ErgoManager : a UIMS for monitoring and revising user interfaces for Web sites

Walter de Abreu Cybis

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

ErgoManager: a UIMS for monitoring and revising user interfaces for Web sites

Walter de Abreu Cybis

N° 5639

July 2005

Thème COG

Rapport de recherche

ISSN 0249-6399

ErgoManager:
un SGIU pour le suivi et l' amélioration d' interfaces utilisateurs de sites Web

Walter de Abreu Cybis

Thème cog C – **Interaction homme-machine**
Projet MErLIn

Rapport de recherche n°5639 – Juillet 2005 - 28 pages

Résumé: Ce rapport décrit la spécification de ErgoManager ainsi que les résultats d' une première étude de validation associé au développement de ce SGIU (Système de Gestion d' Interface Utilisateur). Il a été conçu pour aider les webmestres dans leur tâche d' assurer la qualité « en utilisation » des sites web et présente deux composants de base : ErgoMonitor et ErgoCoIn. ErgoMonitor est un outil de surveillance des niveaux d' utilisabilité qu' un site web procure à ces utilisateurs à partir de l' analyse de fichiers de journalisation. Ce composant met en place une approche d' analyse orientée tâches pour identifier des classes spécifiques des comportements des utilisateurs durant la réalisation des tâches transactionnelles en utilisant le site web. Dans la suite, ErgoMonitor calcule les incidences et les durées des comportements appartenant à ces classes et utilise ces donnés pour produire des métriques d' utilisabilité qui quantifient la productivité moyenne des interactions. ErgoCoIn est un outil EEAO (Évaluation Ergonomique Aidée par Ordinateur) basé sur des listes de vérification, qui se distingue par ses services automatiques de recueil des données du contexte d' opération et de détection des composants des interfaces des sites web. À partir de ces informations il construit des listes de vérification particulièrement adaptées au contexte d' opération du site web en évaluation. En associant ces deux outils, ErgoManager peut assurer une stratégie de garantie de qualité basée sur la confrontation entre les métriques quantitatives d' utilisabilité et les aspects qualitatifs des interfaces utilisateurs.

Mots-clés: SGIU, Métriques d' utilisabilité, Exploration de données, Fichiers de journalisation, Listes de vérification, Sites Web, Commerce Électronique.

ErgoManager:
a UIMS for monitoring and revising user interfaces for Web sites

Abstract: This report describes the specification of ErgoManager as well as the results from the first validation study associated with the development of this UIMS (User Interface Management System) intended to support webmasters at assuring "in use" quality for interactive Web sites. The ErgoManager UIMS aggregates two basic components: ErgoMonitor and ErgoCoIn. ErgoMonitor is a monitoring tool intended to quantify the "average" usability that web sites have been offering to theirs users. It applies task-oriented analysis as a way to identify specific instances of users' behaviors while they are accomplishing transactional tasks with the web site. In the sequence, ErgoMonitor determine the incidence and the duration of these behaviors and use these data to produce usability measures, which quantify the average productivity of interactions. ErgoCoIn is a checklist based CSEE (Computer Supported Ergonomic Evaluation) tool that features automatic services aimed at inquiring context of use aspects and recognizing web page components as a way to tailor focused ergonomic checklists. By integrating these tools, ErgoManager intends to support a quality assurance strategy based on the confrontation between usability quantitative metrics and qualitative aspects of user interfaces.

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

SOMMAIRE

1. Introduction

User Interface Management Systems (UIMS) were proposed in the 80s as an analogy to the Data Base Management System (DBMS), in which a data base administrator has available a collection of tools to create, evaluate, modify, bring up to date and monitor the behavior of a data set during its evolution (Shneiderman, 1998). The underlying idea behind any MS (Management System) is to have an environment that integrates the tools to support developers to accomplish tasks all over the lifecycle of a very changeable system or component, like a database or a user interface. We see a MS as a specialization of a more general concept of CASE (Computer Aided Software Engineering) systems which integrate vertical and horizontal tools to support tasks allover the lifecycle of a general system, not necessarily a changeable one. We advocate that UIMS proposed in the past looked mainly like a CASE (Computer Aided Software Engineering) systems once "before Internet" user interfaces were treated like consolidated entities, and no concerns were directed to support the monitoring and updating activities.

On the other hand, several tools for supporting user interface (UI) evaluation were made available since the 80's (see Scapin et al., 2001). However they were often proposed as isolated or independent tools, i.e., not integrated to an UIMS or CASE environment, and usually they did not address the UI's revision phases. In fact, that phase was not such an important concern in the 80s and 90s, as it is today, mainly in the domain of electronic commerce on Internet.

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

ErgoMonitor is a log analyzer tool aimed at identifying “average” usability levels occurred when users carry out transactional tasks with a Web site (Morandini and Cybis, 2003). In this type of interaction, it becomes possible to identify the users and infer their objectives only by examining data in log files. This allows organizing such data according to a task oriented approach and to treat them following rules aimed at usability metrics production. The values produced by ErgoMonitor represent "average" measures of usability, once they relate to an average operational context, which includes different user types and environment conditions, including high and low capacity processors, modems, screens and others. Even if it is not the ideal information, it is useful enough for a usability engineer, since s/he can get it quickly and at low cost and use these data to monitor the evolution of the average levels of usability during the user interface web site revisions and to specify the usability that its new interfaces must provide to its users.

ErgoCoIn is a computational tool aimed at supporting usability evaluations of web interfaces by means of check-lists (Cybis et al, 2000). The process of ErgoCoIn is based on information about aspects of the web site operational context (characteristic of the users and of their work environment) and on the knowledge about existence of particular HTML interfaces components. This tool features automatic resources to gather this information and by this way, it is able to build objective check-lists with only the questions that apply to the real web site operation context aspects and to the interface components actually present in the pages to be evaluated.

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 a list of web pages that support this task. Moreover, the system will deliver him/her an objective and systematic usability checklist, aimed at helping this professional to identify the usability problems affecting these pages. Once the problems fixed, ErgoManager will be able to signal to webmasters the usability metrics evolution, hopefully towards usual levels.

2. The Ergomonitor Tool

The ErgoMonitor project was thought as a way to help developers and managers to face up the responsibility of continually assuring and improving their web site usability despite the constant updating of actions and information. The general assistance we identified as appropriate for 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, systematic, rapid and non expensive procedures, even if the level of detail and precision are not the highest.

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

2.1. Log file analysis

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 computer identification number (IP);
- Request date, time, type and address (url);
- Request result code and requested document addresses (url) and size;
- Client Browser and operational system identification.

Currently, most popular log analysis tools output can be categorized into the following categories:

- Users perspective: users' geographical region and technical environment;
- Usage/Interaction perspective: most requested pages and documents, date and time of biggest volume of access;
- Maintenance perspective: type and number of errors, components with errors, etc.

WebTrends (WebTrends, 2005) is an example of tool that provides this type of basic requirements. Clicks Counter Pro (MitriDAT, 2005a), between others, supply web developers and managers with users and interactions data in the form of click streams or sequence of clicks on pages and documents accessed by users during interactions. Research Manager Clickstream (Keynote, 2005) is specialized in graphically presenting frequent click streams, in a form of navigation maps in which frequencies are associated with thickness of transition lines. 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. Finally, Audience insite Measures (ComScore Networks, 2005) is an example of system that enriches the user perspective by integrating information from a user database, which is reached by the user IP number.

These data are quantitative, low cost and obtained in a fast and systematic way. They refer to users and interactions, but even though they are quite limited compared to usability evaluation proposals. In fact, a "Usage/Interaction" 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: a "task/usability" oriented perspective. We will be analyzing "tasks" instead of "interactions" and processing "usability measures" instead of "cost-benefit data".

The design of a task/usability oriented perspective depends on the application of two related approaches: the data log task oriented analysis and the usability oriented data log processing.

2.2. The data log task oriented analysis

The **data log task oriented analysis** 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 just by reading log data. This 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 identify in the log data that a user has got access to a registration form, and a few minutes later the system has presented to him/her a confirmation message, it is reasonable to infer that this user was willing to register him/herself. The same holds true for other types of transactions with a "start" and "final" point well distinguishable, like a book order or a product acquisition. Once we know his/her objectives in tasks, we could identify the moment the user had begun and had completed the task and the different paths s/he 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 direct success, success with deviation, success with error, success with help, quitting, canceling (quitting after an error), canceling with help, and so on. Computing the incidence of the alternative successful paths and their duration a system could determine measures of the user efficiency in accomplishing a task. The incidence of failure behaviors could inform about effectiveness, but in these cases, measures will not be very reliable. In fact, there is no way to distinguish between users who did really want to achieve the transaction and were unable to do it, from those who were visiting the sites only to know about its contents and had quitted before to request any execution. Thus, the analysis and treatments presented here can supply webmasters with more reliable measures of efficiency on tasks successfully achieved and less reliable measures on effectiveness over the attempts to carry out the tasks.

2.3. The usability oriented log data processing

The **usability oriented log data processing** refers to the design of a log data abstractions structure composed by: *user' s accesses, user's episodes, user's movements* and *user's behaviors on tasks*.

The first thing to do is to individualize the **user's accesses**. In practice, it is very difficult to do this only from IP numbers, once a same client machine's IP could be shared by several users getting access through the same proxy server (Srivastava, et al., 2000). This is a kind of "middle way" server, placed between clients and content servers. Proxies stores locally the most frequent pages users accessed by they mediation, and so far supply clients with these copies instead of transferring the pages requests to the corresponding content servers. Thus, they are a very useful component for networks operation, performing actions to decrease traffic and time to get information. However, their actions complicate the log data analysis, specifically for user identification as well as for extracting the paths they had been crossing. The usual solution to the user identification problem consists of individualizing a user as a data abstraction composed by <IP number, OS name and Browser name>. This increases the number of differentiating keys, but it is not error prone especially for log file associated to a huge transactional traffic. However, this step could be extremely simplified in case of web sites where user access is controlled by a password. Here, the user's name will be registered in log records and the user identification becomes 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 apart from more than 30 minutes, once most task resuming time falls into this interval (Srivastava, et al., 2000). **User's movements** are in fact, system transitions caused by users' actions and could be viewed 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 resulting from a request done from another page. Movements are classified in relation to a set of anticipated movements. These are viewed as logically authorized movements by the current user interface. Instance of users' movements are: "task entry", "task evolution", "task exit", "return to task", "task accomplishment", "help searching", "error managing", and so on. Since a proxy server supplies clients with a page stored locally, without transmitting this request to the corresponding content server, some users' movements will not be registered into log files. As a consequence, the users' behaviors identification will be compromised, especially those characterized by deviations. In fact, the proxies' action will prevent the measures from being as detailed as they could be. The proxy effect is not verified however, when pages are generated dynamically by the content server. Here, no matter the frequency of users' accesses, all requests will be posted to the content server and will be registered in log files.

A **user's behavior** is an ordered set of user's movements that ends with the task accomplishment or the episode's end. As well as episodes, users' behaviors are characterized according to a set of anticipated behaviors like "immediate success" on task (entry, evolution, accomplishment), "success with deviation" (entry, ..., exit, …, return, …., accomplishment), "success with error" (entry, …, error-recovery, …, accomplishment), "success with help" (entry, …, help-searching, …, accomplishment) and so on.

The incidence and the time of identified behaviors can be now 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) standard.

2.4. Delimiting application of the task&usability perspective

The conceptual architecture and the technological constraints define this approach limitations: it is intended to be applicable to transactional web sites (inter or intranets) where user access is controlled by password and where pages are generated dynamically, as verified in B2B (Business to Business), including ERP (Electronic Resource Planning) internet based systems and some portion of B2C (Business to Consumers).

It is worth to mention that usability measures produced by this approach will be average ones, since the system will consider all tasks trails during a period of time, which refers to different users, pertaining to different profiles and integrated to different software, hardware and physical environments. Taken these limitations into account, the results obtained are expected to be precise enough.

For informational sites or the opened portion of B2C (Business to Consumers) electronic commerce, this conceptual framework is out of focus, since it is impossible, based only in the log data, to infer users' objectives. Even so, the task oriented log data analysis could be useful, if its issues are taken in a relative basis, i.e., compared with the historical values obtained in past for the same limited 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 and investigate a disturbance detected in site usability curve. Supposing that others context of use components are stables, this variation might only be caused by a useless interface users had begun to interact some time ago.

2.5. The ErgoMonitor tool specification

The ErgoMonitor employs both the task oriented analysis and the usability oriented processing to determine usability metrics for a given task and a given user interface for a period of time. The underlying specification prototype presents the following modules: Monitoring Proprieties, Functional Core, Usability Measures Data Base and Monitoring Analysis and Reports.

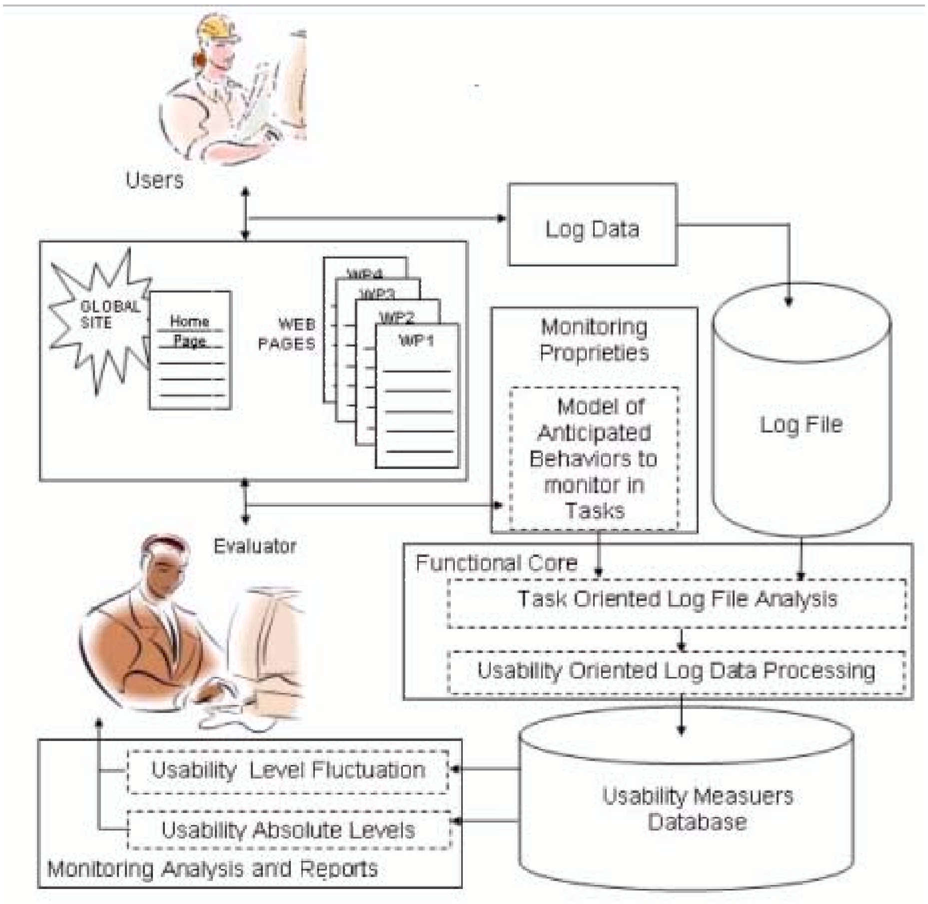


Figure 1 – ErgoMonitor structure overview

ErgoMonitor treatments starts with an analyst examining the web site and defining a model of anticipated (logically authorized by the current user interface) movements and behaviors to detect in the log data for each task to be monitored. The UI analyst will be filling the monitoring properties forms with **tasks markers**, which are meant as mandatory, and optional web pages associated with tasks. Tasks markers are initial page, final page, intermediated pages (optional), error page (optional), help pages (optional). The system is able to build the entire model of movements and behaviors from these tasks markers.

In general, the forms from the Monitoring properties module will be collecting the following data:

- Site name and description;
- Site and log file access data;
- List of task to monitor.
  - Task identification
  - 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 (screen shots, navigation map, 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. Thus, ErgoMonitor will be monitoring usability in less changeable task structures which are supported by more changeable user interfaces. Ideally the task description is filled in only one time and the user interface description each time it is revised.

2.5.1. The Functional Core

Based on the monitoring proprieties and having the log data file as an entry, this module will perform the treatments associated with the log data task oriented analysis and the usability oriented log data processing. The objective here is to build a structure of data abstractions including; users' accesses, user's episodes, users' movements and users' behaviors. Based on user's behaviors incidence and duration, the tool will next compute a set of usability measures about user effectiveness and efficiency on tasks.

The system start by implementing the strategies described in section 2.3, first paragraph, which aims at individualizing users' access and episodes. For each user's episode identified, ErgoMonitor will construct the following data abstractions:

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);
- Return to task = url not associated with the task → Initial page (in a user's behavior not yet concluded);
- Error recovery = 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;

- Immediate Success (IS) = Task entry + Task evolution (optional) + task accomplishment;
- Success with Deviation (SD) = Task entry + Task evolution (optional) + Task exit + Return to task + Task evolution (optional) + Task accomplishment;
- Success with Error (SE) = Task entry + Task evolution (optional) + Error recovery + Task evolution (optional) + Task accomplishment;
- Success with Help (SH)= Task entry + Task evolution (optional) + Help searching + Task evolution (optional) + Task accomplishment;

- Visit (VI) = Task entry + Task exit;
- Quit (QU) = Task entry + Task evolution (optional) + Task exit
- Cancel (CA)= Task entry + Task evolution (optional) + Error managing + Task exit

Based on these behaviors' incidence and time, the ErgoMonitor functional core will compute a set of usability measures, whose construction rules are listed below:

Usability factors
- Amount of Success (#S) = #DS + #SD + #SE + #SH;
- Amount of Failures (#F) = #QU + #CA
- Amount of Task Trials (#T) = #S + #F
- Amount of Access (#A) = #VI + #T;

Usability rates
- Rate of Visits (%V) = # V / #A
- Rate of Task Trials (%T) = # T/ #A
- Rate of Success = (%S) = # S / #T
  - Rate of Immediate Success (%IS) = #IS / # T;
  - Rate of Success with Deviation (%SD) = #DS / # T ;
  - Rate of Success with Help (%SH) = #SH / # T ;
  - Rate of Success with Error (%SE) = #SE / # T ;
- Rate of Failures = (%F) = # F / #T;
  - Rate of Quits (%Q) = #Q/ #T;
  - Rate of Cancels (%C) = #C / #T;

Usability metrics
- Mean Time to Task (MTT) = Σ Time (∀S) / #S (Obs: read "∀S" as "all Success behavior")
  - Immediate MTT = Σ Time (∀DS) / #DS;
  - MTT with deviation = Σ Time (∀SD) / #SD;
  - MTT with error = Σ Time (∀SE) / #SE;
  - MTT with help = Σ Time (∀SH) / #SH;

2.5.2. The Usability Measures Database

This database will be maintained by the Functional Core that will be storing an historic series of values concerning the usability factors, rates and metrics for each transactional task being monitored. These entries will be indexed by task, user interface version and period of time they are related to. So, the database will authorize correlation analysis between interface design and usability measures.

2.5.3. Monitoring Reports

This module will be requesting usability metrics stored on the database to create two types of standard reports: Usability Absolute Level Report and Usability Level Fluctuation Report.

The first report will supply web managers with tables of absolute usability values concerning different usability factors, rates and metrics for different tasks. It is clear that good user interfaces are associated to:

- higher values for usability measures directly related to immediate success behaviors ;
- lower values for behaviors related to success with deviations, error recovery and help searching.

The second report will display a set of graphs corresponding to the evolution of these usability measures for a same task, differentiating data from different user interfaces. A set of warnings will be available to indicate the web developer when system detects decreasing movements of usability lines in the near past.

2.6. The ErgoMonitor's functional core validation study

The procedures associated with the ErgoMonitor's functional core were initially validated in a study at INRIA, during April and March 2005. For this study we decided to apply manually the procedures described above, thus the log date size and the task complexity were supposed to be small and simple. In such conditions it is easy to individualize users, even in a site in which access is not controlled by user ID and password. We were looking for getting access to log data concerning very simple transactional tasks, which were realized by a small population, and in an infrequent basis. We found this kind of tasks being mediated by the École Polytechnique de Montréal Library's web site. They were:

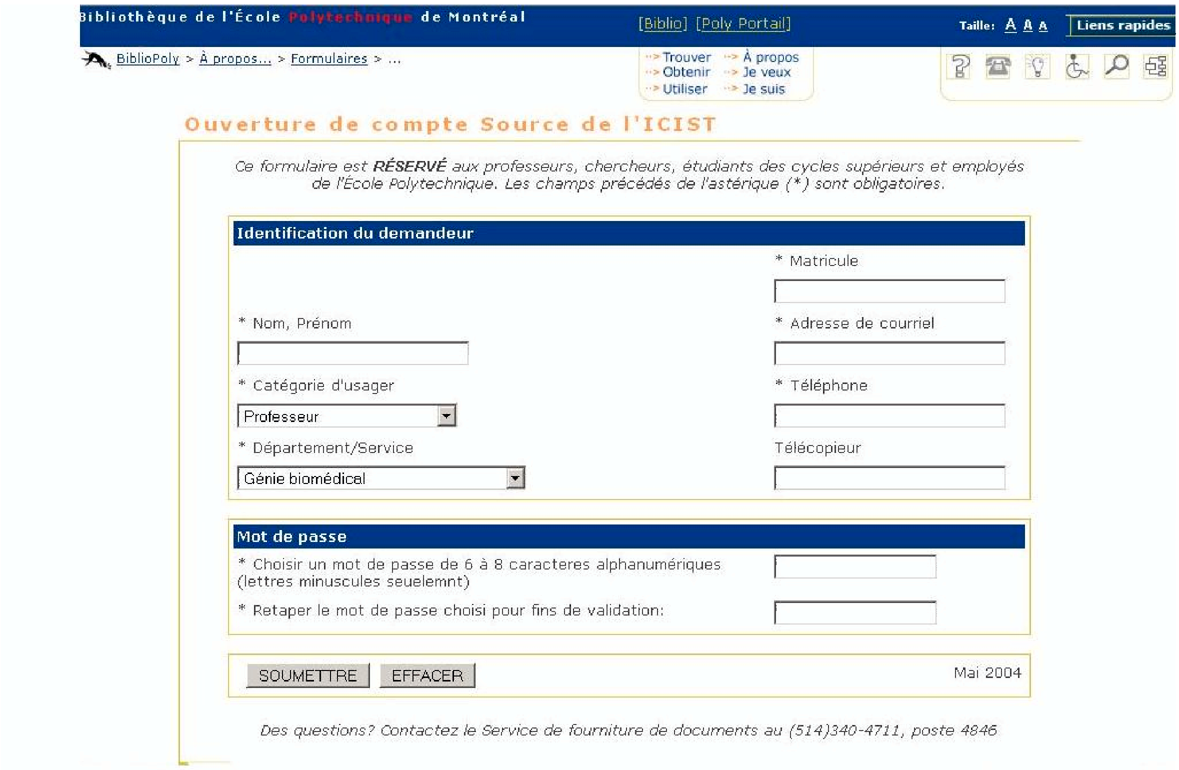
- to subscribe to the « Institut Canadien d'Information en Sciences et Technologie » (ICIST);
- to place reservation requests for books and articles;
- to request a book or an article from a remote storage;

2.6.1. Materials and method

The personnel from the EPM's "Service d'Informatique" provided us with access to the data log associated with the accomplishment of these tasks between November 2004 and February 2005. These files were mounted by filtering the original log files from any other access record than the related to these tasks' markers. This time, the tasks were mediated by users interfaces sharing a same structure featuring: one main form whose fields were mostly mandatory, several error messages associated with the occurrence of empty mandatory fields and syntax error on e-mail address, some general help pages and a final task confirmation page. The main forms associated with the three tasks are presented in Figures (2, 3 and 4).

The task structure was in all cases, too simple, featuring only two mandatory movements; "task entry" (get access to the form to fill) and the "task accomplishment" (to see the confirmation message). There were no "task evolution" movements, and, as in any other task, both the "help searching" and the "error fighting" were optional. Without "task evolution" movements, it was impossible in the validation study to distinguish between the user's behaviors of "Quit" and "Visit", once both behaviors were composed by just one "task entry" and one "task exit".

Another limitation of the study was associated to the way error management was implemented in the main forms. As a way to gain interaction time, the web designer implemented small programs code (Java applets) which were downloaded along with others page's components and had the responsibility to verify if mandatory fields were empty. If it was the case they locally generated the error messages and didn't communicate the content server of these occurrences. Only the fields associated with the e-mail address in the three main forms were subject to syntax verification performed by a piece of code on the server side, which sent a specific error message to the user in case of error. In consequence of this implementation, most of "error recovery" movements were not registered in the log files of EPM's Library and the monitoring of "Canceling" (quitting after an error recovery) behavior was gratefully compromised. Due to the impossibility to monitor the "quitting" and most of "cancelling" behaviors, the measures associated to task efficacy were not considered by this validation study, which indeed, was centered on the procedures associated to measures of efficiency on successfully tasks.



Bibliothèque de l'École Polytechnique de Montréal [Biblio] [Poly Portail] Taille: A A A Liens rapides

BiblioPoly > À propos... > Formulaires > ...

--> Trouver --> À propos
--> Obtenir --> Je veux
--> Utiliser --> Je suis

Ouverture de compte Source de l'ICIST

*Ce formulaire est **RÉSERVÉ** aux professeurs, chercheurs, étudiants des cycles supérieurs et employés de l'École Polytechnique. Les champs précédés de l'astérique (*) sont obligatoires.*

Identification du demandeur

* Matricule
* Nom, Prénom
* Adresse de courriel
* Catégorie d'usager: Professeur
* Téléphone
* Département/Service: Génie biomédical
Télécopieur

Mot de passe

* Choisir un mot de passe de 6 à 8 caracteres alphanumériques (lettres minuscules seuelemnt)
* Retaper le mot de passe choisi pour fins de validation:

SOUMETTRE EFFACER Mai 2004

Des questions? Contactez le Service de fourniture de documents au (514)340-4711, poste 4846

Figure 2 – The ICIST subscribing form

The subscription for the ICIST is authorized to teachers, researchers, graduated students and EPM employers. These users were supposed to fill the form presented in Figure 2 with their identification data: user name, user category, department associated, personal identification number, e-mail address, telephone and fax number. This last field was not mandatory. Eventually, users were asked to fill, in the bottom section of the form, the two replicated fields associated with the password.


Bibliothèque de l'École Polytechnique de Montréal [Biblio] [Poly Portail] Taille: A A A Liens rapi

BiblioPoly > À propos... > Formulaires > ...

--> Trouver --> À propos
--> Obtenir --> Je veux
--> Utiliser --> Je suis

Demande de mise en réserve

*Ce formulaire est **RÉSERVÉ** aux professeurs et chargés de cours de l'École Polytechnique. Les champs précédés de l'astérisque (*) sont obligatoires.*

Identification du demandeur

* Nom de famille | * Prénom | * Département/Service [Autres]

* Catégorie d'usager [Chargé de cours] | * Téléphone (travail) | * Adresse de courriel

Cours

* Code du cours
* Titre du cours
* Année universitaire

* Trimestre ☐ automne ☐ hiver ☐ été

Document(s) à mettre en reserve

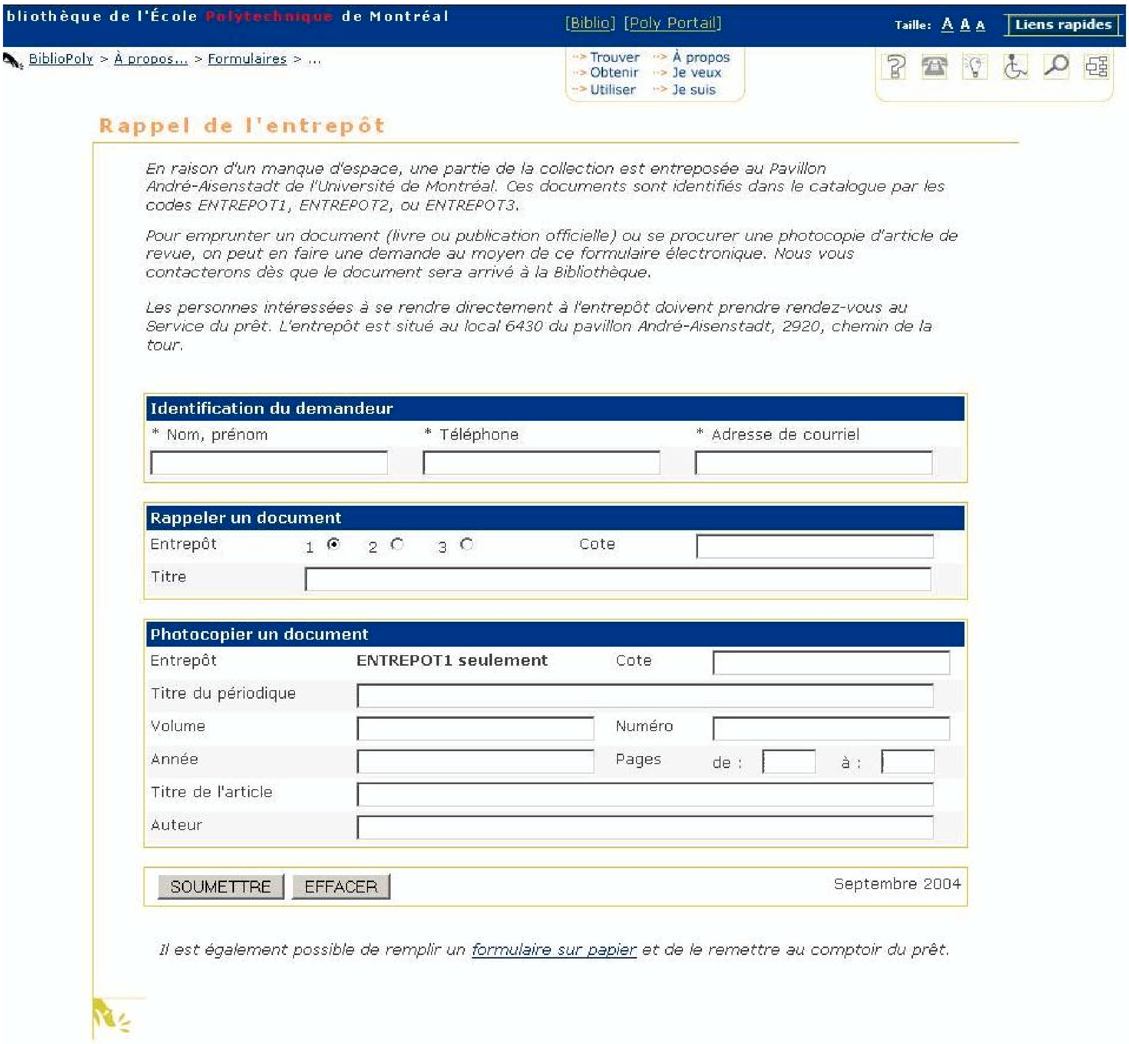
1. Titre	Cote	Année de publication
2. Titre	Cote	Année de publication
3. Titre	Cote	Année de publication
4. Titre	Cote	Année de publication
5. Titre	Cote	Année de publication
6. Titre	Cote	Année de publication
7. Titre	Cote	Année de publication
8. Titre	Cote	Année de publication

SOUMETTRE EFFACER Mai 2004

Des questions? Contactez les Services d'accès à la documentation au (514)340-4711, poste 4849

Figure 3 – Form to receive a request for place a book in reserve.

The task of placing reserve requests for books and articles was authorised to professors and lecturers only, in function to theirs needs during course planning and delivery. They used the form presented in Figure 3, in which the first group of fields was aimed at gathering the same user identification data asked by ICIST form except for the fax number (user name, user category, department associated, personal identification number, e-mail address and telephone number). A second group of fields were aimed at gathering data concerning the course associated to the current reserve request (code, title, year and semester). The last group of fields was aimed at receiving data concerning up to 8 documents to place in reserve (title, cote and publication year).


bliothèque de l'École Polytechnique de Montréal [Biblio] [Poly Portail] Taille: A A A Liens rapides

BiblioPoly > À propos... > Formulaires > ...

--> Trouver --> À propos
--> Obtenir --> Je veux
--> Utiliser --> Je suis

Rappel de l'entrepôt

En raison d'un manque d'espace, une partie de la collection est entreposée au Pavillon André-Aisenstadt de l'Université de Montréal. Ces documents sont identifiés dans le catalogue par les codes ENTREPOT1, ENTREPOT2, ou ENTREPOT3.

Pour emprunter un document (livre ou publication officielle) ou se procurer une photocopie d'article de revue, on peut en faire une demande au moyen de ce formulaire électronique. Nous vous contacterons dès que le document sera arrivé à la Bibliothèque.

Les personnes intéressées à se rendre directement à l'entrepôt doivent prendre rendez-vous au Service du prêt. L'entrepôt est situé au local 6430 du pavillon André-Aisenstadt, 2920, chemin de la tour.

Identification du demandeur

* Nom, prénom | * Téléphone | * Adresse de courriel

Rappeler un document

Entrepôt 1 2 3 Cote

Titre

Photocopier un document

Entrepôt **ENTREPOT1 seulement** Cote

Titre du périodique

Volume Numéro

Année Pages de : à :

Titre de l'article

Auteur

SOUMETTRE EFFACER Septembre 2004

Il est également possible de remplir un formulaire sur papier et de le remettre au comptoir du prêt.

Figure 4 – Form to request a book or an article from a remote storage.

The last task analysed concerned a book or an article requesting from one of the three EPM's library remote storage. To do that, everyone associated with the school was supposed to fill a form composed by three groups of fields (Figure 4). The first one was aimed at gathering user identification data, reduced here to name, phone, and e-mail address. The second group of fields was associated with the book to bring back (title and cote) as well as the corresponding remote storage identification. The last fields group was concerned with the identification of one article to be copied (cote, review title, paper title and author name). In last line of the form, there was a link to download of a .pdf document containing the a printable version of this form that was supposed to be printed, hand filled and delivered in the library office. We assumed here that the form was supporting two tasks: a main on-line request task and a secondary one of downloading the form.

The first operation in this study for validating ErgoMonitor approach was to filter and clean log files from the records not associated with users' movements with the system. In this process we needed to delete:

- robots access: in fact, there is a more or less established set of robots, including those sent by the web master, who "visit" the pages once a day only to detect its availability or to index its contents;

- client calls to page components and server methods : most records in a log data file concern requests to server supplying clients with figure files (.gif .jpg,…), java applets (.js), style sheets (.ccs) and other types of web pages components, as well as, client calls to server methods (.cgi);
- records related to client or server errors: the status-code is a logging element that represents the result of the request made. Client and server errors are identified by three digits codes beginning by 4 (like the famous "404 not found") and 5 respectively (RFC 2616, 1999).

A new file was then created with the remaining records. We had been working with this new file to identify and individualize users' access and episodes, applying heuristics described in section 2.3 from this report. On each user episode, we categorized users' movements and behaviors according to the model of anticipated user movements and behaviors. Next, we computed the time associated with each behavior identified, and finally, we created a spreadsheet with the data. In this electronic document we implemented the usability measures construction rules as defined in section 2.5.1.

2.6.2. The results obtained

The usability measures computed from data log registered between November 2004 and February 2005, concerning the "Subscribing for the ICIST" task revealed that 23 out the 42 access, were simple visits (54,76 %). The remaining behaviors (19) were identified as successful, from which 17 (89,47%) where achieved at first trial (immediate success). One user (5,26% of those who were successful) had deviated from the straightforward way (success with deviation) and one another has made an error in entering an e-mail address (success with error) before successfully accomplishing the task. The immediate success behavior was accomplished in average, within 1'59". The success with deviations behavior took 5'38" for the user and the success behaviors associated with error, 1'19". Evidently, the incidence of just one success with deviation and one another with error do not allow to consider these values as average. In the monitoring period no help searching behavior was observed.

The log data associated with the task of “Placing reserve requests for books and articles” concerned 38 accesses, from which 22 (57,89 %) were simple visits. The remaining ones were characterized as successful behaviors. For 11 (68,75%) of them, users placed reservation requests immediately in 3'58". In 5 cases (32,25%) there were some deviations and the mean time of tasks increased up to 16'45". In the monitoring period there was no observation of any error recovery or help searching behaviors.

It was possible to identify 103 user’s accesses to main form of the task of “Requesting a book or an article from a remote storage”. 75 of them (72,82%) were classified as simple visits. The remaining accesses were characterized as successful on both tasks (the main on-line requesting task and the secondary one of downloading the form). There were 26 successful behaviors to the main task and 2 to the secondary one. In the main task, 23 (88,46%) of success cases were characterized as immediate success and 3 (11,54 %) as success with deviation. The users took 3'26" in average to place requests straightly (immediate success). When there were deviations in task, they took 4'25" in average. Users took 20 seconds in average to complete with immediate success the 2 trails associated with the secondary task. In fact, the very small sample of deviations in main tasks avoid considering these time as average. In the monitoring period there was no observation of any error recovery or help searching behaviors.

Results obtained from applying log data analysis and treatments as specified in ErgoMonitor functional core for these three tasks are synthesized in Table 1.

Task	Access	Visits		Success		Immediate Success			Success with Deviation			Success with Error		
		#	%	#	%	#	%	time	#	%	time	#	%	Time
1	42	23	54,76	19	43,34	17	89,47	1’59”	1	5,26	5’38”	1	5,26	1’19”
2	38	22	57,89	16	42,11	11	68,75	3’58”	5	31,25	16’45”			
3a	103	75	72,82	26	37,34	23	88,46	3’26”	3	11,54	4’25”			
3b				2		2	100,00	20”						

Table 1- Results from applying ErgoMonitor functional core's treatments to the log data of three tasks.

2.6.3. Analysis of results

The study shows that systematization of log data analysis and treatments proposed by ErgoMonitor functional core are possible, depending only on coding of some data searching, cleaning and structuring procedures. They do not require any kind of inference implying artificial intelligence algorithms.

The log data analysis conducted allows to reveal interesting data concerning the usability levels that the EPM Library site offers to users when accomplishing these three tasks. Without any usability test performed, it was possible to know what portion of users did the three tasks directly or with some deviation, and the times associated with theirs behaviors (Table 1). Better than in a test, the measures determined by this study have higher degree of reliability since there was not any interference from the laboratory apparatus on users. As long as a tool is available, this date would be cheaper, faster and easier to obtain than equivalent results issued from any usability tests. On the other hand, in this study we have identified some behaviors observed only two or three times. Consequently, the corresponding average measures obtained could not be considered as representative. In fact, future studies need to be undertaken to establish the conditions a sample (size and gathering period) must fulfill to authorize representative average measures determination.

Beyond validation of functional core specifications, this study aimed also at clearing up the intended strategies for the “Monitoring analysis and report module”. Two types of strategies were previewed: absolute values analysis and values fluctuation analysis.

The variation values analysis are based on the comparison of usability measures computed from different moments and related to different interfaces associated with a same task. This type of analysis could not be performed in this study, since we had no available usability measures obtained for these tasks when supported by others user interfaces.

On the other hand, the absolute values analysis can be based on the comparison of usability measures obtained from different tasks. It is important to mention the quality of each task accomplishment has intrinsic determining features. The first is the user interface itself (with or without problems), but of course, the type of users, as well as the size, availability and organization of information required to perform the task play important roles. We need to analyze carefully the overall set of aspects determining usability before to assign to the user interface the responsibility for the lower values obtained. In this study we looked at two examples of absolute values analysis based in the inter-tasks comparisons.

The first situation concerned the occurrence of a higher “success with deviation” rate for task 2 (place a request to reserve a book), when compared to task 1 and 3 rates (Table 1). The first impulse would be to blame the interface for these deviations. However, examining further the data, we realize that the interface supports reservation request placing in batch, up to 8 books simultaneously. With such a configuration it is natural if users have not available the data to place reserve requests for all books and be required to leave the page to search for a book reference is missing. So, we realize by this situation analysis that the higher rate of deviations in successful behaviors could be considered as normal when a task is supported by a batch structure interface.

The second situation is almost the opposite. It reveals a unit structured user interface causing difficulties to users working in a batch structured task. In fact, we have verified that in task 3 (request a book or an article from a remote storage) the successful behaviors were repeated several times by several users and this was not verified in the others tasks. Examining the user interface associated with this task, we realize that the form to gather users' requests offers fields to input only one request for an article copy. However, people could have the need of copying more than one article from the remote storage in a given situation. This logical consideration explains the existence of repetition in this task and the consequent usability problem of workload to users with such needs. Indeed, he/she will be required to fill his/her identification data at every repetition, as well as to repeat the page access and task confirmation actions every time. The redesign action concerns obviously the implementation of several groups of fields to allow the form to gather data from several articles to copy. Additionally, this case study revealed the need of a new class of usability measures to be included in ErgoMonitor definition. It is associated with repetition of successful behaviors by a same user in a same episode.

2.6.4. Study conclusions

2.6.4.1. New usability factors and rates

The analysis of this situation has motivated the inclusion of new type of behaviors, as well as new usability factors and rates to the ErgoMonitor specification.

New behaviors

- First Success Repetition (1SR) = ∀ Success behavior + ∀ Success behavior *(read ∀ as any type of)*;
- Second Success Repetition (2SR) = 1SR + ∀ Success behavior;
- Third Success Repetition (3SR) = 2SR + ∀ Success behavior;
- ….
- Nth Success Repetition (nSR) = (n-1)SR + ∀ Success behavior;

These behaviors occurrence are governed by a progressive mutually exclusive relationship in a context of a user access: the occurrence of a higher level repetition cancels the occurrence of the lower level one. Thus for example, a Third Success Repetition identification cancels the occurrence of a Second Success Repetition for a given user's episode.

New usability factors

- Amount of Success Repetitions (#SR) = #1SR + 2*#2SR + 3*#3SR + ….. + n* nSR;
- Amount of Users with Repetitions (#UR);
- Max Amount of Repetitions by User (Max#R/U);
- Average Amount of Success with Repetitions by User With Repetitions (AASR) = # SR / #UR ;

New usability rates

- Rate of Success with repetitions (%SR) = # SR / # S

Any way, a good interface is expected to present lowers values to repetition measures.

Applying these new usability factors and rates definition to the case studied allows obtaining the following measures from interaction log data associated with the "Request a book or an article from a remote storage" task :

Usability factors
- Amount of Success Repetitions (#SR) = 3x1 + 10x1= 13;
- Amount of Users with Repetitions (#UR) = 2
- Max Amount of Repetitions by User (Max#R/U) =10;
- Average Amount of Success with Repetitions by User With Repetitions (AASR) = 13 / 2 = 6,5 ;

New usability rates
- Rate of Success with repetitions (%SR) = 13 / 26 = 50 %

Interpreting these numbers it would be able to suggest designers revise the form in a way to implement at least 5 fields, instead of only one. If they had done it before, the amount of task repetitions during the monitoring period could drop down to only 2, concerning only one user.

2.6.4.2. Relationships between movements

Furthermore, this study allowed us to observe the existence of interdependency relationships between movements into a same behavior, which permits to infer the existence of a given movement even if its record isn't in the log file. Specifically, when trying to characterize a behavior as a "success with deviation" we observed the following sequence of movements in an episode: task entry, task exit and task accomplishment. A logical analysis indicates that between the last movements, there was a missing "return to task" record. In fact, after having left a task, the only way to accomplish it requires the user returning to the task. Evidently, this movement has been performed by the user, but the page requested was already stored even in the proxy server facilities or in the user machine's cache memory. Consequently, the web server where the site is hosted was not informed about this movement.

Thus, in some situations it is logically possible to complete movement records missing in the log file, due to interdependences established between them. In the current example, this type of relationship could be stated as: the "task exit" followed by a "task accomplishment" in a same episode presupposes the existence of a "return to task" movement, placed between them.

Based on this specific statement it is possible to distinguish three types of movements:

- Main determining movements: its records in log files are minimally necessary to characterize unequivocally a behavior. (e.g. Task Entry, Task Exit and Task Accomplishment);
- Subordinated determining movements: they are movements that are presupposed by others. Its presence in the log records is not mandatory to characterize unequivocally a behavior. (e.g. Return to task).
- Optional movements: they could not be present in the log file due to differences tasks configurations, specifically, the number of steps of a sequential tasks (e.g Task Evolution)

In consequence, the "Success with Deviation" statement could be revised to be as follows:

- Success with Deviation (SD) = Task Entry + Task Evolution (optional) + Task Exit + Return to task (subordinated) + Task Accomplishment ;

The existence of “subordinated movements” makes the behavior identification easier and more flexible, once reduces the number of conditions a search engine will need to meet to characterize a behavior. However, the fact of having movements records missing in the log data prevents to get more precise usability metrics because the impossibility of knowing the time of movements, even if we could infer about their existence. In fact, if we do not know when exactly a user returned to the task execution, we could not determine the user productive time in task, a fine usability measure proposed by the Music performance measures method (Macleod et al., 1997). It represents the time users were concerned with only the immediate task realization, not taking in account the time users were out of the expected task path, because they were lost, looking for help, recovering incidents and so on. Both measures, productive and no productive user time, would really enhance the quality of usability studies, but unfortunately, the “proxy and cache effect” makes it impossible do determine finest usability measures by means of ErgoMonitor.

Finally, it is worth to mention also that the measures resulting from ErgoMonitor treatments give a partial vision on the web site usability for the monitored tasks. They concern only the efficiency that users demonstrate in successfully accomplishing theirs tasks. A more general overview upon usability needs to aggregate also measures from effectiveness and user satisfaction about the tasks performed with the web site. ErgoMonitor is not supposed to supply webmaster with any user satisfaction measures, but it could give a view about user efficiency, even if it will not be that precise. This feature remains to be validate in future case studies, with more complex tasks.

2.7. Final considerations about ErgoMonitor

Besides its application as a monitoring tool, ErgoMonitor could be used also for assisting the execution of conventional or remote usability tests. In fact, as a test support tool, ErgoMonitor could automatically and at low cost, collect, organize and supply evaluators with a more complete set of usability measures in any kind of tasks. Actually, in a test condition we can extract the users’ identity and objectives as well as the time for theirs tasks trials from the test schedule. The integration of such data into ErgoMonitor processes allows this tool to produce reliable effectiveness measures even for informational and opened web sites. In fact, if we know that a particular user is supposed to perform a particular task, and we verify the inexistence of successful behaviors in his/her task trails, we can not associate this fact to his/her site exploring attitude and, consequently, the user interface is the most accountable.

Due to its conceptual features, ErgoMonitor could be also employed to support free remote usability tests, in which the users will be in their own homes or offices doing tasks from their choice. In this case they will need to inform the system about their intentions, and an ErgoMonitor additional module could be designed to gather and integrate this information to the general treatments. Allowing tests conditions more flexible to users, ErgoMonitor would authorize usability tests results be more reliable.

3. THE ERGOCOIN TOOL

Beyond a tool for processing quantitative usability monitoring, the ErgoManager environment includes another one, aimed at supporting qualitative user interfaces usability evaluations. The ErgoCoIn is a checklists based CSEE (Computer Supported Ergonomic Evaluation) tool originally specified to support tasks oriented usability evaluations of e-commerce web sites (Cybis,et al., 2000). In this section we will be presenting the ErgoCoIn original design revised in order to integrate the ErgoManager UIMS as well as the software components needed to computerize the approach.

ErgoCoIn was originally defined from two main 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. The second consideration is that web sites are often designed along a fast and low cost process supported by non expensive tools which lead designers to carry out numerous and sometimes obvious ergonomic flaws. According to these considerations we had defined some basic requirements for a qualitative usability evaluation approach. The first one concerns the need for supporting web-designers to evaluate their own web sites. Such a method should accommodate this type of evaluator, for example, by providing a minimal and factual amount of ergonomics knowledge directly applicable to the context of use of the interface being evaluated. The second requirement concerns the method application procedures which should be performed in a fast and costless way, for example, having as object of evaluation only the interfaces related to some tasks, not the overall web site.

The associated limit of these original requirements is that the intended method would only consider minimal knowledge about users and tasks (minimal if compared with extensive task analysis, task modeling, etc.) and would not point at all major and complex ergonomics problems. Thus, it was intended as a first step method, aiming at helping designers to identify simple and easy to diagnose problems, just before to ask the help from usability expert professionals. Other limitation of this technique, which is quite compatible with the 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.

Two directions have been considered in this technique design. The one was to use a method known as being fast and cheap, in particular, the checklist-based usability inspection, The other one was to incorporate inquiring techniques (interviews and questionnaires) to supply evaluators with 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. The resulting ErgoCoIn conceptual architecture combines inquiring techniques with checklist-based inspection in an approach able to allow rapid, context focused ergonomic inspections. Three phases of activities were previewed; the contextual analysis; the evaluative inspections and the inspections documentation (Figure 5).

The goal of the **contextual analysis** phase is to collect all information related to the web site context of use that is useful for the usability evaluation process. This concerns applying closed questionnaires to users and designers, as well as, recognizing the site components associated with tasks in evaluation. Here, the evaluator defines the evaluation strategy, formed by a set of tasks that will drive the inspections. The second method’s phase is formed by two types of **evaluative inspections**; one analytical, based on the analysis of information about the context of use; the other empirical, based on examining the web pages components. In the **usability reporting** phase, the evaluator will be reporting the problems identified in these evaluations.

The original ErgoCoIn architecture is naturally fitted to its integration with ErgoMonitor, since both approaches are tasks oriented and aimed at e-commerce applications. Some minor adaptations are necessary. They are described in the next paragraphs along with the software tool specified to support the adaptation.

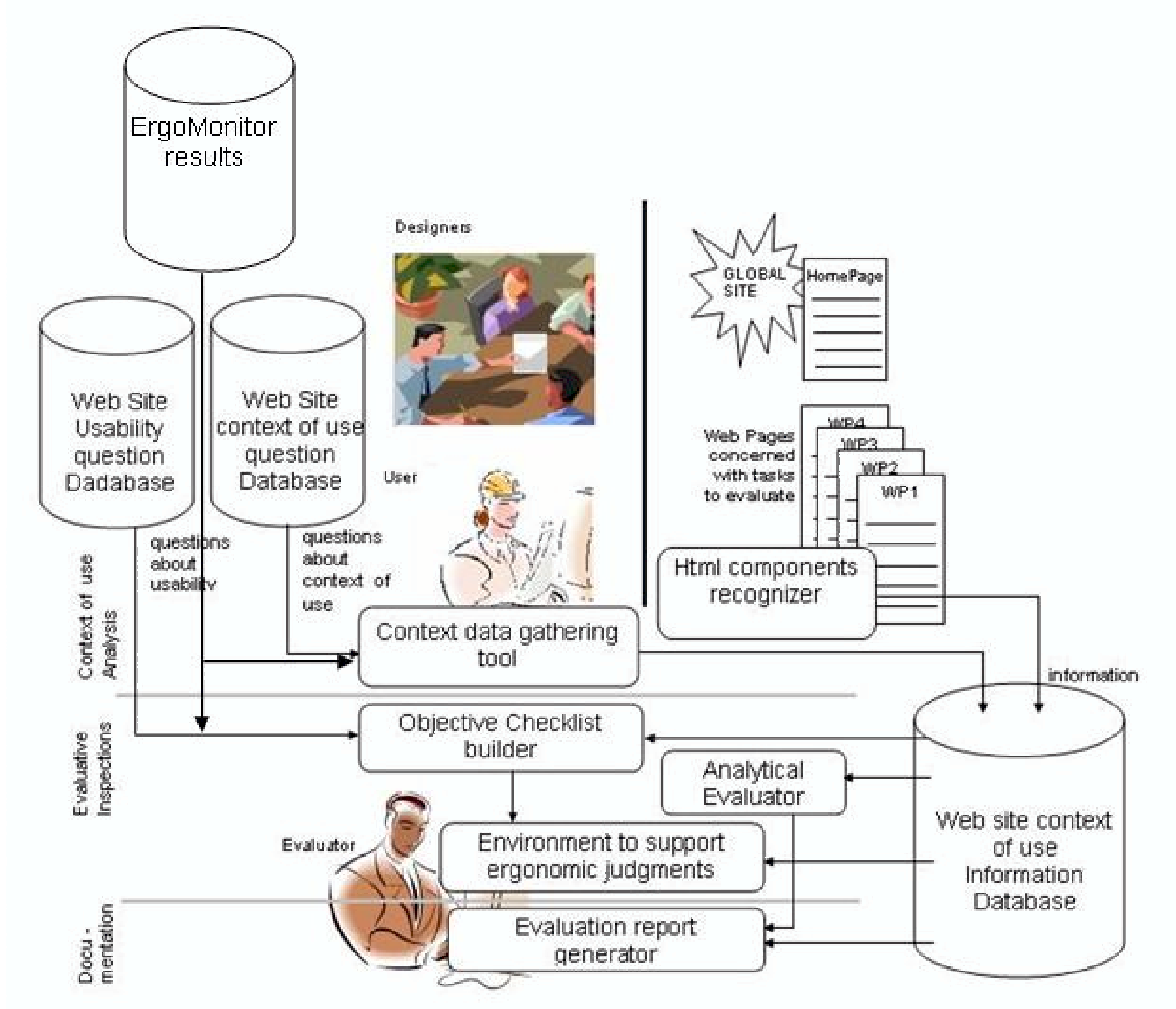


Figure 5 – General ErgoCoIn approach and the software tools.

The base component of ErgoCoIn architecture is a **web site usability questions database**, which integrates questions about the usability of web sites. It results from examining a large collection of ergonomic recommendations applicable to e-commerce web sites (Leulier et al., 1998) later completed with other data collected from different studies (Scapin et al. 2000). This knowledge base was created by reformulating recommendations as questions and associated them to both an ergonomic criterion and to a specific interface component or attribute. While the first index allows defining a management system of relative importance between questions, the second allow systematizing the process of defining questions applicability. In the perspective of integration ErgoCoIn with ErgoMonitor, new knowledge gathering activities will be necessary, specifically in a way of aggregating recommendations concerning specifically transactional tasks. The ISO 9241:17 (Form Filling Dialogues) appears to be the natural knowledge source.

The **Web site context of use question database** contains a set of questions to be included in users and designers interviews/questionnaires. They were defined by analyzing the information demands in each question present on the ergonomic questions database.

As a way to integrate ErgoCoIn to ErgoMananger UIMS, we defined a new database component including the results obtained from ErgoMonitor treatments, specifically those indicating tasks in which usability measures have been presenting decreasing values.

Information from **ErgoMonitor results database** will guide the definition about which tasks to evaluate by means of the ErgoCoIn tool.

In the contextual analysis phase, a **context data gathering tool** will be inviting users and designers to answer on-line questionnaires including closed questions about aspects related to contexts of tasks being evaluated (users, task and environment features). On the other hand, an **html component recognizing tool** will be identifying the existence of specific user interface components on the web pages associated with these pages. These two tools will be feeding a **context of use information database** with descriptions from the site components and the context of use aspects associated with tasks being evaluated.

The analytical evaluation process starts with the action of an **automatic analytical evaluation tool** which will be making comparisons between information provided by users and designers, concerning the intended and the real context of use features. The tool will point out to existence of designer's misconceptions about users' features, and indicate the web site aspects to verify in consequence. Next, an **objective checklist builder** will be assembling checklists concerning the Web Pages related to scenarios of task being evaluated. These checklists can be considered as “focused” ones, once they propose only the questions applicable to the site components being evaluated and arranged according to their levels of importance. The applicability decisions result from processing the site description stored in the context of use database.

The priority between the questions on checklists results from ranking the Ergonomic Criteria (Scapin & Bastien, 1997) according to context of use features. A default E. C. (Ergonomic Criteria) ranking was suggested after analyzing the average B2C e-commerce context of use, in which non professionals users operate sites of virtual stores from theirs 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, for example. The specific ErgoManager's application domain including B2B and ERP user’s profiles, task complexity and equipment configuration had forced a definition of a new E.C. importance ranking. We considered that Work Load criterion is more pertinent than the Guidance criterion in situations where professional users’ work is highly structured and repetitive. Beyond this new ranking, we defined a new relationship between the results of ErgoMonitor and the E.C ranking. In fact, usability problems indicated by ErgoMonitor could be associated with the ranking of E.C., by several ways. In particular we thought of those that are presented in Table 2.

Usability Problem detected by ErgoMonitor	Ergonomic Criteria to examine in priority
- Higher rates of repetitions (see #2.6.4.1)	- Workload
- Higher rates of success with error	- Error protection
- Higher rates of success with help	- Guidance

Table 2- Relationships between ErgoMonitor’s results and ErgoCoin E.C. priority scale.

As consequence of these relationships, the ErgoCoin tool will propose evaluators an E.C. ranking changing command, which will increasing the importance of the criteria associated with usability problems detected by ErgoMonitor. In any case, evaluators will be always authorize to change the ergonomic criteria importance ranking as a way 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 the 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 an **environment to support ergonomic judgments,** in which questions will be followed by information about both usability and the web site context of use.

Finally the system will be supporting evaluation reporting by the means of the **Evaluation report generator** tool. It will be proposing predefined report styles to evaluators who will be in position to produce evaluation reports very fast.

The specification of ErgoCoIn tool, as presented in this report, has not yet been formally validated.

4. CONCLUSION

ErgoManager, the environment for which the specification has been described in this paper, is the result from exploratory studies developed in a INRIA(France)-CNPq (Brazil) project called TVU CE x CI (Techniques de vérification de l'utilisabilité des systèmes interactifs à partir de la confrontation des approches: critères ergonomiques et objets d'interaction) (Cybis et al, 2002). ErgoManager is aimed at supporting 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 an iterative user interface improvement strategy based on verifying the impact that their design decisions have on usability measures. This also means bridging more closely predictive ergonomics (i.e., inspection even before usage) and real usage features (i.e., from actual usage statistics).

The rapid and low cost support from ErgoManager brings important research and development opportunities specifically that related to the construction of a large usability measures database, in which these values will be organized by tasks, user interface design patterns and context of use aspects. Consulting this database, a future usability engineer would be able to appropriately choose and configure a specific user interface design pattern in a way to reach the recommended level of usability for the particular context of use of systems s/he develops.

In order to confirm the possibilities we have been presenting here we are currently attempting to build a first prototype of ErgoMonitor and identifying appropriate web sites for its evaluation.

5. REFERENCES

1. ComScore Networks, 2005. Audience insite Measures , (n.d.). Retrieved February 1st, 2005, from http://www.comscore.com/metrix/aim.asp
2. 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: Proccedings of Brazilian Workshop on Human Factors in Computer Systems. Porto Alegre: Ed. Sociedade Brasileira de Computação, v. I, p. 54-63.
3. Cybis,W., Scapin, D. L. and Morandini, M., Ergomanager : a UIMS for monitoring and revising user interfaces for Web sites, Webist, Miami, USA, May 26-28, 2005.
4. ISO/DIS 9241, 1997. Ergonomic Requirements to Office Work with VDTs, Part 11 - Guidance on Usability.
5. Keynote, 2005. Research Manager Clickstream, http://www.keynote.com/solutions/ solutions_cem_research_mgr_clickstream.html (accessed in 2005)
6. Lalomia, M. J. and SIDOWSKI, J.B. (1990) Measurements of computer satisfaction, literacy, and aptitudes: A review, International Journal of Human-Computer Interaction, vol. 2, p. 231-253, 1990.

7. 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.
8. Macleod M., B., R., Bevan, N., Curson, I. (1997). "The MUSiC performance measurement method." Behaviour and Information Technology 16(4/4): 279-293.
9. MitriDAT, 2005a. Clicks Counter Pro, , (n.d.). Retrieved February 1st, 2005, from http://mitridat.com/products-clicks-counter-pro.html
10. MitriDAT, 2005b. ROI Tracking Pro Pro, , (n.d.). Retrieved February 1st, 2005, from http://mitridat.com/products-roi-tracking-pro.html
11. Morandini, M., Cybis, W.A., 2003. Definition and prototyping of ErgoMonitor. In: Proccedings of International Conference on Human-Computer Interaction, Creta. v.II, p. 1396-1399.
12. Nielsen, J., 1993. Usability Engineering; MK, San Francisco.
13. RFC 2616, Hypertext Transfer Protocol - HTTP/ 1.1: http://www.rfc-editor.org/rfc/rfc2616.txt
14. 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,.
15. 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.
16. Scapin, D. L., Leulier, C., Vanderbonckt, J., Mariage, C., Bastien, C, Palanque, P., Farenc, C., Bastilde, R., 2001., Towards Automated Testing of Web Usability Guidelines, In. Tools for Working with Guidelines London Springer, pp. 293-304.
17. Shneiderman, B. (1998). Designing the user interface: Strategies for effective human-computer interaction (3rd ed.). Reading, MA: Addison-Wesley.
18. Srivastava, J., Cooley, R., Mukund, D., Tan, P.N,. Web Usage Mining: Discovery and applications of usage pattern from Web data. SIGKDD Explorations, v1, n2, p12-23 Jan 2000.
19. WAMMI, 2005. Website Analysis and MeasureMent Inventory (Web Usability Questionnaire), (n.d.). Retrieved February 1st, 2005, from http://www.ucc.ie/hfrg/questionnaires/wammi (accessed in 2005)
20. WebTrends,2005.WebTrends7, (n.d.). Retrieved February 1st, 2005, from http://www.webtrends.com.
21. 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.html#LogFormat.