such pages it also indicates the page previously accessed, which is useful to allow the evaluator to understand from which page the user has taken the wrong path.

Pages				SAM	
Operation	Num	Weight	Total		
Insertions	2	40	80	insertions • http://mobile.lufthansa.com/inf/start.jsp?l=en Source page: http://mobile.lufthansa.com/hpg/start.jsp?l=en • http://mobile.lufthansa.com/inf/relaxed.jsp?l=en Source page: http://mobile.lufthansa.com/inf/start.iso?l=en	110
Deletions	1	30	30		
Reorders	0	30	0		

Fig. 3. An example of SAM analysis report on the pages accessed

5 An Example Application

In order to validate our environment we considered an example application, the mobile version of an airline Web site². We first asked some end users to test it through our logging infrastructure and report on its usability issues, then we asked some evaluators to use the information provided by our tool to identify the application usability problems in order to check whether they would have found similar problems.

The application was accessed remotely by the users without any direct support from the evaluators. Users only received at the beginning some information by email about the purpose of the test and how to start it. Users were free to choose the mobile device used for the test and the exact time and location to carry it out.

The test was carried out by seven users, the average age was 32, and they had not used our tool beforehand. The test consisted in performing five tasks chosen in such a way to consider the various aspects characterising Web access. For each task we defined some custom semantic events to have more control in the analysis of the user logs. The first task was to search for a flight from Paris to London with intermediate stop in Frankfurt for a precise day in economy basic fare. In this way the users had to choose among various options and go through various steps to complete the task. We defined a custom semantic event with the selection of the One Way option to better check that users selected it. The second task was to find information about the lounge in Frankfurt. Here the most difficult part was to find the page regarding lounges. Actually, for this purpose two different paths across the Web site were possible, and we included two custom semantic events to easily identify which one was taken. The third task was to find the time for a flight from Munich to Cracow in a given date. The flight should have been direct, and the last available on the indicated date. Also in this case we introduced a couple of custom semantic events. The fourth task was to find how to reach the Frankfurt airport from Strasburg by bus. In this task there was no form to fill in. The fifth task was to find information about the Boeing 737-300. We introduced one custom semantic event to check the path followed.

Regarding the first task, some users complained that the Web site had the one way option clearly visible and highlighted, thus erroneously leading them to think that it was selected, and they only realized that this was not the case when the application asked them to select the return date. In the second task one user indicated a problem with a pull-down menu to select the class because it was displayed partly off the screen. In the third task the users had some difficulties in specifying all the options requested. For the fourth task the only problem seemed to be the long navigation required to reach the page to complete it. Similar issues were reported also for the fifth task in which some users found complications because of a misleading link label.

Regarding the use of our environment for the test, the users generally appreciated its ease of use and lack of intrusiveness. They only found that the automatic creation of the screendump at each page change caused some delay in the interaction. Some users suggested the possibility of creating more detailed user profiles, which can be considered during the analysis phase.

² <http://mobile.lufthansa.com>

Then, we compared the direct user feedback regarding the mobile application with results obtained with the support of our tool. We thus asked a group with knowledge in usability evaluation to use it in order to identify possible usability problems. We provided them with a short tutorial regarding the tool. We then briefly explained how its various features can be exploited. Then, the evaluators considered each task and the associated user logs in order to identify possible usability problems. We involved six people (three males and three females), average age was 34, all of them with experience in usability evaluation for both desktop and mobile applications, mainly through laboratory user tests or feedback through interviews, questionnaires, and focus groups, thus with little experience in remote usability evaluation, even if they found the approach useful and interesting. They had never used our tool beforehand.

In general, the evaluators found the tool user interface clear and well-structured. They liked the timelines management and the possibility of searching events and pages in them. The evaluators considered how all the users carried out each task through the information provided by the tool. For the first task (Find flight from Paris to London) they soon noticed some long session durations. The occurrence of the second custom semantic event revealed some issues, since it often appeared consistently later than in the optimal session. Likewise, the last custom semantic event associated with the correct flight selection often occurred in an anomalous manner. Some evaluators used the possibility of lining up the timelines according to the pages accessed in the optimal session. This was useful to confirm that in the access to the first two pages the sessions were similar. However, at the page related to the flight selection, the users' behaviour became substantially different. For example, while in the optimal selection the one way option was immediately selected, in some user sessions the user triggered various keyboard events beforehand. This implies that such users made a search without selecting the one way option and then got the request from the application to indicate the date for the return flight. In this case, the involved users either went back to the previous page or stopped their activities for a while. Then, in the end the one way selection event occurred, indicating that they eventually understood the mistake.

In the analysis of the second task (Find information about the lounge in Frankfurt) the evaluators noticed that the users all started from the same page and finished at the right one, but the intermediate pages differed. Indeed, when they searched for the custom semantic event associated with the loading of the page Miles More, they noticed that only one user passed through it. Such differences were well highlighted by the storyboard. The usability issue was that there were two possible ways to accomplish the task, and the shorter was not immediately apparent to the users. Indeed, the path followed in the optimal session requires a link selection (Miles & More) to access information about the lounges, and this was not intuitive.

In the analysis of the third task (Find the flight from Munich to Cracow) the evaluators' analyses focused on the search for the occurrence of the Timetable Page loaded event. Only one user missed that event. Then, the evaluators compared the time taken from when the corresponding page was loaded to when that specific event occurred. For this purpose, some of them combined the search for pages with that for events so as to filter the events to display only the relevant one and align the display

according to the flight timetable page in order to compare how long users visited that page. In order to analyse the use of the pull-down menus, the evaluators focused the analysis on the change event in the timetable page and it was thus possible to detect that the use of these elements was not optimal. Some users did not use the filters to narrow the query results and so it took longer to find the requested piece of information. In particular, the usability problem detected was the lack of meaningful labels in the form elements for filtering according to time and flight type.

For the accomplishment of the last task (Find information on Boeing 737-300) the evaluators noticed that the optimal session took about 90 seconds going through six pages while the user sessions usually took about the double of the time. They compared the paths followed in the navigation by the users and found that they sometimes took the wrong direction and then they had to get back to find the right one. The analysis moved to the screendumps of the pages where such deviations occurred and found that users had to select the link "On Board" to access the relevant part of the Web site but that text was not particularly meaningful to understand this.

In general, the use of the environment was satisfying, and improved over time. The comparison of the issues indicated by the end users and those detected in the evaluation through the tool provided similar results.

6 Conclusions and Future Work

We have presented novel solutions for the interactive visualization and analysis of logs associated with users interactions with Web mobile applications in order to better support their remote usability evaluation. We also carried out a user test on an example application that showed that the tool can be used in real remote tests.

Future work will be dedicated to further applying the environment to the evaluation of mobile Web applications, and investigating the integration of the intelligent log analysis with an automatic analysis of the accessed Web pages to check whether they have been implemented according to usability and accessibility guidelines.

References

1. Carta T., Paternò F., Santana W., Support for Remote Usability Evaluation of Web Mobile Applications, ACM SIGDOC, pp.129-136, ACM Press, Pisa, October 2011
2. Hay B., Wets G. and Vanhoof K., Mining Navigation Patterns Using a Sequence Alignment Method, Knowledge and Information Systems, 2004, Vol. 6, N 2, pp 150-163
3. Nebeling M., Speicher M. and Norrie M. C., W3Touch: Metrics-based Web Content Adaptation for Touch, to appear in Proceedings CHI'13, Paris.
4. Paternò, F., Russino, A., Santoro C., Remote Evaluation of Mobile Applications, Proceedings TAMODIA 07, Springer LNCS N.4849, pp, 155-169, September 2007.
5. Santana V. and Baranauskas M.. Summarizing observational client-side data to reveal web usage patterns. In SAC '10: Proceedings ACM SAC, 2010, pp. 1219-1223, ACM Press.
6. Waterson S., Landay J. A., Matthews T., In the lab and out in the wild: remote web usability testing for mobile devices. CHI Extended Abstract 2002, pp.796-797. ACM Press