

ERGOMANAGER:

a UIMS for monitoring and revising user interfaces for Web sites

Walter de Abreu Cybis

Universidade Federal de Santa Catarina, Brazil & Ecole Polytechnique de Montréal, Canada

Dominique L. Scapin

Institut National de Recherche en Informatique et Automatique, France

Marcelo Morandini

Departamento de Informática, Universidade Estadual de Maringá, Brazil

Keywords: Ergonomics, Usability, Evaluation, Monitoring, UIMS, B2B, ERP, Intranet

Abstract: This paper describes the results of studies dedicated to the specification of ErgoManager, a UIMS (User Interface Management System) specifically intended to support the user interface revision phase over changeable Web sites running B2B, ERP or Intranets transactions. This UIMS contains two basic components: ErgoMonitor and ErgoCoIn. ErgoMonitor applies task-oriented analysis and usability oriented processing on interaction traces stored in log files as a way to identify “average” usability levels that have been occurring when users were accomplishing transactional tasks with a web site. ErgoCoIn is a checklist based CSEE (Computer Supported Ergonomic Evaluation) tool that features automatic services to inquire context of use aspects and to recognize web page components as a way to conduct inspections of only the context pertinent aspects of a Web page. By integrating these tools, ErgoManager aims to support quality assurance strategies over the revision phase of web sites lifecycle by confronting, in an iterative way, usability quantitative metrics and qualitative aspects of user interfaces.

1 INTRODUCTION

In this paper we present the specification of **ErgoManager**, a UIMS (User Interface Management System) specifically intended to support the revision phase of a transactional Web site lifecycle. This environment is being developed through an INRIA-CNPq cooperation agreement (Cybis et al., 2002) and features, in its functional architecture, two basic components: ErgoMonitor and ErgoCoIn.

ErgoMonitor is a tool for identifying “average” usability levels occurred when users have been accomplishing transactional tasks over a Web site (Morandini and Cybis, 2003). It is based on both a task oriented log files analysis technique and a usability metrics oriented log data treatments that allow automatically quantifying usability measures occurred when accomplishments of transactional

tasks. These results could be considered particularly reliable since they are computed a posteriori, i.e. after interactions have been accomplished in the real conditions. Specifically, ErgoMonitor is aimed to signal to a deterioration of usability metrics on transactional tasks as a consequence of ergonomic problems introduced on web pages.

ErgoCoIn is a checklist-based tool aimed at supporting objective ergonomic inspections of e-commerce Web site and pages (Cybis et al., 2000, 2002). The ErgoCoIn tool features automatic inquiring services to identify context of use aspects (users and environment attributes) and to recognize web page components. Consequently, it is able to propose to inspectors only questions applied to the specific task context of use and to the associated Web page components.

By integrating these tools, ErgoManager can present to webmasters a report signaling the deterioration of usability measures over a

transactional task being monitored, along with an objective and systemic usability checklist, aimed to help these professionals to identify the design problems affecting site usability. Once the problems fixed, ErgoManager will be able to signal to webmasters the usability metrics moving back to usual levels states.

2 THE ERGOMONITOR TOOL

The ErgoMonitor project was inspired by a responsibility that developers and managers of e-commerce Web site face up regularly: continually assuring and improving the site usability despite the constant updating of actions and information. The general assistance we identified as pertinent to developers is supplying them with information about usability levels the web site has been offering to its final users. In fact, these professionals' mission would become simpler and more objective if they could continuously know the impacts their design decisions have on Web site usability levels. Specifically, this information should results from reliable, systemic, rapid and non expensive procedures.

However, most popular usability evaluation techniques usually do not match these requirements. Diagnostic evaluation techniques issues are qualitative and most often, subjective, while based on experts judgments. Usability tests produces quantitative and objective results but such technique is quite difficult to set up, time consuming to analyze, and quite expensive.

Log files analysis approaches appear to be good candidates for matching several of the requirements listed above. A log file is a file in which a web server records data related to any request performed by any client. Such data contains (W3C, 2005):

- Client's computer identification number (IP);
- Request date, time, type and address (url);
- Request result code and requested document addresses (url) and size;
- User's technical environment: Browser and operational system of the client computer.

At this time, most popular log analysis tools output can be categorized into the following categories:

- Users perspective: users' technical environment, the address from where they come, urls and documents accessed, frequency and duration of access to different pages, users' profiles (Audience insite Measures (ComScore Networks, 2005);

- Usage/Interaction perspective: most requested pages and documents, date and time of biggest volume of access, path users were crossing over the site (MitriDAT, 2005, Keynote, 2005)
- Maintenance perspective; type and number of errors, components with errors, etc (WebTrends, 2005). Some other tools, like ROI Tracking Pro (MitriDAT, 2005b), support web site return of investment analysis by modeling and processing cost-benefit data in historical series.

These issues are quantitative, low cost and obtained in a fast and systematic way. They refer to users and interactions, but even so they are quite limited compared to usability evaluation proposals. In fact, a "Usage" perspective is too neutral for the goals of usability analysis while we don't know the users' objectives when interacting with a web site.

We argue that it is possible to go further in usability studies by introducing a different perspective for log data analysis and processing: the one formed by the task oriented analysis technique and the usability oriented data log processing.

The **task oriented approach** to analyze data in log files is based on the "inferable task" concept (also call "assumed theoretical task"). It could be seen as a particular type of interaction where we could infer the users' objectives only by reading log data. It could be done by observing the path users have been crossing and the goals they have been accomplishing with the web site. For example, when we verify in the log file that a user has got access to a registration form and some minutes later the system has presented to him/her a confirmation message it is reasonable to infer that this user was willing to register his/herself. The same is true for other type of transactions with and start and final point well distinguishable like a book reservation or a product acquisition. Once we know his/her objectives in tasks we could identify the moment the user had begun and had accomplished it and the different path he/she had crossed during this time. Indeed, the transactional tasks have several associated behaviors or alternative paths which are logically authorized by the user interface, like the direct success, the success with deviation, the success with error, the success with help, the quitting, the canceling (quitting after an error), the canceling with help, and so on. Computing the incidence of the alternative success paths and their time we could determine measures of the user efficiency in accomplishing a task. The incidence of failure behaviors could inform about user effectiveness, but in these cases, we need assume measures will not be so precise. In fact, there is no way to distinguish between users who were really wanting to achieve the transaction and were unable

to do that from those who were visiting the sites only to know its contents and had quit it before to command any execution.

The **usability oriented data log processing** is based on building the following architecture of log data abstractions: *user, user's episodes, user's movements, user's behaviors on task and task's time*. The first thing to do is to identify or individualize the users. In practice, it could be a very difficult task while based only on IP numbers once a same client machine's IP could be shared by several users getting access through the same proxy server. The most common solution consists in defining a user as a data abstraction composed by <IP number, OS name and Browser name>. This increases the number of differentiating index, but it is not error prone especially for log file associated to a huge transactional traffic. This step could be extremely simplified however for web sites where user access is controlled by password. In these cases, the user's name will be registered in log data and the user identification become direct. The next step is to classify all user's movements in each user's episode. User's episodes are commonly defined as sets of interactions far one another from more than 30 minutes. In fact, most task resuming time fall into this interval (Cooley, 1997). User's movements are in fact, system transitions caused by users' actions but could be meant as movements users make with the system. They correspond to a log file entry in which is registered an occurrence of a page display or a document download resulted from a request done from another page. Movements are classified in relation to a set of movements that composes the anticipated behaviors on a task. Typical movements on simple tasks are "task entry", "task exit", "task re-entry", "task accomplishment". A user's behavior is an ordered set of user's movements that ends with the task accomplishment or the episode's end. Depending on its elements, behaviors could be classified as user success on task (entry-accomplishment), the user success with deviation (entry-exit-re-entry-accomplishment), and so on. The incidence and the time of anticipated behaviors are than computed to indicate with which level of resources (time and attempts) the task was accomplished. It is so possible to determine efficiency usability factors and metrics in a very close fashion to those proposed by the ISO 9241:11 (ISO 9241:11, 1997) standards.

This approach is especially useful for analyzing and processing log data from B2B (Business to Business), ERP (Electronic Resource Planning) or web sites (inter or intranets) where the incidence of transactional inferable tasks is large and where user access is controlled by password. The results obtained in these cases are expected to be

precise enough. This is not the case in for informational sites or opened B2C (Business to Consumers) electronic commerce, where it is impossible, based only in the log data, to infer users' objectives. Even so, the task oriented log data analysis could be useful here, if its issues are taken in a relative basis, i.e., compared with the historical values obtained in past for the same context conditions. Here the focus must be turned to the usability level disturbance rather than to the absolute usability level itself. So, a web manager could rapidly identify a disturbance in site usability curve caused by a bad interface users had begun to get access two or three days ago.

The ErgoMonitor applies both the task oriented analysis and the usability oriented processing on log files to determine usability metrics for a given task and a given user interface for a period of time. It is worth to mention that these measures will be average ones, since the system will consider all tasks trails during a period of time, which will refers to different users, pertaining to different profiles and having different physical and software environment. ErgoMonitor processing starts with an analyst examining the web site and defining an inferable tasks model for each task been monitored. This model is composed by a set of user's behaviors, each one consisting in a set of user's movements. In next paragraphs we will detail the specification of the initial ErgoMonitor prototype's modules.

Monitoring properties module: it is composed by forms in which a UI analyst will be filling parameters of current monitoring. Essential data are:

- Site name and description;
- Log file path, site and log file access data;
- List of Inferable Task to monitor.
 - Task Identification
 - Task Pages (or task markers)
 - Initial page (url);
 - Intermediate pages (sequence of urls);
 - Final page (url);
 - Help pages (set of urls);
 - Error pages (set of urls).
- List of Associated User Interfaces
 - Version identification;
 - Date it was made available to users;
 - Description (design pattern, navigation map, screen shots, comments)

By this structure, a web site is viewed as a collection of tasks, each of them being supported by a collection of user interfaces that replace one another in time. So, ErgoMonitor will be monitoring usability in less changeable task structures which are supported by more changeable user interfaces. Ideally the tasks descriptions are filled in only one time and the user interface description each time it is revised.

Functional Core: this module will build the data abstractions presented earlier in this paper: users, users' episodes, user's movements, user's behaviors on task, and task's time. Even if it is easy to figure out several other user movements and behaviors, the first version of ErgoMonitor will monitor specifically the following:

- Movements : url → url (in a user's behavior context)
 - task entry = url not associated with the task → Initial page (no user's behavior opened);
 - task evolution = Initial page → Intermediate pages (in a user's behavior not yet concluded);
 - task exit = Initial page | Intermediate pages → url not associated with the task (in a user's behavior not yet concluded);
 - task re-entry = url not associated with the task → Initial page (in a user's behavior not yet concluded);
 - error managing = Initial page | Intermediate pages → error page (in a user's behavior not yet concluded);
 - help searching = Initial page | Intermediate pages → help page (in a user's behavior not yet concluded);
 - task accomplishment = Initial page | Intermediate pages → accomplishment page (in a user's behavior not yet concluded);
- Behaviors (seq. of movements)
 - Direct Success (DS) = task entry + task evolution (optional) + task accomplishment;
 - Success with Deviation (SD) = task entry + task evolution (optional) + task exit + task re-entry + task evolution (optional) + task accomplishment;
 - Success with Error (SE) = task entry + task evolution (optional) + error managing + task evolution (optional) + task accomplishment;
 - Success with Help (SH) = task entry + task evolution (optional) + help searching + task evolution (optional) + task accomplishment;
 - Visit (V) = task entry + task exit;
 - Quit (Q) = task entry + task evolution + task exit
 - Cancel (C) = task entry + task evolution (optional) + error managing + task exit

Based on these behaviors' incidence and time the functional core will compute the usability *factors*, *rates* and *metrics* listed below:

- Usability factors
 - Amount of Visits (#V)
 - Amount of Success (#S) = #DS + #SD + #SE + #SH;
 - Amount of Failures (#F) = #Q + #C

- Amount of Task Trials (#TT) = #S + #F + #V;
- Usability rates
 - Rate of Visits (%V) = #V / #TT
 - Rate of Success = (%S) = #S / #TT
 - Rate of Direct Success (%DS) = #DS / #TT;
 - Rate of Success with Deviation (%SD) = #SD / #TT ;
 - Rate of Success with Help (%SH) = #SH / #TT ;
 - Rate of Success with Error (%SE) = #SE / #TT ;
 - Rate of Failures = (%F) = #F / #TT
 - Rate of Quits (%Q) = #Q / #TT
 - Rate of Cancels (%C) = #C / #TT
- Usability metrics
 - Mean Time to Task = $\Sigma \text{Time} (\#S) / \#S$;
 - Mean direct time = $\Sigma \text{Time} (\#DS) / \#DS$;
 - Mean time with deviation = $\Sigma \text{Time} (\#SD) / \#SD$;
 - Mean time with error = $\Sigma \text{Time} (\#SE) / \#SE$;
 - Mean time with help = $\Sigma \text{Time} (\#SH) / \#SH$;

Usability Measures Database: This database will be maintained by the Functional Core that will be storing on it values for usability factors, rates and metrics. These entries will be indexed by task, user interface version and period of time analyzes producing them were related.

Monitoring Reports: This module will be requesting usability metrics stored on the database according to parameters selected by UI analyst. By default, the report will present a set of line graphs concerning different usability factors, rates and metrics corresponding to one task, the different user interface versions associated with it and the time they were in service. A set of warnings will be also directed to the web developer when system detects decreasing values of usability level.

3 THE ERGOCOIN TOOL

The design of the usability evaluation technique underlying ErgoCoIn CSEE (Computer Supported Ergonomic Evaluation) tool has been motivated by two considerations.

The first one is that web sites development became accessible (through easily available design tools) to a large spectrum of “designers”, not necessarily highly skilled in computer science or in ergonomics.

A second consideration is that web sites are often designed along a fast and low cost design process supported by non expensive tools which lead designers to carry out numerous and sometimes obvious ergonomic flaws.

Accordingly to these constraints we had defined two basic requirements for a usability evaluation approach. The first one concerns the need to define a method that should accommodate this type of designers, i.e., a method that does not need extensive ergonomics knowledge, but that provides minimal ergonomics knowledge directly into the evaluation context. Of course, the associated limit of this requirement is that the method will not point at all major ergonomics problems (but it is a first start before dealing with the more complex ones, more difficult to diagnose). Furthermore, a method should correspond to a short design process. Two orientations are considered: one is to use a method known as being fast and cheap - i.e. usability inspection; the other one is to incorporate as efficiently and rapidly as possible some of the usual knowledge needed for performing ergonomics evaluations, i.e. information about the users, the tasks, and the site itself through users and designers participation. Of course, the associated limit of this requirement is that the method will only consider minimal knowledge about users and tasks (minimal if compared with extensive task analysis, task modeling, etc.).

ErgoCoIn combines inquiring techniques (interviews and questionnaires) with evaluation techniques in an approach able to allow rapid, context focused ergonomic inspections. The inspection component resulting from examining a large collection of ergonomic recommendations (Leullier et al., 1998) later completed with other data collected from different studies (Scapin et al. 2000), to elaborate checklists for the ergonomic characteristics applicable on e-commerce web sites. These recommendations were formulated as questions and associated with both an ergonomic criterion that allow defining a system of relative importance between questions, and a specific interface attributes that allow insuring fair objectiveness for the evaluation strategy. Interviews/questionnaires and guidance for collecting data from users and designers were defined from analyzing the information demands in each question we elaborated. Finally we specified the ErgoCoIn tool, a software system aimed at minimizing the human effort needed for the context data gathering and the web site inspection. This tool specification follows the two main phases of ErgoCoIn's approach: the web site's contextual analysis and its evaluative inspection.

The goal of the **Contextual Analysis** phase is to collect all information related to the web site operational contexts that are useful for the usability evaluation process. This phase consists of a site description or recognition process and interviews with the users and designers. The first prototype of the ErgoCoIn tool will be supporting and automating these activities. An html component recognizing tool identifies the existence of specific user interface components on the web site pages associated with the main tasks accomplishment. It organizes them according to two categories of descriptions: the global web site descriptions and the individual web pages descriptions. As a consequence of integrating ErgoCoIn into the ErgoManager environment, the html component recognizer will be only considering components over the path concerned with the tasks being monitored by ErgoMonitor. Another ErgoCoIn tool will be proceeding with on-line interviews with designers and users, as a way to obtain information about intended and real context of usage features. Here also, the tool will limit the scope of the interviews to the tasks scenarios being monitored by ErgoMonitor. The information collected in this description phase is registered in a database related to the web site context of use.

The second phase of the method is formed exclusively by **evaluative inspections**. ErgoCoIn tool starts the process performing an automatic analytical evaluation based on the comparison between information furnished by users and designers, concerning the intended and the real context of use features. The system will point out to existence of designer's misconceptions about users' features, and indicate the web site aspects to verify or reformulate in consequence.

Next, the system will be assembling checklists concerning the overall site and the Web Pages features related to task scenarios being monitoring by ErgoMonitor. These checklists can be considered as "objective" ones, once they propose only the site components applicable questions arranged according to their levels of importance. Applicability decisions result from processing the site description stored in the context of use database. Priority decisions results from ranking the Ergonomic Criteria (Scapin & Bastien, 1997) according to context of use features. A default Ergonomic Criteria ranking is suggested as a result of analysis of the average e-commerce context of use, but it can be modified by the evaluators, according to the characteristics of the current web site context of use. In fact, the original importance structure was proposed with a general B2C usage context in mind, in which non professionals users operate sites of virtual stores from their home environments aiming to buy simple products in a relatively low frequent basis. In

such a situation the Guidance criterion should be considered more important than the Work Load criterion. The specific ErgoManager's application domain including B2B or ERP user's profiles, task complexity and equipment configuration had forced an inversion in the relative importance between Work Load and the Guidance criterion. Any way, the ErgoCoin tool will authorize evaluators changing the importance structure at the ergonomic criteria level to accommodate different usage contexts.

The evaluative inspections are performed by an evaluator applying the set of checklists defined in the previous phase. As mentioned before, this process constitutes an evaluative inspection once the evaluator is asked to judge the quality of very precise web site features. The level of judgment proposed by questions was defined in accordance with the level of ergonomic knowledge expected from evaluators (fairly basic usability expertise). Indeed, the questions phrases and associated support information, like justification and examples, were formulated in order to be easily understandable.

The ErgoCoin tool will be supporting the checklists application step by a special work environment in which there will be questions and information about both usability and the web site context of use. The system will be finally supporting evaluation documentation using predefined report styles.

One of the limitations of this technique and tool, which is quite compatible with an integration with ErgoMonitor, is that it can only be applied for web sites that are already running, that have a real user (or group of users) and an available designer. Both of them will be responsible for presenting vital information concerning the context of real and intend web site operation.

4 CONCLUSION

ErgoManager is aimed to support the confronting of two different and complementary usability evaluation issues: quantitative usability metrics and qualitative user interface aspects. Once in use, this environment should allow web developers to implement a continuous user interface improvement strategy based on verifying the impact the user interface design aspects have on usability metrics. This also means bridging more closely predictive ergonomics (i.e., inspection even before usage) and real usage features (i.e., from actual usage statistics)..

REFERENCES

- ComScore Networks, 2005. Audience insite Measures , (n.d.). Retrieved February 1st, 2005, from <http://www.comscore.com/matrix/aim.asp>
- Cooley, R., Srivastava, J., Mobasher, B., Web Mining: Information and Pattern Discovery on the World Wide Web, Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997
- Cybis, W.A., Scapin, D. L. ; Andres, D.P., 2000. Especificação de Método de Avaliação Ergonômica de Usabilidade para Sites/Web de Comércio Eletrônico. In: Proceedings of Brazilian Workshop on Human Factors in Computer Systems. Porto Alegre: Ed. Sociedade Brasileira de Computação, v. I, p. 54-63.
- ISO/DIS 9241, 1997. Ergonomic Requirements to Office Work with VDTs, Part 11 - Guidance on Usability.
- Keynote, 2005. Research Manager Clickstream, http://www.keynote.com/solutions/solutions_cem_research_mgr_clickstream.html (accessed in 2005)
- Leulier, C., Bastien, J. M. C., Scapin, D. L. (1998). Compilation of ergonomic guidelines for the design and evaluation of Web sites. Commerce & Interaction (EP 22287), INRIA Report.
- MitriDAT, 2005a. Clicks Counter Pro, , (n.d.). Retrieved February 1st, 2005, from <http://mitridat.com/products-clicks-counter-pro.html>
- MitriDAT, 2005b. ROI Tracking Pro Pro, , (n.d.). Retrieved February 1st, 2005, from <http://mitridat.com/products-roi-tracking-pro.html>
- Morandini, M., Cybis, W.A., 2003. Definition and prototyping of ErgoMonitor. In: Proceedings of International Conference on Human-Computer Interaction, Creta. v.II, p. 1396-1399.
- Scapin, D. L., Bastien, J. M. C., 1997. Ergonomic Criteria for Evaluating the Ergonomic Quality of Interactive Systems, Behaviour and Information Technology, vol 16, no. 4/5,.
- Scapin, D. L., Leulier, C., Bastien, J. M. C., Vanderdonckt, J., Mariage, C., Farenc, C. Palanque, P., Bastide, R., 2000. Transferring Knowledge of User Interfaces Guidelines to the Web, Workshop ERGO-IHM'2000:Tools For Working With Guidelines, October 7-8, 2000, Biarritz, France.
- WebTrends,2005.WebTrends7, (n.d.). Retrieved February 1st, 2005, from <http://www.webtrends.com>.
- World Wide Web Consortium's (W3C). 2005. Logging Control In W3C httpd, (n.d.). Retrieved February 1st, 2005, from <http://www.w3.org/Daemon/User/Config/Logging.htm> l#LogFormat.