

Remote Usability Analysis of MultiModal Information Regarding User Behaviour

Fabio Paternò, Angela Piruzza and Carmen Santoro

ISTI-CNR

Via G. Moruzzi, 1 Pisa (ITALY)

Fabio.paterno@isti.cnr.it

39 050 315 3066

ABSTRACT

In this paper we describe MultiModal WebRemUsine, a tool for remote usability evaluation of web sites that considers data coming from log files, videos recorded during user tests, and an eye-tracker. The tool performs an automatic evaluation of the usability of the considered web site by comparing such data (which describe the actual behaviour of the users) with that contained in the task model associated with the pages (which describes the expected behaviour of the user). The results of the analysis performed by the tool are provided to the evaluators in terms of task not completed, errors occurring during the performance of tasks, time for completing a task, etc. These results are provided along with information regarding the user behaviour during the task performance. Using such data, evaluators should be in a position to identify problematic parts of the website and make improvements, when necessary. An example of application of the proposed method is also shown in the paper.

Author Keywords

Remote usability evaluation, websites, multimodal data

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The dissemination of Web applications is enormous and still growing. The great penetration of Web sites raises a number of challenges for usability evaluators. In this paper we discuss what information can be provided by automatic tools able to process multimodal information on users gathered from different sources, so as to provide the most effective remote usability evaluation of websites. The collected information ranges from browser logs to videos to eye-tracking data, and the approach proposed tries to integrate such data in order to derive the most complete information for analysing, interpreting and evaluating the users while visiting a website, by taking into account the factors that might affect the performance of the users.

The proposed approach is supported by a tool – MultiModal WebRemUsine, which has been improved over the years in order to include and handle more and more information and provide additional features. In one

of its first versions [4], the tool was just able to automatically analyse the information contained in Web browser logs and compare it with task models specifying the ideal behaviour of users interacting with the application and representing the actual Web site design. The goal was to identify where users interactions deviate from those envisioned by the system design and represented in the model. However, such information can be rather limited because when users visit a web page, their attention can be captured by different areas of the same page and this information cannot be derived just analysing log files, which are only able to track physical interactions of the user with the application (eg.: scrolling, clicking, etc.). Eye tracking is a technique able to allow for deriving the current area of interest of the user by following the user's gaze. Thus, it helps evaluators in discovering the navigation strategies of the users visiting the web site and analysing the impact of different areas of the page. This enables easy identification of possible problematic parts of the page.

However, there are situations in which even the eye-tracker data may result inadequate to provide sufficient information for effective evaluation. Indeed, a user may look at the same portion of the page for quite different reasons, and such reasons could not be discovered by just analysing eye-tracking information. For instance, users might delay in staring at a certain point of the page because they might not be aware of having found the requested information and still look for it or, alternatively, they are aware of having achieved the goal and thus are interested in further reading the information found. In both cases, a video-based analysis might provide useful information for interpreting the different impacts of the same page on users: for instance, it might highlight situations where, although the logged information and user's gaze might make evaluators conclude that the user has successfully completed the expected task, a puzzled expression of the user should force the evaluator to re-interpret the collected data and derive, more realistically, that the user has completed the task but might not realised it, which is still a signal of a usability problem in the page. So, this simple example shows how important is integrating all the data that is possible to capture on the user behaviour (and possibly also on the environmental/contextual conditions in which the user interaction takes place) in order to perform the most comprehensive evaluation.

The novelty of the proposed approach is the integration of data coming from various sources (eye-tracker, webcam, ...) and the data coming from a tool able to know the current goal of the user and the related activities performed by the user in order to reach such goal. While such data have been already used in previous evaluation approaches in a separate manner, in our approach the added value is represented by integrating the two techniques. In this way, the data coming from the various sources allow for better understanding the information regarding the task the user is currently performing. Thus, the evaluator is provided with a more comprehensive picture of the actions performed by the user and consequently with more information in order to effectively interpret and evaluate the related user interface. Moreover, it is worth pointing out that, apart from the data provided by the eye-tracker, which still remains a rather expensive technology, the approach proposed has the remarkable advantage to allow evaluators to identify usability problems even if the analysis is performed remotely, which might contribute to keep at minimum the evaluation costs and allows the users to remain in their familiar environments during the evaluation, improving the trustworthiness of the evaluation itself.

In the paper, we first discuss related work, next we indicate the method underlying our environment and recall the main features of the first versions of WebRemUsine. Then, we present the new features of the new tool, MultiModal WebRemUsine, and report on some example application. Lastly, some conclusions along with indications for future work are provided.

RELATED WORK

Creating a Web site allows millions of potential users, who have diverse goals and knowledge levels, to access the information that it contains. While a Web site can easily be developed using one of the many tools available able to generate HTML from various types of specifications, obtaining usable Web sites is still difficult. Indeed, when users navigate through the Web they often encounter problems in finding the desired information or performing the desired task. With over 30 million Web sites in existence, Web sites have become the most prevalent and varied form of human-computer interface. At the same time, with so many Web pages being designed and maintained, there will never be a sufficient number of professionals to adequately address usability issues without automation [2] as a critical component of their approach. For these reasons, interest in automatic support for usability evaluation of Web sites is rapidly increasing [6][1].

With the advent of the Web and the refinement of instrumentation and monitoring tools, user interactions are being captured on a much larger scale than ever before. In order to obtain meaningful evaluation it is important that users interact with the application in their daily environment. Since it is impractical to have evaluators directly observe users' interactions, interest in remote evaluation has been increasing. In addition, some studies [7] have confirmed the validity of remote

evaluation in the field of Web site usability. Some work [3] in this area has been oriented to using audio and video capture for qualitative analysis performed by evaluators on the result of usability testing. In our case we focus more on quantitative data (eg: log files and data coming from eye-tracking tools like length of the scanpath, duration of fixations, etc.) and provide the support for an intelligent analysis of such data so as to extract useful information for evaluation goals. The aim of this paper is to present a tool for performing remote usability evaluation of Web applications. We describe how it supports analysis of task performance and Web pages accesses by single users and groups of users.

THE ARCHITECTURE

In its first version, the core of the system architecture of our system was mainly composed of three modules (see Figure 1): the ConcurTaskTrees editor (publicly available at <http://giove.cnuce.cnr.it/ctte.html>) developed in our group; the logging tool that has been implemented by a combination of Javascript and applet Java to record user interactions; and 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.

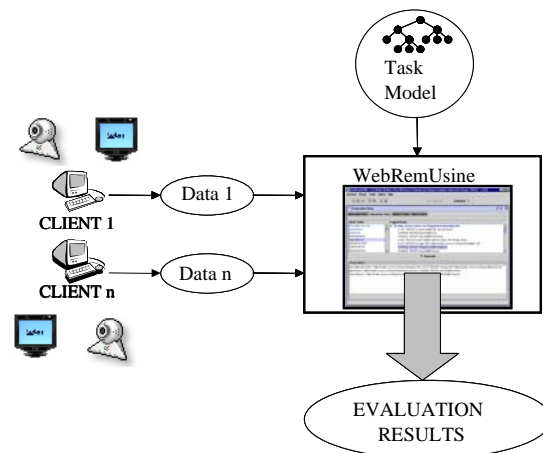


Figure 1: An Overview of the Architecture

The ConcurTaskTrees Environment (CTTE) is a tool for editing and analysing task models, which describe the activities to perform in order to reach user's goals. CTTE supports the ConcurTaskTrees (CTT) notation [5] for specifying task models, by providing a graphical representation of the hierarchical logical structure of the task model. In particular, it is possible to specify a number of flexible temporal relations among tasks (concurrency, choice, enabling, disabling, suspend-resume, order-independence, optionality, ...) and for each task it is possible to indicate the objects to be manipulated and a number of other 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 stores various events detected by a browser, using Javascripts that are encapsulated in the HTML pages and 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. Then, a Java applet stores the log files directly in the application server. Our browser logging tool overcomes the limitations of other approaches to logging: server logs have limited validity since various page accesses are hidden to them because of browser cache memory and they do not detect the local interactions with the user interface elements (checkboxes, type in fields, ...) that also in the case of proxy-based approaches cannot be detected.

The WebRemUSINE analysis can detect usability problems such as tasks with long performance or tasks not performed according to the task model corresponding to the Web site design. The task model describes how activities can be performed according to the current design and implementation. Either the designer or the evaluator can develop it. Since ConcurTaskTrees has various temporal and logical operators, it is possible to describe all the various paths the user can follow to accomplish a task. Deviations from the expected paths can be detected in the logs and represent useful information for identifying any pages that create problems to the user. Moreover, the tool analysis provides information concerning both tasks (such as time performance, errors, ...) and Web pages (such as download and visit times, ...). These results allow the evaluator to analyse the usability of the Web site from both viewpoints, for example comparing the time to perform a task with that for loading the pages involved in such a performance. 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 these 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.

THE UNDERLYING METHOD

The method underlying the tool is composed of two main phases, the preparation and the evaluation.

Preparation Phase

The main goal of the preparation phase is to create an association between the basic tasks of the task model and the events that can be generated during a session with the Web site. This association allows the tool to use the semantic information contained in the task model to analyse the sequence of user interactions. On the one hand, in the task model, all the activities that are

supposed to be performed in order to reach a goal are hierarchically described (for an example of task model see Figure 3). In particular, basic tasks are tasks that cannot be further decomposed, whereas high-level tasks are complex activities composed of sub-activities. On the other hand, the log files are composed of sets of events. If an event is not associated with any basic task, it means that either the task model is not sufficiently detailed, or the action is erroneous because the application design does not call for its occurrence. For example, when a user sends a form then two events are stored in the log: one associated with the selection of the Submit button and the other one with the actual transmission of the form. Thus, in the task model two basic tasks are required: one interaction task for the button selection and one system task for the form transmission, otherwise the model is incomplete.

In the logs, there are three types of events: user-generated events (such as click, change), page-generated events (associated with loading and sending of pages and forms) and events associated with the change in the target task by the user, which is explicitly indicated through selection from the list of supported tasks.

Tasks can belong to three different categories according to the allocation of their performance: user tasks are only internal cognitive activities that thus cannot be captured in system logs, interaction tasks are associated with user interactions (click, change, ...) and system tasks are associated with the internal browser generated events. In addition, the high-level tasks in the model are those that can be selected as target tasks by the user. The different allocation of the various tasks is represented by using different icons in the task model (see Figure 3). In order to ease the development of the task models we also developed a reverse engineering tool able to create automatically the task model corresponding to a web site implemented in (X)HTML. This model can be further refined by the designer, if necessary.

Each event is associated with a single task whereas a task can be performed through different events. For example, the movement from one field to another one within a form can be performed by mouse, arrow key or Tab key. The one-to-many association between tasks and events is also useful to simplify the task model when large Web sites are considered so that we need only one task in the model to represent the performance of the same task on multiple Web pages.

Once the association between tasks and events has been carried out, it is possible to move on the next step of the approach, the proper evaluation, which will be detailed in the next section.

Evaluation Phase

In the evaluation phase the proper automatic analysis is performed, 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. WebRemUSINE displays all results in various formats both textual and graphical. Such information generated is analysed by the evaluators to

identify usability problems and possible improvements in the interface design.

During the test phase all the user actions are automatically recorded, including those associated to goal achievement. The evaluation performed by WebRemUsine mainly consists in analysing such sequences of actions to determine whether the user has correctly performed the tasks complying with the temporal relationships defined in the task model or some errors occurred. In addition, the tool evaluates whether the user is able to reach the goals and if the actions performed are actually useful to reach the predefined goals, or a precondition error occurred, which means that the execution task order did not respect the relations defined in the system design model.

In addition to the detailed analysis of the sequence of tasks performed by the user, evaluators are provided with some results that provide an overall view of the entire session considered:

- The basic tasks that are performed correctly and how many times they have been performed correctly.
- The basic tasks that the user tried to perform when they were disabled, thus generating a precondition error, and the number of times the error occurred.
- The list of tasks never performed either because never tried or because of precondition errors.
- The patterns (sequences of repeated tasks) occurred during the session and their frequency.

Such information allows the evaluator to identify what tasks are easily performed and what tasks create problems to the user. Moreover, revealing tasks that are never performed can be useful to identify parts of the application that are difficult to comprehend or reach. On the basis of such information the evaluator can decide to redesign the site in order to reduce the number and complexity of the activities to be performed.

From the log analysis the tool can generate various types of results:

- *Success*: the user has been able to perform a set of basic tasks required to accomplish the target task and thus achieve the goal.
- *Failure*: the users starts the performance of the target task but is not able to complete it;
- *Useless uncritical task*: the user performs a task that is not strictly useful to accomplish the target task but does not prevent its completion.
- *Deviation from the target task*: in a situation where the target task is enabled and the user performs a basic task whose effect is to disable it. This shows a problematic situation since the user is getting farther away from the main goal in addition to performing useless actions.
- *Inaccessible task*: when the user is never able to enable a certain target task.

A further type of information considered during the evaluation regards the task execution time, and the

duration is calculated for both high level and basic tasks. The set of results regarding the execution time can provide information useful to understand what the most complicated tasks are or what tasks require, in any event, longer time to be performed. Longer execution time does not always imply complicated tasks, for example in some cases downloading time can be particularly high. WebRemusine provides also detailed information regarding downloading time so that evaluators can know its influence on the performance time. A further type of evaluation concerns the time associated with actions that generate errors. By analysing when errors occur, it is possible to determine if the user's performance improves over the session. If the errors concentrate during the initial part of the test and their number decreases over time, we can assume that the user interface is easy to learn.

Modifications of the Evaluation Phase for Enriching Information on Users: Data from Videos and Eye-tracker

In the previous sections we have provided an overview of the system, and, from the point of view of the client side, we have only considered the data coming from the log files. However, such information revealed insufficient to perform an effective evaluation because there might be a number of factors that might affect the performance of a user while interacting with an application and cannot be described just recording log files. Then, integration of such data with videos recorded during the sessions and data from eye-tracker was used to enrich the multimodal information that the system is able to gather on the user.

As for the videos, an association between task and video is automatically performed by the tool, thanks to the information regarding the starting/ending time of the different tasks. Indeed, as the whole session is recorded by a webcam, thanks to such times it is possible to split the video associated with the whole user session into different fragments related to the completion of the various tasks, together with the possibility of visualizing the related video with a suitable player in the tool, that can be activated/stopped by the evaluator. It is worth pointing out that, in order to do this it was necessary to modify the process of recording log files so as to save the information about such times. The data from videos are important in that they can provide more 'contextual' information during the performance of a task. Indeed, since the evaluation is remotely performed, the evaluator is not in a position to understand if any condition might have disturbed the performance of a task while the user visits the web site in his/her own environment. For instance, a high time (or, at least, a time higher than expected) for completing a task might not necessarily be brought about by a usability problem, but it may be caused by interruptions during the session occurring in the user's environment and due to some external factors. Another useful information that can be gained from videos are user comments, which sometimes can reveal that users are aware of having performed an error but cannot undo the related actions.

While videos provide more ‘contextual’ information regarding users, giving the means for correctly interpreting the user’s actions, the eye-tracker provides technical measurements and traces of the routes that users follow while visiting a website. Indeed, the eye-tracker records the pauses and ‘jumps’ that normally occur when a user looks at a page (respectively called fixations and saccades), together with the so-called ‘scanpaths’, the traced routes of sequences of fixations and saccades, revealing the path that the user followed during the visit of the page. By superimposing the scanpath on the page it is possible to understand the strategies used by the user to visit the page. Indeed, the evaluator can understand where the user paused to look and, alternatively, the areas of the page that did not attract his/her attention. Moreover, having in mind the target task that the user should achieve, it might be relevant to analyse the areas around the links that should be followed in order to reach the expected goal, according to the task description specified in the task model. For instance, it might be relevant to analyse the time users spent looking at these area (duration of fixations), as well as the number of accesses to such areas (if any), which is given by the number of fixations. By examining such variables it is possible to understand if users found difficulties in exploring the page: for instance, if the time duration is high, it might be a sign of user’s difficulty in elaborating the information; also the number of fixations might be a sign of problems because the occurrence of some fixations on certain areas might indicate that the users are confused and are not able to find the information that they are looking for. Moreover, also the scanpath might give useful information for the evaluation: for instance, a long scanpath might indicate that the structure underlying the page is rather complicated, while, vice versa, a very short scan path indicates a rather poor structure of the page.

In addition, it is worth pointing out that, in order to manage the information associated with the eye-tracker, it was needed to modify also the logging tool and, in particular, handling scrolling events. Indeed, as soon as a scroll event is recorded, also the extent of the shifting is recorded with respect to the top and bottom corner of the page, so as to reconstruct the actual area that the user was currently looking at. Then, as we considered different sources of data for the evaluation and in order to provide the most flexibility possible, different evaluation options were considered and included in the tool:

- i) *Browser input*, showing information derived from analysing log files;
- ii) *Browser+WebCam* log files and users’ video recorded during the tests;
- iii) *Browser+EyeTracker* input, with log files information and eye-tracker data;
- iv) *Browser+WebCam+EyeTracker* with log files information, videos, and eye-tracker data.

AN EXAMPLE

In this section we show an example of application of the proposed evaluation method. The web site we considered

(<http://www.pisaonline.it>) provides information about Pisa (in Fig. 2 the homepage is shown). The website is divided into four main sections: “Pisa da Visitare” (Visiting Pisa), “Pisa da Vivere” (Living in Pisa), Pisa da Studiare” (Studying in Pisa) e “Pisa Aziende” (Companies in Pisa). For sake of brevity in Figure 3 only the task model related to the “Pisa da Visitare” section is visualised.

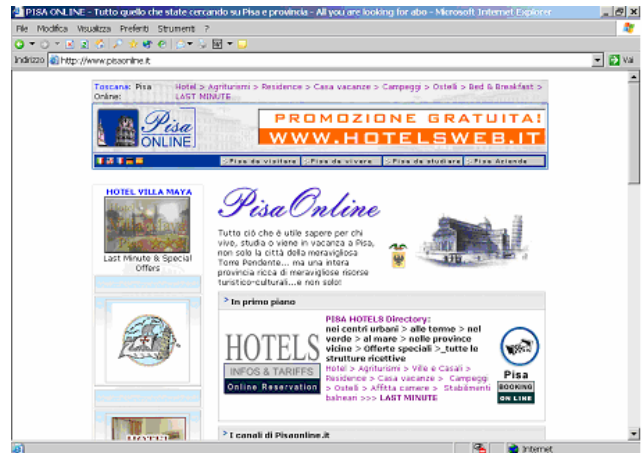


Figure 2: The homepage of the site evaluated

In the model the activities the user should perform to get information about places to visit in Pisa are described. For the evaluation several target tasks were identified, examples are “Trova Info su Tartufo Bianco” (Find information about the white truffle), “Trova gradazione alcolica del Chianti”, (Find alcoholic content of Chianti), etc..



Figure 3: The task model of the “Pisa da Visitare” section

We involved in the test participants aging between 21 and 36. In Fig. 4 the scanpath length of the homepage is visualised: as you can note, it is rather long (16554 pixels).

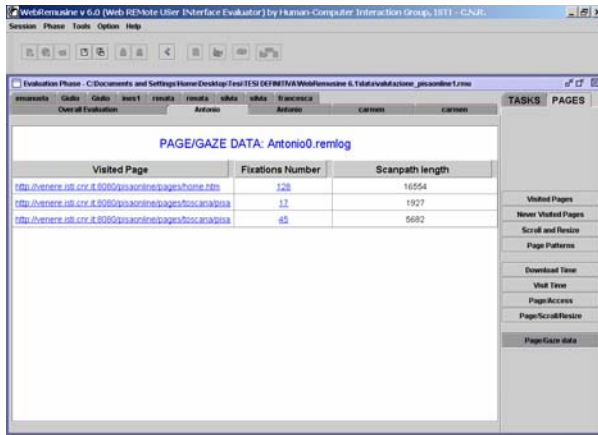


Fig. 4: The tool window associated with Page/Gaze data

Indeed, looking at the image of the scanpath superimposed on the homepage for the task “Find restaurant” (it is visualised in Fig. 5) it is possible to see that the gaze of the user moves from one point to another indicating a possible confusion of the user who seems not able to find the right link for carrying out the task.

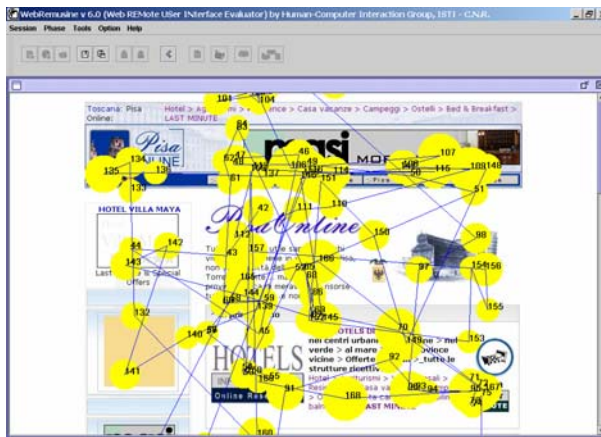


Fig. 5: Scanpath of “Find Restaurant”

Another user (who made the greatest number of errors during the navigation) selected as target task “Find information on the white truffle”, which was associated with the abstract task “Access to Ulisse” (“Ulisse” is the name of an airline magazine providing general tourist information to their passengers). During the test such user performed a number of errors of precondition, because he carried out a number of tasks that were not necessary according to the designer’s task model for achieving the task goal.

While trying to perform the task “Find Information on the White Truffle” the first of such errors was carried out because the user was observed pausing a lot while looking at the area of the page dedicated to the restaurants (a fixation with a relevant duration was registered), whereas s/he did not look at all to the right link (the one related to “Pisa da visitare”) which, according to the designer’s task model, represents the right route for completing the selected task (“Find Information on the White Truffle”). From this it might be derived that the

user was wrong at interpreting the name “White Truffle” and interpreted it as a name of a restaurant.

In addition, after spending long time before selecting “Pisa da Visitare” link it is possible to note from the associated scanpath visualised in Figure 6 that the user found it difficult also to identify the right link among those associated with the “Access to Ulisse” target task. Indeed, in order to access to the section devoted to Ulisse, in the page visualised in Fig. 6 there is a textual link (with label “Ulisse”), another textual link with a different label (“Alitalia Ulisse”), and also an icon with an image associated to Ulisse, and the information available through the last two links is different from that that can be accessed through the first link.

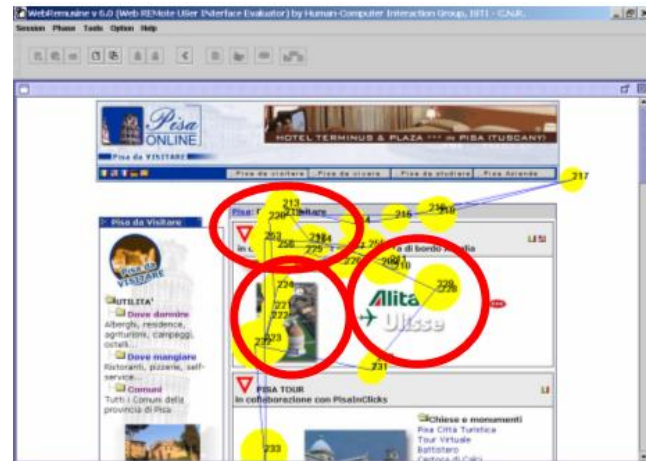


Fig. 6: The ambiguity of links related to the section dedicated to “Ulisse”

Moreover, the experiment highlighted that the majority of users did not select the image link associated with the homepage of the PisaOnline web site (visualised in the top left part of the homepage, see Figure 6) and associated with the related task “Seleziona Home” (Select home). It was rather surprising due to the relevancy of this page within the entire site. The occurrence of such behaviour in almost all users can be interpreted with the fact that the link is rather unclear, and this intuition is reinforced by the image related to the scanpath of users on the page (eg: see figure 6), highlighting that almost all users did not pause on looking at the concerned link. Indeed, it might be explained with the fact that the imagelink might have been confused with a bare decorative image, (especially because it appears on the top part of the page).

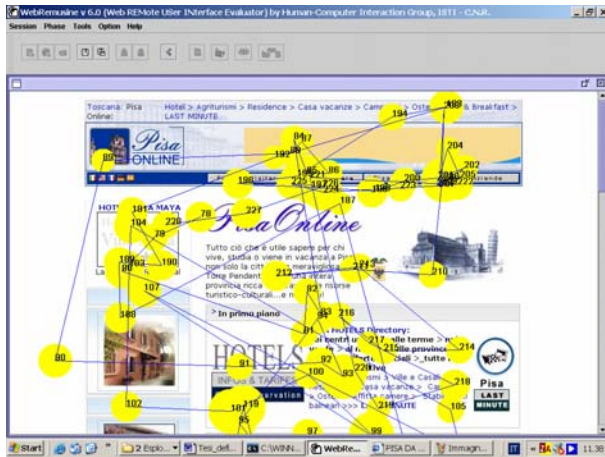


Fig. 7: Scanpath of “Find Alcoholic Content of Chianti”

In addition, users who selected “Find alcoholic content of Chianti wine” (“Trova gradazione alcolica del Chianti”, see Fig. 7) as target task, which is related to the high-level task “Scegli vino” (Select wine) tended to find such information within the section called “Companies in Pisa”, rather than, more correctly, within the “Living in Pisa” section, where the right link actually is. Figure 7 shows that there are many fixations (202,204,205,222,etc.) on the link associated with “Pisa Aziende” (“Companies in Pisa”), which highlighted that the logic followed by users in finding such information while exploring the page was different from that followed by designers.

In another experiment we analysed a different site regarding a publishing house and mainly focused on data recorded by videos. In Figure 8 you can see the evaluation of task/time performed by the tool, regarding a user who explicitly declared at the end of the task that she was wrong at completing the task.



Fig. 8: Task/Time Information with Video of the user

The data from the videos were useful to detect some usability problems. For instance, by examining the tasks that were wrongly performed, it was possible to have further details on the facial expressions of the users, who

sometimes seemed to be confident of their choices, while other times seemed to be quite confused and doubtful, and this information is important when evaluating the user behaviour. Particularly useful information was gained from videos as far as the execution time, which sometimes seemed to be higher than expected: the analysis of the video revealed that users pause at looking the page, then they happen to comment on it, so this is important to understand that it is not completely effective measuring the degree of difficulty of a task by just calculating the time spent in performing it, because users often are distracted/ attracted by portions of the page that are not relevant for carrying out the concerned task, only by curiosity.

CONCLUSIONS

The advances in technology is more and more allowing evaluators to afford sophisticated hardware and software able to collect information about remote users interacting with web pages. In this paper, we propose a method, and the associated tool, for remote evaluation of websites that, through a combination of different sources of data coming from the client side (log files, videos and eye-tracker data) allows the evaluator to get detailed information about the behaviour of the users. More specifically, the evaluator can be informed on the page the user was visiting to perform a certain task, the sequence of actions carried out by the user on the page, together with detailed information of the route traced by the user gazing at the page and background videos on the user during the session.

Such composite information is the input of an automatic tool that has been developed in our group and has shown to be effective in providing evaluators with means for discovering possible problematic areas of the website.

Future work will be dedicated to extending the data regarding the user behaviour and state, including the emotional state, in order to have a more complete analysis of what happens during task accomplishment and better identify the potential usability issues

ACKNOWLEDGMENTS

We thank the SIMILAR European Network of Excellence for partly supporting this work.

REFERENCES

1. Card, S., Pirolli, P., Van der Wege, M., Morrison, J., Reeder, R., Schraedley, P., Boshart, J., (2001) 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.
2. Ivory M. Y., Hearst M. A., (2001) The state of the art in automating usability evaluation of user interfaces. ACM Computing Surveys, 33(4), pp. 470-516, December 2001.
3. Lister M., (2003) Streaming Format Software for Usability Testing, Proceedings ACM CHI 2003, Extended Abstracts, pp.632-633.

4. L.Paganelli, F.Paternò, Tools for Remote Usability Evaluation of Web Applications through Browser Logs and Task Models, Behavior Research Methods, Instruments, and Computers, The Psychonomic Society Publications, 2003, 35 (3), pp.369-378, August 2003.
5. Paternò, F., (1999) Model-based design and evaluation of interactive applications, Springer Verlag, 1999. ISBN 1-85233-155-0.
6. Scholtz, J., Laskowski, S., Downey L., (1998) 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>
7. Tullis, T, Fleischman, S., McNulty, M, Cianchette, C. and Bergel, M., (2002). An Empirical Comparison of Lab and Remote Usability Testing of Web Sites. Usability Professionals Conference, Pennsylvania, 2002.