

Report on Heuristic Evaluation of e-Title prototype

Chandan

Department of Design
IIT Guwahati

Summer Internship: Universitat Pompeu Fabra

Project Guide: Prof. Josep Blat

Interactive Technology Group,
Universitat Pompeu Fabra

Prof. Pradeep Yammiyavar

Department of Design,
IIT Guwahati

Sergio Sayago

Interactive Technology Group,
Universitat Pompeu Fabra

Abstract

This paper provides an overview of well-established user interface evaluation technique: Heuristic Analysis. This background information is used to evaluate the user interface of the e-Title prototype.

Analysis of the eTitle prototype

Heuristic analysis

Heuristic Analysis Methodology

A heuristic analysis is an evaluation which is based upon a set of guidelines. These guidelines are termed 'heuristics'. In a context relevant to user interfaces, these heuristics are a list of interface 'rules', adherence to which has been pre-determined to be indicative of user interface effectiveness.

When performing a heuristic analysis of a user interface, an analyst first selects an appropriate set of heuristics upon which to base the analysis. User interface experts Nielsen and Molich have created a nine point heuristic which has been tested and proven effective for use in user interface analysis.

Once the heuristic has been chosen, the analyst studies an application, taking careful note of any aspects of the interface, which contradict particular guidelines, or of missing interface components which could make the application satisfy the guidelines more readily.

Attention to detail is imperative, as the application must be studied thoroughly if subtle yet important usability considerations are to be noticed. For this reason, Nielsen and Molich (who have formalized a process of effective heuristic analysis) suggest it is desirable for more than one individual to analyze an application. After several independent analyses have been completed, they advise that an interface expert (or alternatively the entire group of independent analysts) combine the results of multiple analyses in order to address the most complete set of interface concerns.

When done with diligent care and attention, heuristic analysis of a user interface can provide detailed and concrete information as to where an application could improve its usability.

The contributions of Nielsen and Molich

Nielsen and Molich developed their generalized nine-point heuristic in 1990. Previous to their research, Nielsen and Molich noted that existing heuristics often contained hundreds or even thousands of guidelines, making heuristic analysis

cumbersome and complicated. Their simplified nine-point heuristic addresses what their years of user interface research identified as critical user interface requirements. Evaluation of an application's user interface with respect to these nine guidelines can capture many frequently occurring interface errors.

To test the effectiveness of a heuristic analysis based upon their nine-point heuristic, Nielsen and Molich set up experiments in which evaluators would use their nine points to conduct a heuristic analysis of a pre-designed system with built in usability concerns. Each test evaluator was able to identify a subset of the interface concerns, but no one analyst was able to identify the complete list. As a result, Nielsen and Molich recommend that a small number of analysts each assess the application independently. These separate assessments can then be combined to address a greater number of concerns. Nielsen and Molich's tests have shown that their nine-point heuristic can be used to effectively identify substantial concerns in most types of screen-based interfaces when a small group (three to five individuals) collates their individual analyses into a comprehensive report.

Merits and shortcomings of heuristic analysis

One of the most attractive aspects of heuristic analysis is that it is relatively easy and inexpensive to carry out. Heuristic analysis can be performed in the early stages of product development so as to greatly reduce the number of interface issues early on in the development process. This of course eliminates the need to go back and change interface elements after large sections of code have already been written. An analysis of interface components can be done even before development has begun, using mockups or interface diagrams, although certainly some usage issues (unreasonable run-time delays unameliorated by warnings or progress indicators, to use Nielsen and Molich's 'MANTEL' example) may not become apparent until the application is in a runnable form.

The process of heuristic analysis is quite convenient in comparison to methods of interface evaluation (observation, interviews, questionnaires) which require costly and time-intensive user participation. Since a team of experts evaluates the interface based on a list of interface rules, adherence to which has been verified to be of benefit to users and enhance ease-of-use in user interfaces, the experts can apply heuristic analysis before requiring user participation, in order to identify fundamental interface flaws.

Of course, the merits of user participatory studies should not be ignored. Even the most careful expert review may overlook certain interface concerns which only a real user who is an actual member of the target user community can identify. However, the fewer heuristically identifiable flaws which remain in an interface if and when user participatory studies are undertaken, the more meaningful and useful eventual actual user feedback will be.

Results obtained from heuristic analysis differ from results obtained from task-oriented evaluation techniques (thinking aloud, cognitive walkthrough, etc.). Heuristic analysis is not a task-oriented evaluation technique, as it evaluates the interface's adherence to a list of rules rather than the interface's tendencies to help or hinder the performance of

required tasks. Task-oriented evaluation techniques yield feedback about the ability to perform specific tasks, which is of great interest, but the holistic nature of heuristic analysis can produce additional interface feedback which is not tied to the performance of specific tasks (feedback on the overall consistency of an application's 'look and feel', for instance can be easily addressed by heuristic analysis, but is not the target focus of a task-oriented evaluation). For this reason, heuristic analysis should be performed in addition to task-oriented evaluation in order to achieve the maximum insight into an application's usability.

A drawback of heuristic analysis is that the results of heuristic analysis provide only a list of mistakes and evaluators' complaints. While this is true, this list of very specific concerns can be used to formulate a definite plan of action. By focusing the analyst's mind on identifying violations of 'rules', heuristic analysis simplifies the discovery process required to identify why exactly portions of an interface may appear confusing. Although admittedly heuristic analysis does not directly propose solutions to identified problems, clarifying the exact nature of each problem is highly beneficial.

Another limitation of heuristic analysis is that the best evaluation is only as good as the set of heuristics used in its generation. For this reason it is crucial to use the most comprehensive and well-tested heuristics available for analysis purposes. This set of heuristics should be as minimal as possible, so as to reduce the evaluator's workload, but must still be detailed enough so as to enable the evaluator to identify critical usability concerns. Nielsen and Molich's heuristics are well respected as valid indicators of usability, but the limitation remains that only interface issues which are addressed by the chosen heuristic can be reported by heuristic analysis.

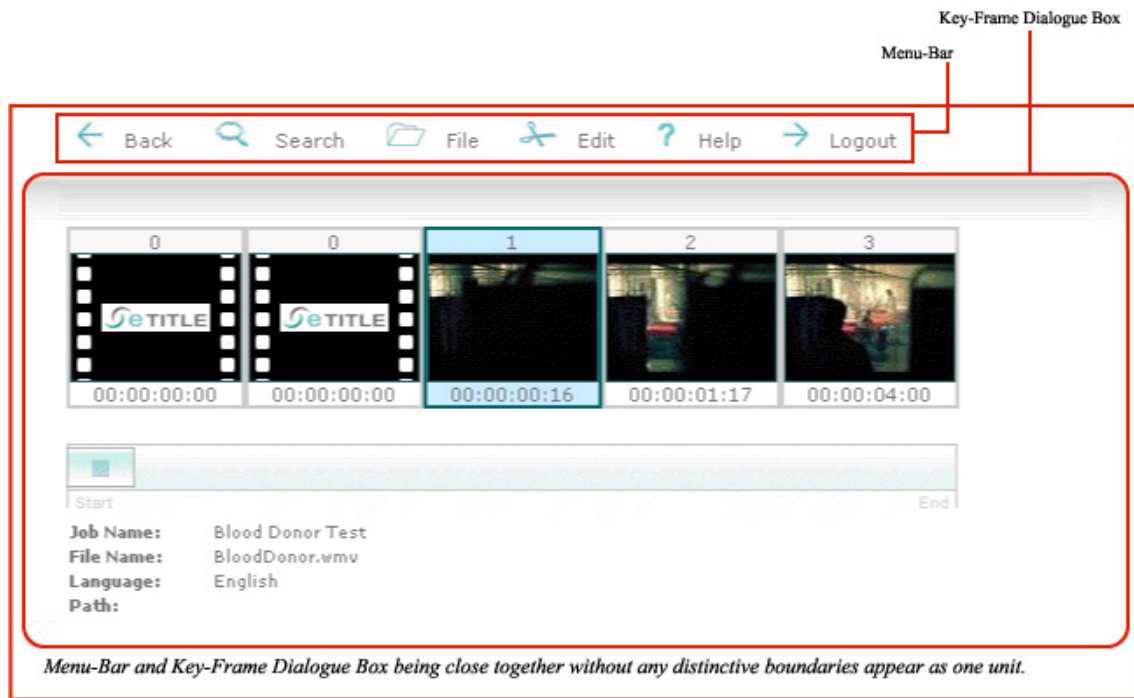
Heuristic analysis of eTitle prototype

To perform this study, the example study described in (a group of individuals was required to use Nielsen and Molich's nine point heuristic to perform a heuristic analysis of a sample system) was used as a model.

Use simple and natural dialogue.

Dialogues should not contain irrelevant or rarely needed information. Every extraneous unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.

- 1) On Screen 4, menu bar is close to key-frame dialogue box and perceived as being grouped with it.



- 2) Graphics: menu bar on all screens (Screen 3,4,5,6) should appear as one distinct group.
- 3) During sign-in on Screen 2, there is no option for users wishing to register/sign up (as new user).

Company Name

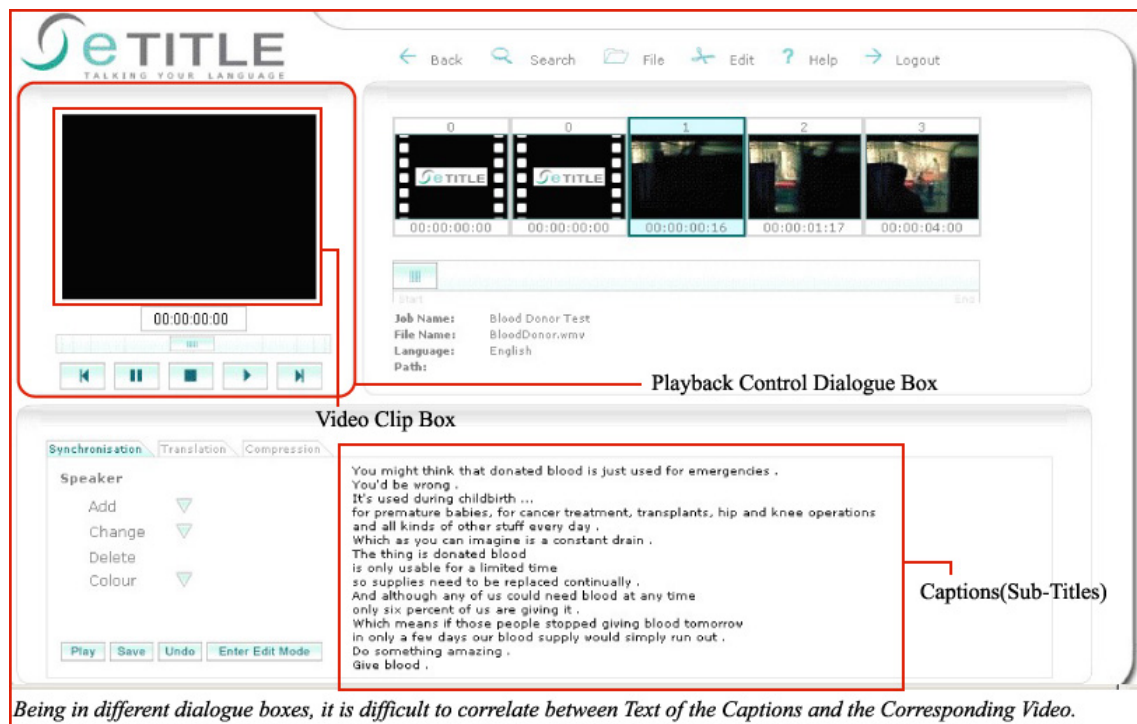
Username

Password

For more details on prices and system specifications please contact: info@etitle.co.uk.

No options for new users to register(or sign in as new user)

- 4) There is no relation displayed between the text and the video clip on screen 4. It is difficult to correlate between the video and the corresponding text in the captions.



- 5) On screen 3, options for Translation and Compression are provided using user-defined Profiles. This multi level navigation is useful for novice users (who can use the default profile: Default_UPF) and also for expert users (to make and use predefined profiles) to save time. This works as a shortcut to avoid the 4th screen for editing.

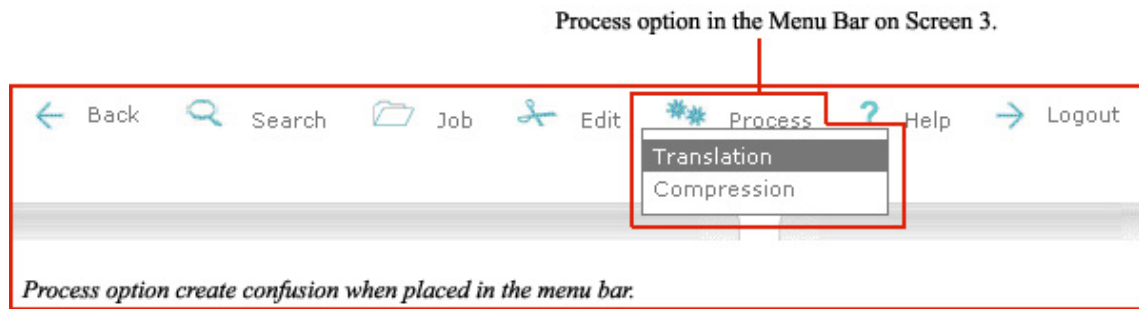
But here Select options are kept on the right end of the Currents Job dialogue-box. And therefore novice users have a tendency to select the job name (by clicking on it) and enter the 4th screen unknowingly which fails the purpose of multi level interaction between user and the system.



- 6) Field area defined for the pull-down labels in menu bars on Screen 3, 4, 5, 6 occupies the area under the adjoining labels.



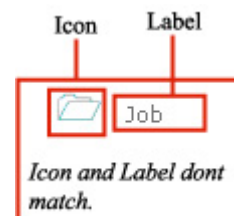
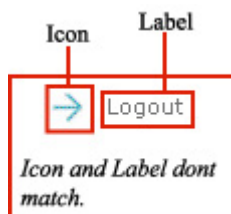
- 7) On Screen 3, Process option does not fit in the menu bar.



Speak the user's language.

The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.

- 1) The system's target users are from different linguistic communities. But there is no option to choose the language of interaction.
- 2) Mapping and Metaphors: Icons for Logout (screen 3,4) and Job (Screen3) don't match with the user expectations and thus are confusing. Icon used for Logout label is more suitable for forward /next. Similarly Icon used for Job is more suitable for File options as per Microsoft Windows User Interface Guidelines.



Minimize the user's memory load.

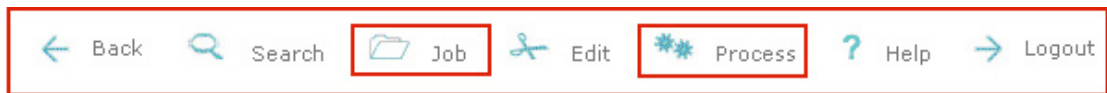
The user's short-term memory is limited. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Complicated instructions should be simplified.

- 1) Information on Screen 3, that the user is going to use most of the times is scattered throughout the layout, forcing user to remember a lot of things. For example, if a user wants to use a profile to translate the clip then he has to take minimum of five steps which is acceptable but yet with a proper layout can be managed in a less number of clear steps (refer Use Simple and Natural Dialogue: fig, point 5).

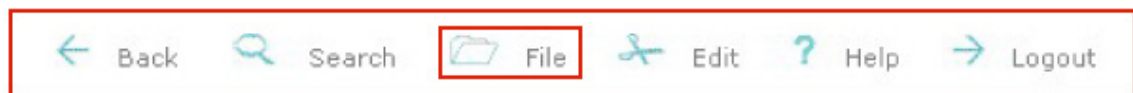
Be consistent.

Users should not have to wonder whether different words, situations, or actions mean the same thing. A particular system action – when appropriate – should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user populations.

- 1) Menu bars on Screen 3, 4, 5, 6 are inconsistent. Screen 3 has got Job and Process labels which are absent in menu bar on Screen 4, 5, 6. Screen 4, 5, 6 have got a File label which is absent in menu bar on Screen 3.



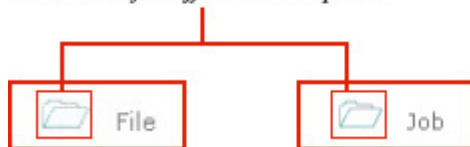
Menu-bar on Screen 3 has Job and Process Options which are absent in the menu-bar on Screen 4.



Menu-bar on Screens 4,5,6 has File Option which is absent in the menu-bar on Screen 3.

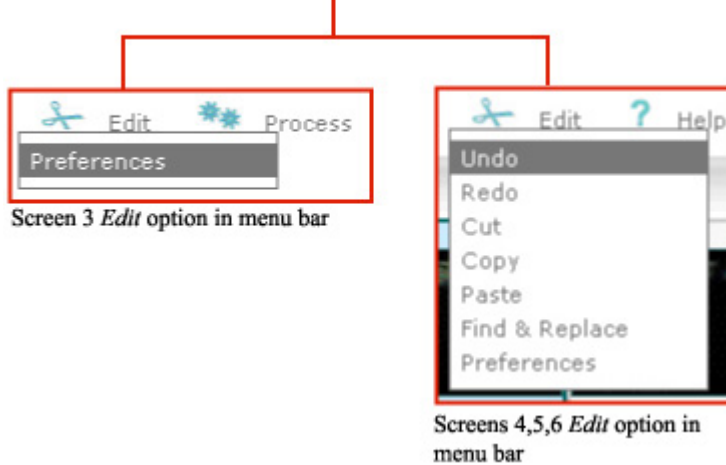
- 2) Same Icons is being used for File label (Screen 4, 5,6) and Job label (Screen3).

Same Icons for different menu options

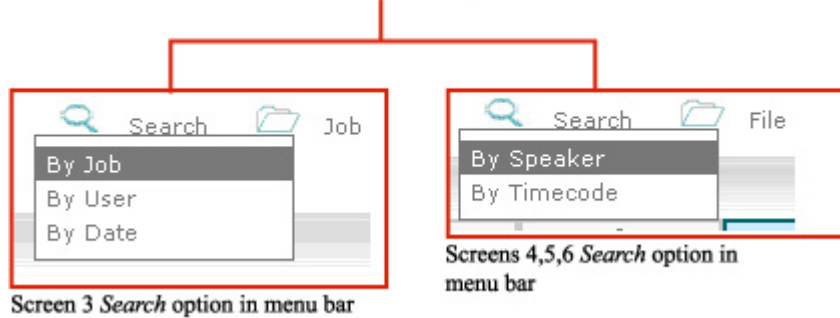


- 3) Same Labels in the menu bar “Edit” and “Search” (on Screens: 3, 4, 5, 6) is being used for different options.

Same "Edit" option in the menu bars have different pull-down menu options



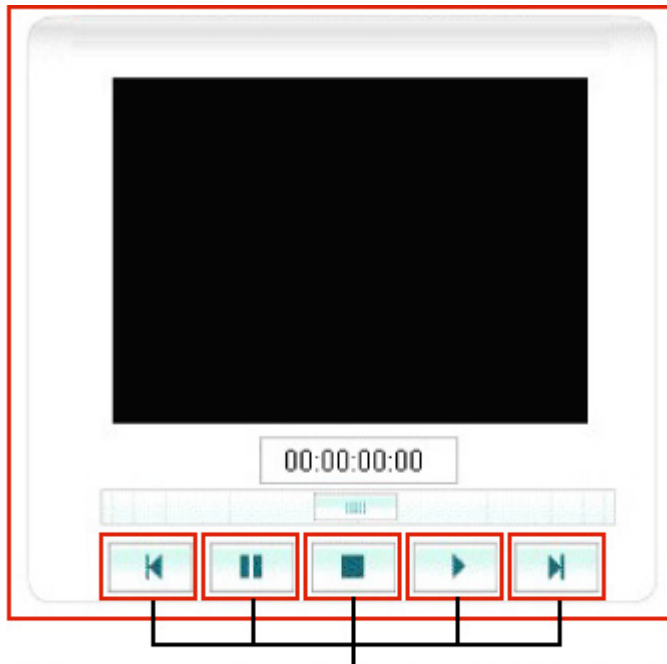
Same "Search" option in the menu bars have different pull-down menu options



Provide feedback to the user.

The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.

- 1) Roll over states for playback control buttons on Screen 4 is not defined. Visual feedback to user action is absent. Also On and Off states are absent.



All playback control buttons (Screen 4: Playback dialogue box) do not have roll-over, on and off states. Also Windows default setting to indicate their function is missing

- 2) The scroll bar over the playback control button on Screen 4: No default Windows feedback is present stating the function of this scroll option. The scroll button also doesn't have a roll over state.

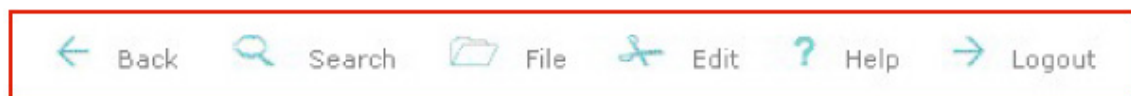


Scroll Bar (Screen 4: Playback dialogue box) does not have roll-over state. Also Windows default setting to indicate its function is missing

- 3) Labels in the menu bars on Screen 3, 4, 5, 6 have got no defined rollover states.

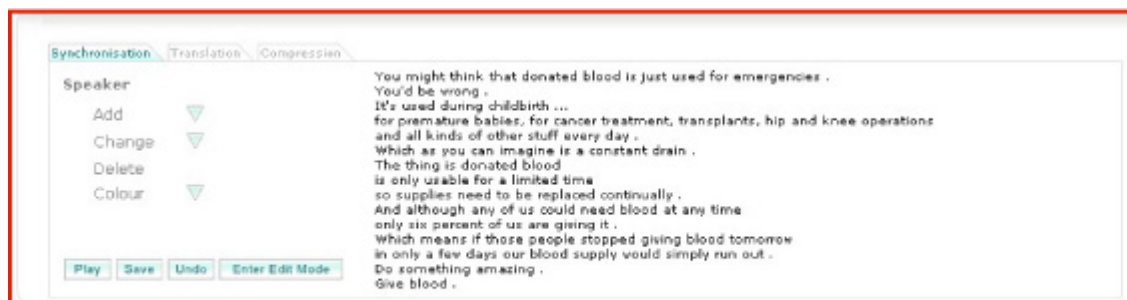


No roll-over states defined for any of the menu options in the menu bar on screen 3.



No Roll-over states defined for any of the menu options in the menu bars on Screens 4, 5, 6.

- 4) All the Buttons in the Synchronization, Translation and Compression dialogue box on Screen 4 have got no roll over state but they do have On and Off states.

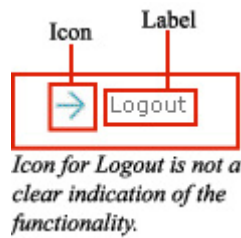


No roll-over states are defined for any of the Buttons in the Synchronization, Translation and Compression dialogue box on Screen 4.

Provide clearly marked exits.

A system should never capture users in situations that have no visible escape. Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue.

- 1) Logout buttons on Screen 3, 4, 5, 6 have got unclear icons which are not matching the label and the user expected metaphors.



Provide shortcuts.

The features that make a system easy to learn – such as verbose dialogues and few entry fields on each display – are often cumbersome to the experienced user. Clever shortcuts – unseen by the novice user – may often be included in a system such that the system caters to both inexperienced and experienced users.

- 1) Through making new profiles, editing existing profiles, saving profiles and using predefined profiles along with translation (fig. 7) and compression (fig. 8) options on screen 3, experienced users can use shortcut to save time. But due to the wrong placement of Select option buttons in the Current Jobs dialogue-box (Screen 3) a novice user could unknowingly go to screen 4 instead of using default profile(the intended setup for novice users).

Provide good error messages.

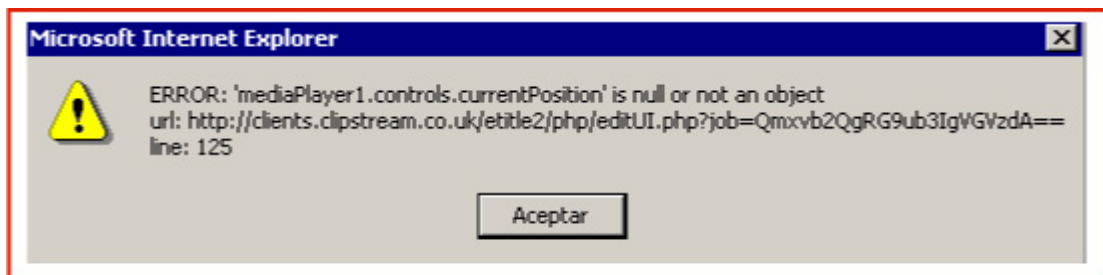
Good error messages are defensive, precise, and constructive. Defensive error messages blame the problem on system deficiencies and never criticize the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.

- 1) On pressing the frame forward button in the playback control option on Screen 4, an error message is generated without specifying the cause and the possible remedy for this error. “Accept” and “Cancel” are the only two options available to users without anymore information in this error message dialogue box.



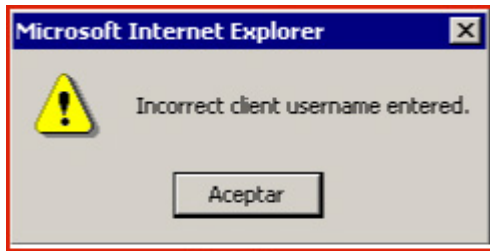
No Information given to user. Only options to choose are Accept and Cancel.

- 2) On pressing back button in menu bar on Screen 4 to go to the previous state, an error message is generated which is not phrased in clear language. There is no feedback stating the reasons of this error and how to avoid it.

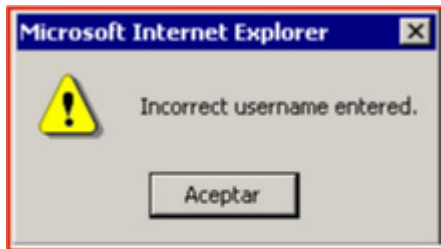


Complex and obscure error message which is too difficult for user to understand. The only possible options are Accept and Cancel.

- 3) Upon signing in with wrong client name, wrong user name or wrong password, the resultant error messages informs that client name, username or password is wrong. But it does not tell the user what will happen if he clicks on the only option available i.e. Accept (apart from default Windows cancel option). Actually Screen 2 goes to its default state on pressing this Accept button on this error message dialogue box without prompting user to re-enter client name, user name and password.



No prompting on what will happen next on pressing accept or cancelling.



No prompting on what will happen next on pressing accept or cancelling.



No prompting on what will happen next on pressing accept or cancelling.

Prevent errors.

Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

- 1) The error messages occurring on Screen 4 as mentioned in the last point (Provide Good Error Messages: 1 and 2) should be avoided as they are because of the systems fault not of users.



Screen 1



Welcome to eTITLE, the World's first intelligent subtitling system.

eTITLE builds on a spectrum of cutting-edge technologies to provide a more cost-effective digital workflow for localising content. Please login below.

Company Name

UPF

Username

UPF

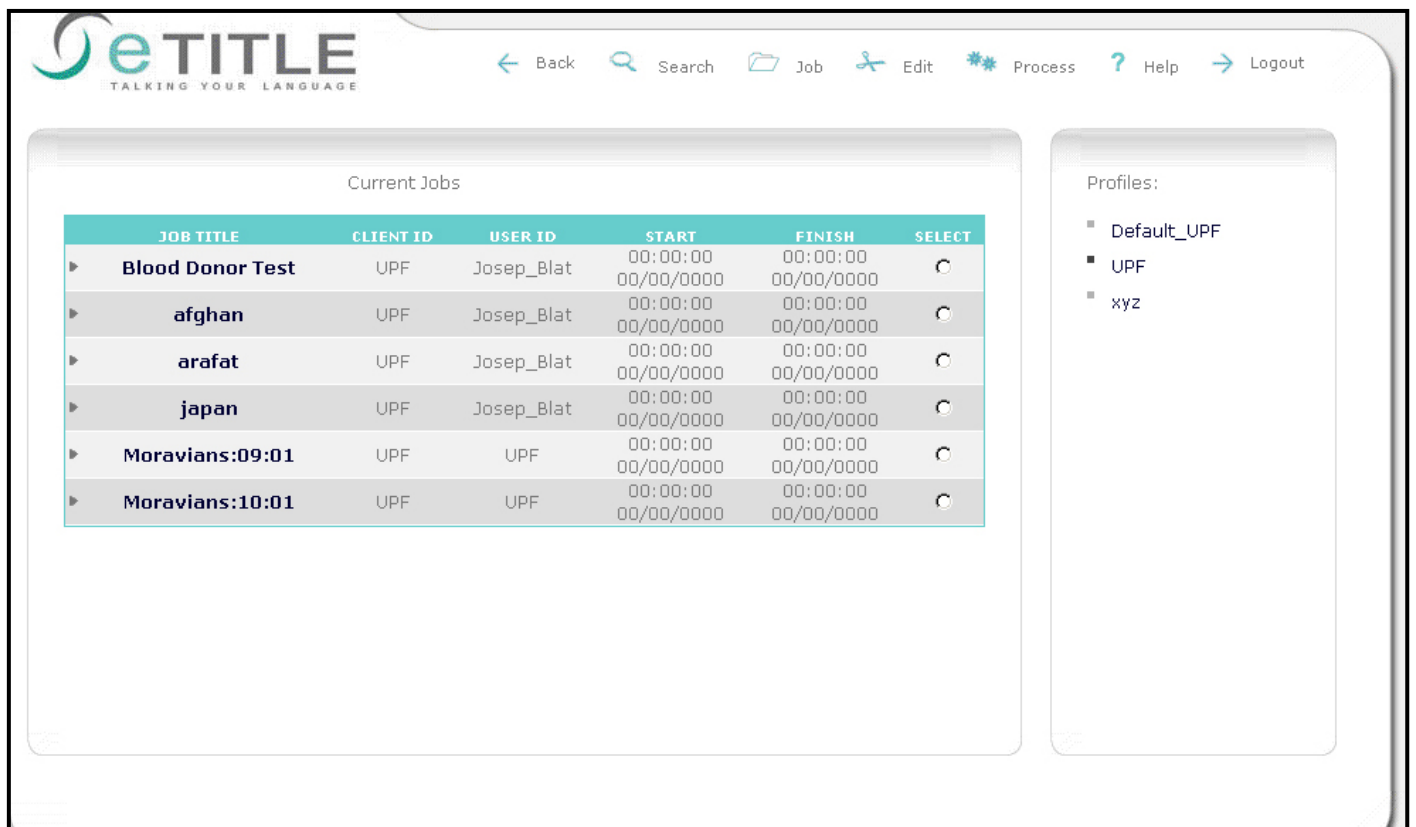
Password

Login >>

TALKING
YOUR
LANGUAGE

For more details on prices and system specifications please contact: info@etitle.co.uk.

Screen 2



Screen 3

Screen 4



[← Back](#)
[Search](#)
[File](#)
[Edit](#)
[? Help](#)
[→ Logout](#)

Please enter new job details.

Job Name

Select Video File

Select Script File

Select Audio File

Start TimeCode

End TimeCode


Select Existing Profile

Default_UPF
UPF
xyz

Create New Profile >>

Create New Job >>

Screen 5



[← Back](#)
[Search](#)
[File](#)
[Edit](#)
[? Help](#)
[→ Logout](#)

Please enter new profile details.

Profile Name

Frame Rate (per second)

Source FPS

Dest. FPS

Genre

Select genre

Word Rate (per minute)

WRPM

Maximum Characters (per screen)

MCPS

Screen Dimensions

Width

Height

Text

Font

Size

Colour

Placement

Alignment

Background

Transparenc

Colour

Background Position

Top Left

px

Top Right

px

Bottom Left

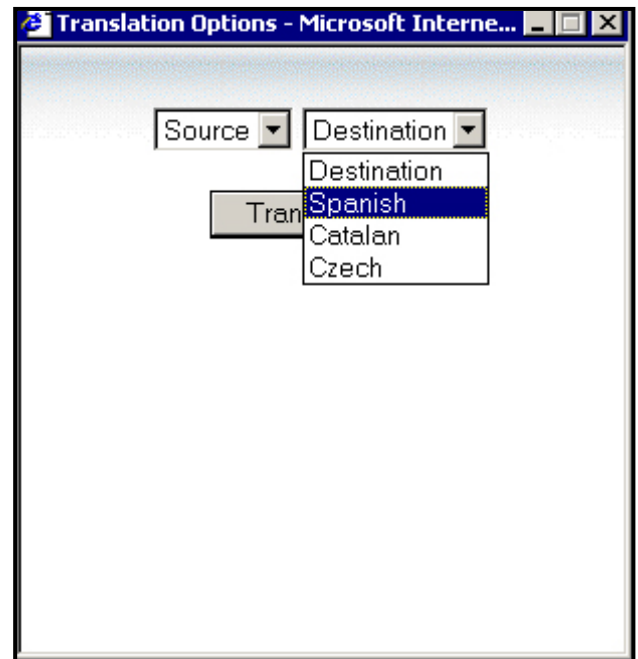
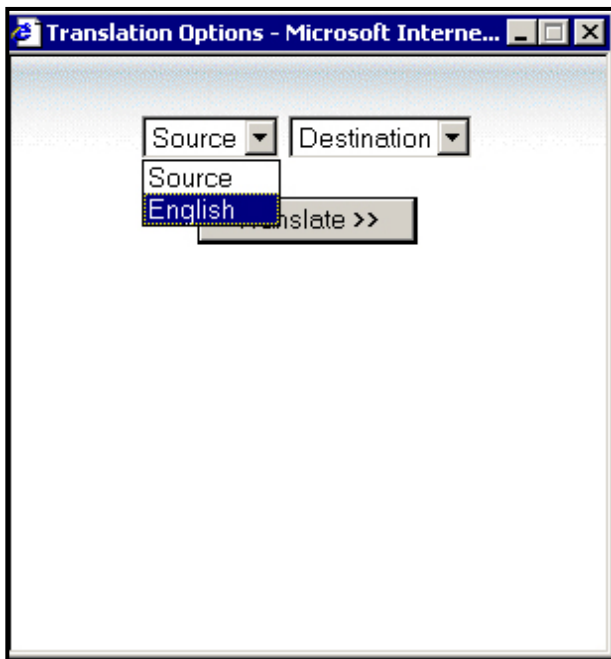
px

Bottom Right

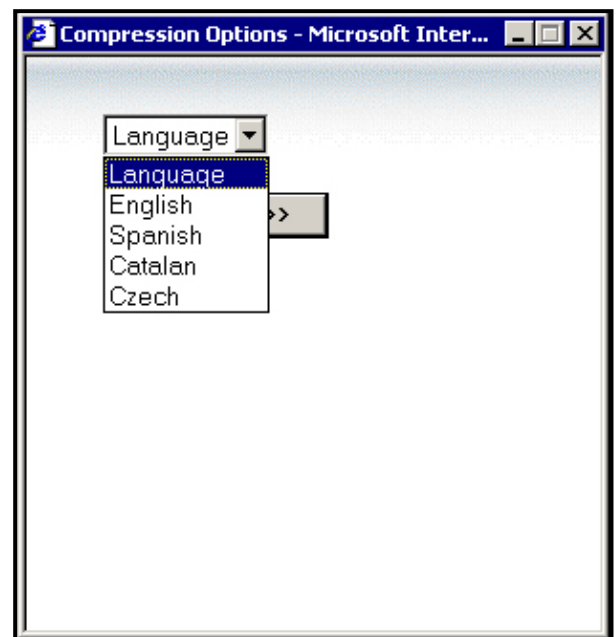
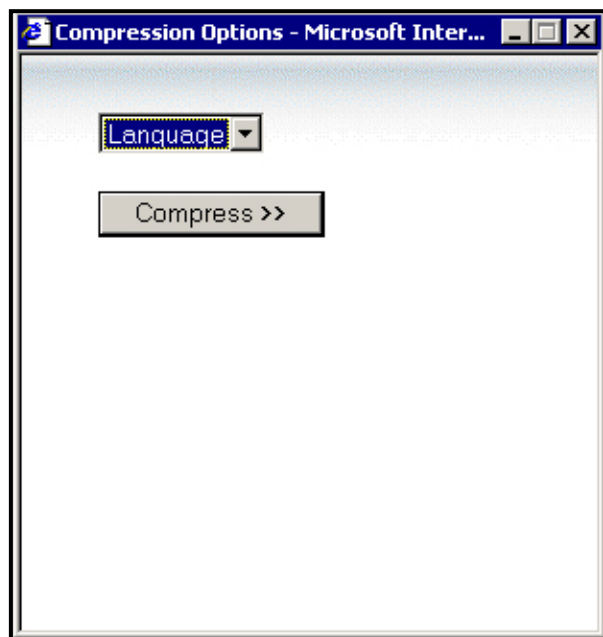
px

Create Profile >>

Screen 6



Translation dialogue-box



Compression dialogue-box

Conclusion:

There are many methods of evaluating the usability of a user interface. Heuristic evaluation using Nielsen and Molich discount usability methods has been done for the eTITLE project in order to identify some specific usability concerns within the interface in a time saving quickest possible way.

The eTITLE system contains a large and impressive amount of functionality, but attempts to evaluate its usability reveal that the complete functionality is not intuitively accessible. While it would easily be possible for a user to memorize the series of steps required in order to perform needed tasks using the system, some adjustments are still required in order for the goal of creating a readily intuitive and user-friendly interactive system.

Systematic performance of well-defined evaluation techniques is valuable in identifying very specific usability concerns within an application. It is hoped that the identification of these specific concerns will guide the direction of further revisions to the eTITLE Prototype in order to enhance the overall usability.

Reference:

- 1) Nielsen, J. (1992c). Finding usability problems through heuristic evaluation. Proc. ACM CHI'92 (Monterey, CA, 3-7 May), 373-380.
- 2) Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. Proc. ACM CHI'94 Conf. (Boston, MA, April 24-28), 152-158.
- 3) Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods. John Wiley & Sons, New York, NY.
- 4) Shneiderman, Ben. Designing the user interface: Strategies for Effective Human-Computer Interaction. 3rd ed. Reading, MA: Addison-Wesley, 1998.
- 5) Nielsen, J. and Molich, R. "Heuristic evaluation of user interfaces." ACM (1990): 249-256.
- 6) Nielsen, J. and Molich, R. "Improving a Human-Computer Dialogue." ACM 33 (March 1990): 338-347.
- 7) Nielsen, J. "Heuristic Evaluation How-to." Accessed April 21, 2003.
<http://www.useit.com/papers/heuristic/heuristic_evaluation.html>.