Chapter 9
Remote Usability Evaluation: Discussion of a General Framework and Experiences from Research with a Specific Tool

Fabio Paternò and Carmen Santoro

Abstract The goal of this chapter 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 to exploit all the possibilities opened up by remote evaluation.

9.1 Introduction

In this chapter, we present and discuss a design space for remote usability evaluation. This type of approach to usability evaluation is characterized 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 analyze user behavior 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, et al. 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.

E. Law et al., *Maturing Usability.*

They differ in 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 chapter, we provide a more updated and focused discussion of the state-of-the-art in remote evaluation through a more refined framework for this area that highlights the important aspects to consider when analyzing approaches within it. More specifically, the framework proposed in this chapter is defined by analyzing various dimensions. The first one considers 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 information provided, which are useful for the evaluation. Another interesting dimension we consider is the type of platform used for interaction. In this regard, we plan to distinguish access through desktop or mobile devices, for example, and discuss how the choice of 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 to obtain the desired information, therefore 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 summarize, the relevant dimensions we have identified for analyzing 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 behavior (graphical logs, voice and/or Webcam recordings, eye-tracking, etc.)
- The type of application considered in terms of implementation environment (Web, java-based, .NET, etc.)
- 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 that are still open.

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 websites that considers information from user tasks, log files, videos recorded during user tests, and data collected by an eye-tracker (Paternò, et al. 2006).

9.2 The Type of Interaction between the User and the Evaluator

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

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 synchronization occurring on *time*. Indeed, while the *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), it is the opposite with *synchronous* evaluations. 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 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 behavior 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 an 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 behavior and then analyze 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 ~~9.7~~ 9.7.

To assess the pros and cons of such options, we can notice that on the one hand, remote observation provides the evaluator with more capabilities for observing the session and 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, because 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 behavior 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 because 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.

9.3 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 nonprofessionals 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 supported modalities.

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 requires 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 the 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, to always be in a position to correctly derive the potential causes of a usability problem that occurred on the client side. As previously mentioned, gathering information about the current environment is extremely important for a mobile user, because the environment can often change, ~~and~~ it becomes less important for a stationary user interacting with a desktop application because the environment is almost fixed.

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

9.3.1 Desktop Applications

~~Because~~ 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 through 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 use on desktop systems due to some software limitations on the underlying hardware (e.g., PC Anywhere only operates on PC platforms).

9.3.2 Vocal Applications

As for the vocal platform, interest is arising in studying this modality because 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, and 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, and fax. Most companies are seeing such systems as an immediate cost saver because call centers are becoming too expensive to be operated by humans.

As speech applications advance, so does the need for a means to evaluate vocal user interfaces (VUIs), 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, and therefore fewer choices can be explored with a VUI in a certain time interval, with respect to what can be done with a corresponding GUI. One of the consequences from the point of view of usability evaluation is that, 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 corresponding GUI, without being necessarily a sign of a bad vocal user interface usability.

Furthermore, it should be kept in mind that only using speech as an interaction medium might represent a burden on users' memory, meaning 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 proven 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 and wireless phones, evaluations of VUIs in lab environments suffer from unrealistic settings because 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, because call recordings tell the VUI designer exactly what happened during each call. They have several shortcomings—they cannot tell the designer what the caller was trying to do, how the caller felt, or why the caller did what s/he did. Another shortcoming is the massive amount of effort required to analyze the calls. Lastly, in call logs, every interactive voice recognition (IVR) platform comes with extensive call-logging capabilities. While surveys and call recordings 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 part of the picture. Call logs do not provide a lot of context for helping to interpret the results. Therefore, because surveys, call recordings, and call logs provide different information for in-use situation analysis, it seems that careful consideration of their combined use as compensation for their various advantages and drawbacks may be a viable solution for the purposes of evaluating VUIs.

An example of tools for VUIs is 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 will be done on this subject.

9.3.3 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 multidevice system, and introduces the concept of interusability to designate the ease with which users can reuse their knowledge and skills for a given functionality when switching to other devices. In their paper, a framework for achieving interusability between devices is proposed. It is based on two components: 1) a theoretical analysis of the cognitive processes underlying device transitions, and 2) 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. (~~Waterson, et al.~~), 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, the work of Stoica, et al. (2005) is worth mentioning, 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, techniques to measure usability-related factors generally include 1) inspection methods, 2) testing methods, and 3) 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. This log file can be combined with a video recording of the visit allowing evaluation of 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 to solve the problem. The experiment was recorded by three video and two audio recorders for further analysis, using the ColAT analysis tool that interrelates activity and 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, and strategies used for solving the puzzles), and behavioral patterns of participants.

AQ1

Indeed, the importance of performing a comprehensive evaluation that can take into account data derived from multiple sources to adequately gain insight into large bodies of multisource 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 visualization, 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 that are better able to support analysis of how task performance varies depending on the context change.

9.4 The Techniques for Collecting Information about the User Behavior

In this section, we discuss the various techniques available for collecting information regarding the user behavior (task performance, use of mouse and keyboard, facial expressions, verbal comments, gestures, gazes, etc.). In this category, we include several techniques for logging low-level user actions, other techniques for gathering users' physiological information, and others capable of recording verbal (and nonverbal) 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 almost without any regard to 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.

9.4.1 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 that 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 because IP addresses

are often used as identifiers. Thus, interpreting the actions of an individual user is extremely difficult, because methods for capturing and generating Web usage logs are not designed for gathering useful usability data, as pointed out by some works (Etgen and Cantor 1999; Davison 1999; Pitkow and Pirolli 1999; Choo, et al. 1998; Tauscher 1999).

Another method is to ask surfers to register online at the first visit and log on with every subsequent visit. In this setting, the Web server can construct an individual profile for each visitor, and track all user behaviors without ambiguity. The Web server stores users' log-on names and 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 their 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 log on before enabling file downloads. However, there are two main limitations. First, Web visitors' choices are greatly reduced if they are required to log on every time they visit the site. It becomes a serious issue for online firms and, even for websites providing free registration, online users may reregister 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 go to other websites. All generated knowledge is limited to only a single website.

9.4.2 Proxy-Based Logging

This solution still supports Web-based applications through an intermediate server between the client and the content server. Proxy servers are even less intrusive and do not require any modification in the Web application to evaluate, but they limit their analysis to the accessed page 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. 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 because the application server and its content do not have to be modified. Second, the proxy allows anyone to run usability tests on any website, even if they do not own that website. 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 side 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 & 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. 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 websites 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.

9.4.3 Logging (Client Side)

In this category, various techniques are considered. Before analyzing 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, because 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 can infer patterns of user behavior that indicate usability problems or other design deficiencies. This possibility has obvious attractions for Web designers, but in HCI usability research some issues have been raised regarding the possibility of identifying usability problems without access to the use context—to the user's 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. One method is to install cookies 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. No data is stored at the server side, but every time the same browser asks for the page or the same website, HTTP sends the cookie to the Web server, which uses it to identify the user and display personalized 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 4K) and therefore, strictly speaking, no Web-mining process can be performed based on such limited information. Second, because the cookies are saved as plain text, they can be easily retrieved at the client computer. 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 side. 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 evaluated pages because it automatically includes 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's 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 analyze the information contained in Web browser logs and compare it with task models specifying the designer model of the possible users' behaviors when interacting with the application to identify whether and where users' interactions deviate from those envisioned by the system design and represented in the model. Within this client-side ~~technique, we also cite 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 behaviors. The authors propose a user-remote tracking framework based on Web services and XML to track every HTTP request from client computers to understand surfers' characteristics. The user-remote tracker is a piece of software installed in the users' browser to keep track of every keyboard input and mouse click from the users. No matter what the users input, all HTTP requests and responses are tracked by the software program, including interactions with Java Applet programs. This program will automatically send the activity log file, together with the user identity, to a central machine for Web-mining (instead of sending such information directly to the Web server). It is that central machine that analyzes clickstreams and generates navigation rules for these users through 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 websites (i.e., websites ~~started~~ with

https://). Here, because the user-remote tracker uses low-level programs to track every user input signal, the activities can be tracked even in this case.

9.4.4 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 the usability of system interfaces, thereby providing an objective source of data that can inform the design of improved interfaces. Evaluators using eye-tracking, however, 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, because 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., colorful 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. Eye-tracking technology, however, has evolved in recent years and there are now more systems that can be used for remote evaluation (see the Tobii system, http://www.tobii.com/) because they can be transported in suitcases and do not require that users wear intrusive equipment. It is only necessary to ~~hold~~ an initial standard training exercise. Nevertheless, one of the most relevant problems with the 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 is actually being processed by the person.

9.4.5 Webcam/Audio Recorders or Microphones

The use of webcams and audio recorders allows for acquiring more contextual information about the data collected. Indeed, as it has been previously mentioned, through logging keystrokes and webpages on a given site, we could infer patterns of user behavior that indicate usability problems or other design deficiencies. In HCI usability research, however, it has been argued that it is not possible to identify usability problems without access to the context of use, to the user's 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 when an error is found, because the evaluator can analyze the video clip 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 the subject's considerations about the system. Furthermore, the use of more than one camera is valuable for capturing some environmental conditions occurring in the testing environment. Work by Lister (2003) has been oriented to using audio and video capture for qualitative analysis performed by evaluators on the results 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 user behavior whenever some measurements (e.g., time needed for completing a task) captured by another software component provide unexpected values.

9.4.6 Sensors

~~By the term *sensors*, we mean more sophisticated~~ research solutions for data acquisition and analysis of some physiological data. Recently, a number of sensors are being used more and more for the evaluation of user interfaces, trying to take into account the emotional dimension of computer-human interaction (e.g., affective user interfaces). Among such measures, we cite physiological signals like ECG, respiration, galvanic skin response, heart rate, and 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, while BVP decreases), and are also easy to be measured with specialized 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 highly specialized equipment necessary, which cannot be assumed available in users' daily environments (although it is slowly appearing and used more and more 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 analyze the user's emotional state.

To summarize, almost all the results obtained with each technique indicated in this section requires additional knowledge about the user from the evaluator to be actually useful for the purposes of the evaluation. Therefore, the big issue is that such data is not informative *per se* about possible usability problems, but requires further comparison with supplementary information. One of the few exceptions can be identified in, for instance, recording users positively (or negatively) commenting on the session while interacting with the application in a remote think-aloud session (which should theoretically provide the evaluator with immediate feedback about the user's satisfaction). In almost all the other cases, a further contextualization

(and integration) of the data collected is needed to correctly evaluate the session state (think about, for example, the uselessness of logging mouse and keyboard actions without contextualizing such actions within the 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, to facilitate the evaluator's work.

9.5 The Type of Application Considered

In this section, we analyze another dimension of the proposed framework—the type of application, considered in terms of the underlying software environment. As with already analyzed dimensions, in this case the consideration of this dimension is also not completely independent from the other ones. Indeed, the type of application considered may prevent (or strongly promote) applying specific techniques mentioned in the previous section, as well as the use of particular interaction platforms. For instance, while in the 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 has only occurred in recent years. It is almost always connected with stationary platforms because only recently prototypical tools that support the evaluation of .NET applications for mobile devices have appeared, and are 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 (e.g., Paternò & Ballardin 2000). Then, with the advent of the Web and the related ease of performing a remote evaluation when the Web is considered (due to the related simplicity in involving a high number of testers with little effort), the majority of methods have considered websites as their primary evaluation targets. Java-based applications indeed have been taken into account, sometimes as a sort of side-effect of the desire to improve 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 (i.e., PDA devices), for which they often provide more robust and supported solutions with respect to Java. An example of logging tools for Microsoft environments is the VibeLog logging tool (http://research.microsoft.com/vibe/), which has been developed at Microsoft Research to evaluate the ways that work practice might change as users move between various-sized displays during their work day. The logging tool is married with ethnographic research data, which should provide good indications of what parts of the Windows and Office designs 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 development, with an eye toward designing more elegant UIs.

9.6 The Type of Evaluation Results

Before analyzing the last dimension of the proposed framework in depth—the type of results an evaluation can deliver (i.e., qualitative vs. quantitative data)—we judge it useful to mention 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 in this respect. In *formative* evaluations, the objective is to collect information about design flaws and inform redesign. Therefore, it is particularly important that participants feel free to criticize a system and 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 take it 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, recording problems encountered and then sending them along with ratings of the accessed websites. Both remote and local evaluations provided considerable quantities of qualitative data, but the local evaluations provided far richer data because the researchers were 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 analyze what had caused them.

On one hand, achieving the rich interaction between participants, researchers and developers, as requested by *formative* evaluation, is very difficult in remote evaluation situations. With high quality video conferencing, broadband connections, and remote recording systems, however, it might be possible to conduct remote evaluations that capture a 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 behavior.

We agree with this position to some extent because, in our opinion, remote evaluation can provide different types of results, which can, in turn, be used for different purposes, both summative and formative. In this section, we are going to analyze the type of results an evaluation can deliver. In particular, a discussion about the type of information that can be useful to obtain in order to analyze 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 9.4 and the evaluation results analyzed in this section (for instance, sensing technologies deliver quantitative data about the user's emotional and physical state), while in other case such correspondence is not so straightforward.

9.6.1 Task-Related Information

Many applications are task-oriented and therefore 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, ...). One issue is how to know what the desired tasks are. One possible solution is to ask users to explicitly indicate the tasks at the beginning of the session. The issues associated with user errors are related to the issues associated with task performance, because user errors 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 analyze the performance of short basic tasks, and in other cases it is important to focus on the performance of high-level complex activities.

During an analysis of task performance, it can be useful to analyze when it deviates from the ideal expected behavior, and to what extent. The evaluator then has to understand the reasons for such mismatch and needs to go back and analyze what happened for each action in the user session and what factor triggered the deviation.

9.6.2 Qualitative Information

Under this heading, 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, because 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 collected data that may be too ambiguous to be interpreted—generally quantitative data. An example in which this strategy might disambiguate other data, is the case of a user spending a 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 just had problems in finding the concerned

information—in this case, the webcam can help in correctly interpret the feeling of the user (engaging or not the visit).

9.6.3 Presentation-Related Data

In this section, we analyze the results that the evaluation should deliver about the usability of the user-interface presentation (e.g., for GUIs—layout, choice of widgets, colors, 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 & others 2002) provides representations consisting of nodes representing visited webpages, and arrows representing the traffic between the pages. Entry pages are green and exit pages are cyan. Thicker arrows represent heavier traffic. Arrow color is used to indicate time spent on a page before transitioning, where the closer the arrow is to red, the longer the user 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.

9.6.4 Quantitative Cognitive-Physiological Information

Quantitative psycho-physiological measurements can provide useful information about more general, qualitative information on a human's feeling in a specific situation. For instance, with a growing population of elderly persons today, 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 to enable the possibility of assessing elderly mood in a non-intrusive manner. 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 have to be analyzed to reliably convey short-term mood changes (that might be relevant for the relatives and form a basis for an enhanced feeling of connectedness), as well as long-term trends. When the shape of people is mostly visible, computer vision tools can be used to classify their posture and gait, and posture changes over 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, and classify the facial orientation (frontal, profile) to provide the process of facial expression analysis with suitable data. In addition, faces are processed for expression/recognition/authentication. If a person is not visible or the user does not like a camera to be used (e.g., in the bathroom), 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 being analyzed. The duration of the scanpath is a productivity measure and can be compared with a theoretical optimal duration. Even the ratio between saccades and fixations can be a useful index for comparing the percentage of time spent in looking for information (saccades) and that during which information is acquired (fixations).

9.7 An Example Tool for Remote Evaluation: MultiModalWebRemUsine

In this section, we discuss an example of a tool for remote evaluation according to the framework presented in the chapter. The basic idea of this tool is to analyze 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 users to 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 to achieve the user's goal. If the performed task sequence diverts from those enabled by the task model, there is clearly a mismatch that needs to be analyzed by the evaluators. 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 & Paternò 1998), mainly addressed the issue of using task models for analyzing user logs without considering its use as remote evaluation. The next version, RemUSINE (Paternò & 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. It was, for example, compared with a video-based evaluation. It turned out that for evaluating a small number of sessions, the video-based evaluation was more efficient because RemUSINE required some time to enable the automatic evaluation given that the evaluator first has 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, in some cases, video analysis 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.

Given the explosion of the Web, which has become the most common user interface, we thought it useful to develop a new version (WebRemUSINE) aimed at evaluating this type of application (Paganelli & Paternò 2003). We also had to decide how to log user interactions. 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 analyzed—visited pages, never visited pages, extent of scrolling and resizing, page patterns, and download and visit time. Some information is provided, along with summary data regarding the content of the page. Thus, the visit time is provided and also indicates the number of forms, links, and words in the page so the evaluator can compare the visit time with the quantity of information available in the page. The latest version of the tool (MultimodalWebRemUSINE) aims to exploit the possibilities opened up by recent technologies to gather a richer set of information regarding user behavior. Thus, the traditional graphical logs can be analyzed together with the logs from webcams and portable eye-trackers, which do not require the use of intrusive equipment.

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

In general, MultimodalWebRemUsine is based on a comparison of planned user behavior and actual user behavior. Information about the planned logical behavior of the user is contained in a (previously developed) task model, while data about actual user behavior is provided by the other modules supposedly available within the client environment (the logging tool, the webcam 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 analyzes 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 if 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, the corresponding basic task is first identified and then 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, to allow the evaluator to check ~~to see~~ if the target task has been completed. Otherwise, some error will be notified in the

report analyzing 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), they usually 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 realize that both fields have to be filled in before connecting to the mailbox, then some precondition errors can occur (for example, the user sends the request without first proving 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 to how to represent user sessions and related data in such a way that eases their analysis. Figure 9.1 shows the type of representations designed. It is possible to show data related to several sessions in different ways at the same time. In Figure 9.1, we analyze the parts of the sessions about users who want to become member of an association. As you can see from the selected radio buttons, the deviation graph is shown for the first two users, while the state graph is visualized for the other ones. 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 color of the line underlying the white circles is used to indicate whether the user is correctly or incorrectly performing the task.

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 understanding what the most complicated tasks are or what tasks require a longer time to be performed. In Figure 9.2, a screenshot of the tool is presented. As you can see, whenever an inexplicably lengthy time period for carrying out a certain task is registered by the tool, the evaluator can activate the related video recorded through a webcam to gather further information.

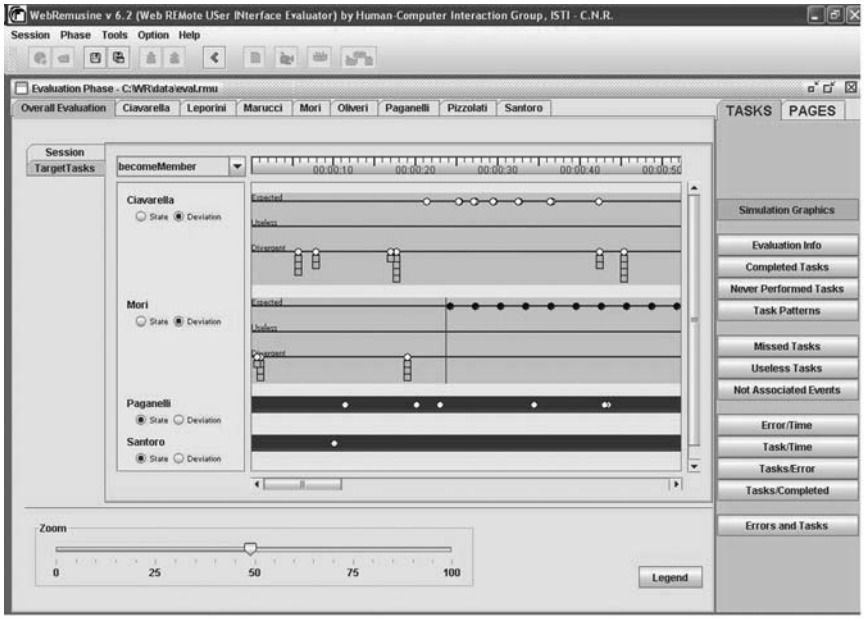

Fig. 9.1 Representation of user sessions in MMWebRemUsine

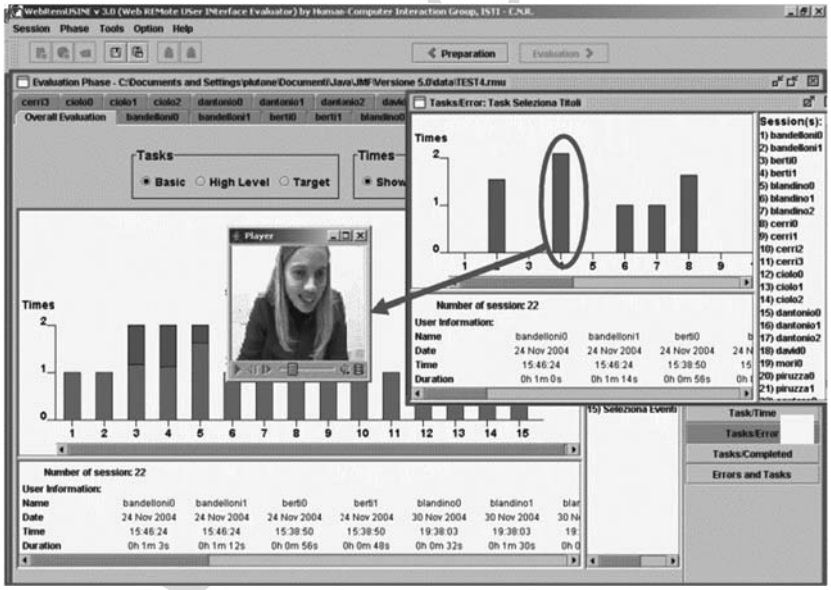

Fig. 9.2 Highlighting a video within the MultiModalWebRemusine environment

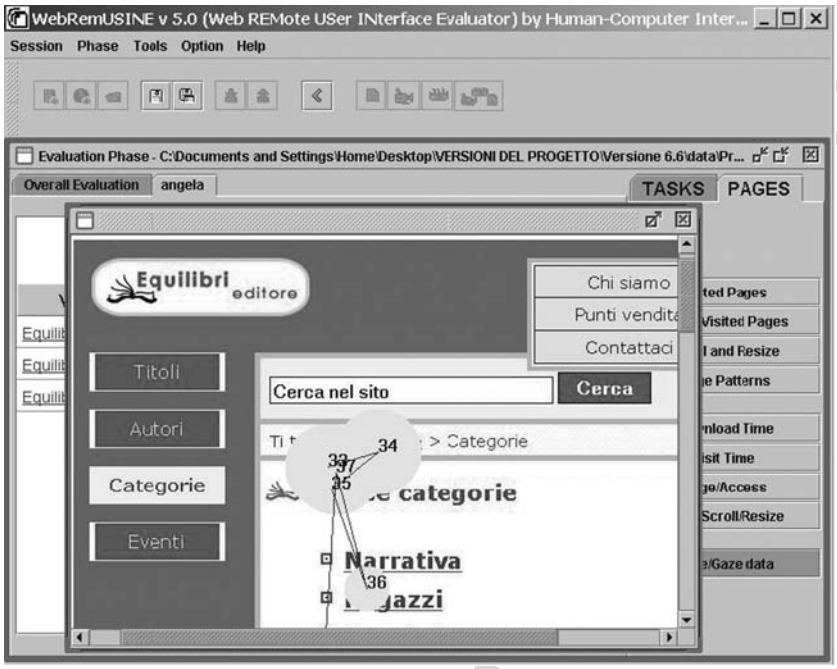

Fig. 9.3 An example of visualization of fixations (the yellow areas) and scanpaths (the paths in-between) registered by the eye-tracker

The approach supported by MultiModalWebremusine is also able to provide usability data associated with every single presentation. Moreover, it is worth noting that ~~because~~ this approach can correlate task-based measures with presentation-related data, it can also analyze the usability of the website 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.

Eye-tracking is a technique that has been used within MultiModalWebRemUsine. In Figure 9.3, a screenshot has been taken that shows how the registrations of the eye-tracker are visualized 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.

9.8 Discussion and Interrelationships between the Framework Dimensions

We can discuss possible interrelationships among the different dimensions that we have identified while analyzing 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, with the user context, which is very important for mobile applications and less important for desktop systems.

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 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) that 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 technique for collecting information about the user, in automatic collection of data the range of techniques can greatly vary, as 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 the case of Web-based applications, the use of server-side and client-side logging techniques is a well-known and established approach, while 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.

9.9 Conclusions and Future Challenges

In this chapter, 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. These methods have been receiving more and more interest due to the improvements in techniques that are able to capture information regarding user behavior and the validity of the data that are collected *in the field*. Therefore, the chapter tries to shed some light on the different

methods through a common framework in which it is possible to compare and contrast 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, this chapter is useful in 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 gathered data regarding user behavior and state—including emotional state—so a more complete analysis of what happens during user sessions can be done and potential usability issues can be better identifed. Another novel emerging application area is that of mobile applications, in which is important ~~in understanding~~ how task performance varies depending on the changes in the context of use. To make the discussion more concrete, we have also reported our experience with our tool in the area of remote evaluation and analyzed it according to the dimensions of the logical framework proposed.

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)*, 34–41.

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., & 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 ASIS 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: *Proceedings of Ninth Annual Conference of the Internet Society (INET'99)*, San Jose, CA, June 1999.

Denis, C., & Karsenty, L. (2003). Inter-usability of multi-device systems: A conceptual framework. In A. Seffah & H. Javahery (Eds.), *Multiple User Interfaces: Engineering and Application Framework* (pp. 375–385). New Jersey: John Wiley and Sons.

Etgen, M., & Cantor, J. (1999). What does getting WET (Web Event-Logging Tool) mean for Web 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. (1996). Remote evaluation: The network as an extension of the usability laboratory. In: *Proceedings of CHI 1996* (pp. 228–235).

Ho, S.Y. (2005). An exploratory study of using a user remote tracker to examine Web users' personality traits. In *Proceedings of the 7th International Conference on Electronic commerce, ICEC'05* (pp. 659–665), August 15–17, 2005, Xi'an, China. ACM Press.

Hong, J.I., & Landay, J.A. (2001). WebQuilt: a framework for capturing and visualizing the Web experience. In *Proceedings of WWW 2001 Conference* (pp. 717–724).

IQ Services (2006). *IQ Services: Interactive Quality Services, Inc*. Access at: 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), 470–516.

Lecerof, A., & Paternò, F. (1998), Automatic support for usability evaluation. *IEEE Transactions on Software Engineering, 24 (10)*, 863–888.

Lister M. (2003). Streaming format software for usability testing. In *Proceedings ACM CHI 2003, Extended Abstracts* (pp. 632–633).

Paganelli, L., & Paternò F. (2003). Tools for remote usability evaluation of Web applications through browser logs and task models. *Behavior Research Methods, Instruments, and Computers, 35 (3)*, 369–378.

Paternò, F., & Ballardin, G. (2000). RemUSINE: a bridge between empirical and model-based evaluation when evaluators and users are distant. *Interacting with Computers, 13(2)*, 229–251.

Paternò, F., Piruzza, A., & Santoro, C. (2006). Remote Web usability evaluation exploiting multimodal information on user behavior. In *Proceedings CADUI 2006*, Bucharest. Springer-Verlag.

Petrie, H., Hamilton, F., King, N., & Pavan, P. (2006). Remote usability evaluations with disabled people. In *Proceedings of CHI 2006* (pp. 1131–1141), Montréal, Québec, Canada, April 22–27, 2006.

Scholtz, J., Laskowski, S., & Downey L. (1998) Developing usability tools and techniques for designing and testing Websites. In *Proceedings HFWeb'98* (Basking Ridge, NJ, June 1998). Access at: http://www.research.att.com/conf/hfWeb/ proceedings/scholtz/index.html

Stoica, A., Fiotakis, G., Simarro-Cabrera, J., Frutos, H.M., Avouris, N., & Dimitriadis, Y. (2005). Usability evaluation of handheld devices: A case study for a museum application. In *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. (2006). Replayer: Collaborative evaluation of mobile applications. Presented in *CHI'06 Workshop on Information Visualization and Interaction Techniques for Collaboration across Multiple Displays*, Montreal, Canada.

Tullis, T, Fleischman, S., McNulty, M, Cianchette, C., & Bergel, M. (2002). An empirical comparison of lab and remote usability testing of websites. In: Proceedings of *Usability Professionals Conference*, Pennsylvania, 2002.

Waterson, S., Landay, J.A., & Matthews, T. (2002). In the lab and out in the wild: Remote Web usability testing for mobile devices. In: *Proceedings of CHI 2002* (pp. 796–797), *April 20–25*, Minneapolis, USA.

West, R., & Lehman, K.R. (2006). Automated summative usability studies: An empirical evaluation. In *Proceedings of CHI 2006* (pp. 631–639), *April 22–27, 2006*, Montréal, Québec, Canada, ACM Press.

Chapter-9

Query No.	Page No.	Line No.	Query
AQ1	203	35	In who's paper–Stoica?