A Prospect of Websites Evaluation Tools Based on Event Logs

Vagner Figuerêdo de Santana[1] **, and M. Cecilia C. Baranauskas**[2]

[1] Institute of Computing, UNICAMP, Brazil, v069306@dac.unicamp.br
[2] Institute of Computing, UNICAMP, Brazil, cecilia@ic.unicamp.br

Abstract: The variety of websites evaluation tools based on event logs is growing, but some of their limitations are already visible (e.g., the need for task model, plug-in dependency, use of simulated tasks, separation of accessibility from usability, etc). Some of these characteristics result in coupled systems and make the configuration and use of these tools more expensive. This work aims to show the main features and weaknesses of these tools. One expects that the discussion and the requirements pointed out in this paper could help developers of evaluation tools so they could reuse consolidated ideas and avoid identified weaknesses.

Keywords: website evaluation tools, log based evaluation, event loggers

1. Introduction

In the last years many researchers have been working on automatic and remote usability evaluation of user interfaces focusing on tests with more users without resulting in costly evaluations (Ivory, 2001). The total or partial use of automatic usability evaluation methods can reduce time and costs involved in the development of a Web application, as it liberates specialists from repetitive tasks as manual log analysis (Paganelli, 2002). While researchers have focused on usability issues, less has been done specifically to increase the accessibility level of the applications. In addition, the integration of Accessibility and Usability (A&U) concepts is not usual in user interface (UI) evaluation tools.

One of the aspects that make interesting the approach of automatic evaluation is the remote evaluation during real use, and involving users in their regular environments. In addition, remote analysis of A&U make unnecessary the presence of participants in the test places, which could be a barrier. Moreover, tests in controlled environments are artificial and may influence the results (Rubin, 1994). Thus, the UI evaluation in real situations points to the use of event logs, since it makes possible to capture data while users work on their regular tasks, in their usual environments (Guzdial, 1993).

Interface events have a duration ranging from 10 milliseconds to one second (Hilbert, 2000), showing that the number of events that occur during few minutes simple tasks can be huge. Thus, due to the typical amount of interface events, the use of automatic evaluation tools is usually necessary so that information extracted from the events reach a level of abstraction that is useful to specialists (Hilbert, 2000). This paper identifies solutions, limitations and gaps of website evaluation tools based on event logs; it is organized as follows: section 2 presents ideas well succeeded and the gaps identified; section 3 discusses the results of this work; section 4 presents conclusions and points to new directions.

2. Event logs and websites evaluation: identified solutions, limitations, and gaps

UI events are natural results of using windows based interfaces and their components (e.g., key strokes, mouse clicks) (Hilbert, 2000). Since it's possible to record these events and they indicate the user behavior during the interface usage, they represent an important source of information regarding usability. Event logs produce results as frequency of use of certain functions, places where users spend more time and the sequence they complete tasks (Woo, 2004).

According to Hong et al. (Hong, 2001), the goal of evaluation software based on event logs is to use a capture method that is easy to apply to any website, and that is compatible with various operating systems and web browsers. This section summarizes the main issues present in the following tools: WET (Etgen, 1999), WebRemUSINE (Paganelli, 2002), and WebQuilt (Hong, 2001). Also, we combine some ideas with other works related to the evaluation of websites as, for example, NAUTICUS (Correani, 2006), a tool that integrates A&U and points to a promising direction for new tools. Our analysis is organized according to eight aspects: the configuration needed to use the tools, the capture mechanisms, the ways logs are stored and transmitted, the dependence on users' actions, the changes needed at the UI, scalability, levels of abstraction, and integration of A&U. We present the solutions, limitations we found, and gaps to be investigated.

1) Regarding the configuration of the environment, i.e. the set of tasks needed for a tool to be used to evaluate a website, we can point out:

Solutions – Events capture on the client-side at WET depends only on the native script languages of the newer browsers (e.g., JavaScript), avoiding the need of plug-ins installation. Other solutions that make possible to capture across different websites without requiring any change on evaluated Web pages are the use of a proxy logger to capture server-side data (Hong, 2001), or the use of specific browser to capture client-side data (Claypool, 2001).

Limitations – Require maintainers of the evaluated website to change their Web pages so that the data capture starts, as in (Paganelli, 2002; Etgen, 1999); or that files must be hosted on the server of the evaluated websites (Etgen, 1999). Other important points are the dependency of task models (Paganelli, 2002) and dependency of specific software installed, as in (Paganelli, 2002; Claypool, 2001).

Gap - Events capture at client-side across different websites without requiring any change in the evaluated web pages or using specific browsers.

2) Regarding server-side vs. client-side capture, server-side capture concerns the use of logs generated only on the server, while client-side capture refers to data obtained through the recording of information available only on the client's device.

Solutions – Use of events triggered at client-side interface, because there are more detailed data sources of how the user uses a Web site (Paganelli, 2002; Etgen, 1999). Also, the use of a proxy logger to record all communication between the client's browser and the visited Web pages, without requiring any change in the evaluated pages or any plug-in installed in the client's device (Hong, 2001).

Limitations – The limitations related to the data capture deals with the storage of data too, because the amount of data obtained through events on the client-side is representative and the space available to store them in client's device is restricted, as in (Etgen, 1999). On the other hand, when there is space available on the client-side, the problem arises from the time spent on the log transmission at the end of a test session, as can occur in (Paganelli, 2002). An alternative would be to use only server-side data, but it can lead to less accurate results (Claypool, 2001).

Gap – The combination of server-side data and the client-side data, since server data may indicate events not present on the client-side (e.g., response times of accessing a link). On the other hand, client-side data have more detailed information about the user route in a Web page.

3) Storing and transmitting logs refers to strategies used to record event logs, including the location, data storage capacity, and the transfer methods to a server.

Solutions – Use of a proxy logger that captures the data about client requests and server responses (Hong, 2001). Thus, the data storage capacity depends on the tool server capacity and it's not necessary to send data to another element. To avoid the limit of available space in cookies, some components (e.g., Java applets) can be used to store and transmit client-side events (Paganelli, 2002).

Limitations – When recording data on a participant machine, the tool becomes dependent on the available space in that device, as in (Etgen, 1999). Moreover, the transmission of such data at the end of a test session, as in (Paganelli, 2002; Etgen, 1999), may require too much bandwidth connection, and thus interfere with the use of the interface. Another identified limitation deals with the use of software components that are plug-ins dependent (e.g., Java applets), as in (Paganelli, 2002). This can be an obstacle, since the user may not have knowledge to install the plug-in or doesn't have a broadband connection to download it quickly.

Gap – The use of cyclic transmission of event logs from client-side to the server, avoiding the limit of available space in cookies and the time required to transfer captured data in batch at the end of a test session.

4) By dependence on user actions we mean what is necessary from the participants for the tool to start the data capture (e.g., acceptance of the use).

Solution – The user accesses the first URL from the main page of the tool and then no further action is required until the end of the test session (Hong, 2001).

Limitation – Interfere the common use of the interface, requiring that the user select a task from a list or select options to start and stop the capture every test session. This may call the user's attention or even not allow that the data between these actions be captured, as in (Paganelli, 2002; Etgen, 1999).

Gap – An alternative to requiring a user action at the beginning of each test session is to make the acceptance of a test to be valid for several sessions, until the user chooses to stop the data capture.

5) Regarding the impacts that the evaluated UI suffers when a tool is in use, the key metric used is how the presentation of the tool interfere with the use of the website under evaluation.

Solutions – Display options to indicate the beginning and the end of capture (Etgen, 1999) or using a Web page, at the beginning of each session (Hong, 2001).

Limitations – The interface changes concern the reduction of the useful space for the UI to display any component of the evaluation tool, as occurs in the presentation of the list of tasks in (Paganelli, 2002). Another issue is the change of URLs used in the pages processed by the tool, as in (Hong, 2001).

Gap – Ensure that the tools' controls are always visible, don't compete with other UI elements, and don't reduce the useful area of the Web page.

6) Scalability refers to how the evaluation tools deal with many evaluation sessions or long test sessions.

Solutions – The use of specific software (e.g., web browser, applet Java) that can access participant's file system to record logs and then avoid data storage capacity of cookies, as in (Paganelli, 2002). Or use of a proxy logger to keep track of user requested data (Hong, 2001), depending only on the tool's server.

Limitation – The biggest barrier found was the space available for storage of logs in cookies, as it is not possible to record all usage, it forces the events captured to be a reduced sub set of existing events, as in (Etgen, 1999).

Gaps – Avoid limits of cookies without making the tool plug-ins dependent and avoid to transfer logged data only at the end the tests sessions.

7) As concerns the strategies used to achieve higher levels of abstraction from logs of events (e.g., implicit interest, use of grammars to relate events).

Solutions – Using task models and grammars that enable events of Web pages to be represented as higher levels actions (Paganelli, 2002) and conversion of low level events into sequences of actions (Hong, 2001).

Limitation – Dependency of specific task models and/or grammars to reach higher levels of abstraction, as in (Paganelli, 2002).

Gap – The use of methods that do not depend on task models to obtain higher level information. A possible alternative to these models is the sequence characterization based on the Markov Chains (Hilbert, 2000).

8) Finally, the integration of A&U is necessary, because if developers evaluate them separately, then some problems may appear regarding different priorities for guidelines interfering in the way the target audience uses an interface.

Solutions – Development and use of criteria covering aspects of A&U and evaluation of structure and content of pages (Correani, 2006).

Limitation – The identified limitation found in NAUTICUS is the dependence of a dictionary of terms commonly used in bad structured pages.

Gap – The integration of A&U was not found in websites evaluation tools based on logs, thus becoming the main gap of the studied tools.

3. Discussion

In this study we could identify characteristics that can strengthen evaluation tools, making them more robust, less costly, and easier to use and reuse. The integration of A&U has been identified as the main gap in the analyzed tools. We propose requirements that a logs based websites evaluation tool should have (Table 1). The requirements are organized in the Semiotic Ladder (SL), an artifact of Organizational Semiotics (Stamper, 1993) which provides a framework for analysis of information systems under six different layers. The SL structure allows the clarification of concepts and the analysis of information, considering from the Information Technology (IT) Platform (i.e., physical layer, empirical layer, and syntactic layer) to the Human information Functions (i.e., semantic layer, pragmatic layer, and social layer).

Table 1. Requirements for websites evaluation tools based instantiated in the SL

	Human Information Functions
Social	Focus on the integration of A&U for the target audience of the evaluated website. Enable remote testing during real use of the evaluated website. Interfere with the Web page as minimum as possible.
Pragmatic	Provide controls representing the status of the tool and user context during the test session. The tool should use two actions: one to start the capture, which stay valid for future sessions, another to interrupt the capture, which may occur at any time.
Semantic	Provide high levels of abstraction without depending on specific task models, grammars, or events.
	Information Technology Platform
Syntactic	Use all available data (e.g., client-side events and server-side logs) in order to obtain correlations between them. The combination of the available data in different components can reveal information impossible to obtain independently.
Empirical	Prevent that processing or transmitting logs interfere with the use of evaluated interface. The tool should implement safe and effective techniques without impacting on the website usage.
Physical	Do not depend on resources or specific configuration of the participants devices (e.g., disk space, bandwidth, etc). The evaluation tool should include mechanisms to achieve their goals in different configurations of hardware and software.

Analysis shows that the tools studied meet, mainly, the requirements associated with layers related to IT Platform (i.e., physical, empirical, and syntactic layers). The requirements of the Human Information Functions (i.e., semantic, pragmatic, and social layers) are hardly addressed. This suggests that new tools must take them into account to achieve goals related to these layers.

4. Conclusion

This paper presented a study of websites evaluation tools based on event logs, as well as different solutions to problems encountered at the event capture, log transmission, etc. The main results of this work are related to the identification of requirements and promising investigation subjects. Thus, developers of new websites evaluation tools can avoid the weaknesses and address some unexplored aspects such as the integration of A&U, for example. Based on the characteristics present in websites evaluation tools based on event logs, a set of requirements was proposed and have to be addressed to avoid the limitations discussed in this work.

The next steps of this investigation involves the development of a website evaluation tool based on logs that combine the solutions in order to fill as much as possible the identified gaps, taking into account the requirements listed.

References

Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: IUI ’01: Proceedings of the 6th Int. conference on intelligent user interfaces, New York, NY, USA, ACM (2001) 33–40.

Correani, F., Leporini, B., Paternò, F.: Automatic inspection-based support for obtaining usable web sites for vision-impaired users. Universal Access in the Information Society **5**(1) (2006).

Etgen, M., Cantor, J.: What does getting wet (web event-logging tool) mean for web usability? In: Proceedings of 5th Conference on Human Factors & the Web. (1999).

Guzdial, M.: Deriving software usage patterns from log files. Technical report, Georgia Institute of Technology (1993).

Hilbert, D.M., Redmiles, D.F.: Extracting usability information from user interface events. ACM Comput. Surv. Vol. 32 (4) (2000) 384–421.

Hong, I., J., Heer, J., Waterson, S., Landay, A., J.: Webquilt: A proxy-based approach to remote web usability testing. ACM Transactions on Information Systems 19 (3) (2001) 263–285.

Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. Vol. 33 (4) (2001) 470–516.

Paganelli, L., Paternò, F.: Intelligent analysis of user interactions with web applications. In: IUI ’02: Proceedings of the 7th int. conf. on intelligent user interfaces, ACM (2002) 111–118.

Rubin, J.: Handbook Of Usability Testing: How to plan, design, and conduct effective tests. 1st edn. John Wiley & Sons Inc (1994).

Stamper, R.: A semiotic theory of information and information systems / applied semiotics. In: Invited papers for the ICL/University of Newcastle Seminar on “Information”, (1993).

Woo, D., Mori, J.: Accessibility: A tool for usability evaluation. In Masoodian, M., Jones, S., Rogers, B., eds.: APCHI. Volume 3101 of LNCS, Springer (2004) 531–539.