Integration of Multiple Techniques to Support Usability Evaluation of Web Applications

Fabio Paternò & Laila Paganelli
CNUCE - C.N.R.
Pisa, Italy
{ fabio.paterno, laila.paganelli}@cnuce.cnr.it

Abstract

In this paper, we discuss some issues related to automatic support to usability evaluation. The discussion is based on our experience with WebRemUSINE, a tool that we have designed and developed in order to perform intelligent analysis of Web browser logs using the information contained in the task model of the application. This approach supports remote usability evaluation of Web sites.

Keywords

Remote evaluation, Usability, Task models, Tools.

INTRODUCTION

Creating a Web site allows millions of potential users with various goals and knowledge levels to access the information that it contains. For this reason, interest in usability evaluation of Web sites is rapidly increasing.

There are many motivations for automatic tools able to support the evaluation process [7]. The total or partial automation of usability evaluation can reduce the time and costs involved and release evaluators from repetitive and tedious tasks. A number of tools for usability evaluation of traditional graphical applications have been proposed. However, the different nature of Web interfaces requires specific tools. The goal is not to provide designers with an overall, definitive evaluation; rather, a more meaningful approach is to provide a number of pieces of information that can be helpful to evaluators and developers in order to improve their applications.

We have developed a method and an associated tool to detect usability problems in Web interfaces through a remote evaluation. Our approach combines two techniques that usually are applied separately: empirical testing and model-based evaluation. The reason for this integration is that models can be useful to detect usability problems but their use can be much more effective if they can be related to the actual use of a system. Our tool is able to analyse the possible inconsistency between the actual user interactions and the task model of the Web site that describes how its concrete design assumes that activities should be performed. To support remote evaluation, we have developed a technique that allows recording user actions during a site visit. The analysis of the logged data is based on the comparison of the traces of actions performed with the structure of the task model. This analysis provides evaluators with a number of results that are related to the tasks that users intend to perform and the Web pages and their mutual relationships.

RELATED WORKS

In recent years, interest in automatic support for usability evaluation of Web sites has been increasing. The methods for usability evaluation of Web sites can be classified into two types of approaches: empirical evaluation, where users are directly involved to some extent, and analytical evaluation where various combinations of criteria, guidelines and models are applied to the assessment of the site directly by the evaluators.

In the former group there are techniques based on the analysis of Web server logs whose effectiveness is strongly limited by the validity of the data (that cannot capture the accesses to the pages stored into the browser cache) and the impossibility to capture local user interaction with the user interface techniques (menus, buttons, fill in text, …). To overcome these limitations another approach, WebVip (Web Visual Instrumenter Program) [11], has been developed at NIST. This tool allows logging of user interactions and the resulting log files can be analysed through VISVIP [4] a graphical tool that visualises the paths followed by the users during the site visit. The logging tool proposed requires a number of modifications in the HTML pages that must be evaluated because each tag representing a user interface component calls for adding Javascript code to record the interaction. Because of the many modifications required, WebVip needs a copy of the entire site. Unfortunately copying the entire site can generate many problems. Also WET [5] considers client-side logs but they are obtained more efficiently without requiring copy of the entire site. In this case it is sufficient to include the javascript file in the heading of the page. This javascript file includes the specification of the events that can be detected and the handling functions able to capture them. In WET only the *click*, *change*, *mouseover* and *page load* events are recorded. This limitation is due also to the lack of automatic tools able to analyse the data. Since the analysis is performed manually it is important to have readable log files with content useful for the evaluator. Adding all possible events in the log file increases the complexity of its content since one user interaction can correspond to many events. Other techniques are based on the use of questionnaires but they have limited ability to find detailed usability problems.

In the latter group we have tools such as Bobby [1] that aims to support verification of application of accessibility guidelines. WebSat [11] provides a usability evaluation by analysing the HTML code through six categories of usability guidelines (accessibility, form use, performance, maintainability, navigation, readability). Design Advisor [6] is based on guidelines derived from the use of eye-tracking techniques that identify which interface elements attract user attention (animations, images, colours, …) by identifying the scanning path on the Web page. In Web Criteria [12] an expert user model is applied. Instead of real users, the simulated user always follows an ideal path where no errors are performed and the shortest path is always selected.

In our case we follow a hybrid approach because our environment is able to analyse data relative to user interactions and then compare them to the task model corresponding to the design of the Web site. To this end, the first issue we addressed was what type of log files to consider. To detect user actions we have used a technique similar to that used in Wet. Our tool exploits the possibility of defining handlers of browser events in order to record user interactions. The differences between Wet and our logging tool are two: Wet records only some event types because the analysis of the log files is manual whereas in our case the analysis is automatic and it is possible to record a wider set of events; in addition, to save log files WET uses cookies but with this technique it is possible to save only a limited amount of information (about 4K per cookie, with a maximum of 20 cookies per site) whereas our method uses a Java applet that is able to save log files in the server without limiting the amount of data, thus even with long user session there is no loss of data.

How user interactions can be considered as implicit interest indicators has also been demonstrated in [3] where a web browser was equipped to log user interactions, and the findings were compared with explicit ranking automatically requested of the users whenever they changed pages. By comparing the results, what emerged is that time on page, moving mouse, mouse clicks and scrolling are good indicators of user interest. Another interesting approach is discussed in [2]; it is based on the combined use of browser logging tools and eye trackers. However, this approach requires the use of devices that are still rather expensive.

THE METHOD

The solution that we have adopted in WebRemUSINE [9] to identify user intentions is to display the high-levels tasks that are supported by the Web site asking the user to indicate explicitly what task they want to perform. During the testing, since we perform remote evaluation without direct observation of the user interactions, it is important to obtain logs with detailed information. We have designed and implemented a logging tool able to record a set of actions wider than those contained in server logs. WebRemUSINE compares the logs with the task model and provides results regarding both the tasks and the Web pages supporting an analysis from both viewpoints.

The method is composed of three phases: *Preparation*, which consists of creating the task model of the Web site, collecting the logged data and defining the association between logged actions and basic tasks; *Automatic analysis*, where WebRemUSINE examines the logged data with the support of the task model and provides a number of results concerning the performed tasks, errors, loading time, *Evaluation*, the information generated is analysed by the evaluators to identify usability problems and possible improvements in the interface design.

The environment is mainly composed of three modules: the ConcurTaskTrees editor (publicly available at http://giove.cnuce.cnr.it/ctte.html); the logging tool that has been implemented by a combination of Javascript and Java applet to record user interactions; WebRemUSINE, a java tool able to perform an analysis of the files generated by the logging tool using the task model created with the CTTE tool.

Task models describe the activities to perform in order to reach user's goals. We have used the ConcurTaskTrees (CTT) [10] notation to specify them. This is a notation where it is possible to graphically represent the hierarchical logical structure of the task model. It is possible to specify a number of flexible temporal relationships among such tasks (concurrency, enabling, disabling, suspend-resume, order-independence, optionality, …) and for each task it is possible to indicate the objects that it manipulates and a number of attributes. The notation also allows designers to indicate how the performance of the task should be allocated (to the user, to the system, to their interaction) through different icons.

The logging tool is able to store various events detected by a browser. The Javascripts are encapsulated in the HTML pages and are executed by the browser. When the browser detects an event, it notifies the script for handling it. By exploiting this communication, the script can capture the events detected by the browser and add a temporal indication. Our tool works for the two main Web browsers (Micorosft IE and Netscape Communicator).

WebRemUSINE performs an automatic evaluation of a Web site providing the evaluator with a set of measures, concerning also group of users, useful to identify usability problems. The input for the tool are the task model and the log files recorded during the test sessions. WebRemUSINE is composed of two submodules:

- *The preparation module*, this module filters the information recorded during the testing, then the evaluator has to associate each basic task with the corresponding event. All the event-basic task associations are recorded in a file.

- *The evaluation module*, it has three inputs: the task model, the log files and the event-basic tasks associations. This information is useful to analyse the logs with the support of the task model and identify errors performed by the user during the navigation. By following the sequence of events stored in the log it is possible to identify the corresponding tasks (through the event-basic tasks association) and comparing the sequence with the temporal relationships among the tasks it is possible to identify the tasks performed correctly and those that generate errors. It is also possible to calculate the completion time for the relative tasks. All results are displayed by WebRemUSINE in various formats both textual and graphical.

The WebRemUSINE analysis can point out usability problems such as tasks with long performance or tasks not performed according the task model corresponding to the Web site design. These elements are useful to identify the pages that create problems to the user. As previously explained, log files store both user interactions (mouse movements, keyboard input, link selection) and browser behaviour (start and end of page loading). The events corresponding to user interactions are associated with interaction tasks whereas the internal browser events are associated with system tasks. Thus the evaluation performed provides information concerning both tasks and Web pages. WebRemUSINE also identifies the sequences of tasks performed and pages visited and is able to identify patterns of use, to evaluate if the user has performed the correct sequence of tasks according to the current goal and to count the useless actions performed. In addition, it is also able to indicate what tasks have been completed, those started but not completed and those never tried. This information is also useful for Web pages: never accessed web pages can indicate that either such pages are not interesting or that are difficult to reach. All the results can be provided for both a single user session and a group of sessions. The latter case is useful to understand if a certain problem occurs often or is limited to specific users in particular circumstances.

CONCLUSIONS

In this position paper we have discussed our solution to perform remote testing of Web sites and analyse the results with the support of automatic tools.

Once the initial preparation phase has been completed, this approach allows evaluators to analyse large number of sessions without additional effort.

Current work is dedicated to include automatic support for generation of Web sites task models and for associating basic tasks to events in the log files in order to ease the preparation phase. We are also investigating how to improve the representation of the data related to a user session and its automatic analysis. Current work is also devoted to the inclusion in WebRemUSINE of an automatic analysis of the features of the Web pages. We hope to discuss these issues at the workshop and will be glad to contribute to any related publication that will be planned.

REFERENCES

[1] Bobby. http://www.cast.org/bobby

[2] S.Card, P.Pirolli, M. Van der Wege, J.Morrison, R.Reeder, P.Schraedley, J.Boshart, *Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability*, Proceedings ACM CHI 2001, pp.498-504.

[3] M.Claypool, P.Le, M.Wased, D.Brown, *Implicit Interest Indicators*, Proceedings ACM IUI'01, pp.33-40.

[4] J. Cugini, J. Scholtz *VISVIP: 3D visualization of paths through web sites*. Proceedings of the International Workshop on Web-Based Information Visualization (Florence, Italy, September 1999), pp. 259-263. Institute of Electrical and Electronics Engineers. http://www.itl.nist.gov/iad/vug/cugini/webmet/visvip/webvis-paper.html

[5] M. Etgen, J. Cantor *What does getting WET (WebEvent-logging Tool) mean for web usability?*. Proceedings of HFWeb'99 (Gaithers-burg, Maryland, June 1999). http://zing.ncsl.nist.gov/hfweb/proceedings/etgen-cantor/index.html.

[6] P. Faraday *Visually critiquing web pages*. Proceedings of HFWeb'00 (Austin, TX, June 2000). http://www.tri.sbc.com/hfweb/faraday/faraday.htm

[7] M. Ivory, M. Hearst *State of the Art in Automated Usability Evaluation of User Interfaces*. 2000. University of California, Berkeley

[8] G. Lynch, S. Palmiter, C. Tilt *The max model: A standard web site user model*. Proceedings of HFWeb'99 (Gaithers-burg, Maryland, June 1999). http://www.nist.gov/itl/div894/vvrg/hfweb/proceedings/lynch/index.html

[9] L.Paganelli, F. Paternò, Intelligent Analysis of User Interactions with Web Applications. Proceedings ACM IUI 2002, pp.111-118, ACM Press.

[10] F. Paternò, Model-based design and evaluation of interactive applications, Springer Verlag, 1999. ISBN 1-85233-155-0.

[11] J. Scholtz, S. Laskowski, L. Downey *Developing usability tools and techniques for designing and testing web sites.* Proceedings HFWeb'98 (Basking Ridge, NJ, June 1998). http://www.research.att.com/conf/hfweb/proceedings/scholtz/index.html

[12] WebCriteria. *Max, and the objective measurement of web sites.* December 1999. http://www.webcriteria.com/pdf/max_102.cfm