FP6-507609

SIMILAR

The European research taskforce creating
human-machine interfaces SIMILAR
to human-human communication

Network of excellence
FP6 - IST

Deliverable #99
Tool and report on remote usability testing support

Due date of deliverable: November, 2006
Actual submission date: January 15, 2007

Start date of project: 1st December 2003 Duration: 48 months

Organisation name of lead contractor for this deliverable: ISTI-CNR
Fabio Paterno

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Project ref. no.	FP6-507609
Project acronym	SIMILAR
Deliverable status	R
Contractual date of delivery	30 November 2006
Actual date of delivery	November 2006
Deliverable number	D99
Deliverable title	Tool and report on remote usability testing support
Nature	Report
Status & version	Final
Number of pages	23
WP contributing to the deliverable	SIG7
WP / Task responsible	NISLab/ISTI-CNR
Editor	N/A
Author(s) (alphabetic order)	Fabio Paternò and Carmen Santoro
EC Project Officer	Mats Ljungqvist
Keywords	Remote usability, tools, methods, usability evaluation
Abstract (for dissemination)	The goal of this report is to present a design space for tools and methods supporting remote usability evaluation of interactive applications. This type of approach is acquiring increasing importance because it allows usability evaluation even when users are in their daily environments. Several techniques have been developed in this area for addressing various types of applications that can be used in different contexts. We discuss them within a unifying framework that can be used to compare the weaknesses and strengths of the various approaches and identify areas that require further research work in order to exploit all the possibilities opened up by remote evaluation.

Deliverable D99

Tool and Report on Remote Usability Testing Support

ISTI-CNR, HIIS Laboratory, Pisa, Italy

November 2006

Contents

Remote Usability Evaluation: Discussion of a General Framework and Experiences from Research with a Specific Tool

Fabio Paternò, Carmen Santoro

ISTI-CNR, Pisa Italy

Abstract

The goal of this report is to present a design space for tools and methods supporting remote usability evaluation of interactive applications. This type of approach is acquiring increasing importance because it allows usability evaluation even when users are in their daily environments. Several techniques have been developed in this area for addressing various types of applications that can be used in different contexts. We discuss them within a unifying framework that can be used to compare the weaknesses and strengths of the various approaches and identify areas that require further research work in order to exploit all the possibilities opened up by remote evaluation.

1 Introduction

In this report we present and discuss a design space for remote usability evaluation. This type of approach to usability evaluation is characterised by the fact that users and evaluators are separated in time and/or space (Hartson et al., 1996). Thus, it still requires the involvement of these two actors (user and evaluator) but it relaxes the constraint that they need to be present at the same time in the same place. The motivations for remote evaluation are various:

- Usability laboratories can be expensive to set up because they require dedicated sites with specific equipment;
- Moving users to the usability laboratory can be difficult and expensive as well, in particular for expert users, whose time is costly. Indeed, it can be difficult to find an adequate number of users willing to move to a usability lab for a test;
- Remote evaluation can be useful to analyse user behaviour in their daily environment (e.g. workplace, home, and so on), thus in more realistic settings;
- It facilitates the possibility of a continuous evaluation, even after the first release of the application.

Some studies have investigated to what extent remote evaluations yield results similar to lab testing. For example, Tullis (Tullis and others, 2002) found that remote evaluation in the field yielded results that were largely similar to studies in the lab. There are several methods that support some kind of remote usability evaluation. They differ for the type of information that is made available to the evaluator and how it is provided to them. Ivory and Hearst (2001) wrote an interesting review of the state of the art on automating usability evaluation of user interfaces in which some methods and tools in the area of remote evaluation were considered as well. In this report, we provide a more updated and focused discussion of the state of art in

remote evaluation through a more refined framework for this area that highlights the important aspects to consider when analysing approaches within it. More specifically, the framework proposed in this report is defined analysing various dimensions. The first one regards the type of interaction that occurs between the user and the evaluator, and is strongly connected with the possibility of having a co-presence (in terms of time) between the user and the evaluator. Another dimension involves the techniques that can be used to gather information on user sessions (server/proxy/client logs, Webcams, eye-trackers, and other sensing technologies) and the provided information, useful for the evaluation. Another interesting dimension we consider is the type of platform used for interaction: with this regard, we plan to distinguish, for example, access through desktop or mobile devices and discuss how the choice of the platform affects the aspects to consider in the evaluation. The last dimension regards the type of application considered (for instance: Java-based, Web-based, etc.) A discussion about the potential correspondences between such dimensions should shed some light on which techniques/technologies evaluators should direct their attention in order to obtain the desired information, so providing them with a better understanding of techniques for remote evaluation of user sessions and how to use them to identify problematic parts of interactive applications and make improvements accordingly (when necessary).

To summarise, the relevant dimensions we have identified for analysing the different methods for assessing remote usability evaluation are:

- The type of interaction between the user and the evaluator;
- The platform used for the interaction (desktop, mobile, vocal, etc.);
- The techniques used for collecting information about the users and their behaviour (graphical logs, voice and/or Webcam recordings, eye-tracking, ..);
- The type of application considered in terms of implementation environment (Web, java-based, .NET, ..);
- The type of the evaluation results (task performance, emotional state, ..) provided.

In the next sections, we use the framework composed of such dimensions to discuss a number of techniques that can be used to perform remote evaluation of user sessions (logging technology, interaction platform, semantic analysis) along with a review of the most relevant works in the area together with a discussion about issues that have been resolved from the perspective of usability evaluation, and problems still open.

In order to make the discussion more concrete we also discuss our own experiences in this area, including our method (and the related tool) for remote usability evaluation of Web sites that considers information from user tasks, log files, videos recorded during user tests, and data collected by an eye-tracker (Paternò et al, 2006).

2 The Type of Interaction between the User and the Evaluator

There are different methods and techniques that can be applied in order to perform a remote evaluation. One important dimension that can be used to classify them is how users and evaluators actually interact between them.

Bearing in mind that remote evaluation assumes that users and evaluators are separated in *space*, the type of interaction occurring between users and evaluators strongly depends on the type of synchronisation occurring on *time*. Indeed, while *asynchronous* evaluation method assumes that evaluators might not be necessarily available at the time when the user session is

taking place (and therefore, there is no possibility for the evaluator/moderator to deliver immediate input for impromptu changes), with *synchronous* evaluations it is the opposite: as an exemplary case of synchronous evaluation we mention collaborative usability evaluation methods via the network, in which evaluators in usability labs are connected to remote users via commercially available teleconferencing software (e.g., Microsoft Netmeeting) supporting real-time application sharing, audio links, shared drawing tools, and/or file transfer capabilities. Below we describe the various possibilities, for both synchronous (first bullet) and asynchronous evaluation (second-fourth bullets):

- *Remote observation*; this implies that users and evaluators are separated in space but they are active at the same time and connected through some tool (for example video conference tools) that allows the evaluator to observe the actual user behaviour in real time;
- *Remote questionnaires*; this is a technique that allows users to provide their feedback through a series of questions made available electronically;
- *Critical incidents reported by the user*; in this case the user directly reports to the evaluator when some incident occurs;
- *Automatic data collection*; this is the method that has stimulated the most interest because there are many ways to collect data regarding user behaviour and then analyse it. The potential information ranges from browser logs to videos taken by Web-cams to eye-tracking data. This case also includes our approach (Paternò et al, 2006), which will be described in Section 7.

In order to assess pros and cons of such options, we can notice that, on the one hand, remote observation provides the evaluator with more capabilities not only for observing the session but also for intervening *during* the session; furthermore, the simultaneous presence of evaluator and user brings the additional advantage of not requiring a particularly strong effort for an *a posteriori* analysis of the collected data, since most of this work should be already carried out by evaluators during the session. On the other hand, remote observation strongly limits the number of users that can be evaluated at a time, and, additionally, it might also happen that the behaviour of the users might be affected to some extent by their awareness of being currently observed by the evaluator.

Remote questionnaires and critical incidents are useful in that they report aspects that the users themselves noticed and judged relevant from the point of view of usability. However, the result of such techniques might be compromised by the fact that the reporting time is generally postponed with respect to when the problem appeared, therefore retrospective reporting and questionnaires might be subjected to loss of detail, which can hinder the reconstruction of the original problem.

The last technique (automatic data collection) on the one hand guarantees gathering a vast amount of detailed data, which, on the other hand, generally claim a non irrelevant effort and time for being correctly interpreted by humans, in absence of appropriate automatic data analysis techniques.

3 The Platform Used for the Application Interaction

One of the main characteristics of the rapid evolution of Information and Communication Technologies is the wide availability of various types of interaction platforms. The desktop is no longer the only device that even non professionals use for accessing their applications. There is a wide variety of interaction platforms on the market, which can largely differentiate in terms of interaction resources (such as screen size, etc.) and modalities supported.

Heterogeneous platforms raise specific issues that should not be neglected for the purposes of remote evaluation. For instance, mobile systems are typically used in highly dynamic contexts, and remotely evaluating mobile users require the use of specific techniques able to capture and identify usability problems that might be experienced in mobile use. One exemplary issue in remote usability evaluation involving mobile users is that they are physically moving and such changes in the context might imply a number of known and unknown variables potentially affecting the set-up (for instance, when increasing the amount of physical activity, a significantly increased subjective workload might be experienced by the users). In addition, the use of a particular platform should also be considered with the objective of identifying appropriate means for collecting user data in the remote site. For instance, eye-tracking systems are clearly useless for recording user interactions with only-voice applications. Therefore, a current issue for this dimension is represented by the capability of the different techniques for remote evaluation to dynamically vary the information that should be collected about the users, so as to cope with the potential issues that the specific platform in use can introduce. The expected objective is providing the evaluator with the most comprehensive picture of all the aspects that might have affected the interaction, so always being in a position to correctly derive the potential causes of a usability problem occurred on the client side. For instance, as we previously mentioned, if gathering information about the current environment is extremely important for a mobile user, since the environment can often change, it becomes less important for a stationary user interacting with a desktop application, as in this case the environment is almost fixed.

In this section we analyse how the remote evaluation methods address the issues raised by the specific platforms.

3.1 Desktop Applications

Since most applications have been developed for the desktop, the majority of remote evaluation methods have addressed this type of platform.

In (Hartson et al. 1996) one of the first examples of remote-control evaluation is described. The remote-control method checks a local computer from another computer at a remote site. The user is separated from the evaluator in space and possibly in time. The two computers can be connected through the Internet or a direct dial-up telephone line with commercially available software (e.g., Timbuktu TM, PC Anywhere). Using this method, the evaluator's computer is located in the usability lab where a video camera or scan converter captures the users' actions. The remote users remain in their work environment and audio capture is performed via the computer or telephone. If the audio capture is via telephone, the evaluator and user remain connected at the same time. Alternatively, equipment in the usability lab could be configured to automatically activate data capture tools based on the use of a particular application. This is an example of quite a flexible technique for asynchronous remote evaluation which is restricted to be used on desktop systems due to some software limitations on the underlying hardware (e.g., PC Anywhere only operates on PC platforms).

3.2 Vocal Applications

As for the vocal platform, interest is arising in studying this modality since the associated technology is becoming more reliable and robust in different systems in everyday use. The most exemplary case is that of the automatic response systems commonly used for completing user tasks such as banking, paying bills, receiving train/flight information. Such systems can accept both speech and touch tone inputs and in response provide relevant information

through voice, email, text messaging, fax. Most companies are seeing such systems as an immediate cost saver since call centres are becoming too expensive to be operated by humans.

As speech applications advance, so does the need for a means to evaluate VUIs (Vocal User Interfaces), in order to be able to assess how a user interacts with a vocal application. In this respect we have to say that sometimes methods that are commonly applied in GUI evaluation are also applied to the evaluation of VUIs, although this translation is not a perfect fit. Indeed, the sequential nature of speech means that VUIs are inherently more restrictive than GUIs, therefore fewer choices can be explored with a VUI in a certain time interval, with respect to what can be done with a correspondent GUI. One of the consequences from the point of view of usability evaluation is that, for instance, the number of tasks carried out in a certain interval of time by interacting with a VUI will deliver a lower value when compared with a correspondent GUI, without being necessarily a sign of a bad usability of the vocal user interface considered.

Furthermore, it should be born in mind that using only the speech as interaction medium might represent a burden on users' memory, which means that not only VUI users should be focused on a smaller set of choices and in a narrower context, but also that, without visual cues and a well-established mental model, they are even unlikely to understand what choices are available to them. The consequence is that, without careful design, these limitations can severely diminish the general usability of the vocal application.

Despite the limitations noted above, well-designed voice applications have proved to be both engaging and effective: some novel evaluation methods for these interfaces are under development and several experiments in the lab have already been done, although, with the proliferation of cellular phones and wireless phones, evaluations of VUIs in lab environments suffer from unrealistic settings, as they are very different from the real contexts of use.

Several techniques can be envisaged for evaluating vocal user interfaces and, more specifically, automatic response systems: among them, we cite surveys, call recordings and call logs. Surveys are issued after a call is completed, but sometimes callers do not complete the call, hence never reaching the survey. As for call recordings, most VUI systems record caller interactions, since call recordings tell the VUI designer exactly what happened during each call. They have several shortcomings: they can not tell the designer what the caller was trying to do, how the caller felt, why the caller did what s/he did; another shortcoming is the massive amount of effort required to analyse the calls. Lastly, in call logs, every IVR (Interactive Voice Recognition) platform comes with extensive call logging capabilities. While surveys and call recording typically result in qualitative data, call log data is typically quantitative e.g.: average call length, time on hold, abandon rate, etc. Due to the enormous amount of data that can be collected, data mining techniques are suitable for processing such data. Call logs identify where in the VUI callers have difficulty, but this is only a part of the picture. Call logs do not provide a lot of context to help in interpreting the results. Therefore, since surveys, call recordings and call logs provide different information for in-use situation analysis, it seems that in order to compensate for their various advantages and drawbacks, careful consideration of their combined use may be a viable solution for the purposes of evaluating VUIs.

Example of tools for VUIs are ClickFox (2002), which aims to answer questions like: what is the main cause of customer hang-ups? What are callers doing most frequently at critical decision points? Are callers using the system in the way that you expected? Another example is provided by IQ Services (2006), which is able to log and record each call, allowing IQ Services' analysts to duplicate and experience system errors; after the test is completed

designers receive online test results, step-by-step logs and online playback of each digital test call recording.

To conclude, while there are not many works on remote usability evaluation for vocal applications, the naturalness of this kind of interaction and its quick diffusion in a number of applications covering different devices, make us expect that further research effort will be put on this subject.

3.3 Mobile Applications

In mobile applications it is important to understand the influence of the context of use, which is composed of three main parts: the user, the device and the environment. Thus, one issue is to understand how usability is affected by dynamic changes of any of these components. Regarding evaluating interaction with mobile devices, the work of Denis and Karsenty (2003) focuses on the usability of a multi-device system and introduces the concept of inter-usability to designate the ease with which the users can reuse their knowledge and skills for a given functionality when switching to other devices. In their paper a framework for achieving inter-usability between devices is proposed. It is based on two components: (i) a theoretical analysis of the cognitive processes underlying device transitions, and (ii) an exploratory empirical study of the problems in using functionalities across multiple devices. Another work in this area is the paper by Waterson et al. (2002), where the authors discuss a pilot usability study using wireless Internet-enabled personal digital assistants (PDAs), in which they compare usability data gathered in traditional lab studies with a proxy-based clickstream logging and analysis tool. They found that this remote testing technique can more easily gather many of the content-related usability issues, whereas device-related issues are more difficult to capture. Lastly, worth mentioning is the work of (Stoica et al., 2005), in which the authors describe a usability evaluation study of a system which permits collaboration of small groups of museum visitors through mobile handheld devices (PDAs). As the authors point out, in general, techniques to measure usability-related factors include (i) inspection methods, (ii) testing methods and (iii) inquiry methods. For systems including mobile devices, a combination of these techniques is sometimes used. As usability evaluation methodology, they propose a combination of a logging mechanism and an analysis tool (the ColAT environment (Avouris et al., 2004)), which permits mixing of multiple sources of observational data, a necessary requirement in evaluation studies involving mobile technology, when users move about in physical space and are difficult to track. The museum system evaluated is based on a client–server architecture and an important characteristic of the application is that the server produces a centralized XML log file of the actions that take place during the visit, and this log file can be combined with video recording of the visit allowing evaluation of the activity during the visit. In the experiment shown in the paper, different teams gathered the clues and then each group had to discuss and discover collaboratively what the combined clues were in order to solve the problem. The experiment was recorded by 3 video and 2 audio recorders for further analysis using the ColAT analysis tool, which interrelates activity logs video and observers notes in the same environment. So, through ColAT the actions that the users performed during the use of the PDAs, which were logged by the server, were synchronized with the videos. The methodology was able to deliver data useful for deriving quantitative information (e.g. total and average times for solving the puzzles, etc.); aspects related to group activities (number of exchanges between the group, strategies used for solving the puzzles, ..), behavioural patterns of participants.

Indeed, the importance of performing a comprehensive evaluation able to take into account data derived from multiple sources in order to adequately gain insight into large bodies of

multi-source data, especially when mobile applications are considered, is quite clear. An example of this trend can also be found in the work of Tennent et al. (2006), in which the authors present Replayer, which consists of a number of tools (two video players, an audio player, an aggregate log visualisation, a text search tool and a playback control tool), and a collaborative tool for analysis of recorded data of mobile applications. The tool was designed in the effort to provide analysts from a variety of disciplines (each using distinct sets of skills to focus on specific aspects of the problem) with the ability to work cooperatively.

One of the emerging needs in this area is for tools able to better support analysis of how task performance varies depending on the context change.

4 The Techniques for Collecting Information about the User Behaviour

In this section we discuss the various techniques available for collecting information regarding the user behaviour (task performance, use of mouse and keyboard, facial expressions, verbal comments, gestures, gazes, …). In this category we include several techniques for logging user low-level actions, others techniques allowing for gathering users' physiologic information, and others capable of recording verbal (and non-verbal) cues coming from the user's side (collected through a webcam and/or a microphone).

It is worth pointing out that, while there are techniques that rely on commonly available support and can be used without almost any regard with the particular platform considered (see for instance server side logging techniques), other techniques (e.g. eye-tracking) require specific hardware, whose use cannot neglect the particular platform in use.

4.1 Logging (Server Side)

This technique refers to Web-based applications and allows for collecting data at the server side. Its effectiveness is strongly limited by the impossibility to capture local user interaction with the user interface techniques (menus, buttons, fill in text, use of anchor links within the same page or "back" button,…) and by the validity of the server logs, which cannot capture the accesses to the pages stored into the proxy servers and the browser cache. For instance, if the requested page is in the browser cache then the request will never reach the server and is thus not logged. Moreover, multiple people can also share the same IP address, making it difficult to distinguish who is actually requesting what pages. Dynamically assigned IP addresses, where a computer's IP address changes every time it connects to the Internet, can also make it quite difficult to determine what an individual user is doing since IP addresses are often used as identifiers. Thus, interpreting the actions of an individual user is extremely difficult, as methods for capturing and generating Web usage logs are not designed for gathering useful usability data, as pointed out by some work (Etgen and Cantor 1999; Davison 1999; Pitkow and Pirolli 1999; Choo et al. 1998; Tauscher 1999).

Another method is to ask surfers to online register at the first visit and logon every other visit. In this setting, the Web server can construct an individual profile for each visitor, and track all user behaviours without ambiguity. The Web server stores users' logon name, their personal information, such as age, gender and occupation, and the visited pages. Such datasets are very rich, and statistics on types of Web surfers, their interests and browsing habits can be generated with the Web mining process. This technique is widely adopted by firms selling digital information products (e.g. online newspapers), which request the users to logon before enabling file downloads. However, there are two main limitations. First, Web visitors' desires

are greatly reduced if they are required to logon every time they visit the site. It becomes a serious issue for online firms and, even for Web sites providing free registration, online users may re-register or provide fake details. The statistics will become blurred, and this will result in invalid and confusing conclusions. Second, the online firms cannot keep track of the visitors once they leave to other Web sites. All generated knowledge is limited to only a single Web site.

4.2 Proxy-Based Logging

This solution still supports Web-based application through an intermediate server between the client and the content server. Proxy servers are even less intrusive and not require any modification in the Web application to evaluate but they limit their analysis to the page accessed and are not able to capture the local user interactions. The proxy approach has three key advantages over the server-side approach. First, the proxy represents a separation of concerns, then any modifications needed for tracking purposes can be done on the proxy, leaving the application server to deal with just serving content, which makes it easier to deploy, as the application server and its content do not have to be modified. Second, the proxy allows anyone to run usability tests on any Web site, even if they do not own that Web site. Lastly, having testers go through a proxy allows Web designers to "tag" and uniquely identify each tester. Furthermore, a proxy logger also has advantages over client-side logging. For example, it does not require any special software on the client beyond a Web browser, making it faster and much simpler to deploy. Therefore, the proxy makes it easier to test a site with different test participants, operating systems and Web browsers than a client-side logger does, so allowing testing with a more realistic sample.

An example of this kind of solution can be found in WebQuilt (Hong and Landay, 2001), which uses a proxy logger to capture user accesses on the Web. As a proxy, it lies between clients and content servers, with the assumption that clients will make all requests through the proxy. Traditionally, proxies are used for things like caching and firewalls, in WebQuilt the Web proxy is used for usability purposes, with special features to make the logging more useful for usability analysis. However, although the proxy-based technique seems quite appealing, there are still limitations on what the WebQuilt proxy logger can capture. The most pressing of these cases is links or redirects created dynamically by JavaScript and other browser scripting languages. As a consequence, the JavaScript generated pop-up windows and DHTML menus popular on many Web sites are not captured by the proxy. Another situation that WebQuilt cannot handle is server-side image maps. Other elusive cases include embedded page components such as Java applets and Flash animations. As technologies change and develop, the proxy will need to be updated to handle these new cases.

4.3 Logging (Client Side)

In this category various techniques are considered. Before analysing them, it is important to remember that client logging is a technique that can be applied not only to Web applications but also to Java and Microsoft applications with similar results, as many tools have been developed for this purpose as well.

In addition, it has been pointed out that through logging user interactions with a given application, we could infer patterns of user behaviour that indicate usability problems or other design deficiencies. This possibility has obvious attractions for Web designers, but in the HCI usability research some issues have been raised regarding the possibility of identifying usability problems without access to the use context, to the users tasks and goals and to the

user's own reports of what counts as a problem for them. Thus, logging techniques alone are unlikely to provide useful results to the evaluators.

Cookies. A method is to install cookie at Web client computers. A cookie is a small text file that the Web server embeds in the browser for identifying the user. If the user provides his name, when he comes to a new site supporting cookies, his name is stored in a plain text file at the client computer. Therefore, no data is stored at the server side, but every time the same browser asks for the page or the same Web site, HTTP sends the cookie to the Web server which use it to identify the user and display personalised information, such as name-calling greetings. One of the advantages of using cookies is the ease of implementation. However, there are two drawbacks. First, the amount of information stored in cookies is limited (the average size is about 4Kbyte) and therefore, strictly speaking, no Web mining process can be performed based on such a limited information. Second, since the cookies are saved as plain text, they can be easily retrieved at the client computers: hence security and privacy can be at risk.

Client-side logs. They capture more accurate, comprehensive usage data than server-side logs because they allow all browser events to be recorded, and it might provide useful insight for usability evaluation. One alternative to gathering data on the server is to collect it on the client. Clients are instrumented with special software so that all usage transactions will be captured. More specifically, clients can be modified either by running software that transparently records user actions whenever the Web browser is being used (as in Choo et al., 1998), by modifying an existing Web browser (as in (Tauscher, 1999)), or by creating a custom Web browser specifically for capturing usage information (as with (Vividence, 2000)). The advantage of client-side logging is that literally everything can be recorded, from low-level events such as keystrokes and mouse clicks to higher level events such as page requests. All of this is valuable usability information. However, there are some potential drawbacks to client-side logging. First, special software must be installed on the client, which end-users may be unwilling or unable to do. This can severely limit the usability test participants to experienced users, which may not be representative of the target audience. Second, there needs to be some mechanism for sending the logged data back to the team that wants to collect the logs. Third, the software, in some cases, is platform-dependent, meaning that the software only works for a specific operating system or a specific browser.

(Paganelli and Paternò, 2003) developed a tool for performing client logging of Web applications: the main advantages are that it does not require expensive equipment, and facilitates the problem of modifying the pages evaluated since it automatically included JavaScript code in all the pages that have to be evaluated. Such Javascript snippets are able to adapt to the various features of different browsers. Using a browser log-based analysis, the evaluator can accurately measure time spent on tasks or particular pages as well as study the use of the back button and user clickstreams. It is also possible to precisely identify the downloading time and the time when the page is visible to the users. In addition, their tool is able to automatically analyse the information contained in Web browser logs and compare it with task models specifying the designer model of the possible users behaviours when interacting with the application in order to identify whether and where users interactions deviate from those envisioned by the system design and represented in the model. Within this client-side techniques we cite also the work (Ho, 2005), developed in the e-commerce domain area, which is about the use of a user remote tracker to examine Web Users' characteristics, trying to draw a linkage between Web customers' characteristics and their browsing behaviours. The authors propose a user remote tracking framework based on Web services and XML in order to track every HTTP request from client computers to understand surfers'

characteristics. The user-remote tracker is a piece of software installed in the user browsers to keep track of every keyboard input and mouse click from the users. No matter the users input, all HTTP requests and responses are tracked by the software program, including interactions with Java Applet programs. This piece of program will automatically send the activity log file, together with the user identity, to a central machine for Web mining process (instead of sending such information directly to the Web server). It is such central machine that analyses click streams and generates navigation rules of these users through some algorithms. There are several advantages with this user-remote tracker. First, it can follow users everywhere. Second, while server logging cannot track the interaction between a user and an applet program, the tracker can solve this problem. Third, in the traditional data collection method, it is possible to get little information once the users enter the secure Web sites (i.e. Web sites started with https://). Here, since the user-remote tracker uses low-level programs to track every user input signal, the activities can be tracked even in this case.

4.4 Eye-Trackers

Eye trackers are a technique for measuring users' eye movements so that it is possible to know both where a person is looking at any given time and the sequence in which their eyes are shifting from one location to another, on the screen. Tracking people's eye movements can help evaluators understand visual information processing and the factors that may impact upon the usability of system interfaces, so providing an objective source of data that can inform the design of improved interfaces. However, evaluators using eye-tracking should take into account the limits of such technology and how such limits impact the data collected. For example, an appropriate minimum threshold time for a fixation should be carefully identified since interpretations can vary a lot according to the time set to detect a fixation in the eye-tracking system. Moreover, eye trackers might have difficulty tracking participants who have lenses. Furthermore, visual distractions (e.g., colourful or moving objects around the screen or in the testing environment) should also be eliminated, as these will inevitably contaminate the eye-movement data. Also, eye tracking generates huge amounts of data, so it is essential to automatically perform filtering and analysis. However, eye tracking technology has evolved in recent years and there are now more systems that can be used for remote evaluation (see for example the Tobii system http://www.tobii.com/) since they can be transported in suitcases and do not require that users wear intrusive equipment. It is only necessary an initial standard training exercise. Nevertheless, one of the most relevant problem with eye-tracking technique remains the fact that it is possible to know what users see but not what users think about what they see, in other words, how data are actually being processed by the person.

4.5 Webcam/Audio Recorders or Microphones

The use of Webcam and audio recorders allows for acquiring more contextual information about the data collected. Indeed, as it has been previously mentioned, through logging keystrokes and Web pages on a given site, we could infer patterns of user behaviour that indicate usability problems or other design deficiencies, but in the HCI usability research it has been argued that it is not possible to identify usability problems without access to the use context, to the users tasks and goals and to the user's own reports of what counts as a problem for them. Webcam-based videos are very valuable when further analysis is necessary whenever an error is found, as the evaluator can analyse the videoclip and convert it into a usability problem description or use it in any case to understand the reason of a usability problem. For instance, videos can be valuable in capturing facial movements/expressions,

verbal/vocal signals and expressions, non verbal communication, body language and posture. Moreover, facial expressions may provide indications of the immediate appreciation of the system by showing the instantaneous reactions to the system, and also might reflect subject's considerations about the system. Furthermore, especially the use of more than one camera is valuable for capturing some environmental conditions occurring in the tester environment. Work by Lister (2003) has been oriented to using audio and video capture for qualitative analysis performed by evaluators on the result of usability testing.

Also in the work of Paternò et al. (2006) Webcams are used to record the users (not the users' screens) to provide valuable information for interpreting problematic parts of the user interaction: for instance, in this work, videos are also used to check the user behaviour whenever some measurements (e.g. time needed for completing a task) captured by another software component provide unexpected values.

4.6 Sensors

Under this voice we mean more sophisticated research solutions for data acquisition and analysis of some physiological data. Recently, a number of sensors are being more and more used for the evaluation of user interfaces, trying to take into account the emotional dimension of computer-human interaction (e.g. affective user interfaces). Amongst such measures we cite physiological signals like ECG, respiration, galvanic skin response, heart rate, skin temperature. Most of them, such as Galvanic Skin Response (GSR), Heart Rate (HR) and Blood Volume Pulse (BVP) are generally chosen as good, physically non-invasive indicators of stress (under stress, GSR and HR increase, whereas BVP decreases), and are also easy to be measured with specialised equipment. In this respect, we mention the ProComp system, manufactured by Thought Technology Ltd (http://www.thoughttechnology.com), or the BIOPAC system (http://www.biopac.com/), which allows for recording different kinds of data: physiological signals, vocal/verbal signals, and non verbal signals (posture, gaze direction, facial movements). Unfortunately, the use of sensors in remote usability evaluation is currently suffering the limitation of the high specialised equipment necessary, which cannot be assumed available in users' daily environments (although it is slowly appearing and more and more used in telemedicine applications). However, more research effort is envisaged in the next years on this subject for the useful information that it can provide to analyse user emotional state.

To summarise, almost all the results obtained with each technique indicated in this section requires from the evaluator additional knowledge about the user in order to be actually useful for the purposes of the evaluation. Therefore, the big issue is that such data are not informative "per se" about possible usability problems, but require further comparison with supplementary information. One of the few exceptions can be identified in, for instance, recording user positively (or negatively) commenting on while interacting with the application in a remote think aloud session (which should theoretically provide the evaluator with an immediate feedback about user's satisfaction). In almost all the other cases a further contextualisation (and integration) of the data collected is needed in order to correctly evaluate the session state (think about, for example, the uselessness of logging mouse and keyboard actions without contextualising such actions within teh current user intention). One of the current issues is identifying techniques enabling an easy synchronisation and aggregation of all such different sources of information in some semantic context, so as to facilitate the evaluator's work.

5 The type of Application Considered

In this section, we analyse another dimension of the proposed framework: the type of application, mainly considered in terms of the underlying software environment. As it happened with already analysed dimensions, also the consideration of this dimension is not completely independent from the other ones. Indeed, the type of application considered may prevent (or strongly promote) not only applying some specific techniques mentioned in the previous section, but also the use of particular interaction platforms. For instance, while in case of web-based applications we have seen that there are several options about where the logging tool should work (e.g.: server, client, proxy, ..) regardless of the particular platform at hand, the consideration of .NET-based applications for remote evaluation only belongs to recent years and it is almost connected with stationary platforms since only recently prototypal tools to support the evaluation of .NET applications for mobile devices appeared, still limited in terms of the information they are actually able to provide.

Indeed, the first applications that were evaluated with some type of remote evaluation were graphical applications, often implemented in languages such as Java (see for example Paternò and Ballardin, 2000). Then, with the advent of the Web and the related easiness of performing a remote evaluation when Web is considered (due to the related simplicity in involving a high number of testers with little effort), the majority of methods have considered Web sites as their primary evaluation targets. Java-based applications indeed have been taken into account, sometimes as a sort of side-effect in the willingness of improving the flexibility of techniques considered for Web applications whenever applets are also included. An example of this can be found in the already mentioned work of Ho (2005), developed in the e-commerce domain area: it is about the use of a user remote tracker to examine Web users' characteristics, trying to draw a linkage between Web customers' characteristics and their browsing, and with the capability of tracking client-side logs including interactions with Java Applet programs. Microsoft .NET applications have been considered as well, for example for PDA devices, for which they often provide more robust and supported solutions with respect to Java. An example of logging tool for Microsoft environments is the VibeLog logging tool (http://research.microsoft.com/vibe/), which has been developed at Microsoft Research to research the ways that work practice might change as users move in between various sized displays during their work day by means of marrying the logging tool with ethnographic research data, which should provide good indications of what parts of the designs of Windows and Office do not scale well across different display sizes. This analysis is used to understand where they should orient their research efforts in novel visualization and interaction, with an eye toward designing more elegant UIs.

6 The type of Evaluation Results

Before analysing in depth the last dimension of the proposed framework, the type of results an evaluation can deliver (for instance qualitative vs. quantitative data), we judge useful mentioning the work by Petrie et al. (2006) about remote evaluation. In this work the authors highlight how both formative and summative evaluations can be supported by remote techniques. Indeed, in summative evaluations one of the main goals is to understand whether the users can install and run a system on their own and on their own machines, and how they rate the key functions of the system. Therefore, the disparate environments and configurations that can be reached with remote evaluation can provide highly reliable data with this respect. In formative evaluations the objective is to collect information about design flaws and inform

re-design; therefore, it is particularly important that participants feel free to criticise a system and to avoid evaluator bias, and this may be easier if they are in the privacy of their own environment, rather than the potentially more threatening situation of the usability laboratory.

However, the authors make a step further claiming that, in particular, remote asynchronous techniques in which the evaluator cannot intervene during the user sessions are especially useful for summative evaluation. To support this idea, the authors report on two evaluations conducted with disabled users, In both cases they performed both a local and a remote evaluation. The technique used for remote evaluation was in one case making notes on problems encountered and then sending them to the evaluator, and in the other case record problems encountered and then sending them along with ratings of the web sites accessed. Both remote and local evaluations provided considerable quantities of qualitative data, but the local evaluations provided far richer data as the researchers are able to record problems that the participant may not have been aware of, and are in a position to prompt the participant to explore these problems, comment on them, and analyse what had caused them.

Therefore, on the one hand, achieving the rich interaction between participants, researchers and developers, as requested by formative evaluation, is very difficult in remote evaluation situations, although with high quality video conferencing, broadband connections and remote recording systems it might be possible to conduct remote evaluations that capture rich set of data. On the other hand, if the evaluation is summative, a remote evaluation may be quite appropriate because it adequately shows real user behaviour.

We agree with this position to some extent since in our opinion remote evaluation can provide different types of results, which can in turn be used for different purposes, both summative and formative. Indeed, in this section we are going to analyse the type of result an evaluation can deliver. In particular, a discussion about the type of information that can be useful to obtain in order to analyse the multimodal data regarding user sessions is provided. This information can be quantitatively determined by specific software and highlighted during the evaluation (tasks not completed, errors occurring during the performance of tasks, time for completing a task, etc.) together with other information that deals with intrinsic qualities of the user interface (e.g.: time needed for the performance of a task). It is not surprising that some relations exist between the evaluation techniques mentioned in Section 4 and the evaluation results analysed in this section (for instance, sensing technologies delivers quantitative data about user emotional and physical state), while in other case such correspondence is not so straightforward.

6.1 Task-Related Information

Many applications are task-oriented and consequently some important aspects to consider are whether the users are able to accomplish the desired tasks and information regarding task performance (task duration, number of actions, …). Here one issue is how to know what the desired tasks are. One possible solution is to ask users to explicitly indicate them at the beginning of the session. Associated to the issues related to task performance there is that corresponding to user errors: , which are actions not useful for the current task. The errors are good indicators of bad usability and difficulties in task accomplishment. Tasks can be considered at various granularities. In some cases, it can be interesting to analyse performance of short basic tasks, in other cases it is important to focus on high-level complex activities to perform.

During analysis of task performance it can be useful to analyse when it deviates from the ideal expected behaviour and to what extent. Then, the evaluator has to understand the reasons for

such mismatch and needs to go back and analyse what happened for each action in the user session and what factor triggered the deviation.

6.2 Qualitative Information

Under this voice we mean all the techniques that allow evaluators to collect qualitative data from the users. As we already mentioned, qualitative data are quite relevant especially if the kind of evaluation is formative, therefore the richness of the qualitative data is very important in understanding how to improve the system. For example, gathering informal and spontaneous comments in natural language from the users undoubtedly offers valuable information to the evaluators for improving the resulting design. Also, since this information can provide a rich contextual knowledge about the situation currently occurring during the user's interaction, it may also be used for cross-checking other data collected and which may be found ambiguous to be interpreted –generally, quantitative data. An example in which this strategy might disambiguate other data is the case of a user spending long time visiting a page: considering only this quantitative information (registered by the browser logging tool) would not allow the evaluator to assess whether the users found the information very important or they just had problems in finding the concerned information: in this case the Web-cam can help in correctly interpret the feeling of the user (engaging or not the visit).

6.3 Presentation-Related Data

In this section we analyse the results that the evaluation should deliver about the usability of the user interface presentation (which means, e.g. for GUIs: layout, choice of widgets, colours, labels, etc.). There are tools that link the task performance with the user interface elements supporting such performance; in other cases the tools are able to provide reports that highlight the user interface elements that might be problematic from the usability point of view. For example, WebQuilt (Waterson and others, 2002) provides representations consisting of nodes representing visited web pages, and arrows representing the traffic between the pages. Entry pages are coloured green, and exit pages cyan. Thicker arrows represent heavier traffic. Arrow colour is used to indicate time spent on a page before transitioning, where the closer the arrow to red, the longer spent in transition. The designer's path is highlighted in blue. There is a slider along the left hand side that allows the designer to zoom into the graph, viewing actual images of the pages users saw, and where they clicked.

6.4 Quantitative Cognitive-Physiological Information

Quantitative psycho-physiological measurements can provide useful information about more general, qualitative information on human's feeling in a specific situation. For instance, nowadays, with a growing population of elderly persons, this result is expected to be more and more applied in the field of elderly care/assistance, where there has been an increasing interest in investigating algorithms able to enable the possibility of assessing elderly mood in a non intrusive manner. In order to make state-of-mind information available, sensor technology can be employed. Various psycho-physiological signals are known in literature that can convey the presence of strong emotions or stress (Cacioppo et al., 2000): skin conductance, muscle tension, heart rate and heart rate variability. Such signals can now be measured in an unobtrusive manner. The measured signals then will have to be analyzed in order to reliably convey short term mood changes (that might be relevant for the relatives and form a basis for an enhance feeling of connectedness) as well as long term trends. When the

shape of the people is mostly visible, computer vision tools can be used to classify their posture, and gait and posture changes along time. This information can be exploited, for example, to predict (by gait analysis) and detect (by analyzing posture changes) falls. Computer vision techniques can be used to detect the head position in real time, classify the face orientation (frontal, profile) in order to provide the process of facial expression analysis with suitable data. In addition, faces are processed for expression/recognition/authentication. In case a person is not visible or the user does not like a camera to be used (e.g. in the bath room), speech/audio tracking is an alternative.

Eye-tracking systems can provide many interesting pieces of information derived from fixations and saccades. Long fixations can indicate that users spend too much effort to interpret or to process what they are looking at. The number of fixations is often related to the user efforts to process the content of the screen area under analysis. The duration of the scanpath can be considered as a productivity measure and can be compared with a theoretical optimal duration. Even the ratio between saccades and fixations can be a useful index to compare percentage of time spent in looking information (saccades) and that during which information is acquired (fixations).

7 An Example Tool for Remote Evaluation: MultiModalWebRemUsine

In this section we discuss an example tool for remote evaluation according to the framework presented in the report. The basic idea of this tool is to analyse user logs through the semantic information contained in task models. Thus, on the one hand, we have a task model that describes how designers expect that users perform their activities, and, on the other hand, there are logs indicating the actions performed by the users while interacting with the application. Each user session can be defined through the sequence of the corresponding user actions, which can be associated with a corresponding sequence of basic task performance in order to achieve the user goal. If the task sequence performed diverts from those enabled by the task model there is clearly a mismatch that needs to be analysed by the evaluators because either the task model is too rigid or there is something unclear in the user interface, which prevents the user from performing the expected sequences of tasks.

Various versions of the tool have been developed, which vary for the type of application addressed and the type of results provided. The first version USINE (Lecerof and Paternò, 1998) mainly addressed the issue of using task models for analysing user logs without considering its use as remote evaluation. The next version, RemUSINE (Paternò and Ballardin, 2000), was developed for remotely evaluating desktop Java applications and was tested in industrial sites providing useful information regarding its possibilities, even in comparison with other methods. For example, it was compared with a video-based evaluation. It turned out that for evaluating a small number of sessions the video-based evaluation was more efficient since RemUSINE required some time to enable the automatic evaluation given that the evaluator has first to provide the task model of the designed application and create mappings between basic tasks and log events. On the positive side, it was noted that video analysis in some cases is not able to detect quick user actions (such as some user clicks) and is not usable for evaluations when users are located far from the evaluator.

Then, given the explosion of the Web, which has become the most common user interface, we thought useful to develop a new version (WebRemUSINE) aimed to evaluating this type of application (Paganelli and Paternò, 2003). We had also to decide how to log user interactions and for this purpose we implemented an efficient, interoperable, client-side logging system. In

addition to information regarding task performance, the web-oriented version provides a lot of information regarding the Web pages analysed: visited pages, never visited pages, extent of scrolling and resizing, page patterns, download and visit time. Some information is provided along with summary data regarding the content of the page. Thus, for example, the visit time is provided indicating also the number of forms, links and words in the page so that the evaluator can compare the visit time also with the quantity of information available in the page. The last version of the tool (MultimodalWebRemUSINE) aims to exploit the possibilities opened up by recent technologies to gather richer set of information regarding user behaviour. Thus, the traditional graphical logs can be analysed together with the logs from webcams and from portable eye trackers, which do not require the use of intrusive equipment.

In summary, the changes on the tool mainly aimed at fulfilling the evolving needs of usability evaluators. Indeed, the tool started from the original vision of having cost-effective techniques for usability evaluation able to analyse data about the product usage in a real-world environment. To this end, remote evaluation is valuable when trying to keep the budgets down while staying competitive in the marketplace (which is especially relevant for companies). Also, the necessity to reach larger, more diverse and disperse pool of participants stimulated the attention to web applications. Indeed, in these times of global customers and development organizations, there is a clear correlation between the globalization of the product market and the potentialities (and challenges) of remote evaluation. Next, the tool kept evolving on these directions with an eye toward the improvements (in terms of robustness and affordability) of technology and broadband infrastructure available, which were efficiently exploited for enriching the tool with multimodal information on the user's behaviour. The objective was to compensate the recognised evaluator's decreased ability - typically connected with remote usability evaluation techniques- to interpret the motivations underlying a certain user behaviour, due to separation in space (and sometimes also in time) between the user and the evaluator.

In general, MultimodalWebRemUsine is mainly based on a comparison of planned user behaviour and actual user behaviour. Information about the planned logical behaviour of the user is contained in a (previously developed) task model, while data about the actual user behaviour is provided by the other modules supposed to be available within the client environment (the logging tool, the Web cam and the eye-tracker).

Before starting the test, users have to explicitly indicate the target task; after that, all the user actions will be automatically recorded. The evaluation then analyses the user's sequences of actions to determine whether the user has correctly performed the tasks in accordance 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, by means of an internal task model simulator: for each action in the log, first the corresponding basic task is identified and next there is a check to see whether that task was logically enabled. If no error occurs, the list of the basic tasks that have been enabled after its performance is provided together with the updated list of high-level tasks already accomplished, so as to allow the evaluator to check if the target task has been completed. Otherwise, some error will be notified in the report analysing the user session. An example of error is a precondition error, which means that the actual user's task performance did not respect the relations defined in the system design model. For example, if people want to access a remote service (such as Web access to emails), usually they have to provide username and password and then activate the request through a button. If the user interfaces elements are not located in such a way that the user can easily realise that both

fields have to be filled in before connecting to the mail box, then some precondition errors can occur (for example the user sends the request without proving first the password). Such types of errors can be detected through this type of approach.

From the log analysis the tool can generate various indications:

- *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.

Recently, we have paid attention on how to represent user sessions and related data in such a way to ease their analysis. Figure 1 shows the type of representations designed. It is possible to show at the same time data related to several sessions in different ways. In the example in Figure 1 we analyse the parts of the sessions about users who wanted to become member of an association. As you can see from the selected radiobuttons, for the first two users the deviation graph is shown, while for the other ones the state graph is visualised. In both types of graphs the white circles are associated with the basic tasks performed and their positions indicate when they have been accomplished. In the first diagram (deviation diagram) there are three lines: one for the basic tasks correctly performed, one for those uselessly performed and one for the tasks that have diverted the user from achieving the current goal. In the state diagram the colour of the line underlying the white circles is used to indicate whether the user is correctly or wrongly performing the task

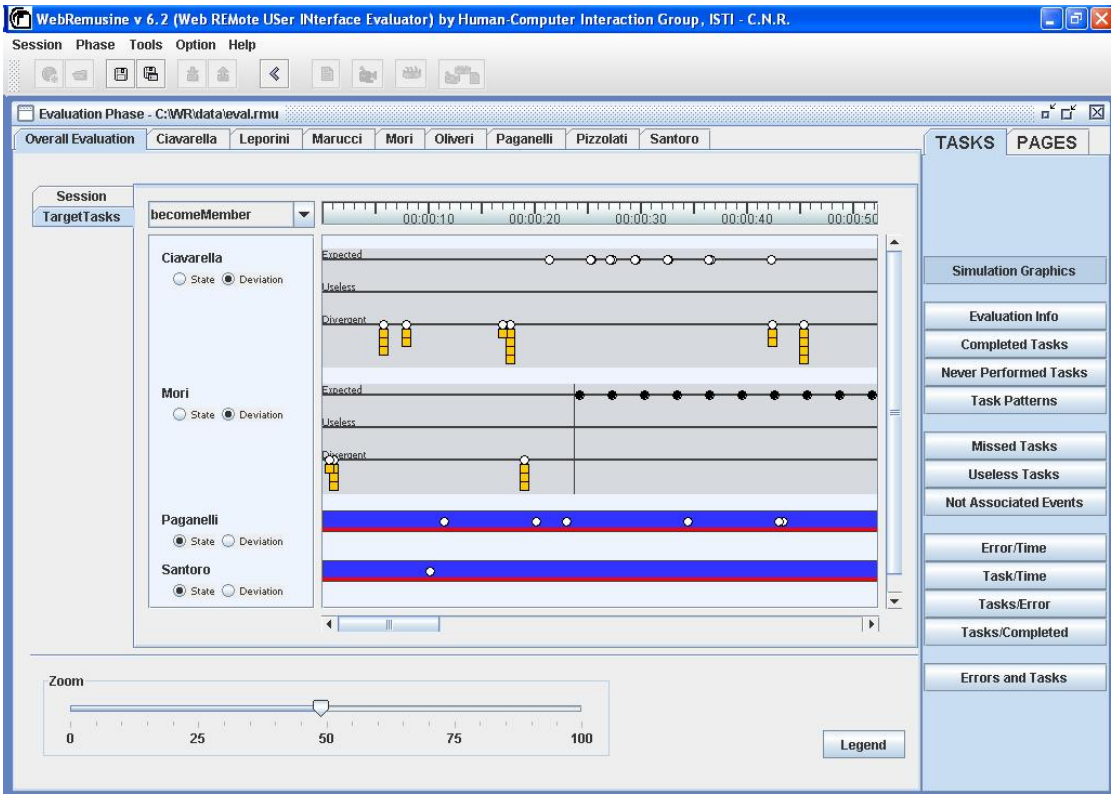

Figure 1: Representation of user sessions in MMWebRemUsine.

A further type of information considered during the evaluation regards the task execution time. In case of tasks correctly performed, the tool calculates the global time of performance. This information is calculated by examining the temporal information associated with each event and stored in the logs. 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. In Figure 2 a screenshot of the tool is presented: as you can see, whenever an inexplicably too high time for carrying out a certain task is registered by the tool, the evaluator can activate the related video recorded through a Web cam to gather further information.

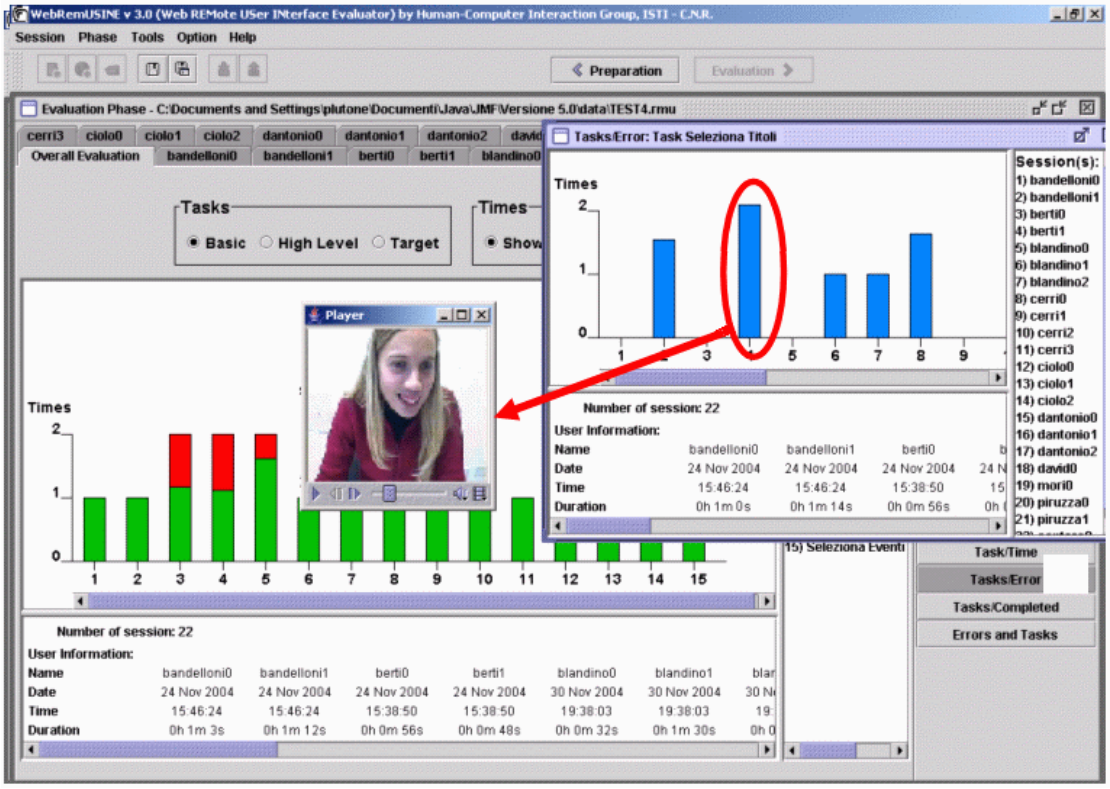

Figure 2: Highlighting a video within the MultiModalWebRemusine environment.

The approach supported by MultiModalWebremusine is also able to provide usability data associated with every single presentation. Moreover, it is worth noting that, as this approach can put in correlation task-based measures with presentation-related data, it can also analyse the usability of the Web site from both viewpoints. For example, the tool can compare the time to perform a task with that for loading the page(s) involved in such a performance.

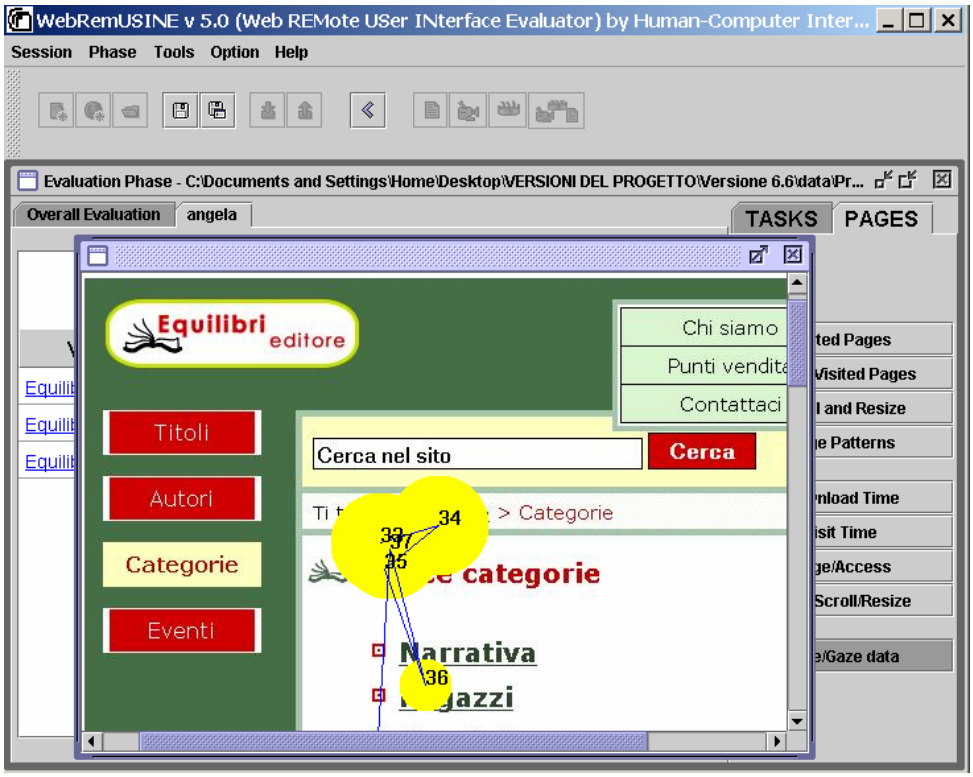

Figure 3 An example of visualisation of fixations (the yellow areas) and scanpaths (the paths in-between) registered by the eye-tracker.

The eye-tracking is a technique that has been used within MultiModalWebRemUsine: in Figure 3 a screenshot has been taken, showing how the registrations of the eye-tracker are visualised within the MultiModalWebRemUsine environment. As you can see, each fixation is represented by an area whose size is proportional to its recorded duration, while the lines connecting such areas (scanpath) highlight the path the user followed while visiting the page.

8 Discussion and Interrelationships between the Framework Dimensions

We can discuss possible interrelationships among the different dimensions that we have identified while analysing several contributions in the area of remote usability evaluation. First of all, the choice of the platform type might substantially limit –in terms of hardware and software– the type of technology and/or techniques that can be used for remote evaluation, as well as put less/more emphasis on the relevance of particular information for the purposes of the evaluation. Indeed, apart from the well-known differences between the type of applications that might be supported by a cellphone and by a desktop system, due to the diversities in hardware and software capabilities, the type of platform used may also affect the relevance of a piece of information with respect to another one. This is the case, for instance, of the user context, which is very important for mobile applications and less important for, e.g. desktop systems.

Furthermore, another observation is the fact that the same type of information useful for the evaluation –e.g. user workload– can be gained with different techniques. For instance, a possible indication for user workload might be a user blinking very frequently (information

that can be gained from an eye-tracker), but some physiological data can –more reliably– signal this workload. Another example regards the user feedback, which can be gained with different means: a user nodding (captured by the Webcam) might be a sign of a good feedback, as well as a user using some positive vocal expressions. In other cases there is information useful for evaluation (e.g.: task-based data) which cannot be derived without explicitly including additional information (task specification).

Moreover, the type of interaction between the evaluator and the user might also affect the use of the particular technique(s) adopted, as well as the quantity (and also the quality) of information collected for the evaluation. For instance, while the use of remote questionnaires indicates a specific type of techniques for collecting information about the user, in case of automatic collection of data the range of techniques can greatly vary and so does the range of the evaluation results that can be derived from interpreting the collected data.

Lastly, as we already noticed, the type of application considered may prevent (or strongly encourage) the application of certain techniques, as well as the use of specific interaction platforms. For instance, in case of web-based applications, the use of server-side and client-side logging techniques is a well-known and established approach, whereas the consideration of user interactions with additional software components like .NET for remote evaluation only belongs to recent years, and is often restricted only to specific types of platforms.

9 Conclusions and Future Challenges

In this report we have described a framework composed of different dimensions that we have identified as relevant in the area of remote usability evaluation. This type of evaluation is becoming more and more important in a time of globalization of companies and their customers. We have used such a framework to review a large spectrum of methods that have been proposed in this area, which has been receiving more and more interest due to the improvements of the techniques able to capture information regarding the user behaviour and the validity of the data which are collected "in the field". Therefore, the report is aimed to put some light on the different methods through a common framework on which it is possible to reason and to compare current works in the area of remote evaluation, as well as delineate possible future trends in the research agenda of remote usability evaluation. In addition, the report is useful for identifying which are the current strategies for compensating some traditional weaknesses in this type of evaluation. For instance, future work should be dedicated to extending the data detected regarding the user behaviour and state, including the emotional state, in order to have a more complete analysis of what happens during user sessions and better identify the potential usability issues. Another novel emerging application area is that of mobile applications in which is important to understand how task performance varies depending on the changes in the context of use. In order to make the discussion more concrete we have also reported our experience with our tool in the area of remote evaluation and analysed it according to the dimensions of the logical framework proposed.

10 References

Avouris N., Komis V., Margaritis M., Fiotakis G., (2004), An Environment for Studying Collaborative Learning Activities, Journal of Educational Technology & Society, Special Issue on Technology – Enhanced Learning, 7 (2), pp. 34-41, April 2004.

Cacioppo, J.T., Berntson, G.G., Larsen, J.T., Poehlmann, K.M., & Ito, T.A. (2000). The psychophysiology of emotion. In: M. Lewis, R.J.M. Haviland-Jones (Eds.), The Handbook of Emotions (2nd Ed.; pp 173-191). New York: Guilford Press.

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.

Choo, C. W., Detlor, B., and Turnbull, D. 1998. A Behavioral Model of Information Seeking on the Web—Preliminary Results of a Study of How Managers and IT Specialists Use the Web. In Proceedings of the 61st Annual Meeting, 35, pp. 290–302.

ClickFox (2002), ClickFox Inc., http://www.clickfox.com.

Davison, B. 1999. Web Traffic Logs: An Imperfect Resource for Evaluation. In Ninth Annual Conference of the Internet Society (INET'99). San Jose, CA, June 1999.

Denis, C. and Karsenty, L., Inter-usability of Multi-device Systems: A Conceptual Framework, in A. Seffah and H. Javahery (eds.), Multiple User Interfaces: Engineering and Application Framework, John Wiley and Sons, New Jersey, 2003.

Etgen, M. and Cantor, J. 1999. What Does Getting WET (Web Event-Logging Tool) Mean forWeb Usability? In Proceedings of the Fifth Conference on Human Factors and the Web. Gaithersburg, MD, June.

Hartson, R. H. Castillo, J. C., Kelso, J. T., Neale W. C.. Remote Evaluation: The Network as an Extension of the Usability Laboratory. CHI 1996, pp. 228-235

Ho, S.Y., An Exploratory Study of Using a User Remote Tracker to Examine Web users' Personality Traits. Proceedings of the 7th International Conference on Electronic commerce, ICEC'05, August 15–17, 2005, Xi'an, China, ACM Press, pp. 659 – 665.

Hong, J.I. and Landay, J.A. WebQuilt: a Framework for Capturing and Visualizing the Web Experience. WWW 2001 conference: pp.717-724.

IQ Services (2006) IQ Services: Interactive Quality Services, Inc. http://www.iq-services.com/

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.

Lecerof, A., Paternò, F. (1998), Automatic Support for Usability Evaluation, IEEE Transactions on Software Engineering, Vol.24, N.10, pp. 863-888, IEEE Press, October 1998.

Lister M., (2003) Streaming Format Software for Usability Testing, Proceedings ACM CHI 2003, Extended Abstracts, pp.632-633.

Paganelli, L. and Paternò F., 2003. Tools for Remote Usability Evaluation of Web Applications through Browser Logs and Task Models, Behavior Research Methods, Instruments, and Computers, 2003, 35 (3), pp.369-378, August 2003.

Paternò, F. Ballardin, G., RemUSINE: a Bridge between Empirical and Model-based Evaluation when Evaluators and Users are Distant, Interacting with Computers, Vol.13, N.2, pp. 229-251, Elsevier, 2000.

Paternò, F., Piruzza, A., Santoro, C.. Remote Web Usability Evaluation Exploiting MultiModal Information on User Behaviour, Proceedings CADUI 2006, Bucharest, Springer Verlag, June 2006.

Petrie, H., Hamilton, F., King, N. and Pavan, P., Remote Usability Evaluations with Disabled People, CHI 2006 Conference Proceedings, Montréal, Québec, Canada, April 22-27, 2006.

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

Stoica, A., Fiotakis, G., Simarro-Cabrera, J., Frutos, H.M., Avouris, N., and Dimitriadis, Y., Usability Evaluation of Handheld Devices: A Case Study for a Museum Application, Proceedings of PCI 2005, Volos, November 2005.

Tauscher, L.M. 1999. Evaluating History Mechanisms: An Empirical Study of Reuse Patterns in WWW Navigation. MS Thesis, Department of Computer Science, University of Calgary, Alberta, Canada.

Tennent, P., Chalmers, M., Morrison, A., Replayer: Collaborative evaluation of mobile applications, CHI'06 Workshop on Information Visualization and Interaction Techniques for Collaboration across Multiple Displays, Montreal, Canada.

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.

http://www.vividence.com/resources/public/solutions/demo/demo-print.htm.

Waterson, S., Landay, J.A., Matthews, T., In the Lab and Out in the Wild: Remote Web Usability Testing for Mobile Devices. CHI 2002, April 20-25, Minneapolis, USA.

West, R., and Lehman, K.R., Automated Summative Usability Studies: An Empirical Evaluation. CHI 2006, April 22–27, 2006, Montréal, Québec, Canada, ACM, pp. 631-639.