

# Ergonomic criteria for the evaluation of human-computer interfaces

Christian Bastien, Dominique Scapin

▶ **To cite this version:**

Christian Bastien, Dominique Scapin. Ergonomic criteria for the evaluation of human-computer interfaces. RT-0156, INRIA. 1993, pp.79. inria-00070012

**HAL Id: inria-00070012**
**https://inria.hal.science/inria-00070012**

Submitted on 19 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

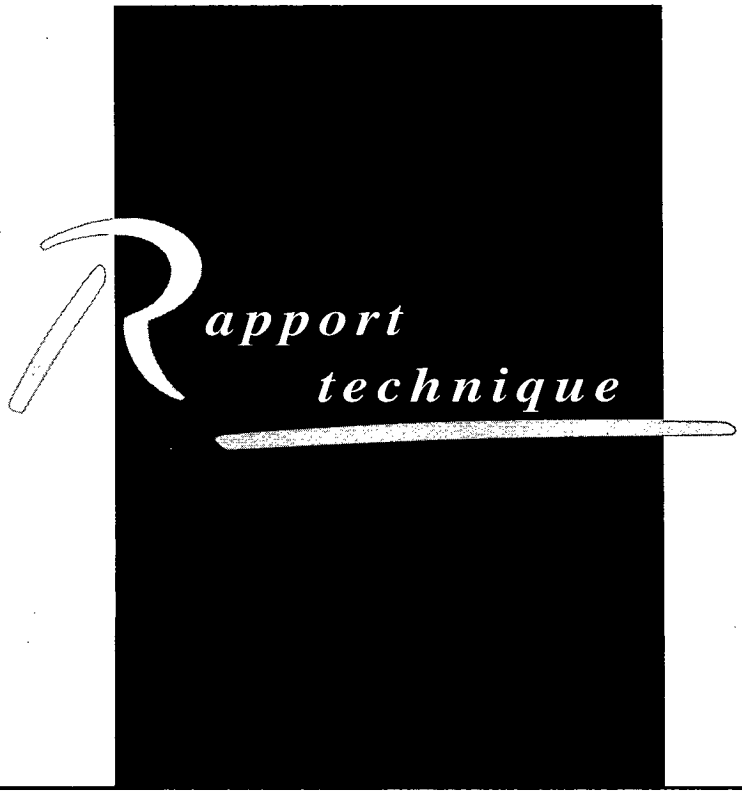
# *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*

J.M. Christian BASTIEN
Dominique L. SCAPIN

**N° 156**
Juin 1993

PROGRAMME 3

Intelligence artificielle,
Systèmes cognitifs et
Interaction homme-machine






# Ergonomic Criteria for the Evaluation of Human-Computer Interfaces

•

# Critères Ergonomiques pour l'Évaluation d'Interfaces Utilisateurs

(version 2.1)

**J. M. Christian Bastien and Dominique L. Scapin**

**May/Mai 1993**

**Abstract**

This technical report presents first a brief summary of the research conducted towards the design of ergonomic criteria for the evaluation of human-computer interfaces (HCI), and then, the full description of the most recent set of criteria (version 2.1) both in English and French. The summary outlines the context in which the criteria were developed, the goal of the criteria approach, the experiments conducted, and the results obtained. The set of ergonomic criteria that resulted from this work consists of a list of 18 elementary criteria (including 9 main criteria). The criteria are presented along with their definitions, rationales, examples of guidelines, and comments setting out the distinctions between some of them.

**Key-words:** Analytic evaluation, User-interface evaluation, Ergonomic criteria, Normative dimensions, Standards.

**Résumé**

Ce rapport technique présente d'une part un bref résumé des travaux menés dans le cadre de la conception de critères ergonomiques pour l'évaluation des interfaces utilisateur (HCI) et d'autre part, la dernière version (version 2.1) du jeu de critères en anglais et en français. Le résumé expose dans ses grandes lignes le contexte dans lequel s'inscrivent les critères, l'objectif de l'approche par critères, les expériences réalisées et les résultats obtenus. Le jeu de critères issu de ces travaux est constitué de 18 critères élémentaires (dont 9 critères principaux). Les critères sont présentés avec leurs définitions, justifications, exemples de recommendations et commentaires permettant d'expliciter les distinctions entre certains d'entre eux.

**Mots-clés:** Évaluation analytique, Évaluation d'interfaces utilisateur, Critères ergonomiques, Dimensions normatives, Standards.

## CONTENTS

Abstract ..... iii

Résumé ..... iii

I. Introduction ..... 1

II. The Ergonomic Criteria (English Version) ..... 7

1. Guidance ..... 9
1.1. Prompting ..... 10
1.2. Grouping/Distinction of Items ..... 13
1.2.1. Grouping/Distinction by Location ..... 14
1.2.2. Grouping/Distinction by Format ..... 15
1.3. Immediate Feedback ..... 16
1.4. Legibility ..... 18

2. Workload ..... 20
2.1. Brevity ..... 21
2.1.1. Concision ..... 22
2.1.2. Minimal Actions ..... 23
2.2. Information Density ..... 25

3. Explicit Control ..... 26
3.1. Explicit User Action ..... 27
3.2. User Control ..... 28

4. Adaptability ..... 29
4.1. Flexibility ..... 30
4.2. User Experience ..... 32

5. Error Management ..... 34
5.1. Error Protection ..... 35
5.2. Quality of Error Messages ..... 36
5.3. Error Correction ..... 37

6. Consistency ..... 38

7. Significance of Codes ..... 39

8. Compatibility ..... 40

III. The Ergonomic Criteria (French Version) ..... 43

1. Guidage ..... 45
1.1. Incitation ..... 46

1.2. Groupement/Distinction entre Items ..... 49
1.2.1. Groupement/Distinction par la Localisation ..... 50
1.2.2. Groupement/Distinction par le Format ..... 51
1.3. Feedback Immédiat ..... 53
1.4. Lisibilité ..... 55

2. Charge de Travail ..... 57
2.1. Brièveté ..... 58
2.1.1. Concision ..... 59
2.1.2. Actions Minimales ..... 61
2.2. Densité Informationnelle ..... 63

3. Contrôle Explicite ..... 64
3.1. Actions Explicites ..... 65
3.2. Contrôle Utilisateur ..... 66

4. Adaptabilité ..... 67
4.1. Flexibilité ..... 68
4.2. Prise en Compte de l'Expérience de l'Utilisateur ..... 70

5. Gestion des Erreurs ..... 72
5.1. Protection Contre les Erreurs ..... 73
5.2. Qualité des Messages d'Erreur ..... 74
5.3. Correction des Erreurs ..... 75

6. Homogénéité/Cohérence ..... 76

7. Signifiance des Codes et Dénominations ..... 77

8. Compatibilité ..... 78

# I. INTRODUCTION

The design of ergonomic criteria is part of a wider research project[1] aiming at the development of methods and tools that would incorporate human factors considerations into the process of designing and evaluating human-computer interfaces (HCI). While other parts of the research work concern task issues, expertise issues, interface modelling issues, etc., the design of criteria is viewed as a means of defining and operationalising dimensions of usability. With respect to the evaluation, the design of criteria represents a way of improving the completeness and explicitness of the diagnosis, of standardising the format of the evaluation, and of better documenting the evaluation. Criteria can also be helpful for teaching HCI issues.

Other requirements for an evaluation method based on ergonomic criteria were that it should:

- be based on analysis of the interface rather than user testing, owing to cost and time constraints;
- be usable also by non-human factors specialists (e.g., interface designers) for whom direct availability of human factors results is important;
- be explicit so as to permit measurements, and sufficiently standardised to be replicated.

Normative dimensions such as criteria can be based on different design strategies. For instance, one can start from available knowledge in domains such as cognition, reasoning, memory, language, etc.; or start from currently published dimensions or heuristics with the goal of organising them into a common structure; or starting from experimental data and recommendations. The strategy chosen here was a bottom-up or inductive strategy (Scapin, 1990b). Available experimental data and recommendations were first translated into rules, then iteratively grouped into sets that were characterised by specific "criteria" which were supposed to best describe the rationale for using such recommendations (initially as a way of accessing a data base). After a few iterations, using also decisions based on group agreement, a set of criteria was established (Scapin, 1990b; 1990c).

The validity and reliability of a tool for the evaluation of HCI are important issues.

The validity of the criteria, i.e. the extent to which they allow one to evaluate the ergonomic quality of an interface comes from the strategy adopted in their design. As indicated above, the set of ergonomic criteria is based on a rather large set of individual recommendations that they characterise and synthesise. In this sense the criteria may be said to be valid.

Although the set of criteria has already been found to allow an accurate description and classification of the usability problems found by experts in an evaluation task (Pollier, 1992), a more direct assessment of its reliability was needed.

---

1 The research project has been summarised in Scapin (1990a). A more detailed presentation of this project can be found in Scapin (in press).

Reliability (or, in this case, evaluators' agreement) can be established by determining the consistency with which the criteria can be used: for instance, if evaluators use the criteria in the same way, i.e. if they uncover the same design flaws, or if they classify them under the same criteria, then the criteria may be said to provide reliable performances. Consistency of use can thus be assessed in different tasks (e.g., classification of design flaws, aid for the evaluation). The factors determining the reliability of the criteria are: the quality of the definitions; the explicitness of the rationale; the relevance of the examples provided with each criterion; and the distinctions given between criteria that are conceptually "close" to one another.

A first assessment of the reliability of the set of criteria was performed in a classification task (Bastien and Scapin, 1992). The rationale of the experiment was that if the criteria were sufficiently well defined and exemplified, then evaluators would be able to identify them correctly, and assign them appropriately to interface problems: in other words the set of criteria would be reliable, and the agreement between would be high. Twenty-four subjects (12 human factors specialists and 12 non-specialists) were asked to identify which criterion was violated for each of thirty-six usability problems they were presented with. The results of this study showed no statistical difference between groups both in terms of performance times and in terms of correct identifications. The mean percentage of correct identifications was 63,7% and 56% for the human factors specialists and the non-specialists respectively. A detailed examination of the data and an analysis of confusion matrices allowed the identification of categories of well defined criteria and categories of criteria that were to benefit from improvements in their definition. Even though the experimental situation was very restrictive, the overall assignment performance was above average. The improvement of the set of ergonomic criteria proceeded through the inclusion of new examples, the addition of comments allowing a better distinction between criteria, and the refinement of some definitions.

The effectiveness of ergonomic criteria as a tool or a guide for the evaluation of user interfaces was assessed in another experiment (Bastien and Scapin, 1993). Two groups of human factors specialists evaluated the interface of a musical database application. After an exploration-diagnosis phase in which all the evaluators' actions and comments were recorded along with the corresponding states of the application, the participants re-evaluated the same interface states through the replay of their first exploration. This second evaluation was conducted with or without ergonomic criteria depending on the group. Preliminary results show that in the first phase, the number of usability problems uncovered and the proportions of usability problems found with respect to the size of the aggregates of evaluations were similar for both groups. In the second phase of the experiment, the use of criteria increased both the evaluation diagnosis and the proportions of problems with respect to the size of the aggregates. The criteria were thus helpful for the diagnosis of design flaws : they are helpful in organising the evaluation, in increasing its completeness and in decreasing the number of experts necessary to get a given proportion of design flaws.

We also believe (but it remains to be tested) that even non-usability specialists should improve significantly the quality of the design and of the evaluation of HCI by using the set of ergonomic criteria.

The ergonomic criteria may be considered to belong to the category of "discount methods". They constitute a support for fast evaluation of the most well-known problems and reduce the evaluation costs: first of all, the "a priori" analytic approach used with the criteria is less costly than the running of users tests (which are know to be time consuming and resources demanding)[2], especially for a first global evaluation; secondly, the cost reduction results also from the reduction of the number of usability specialists necessary to perform an evaluation.

To sum up, interesting features of the criteria appear to be: their completeness, their independence/distinctiveness, and their overall applicability.

*Completeness* - The very large set of recommendations that gave rise to the criteria hopefully guarantees to some degree their completeness. In other words the criteria may be considered to synthesise most of the recommendations currently available in the domain of user interface software design. Moreover, through the various experiments, no other criterion was ever suggested by the participants: all the problems uncovered could be classified under one of the various criteria.

*Independence/distinctiveness* - As described before, a fairly good assignment performance was obtained in the classification experiment: there was usually a univocal classification which supports independence of the criteria. In cases where classification errors occurred, i.e. confusions between criteria, distinctions between them were refined. Although the refinement of the definitions as well as of the examples and comments is still in progress, the present criteria may be said to be independent and distinct. This state of affairs allows us to characterise each of our database recommendations by a single criterion.

*Overall applicability* - In the above experiments, all criteria were used and none was ever rejected as inappropriate or questionable, or context dependent.

## Limitations and cautions

For now, and even though the present set of criteria may be said to be useful, to be usable to some extent, to guide the evaluation, to be relatively reliable and valid, it does not yet represent a fully developed evaluation method: it does not for example specify ways to explore the interface nor does it operationalise each criterion.

A criteria-based evaluation method constitutes an analytic approach. As such, the criteria are not intended to replace other evaluation methods (e.g., "model-based" methods, questionnaire, interview, etc.). For instance, user testing may remain necessary, especially for complex problems, or for new problems, i.e. problems or questions for which no recommendations exist. The criteria should rather be viewed as a supplement to other evaluation methods. The criteria approach may be a means for ensuring compliance with software design guidelines, and as such, it could be used before user testing to uncover and to fix existing design flaws.

---

2 Of course, user performance testing is necessary and useful (even in cost/benefit terms) especially for answering precise evaluation questions, rather than for a first general scan at usability problems.

## Future work

The set as such has been used as a grid for the diagnosis and the reporting of interface evaluations. However, future work should develop the present set of criteria into a full evaluation method. To this end, work is being carried out in the following directions:

- extending the content of each criterion; completing the level of detail of the criteria i.e., including a full set of specific "rules" for each of the criteria; identifying ways of supporting each of the criteria;

- defining a set of priorities for the evaluation (which criterion first, which next, etc.). Such priorities should be defined as a function of the evaluation goals (e.g., *Guidance* should have priority over *Flexibility* for inexperienced users; if high performance is requested, then *Work load* should have a high priority, etc.);

- defining the pre-requisites to the evaluation, i.e., all the task and user related characteristics that are needed to apply the criteria. At this point, links should be identified with formal task analysis methods;

- defining ways to examine systematically the interface states and elements (screen, windows, sequences of inputs, etc.). This requires appropriate models of the interface and of its behaviour on which to apply the criteria;

- selecting tools supporting a full evaluation procedure (hypertext, expert-system, etc.).

In addition, the reliability of further developments of the method will have to be assessed with real users (designers and end-users). The use of the method and its influence (cost/diagnosis, etc.) on the software produced will also need to be evaluated. The criteria approach should also be compared with other evaluation methods to assess their relative efficiency, but also to identify their relative contributions to the interface evaluation process.

## The set of ergonomic criteria

The set of criteria presented in the following sections consists of eight main criteria, some of which are subdivided into sub-criteria. The set is thus composed of 18 elementary criteria, i.e. criteria that are not divided into sub-criteria. In the text these criteria are indicated by an asterisk.

Each elementary criterion is presented with a definition, a rationale, examples of guidelines, and comments:

- *Definitions:* In defining the criteria, precisions that were made, and conventions that were adopted were based on experimental results. All definitions were written so as to allow as great a distinction as possible to be made between criteria.

- *Rationales:* The rationales explain the basis for using the criteria, basically their influence on users' performances.
- *Examples of guidelines:* A small set of examples, i.e., specific guidelines, has been selected for each criterion.
- *Comments:* Comments were added to help distinguish the criteria that have close conceptual relationships, by making their differences more explicit.

## REFERENCES

Bastien, J. M. C., & Scapin, D. L. (1992). A validation of ergonomic criteria for the evaluation of human-computer interfaces. *International Journal of Human-Computer Interaction, 4(2),* 183-196.

Bastien, J. M. C., & Scapin, D. L. (1993, April). *Preliminary findings on the effectiveness of ergonomic criteria for the evaluation of human-computer interfaces.* Short paper presented at the Conference on Human Factors in Computing Systems, INTERCHI '93, Amsterdam, The Netherlands.

Pollier, A. (1992). Évaluation d'une interface par des ergonomes: Diagnostics et stratégies [User interface evaluation by human factors specialists : diagnoses and strategies]. *Le Travail Humain, 55(1),* 71-96.

Scapin, D. L. (1990a). Aiding mechanisms for the design of user interfaces: some research issues, *First International Conference on Automation Technology,* July 4-6, 1990, Taipei, R.O.C.

Scapin, D. L. (1990b). Des criteres ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs [Ergonomic criteria for the evaluation and design of user interfaces]. In *Actes du XXVI Congrès de la SELF.* Montréal, Canada: Institut de Recherche en Santé et en Sécurité du Travail du Québec.

Scapin, D. L. (1990c). Organizing human factors knowledge for the evaluation and design of interfaces. *International Journal of Human-Computer Interaction, 2(3),* 203-229.

Scapin, D. L. (in press). Situation et perspectives en ergonomie du logiciel [Situations and perspectives in software ergonomics]. In J. C. Sperandio (Ed.), *L'ergonomie dans la conception des projets informatiques.* Paris : Octarès.

# II. The Ergonomic Criteria (English Version)

# 1. GUIDANCE

*Definition :*

User *Guidance* refers to the means available to advise, orient, inform, instruct, and guide the users throughout their interactions with a computer (messages, alarms, labels, etc.), including from a lexical point of view. The criterion *Guidance* is subdivided into four criteria: *Prompting*, *Grouping/Distinction of Items*, *Immediate Feedback*, and *Legibility*.

*Rationale(s) :*

Good guidance facilitates learning and use of a system by allowing the users: to know at any time where they are in a sequence of interactions, or in the accomplishment of a task; to know what the possible actions are as well as their consequences; and to obtain additional information (possibly on demand). Ease of learning and ease of use that follows good guidance lead to better performances and fewer errors.

## 1.1. PROMPTING *

*Definition :*

As it is used here, *Prompting* has a broader definition than usual. Here it refers to the means available in order to lead the users to making specific actions whether it be data entry or other tasks. This criterion also refers to all the means that help users to know the alternatives when several actions are possible depending on the contexts. *Prompting* also concerns status information, that is information about the actual state or context of the system, as well as information concerning help facilities and their accessibility.

*Rationale(s) :*

Good prompting guides the users and saves them, for instance, from learning a series of commands. Good prompting also allows users to know exactly the current mode, where they are in the dialogue, as well as the actions that resulted in that context. Thus good prompting helps users to navigate in the application or the system and helps to reduce errors.

*Examples of guidelines:*

- For data entry, provide the user with the required formats and acceptable values ; e.g., include in a field label additional cueing of data format (e.g., Date (mm/dd/yy) : _ _ / _ _ / _ _ ).
- Display measurement units for data entry.
- Indicate all status information (e.g., modes, values, etc.).
- For each data field, display an associated label.
- Provide cues on the acceptable length of entries.
- Provide a title for each window.
- Provide on-line help and guidance.

*Comments :*

• ***Prompting vs Grouping/Distinction by Format.***

*Prompting* helps in a direct manner by suggesting to the users possible actions that may be taken. For instance, when several actions are possible (e.g., confirm, cancel, copy, etc.), only the available actions could be displayed, or indicated in a message.
Less direct means are also available to guide the users. In the above example, if all options are presented, different formats could be used to indicate those which are currently available.
When several choices or actions are possible, the most probable or most logical action can be presented in a format that distinguishes it from other options, such as a bold border encircling a push button (the default option).
It is a case of *Grouping/Distinction by Format*, not a case of *Prompting* since the carriage-return option is not explicit.

---

* Elementary criterion

*Prompting* consists in directly presenting the different possible actions whereas *Grouping/Distinction by Format* will indicate particular characteristics of these possible actions.

• ***Prompting vs Immediate Feedback***

The distinction between *Prompting* and *Immediate Feedback* is sometimes subtle. For example, following a user action, a message can indicate that the command has been executed and also which action must be taken to continue the transaction. Thus a message can be a feedback for the previous action(s) and a prompting for the subsequent action. *Immediate Feedback* concerns messages that inform the user about the previous actions. *Prompting* concerns messages that either guide the users for future actions or suggest performing a specific action.

• ***Prompting vs Legibility***

The lexical aspects of the prompts (instructions, messages indicating to the users what needs to be done, etc.) which can have an effect on the readability (e.g., font size, type of font, etc.), concern the criterion *Legibility*.

• ***Prompting vs Concision***

Good prompting, such as instructions, messages indicating to the user what needs to be done may be too long. This flaw in the prompting concerns the criterion *Concision*. In the same way, good prompting can indicate the way data must be entered, but at the same time a problem of *Concision* may be introduced if, in addition to the data, the user must enter measurement units.

• ***Prompting vs Error Protection***

Error protection can be achieved by various means. Automatic mechanisms can be implemented that control the users' inputs. For instance, after users' entry, the computer will display an error message if the data entry format is not appropriate. This case would concern the criterion *Error Protection*. Another way of preventing errors would be to give users information on the type and format of the data to be entered. In this case *Prompting* is concerned. These two means of preventing errors can coexist.

• ***Prompting vs Quality of Error Messages***

An error message can include prompting about the means to correct errors. It is indeed a case of *Quality of Error Messages*, not a case of *Prompting*. *Prompting* only refers to the guidance provided to the users in error-free situations.

• ***Prompting vs Consistency***

As long as it is a matter of comparison between several transactions, objects, etc., including the ones with a prompting-oriented goal, it concerns the criterion *Consistency*. For instance, providing a single identification for each

display in a consistent location at the top of the display frame concerns *Consistency*.

- ***Prompting vs Compatibility***

When the wording used for prompting differs from the users' language, it concerns the criterion *Compatibility*.

## 1.2. GROUPING/DISTINCTION OF ITEMS

*Definition :*

The criterion *Grouping/Distinction of Items* concerns the visual organisation of information items in relation to one another. This criterion takes into account the topology (location) and some graphical characteristics (format) in order to indicate the relationships between the various items displayed, to indicate whether or not they belong to a given class, or else to indicate differences between classes. This criterion also concerns the organisation of items within a class.

The criterion *Grouping/Distinction of Items* is subdivided into two criteria: *Grouping/Distinction by Location* and *Grouping/Distinction by Format.*

*Rationale(s) :*

The users' understanding of a display screen depends, among other things, on the ordering, the positioning and the distinction of the objects (images, texts, commands, etc.) that are presented. Users will detect the different items or groups of items, and learn their relationship(s) more easily if, on the one hand, they are presented in an organised manner (e.g., alphabetic ordering, frequency of use, etc.), and if on the other hand the items or groups of items are presented in formats or coded in ways that indicate their similarities or differences. Also, the learning and remembering of items or groups of items will be improved. The grouping/distinction of items leads to a better guidance.

## 1.2.1. GROUPING/DISTINCTION BY LOCATION *

*Definition :*

The criterion *Grouping/Distinction by Location* concerns the relative positioning of items in order to indicate whether or not they belong to a given class, or else to indicate differences between classes. This criterion also concerns the relative positioning of items within a class.

*Rationale(s) :*

The users' understanding of a display screen depends, among other things, on the ordering of the objects (images, texts, commands, etc.) that are presented. Users will detect the different items more easily if they are presented in an organised manner (e.g., alphabetic ordering, frequency of use, etc.). Also, the learning and remembering of items will be improved. The grouping/distinction by location leads to a better guidance.

*Examples of guidelines:*

- Organise items in hierarchical lists.
- Organise the options of a menu dialogue as a function of the objects to which they apply.
- When several options are presented, their organisation must be logical, i.e., the organisation must represent a significant or relevant functional organisation (alphabetic order, functional, frequency of use, etc.).

*Comments :*

- ***Grouping/Distinction by Location vs Grouping/Distinction by Format***

The grouping/distinction of items can be achieved by format and/or by location. Location and format correspond to different display features (topology vs. added graphics). For instance, menu options can be distinguished either or both with location (e.g., most frequent options at the top, less frequent options further down) and format (e.g., a line separator between a set of options concerning text layout and a set of options concerning character types).

* Elementary criterion

## 1.2.2. GROUPING/DISTINCTION BY FORMAT *

*Definition :*

The criterion *Grouping/Distinction by Format* concerns more precisely graphical features (format, colour, etc.) that indicate whether or not items belong to a given class, or that indicate distinctions between different classes, or else distinctions between items of a given class.

*Rationale(s) :*

It will be easier for the user to know the relationship(s) between items or classes of items if different formats or different codes illustrate their similarities or differences. Such relationships will be easier to learn and to remember. A good grouping/distinction by format leads to a better guidance.

*Examples of guidelines:*

- Provide clear visual distinction of areas having different functions (command zone, message zone, etc.).
- Provide clear visual distinction of data fields and their labels.

*Comments :*

• ***Grouping/Distinction by Format vs Prompting***

*Prompting* helps in a direct manner by suggesting possible actions. For instance, when several actions are possible (e.g., confirm, cancel, copy, etc.), only the available actions could be displayed, or indicated in a message. Less direct means are also available to guide the users. In the above example, if all options are presented, different formats could be used to indicate those which are currently available.
When several choices or actions are possible, the most probable or most logical action can be presented in a format that distinguishes it from other options, such as a bold border encircling a push button (the default option). It is a case of *Grouping/Distinction by Format*, not a case of *Prompting* since the carriage-return option is not explicit.
*Prompting* consists in directly presenting the different possible actions whereas *Grouping/Distinction by Format* will indicate particular characteristics of these possible actions.

• ***Grouping/Distinction by Format vs Grouping/Distinction by Location***

The grouping/distinction of items can be achieved by format and/or by location. Location and format correspond to different display features (topology vs. added graphics). For instance, menu options can be distinguished either or both with location (e.g., most frequent options at the top, less frequent options further down) and format (e.g., a line separator between a set of options concerning text layout and a set of options concerning character types).

---

* Elementary criterion

## 1.3. IMMEDIATE FEEDBACK *

*Definition :*

*Immediate Feedback* concerns system responses to users' actions. These actions may be simple keyed entries or more complex transactions such as stacked commands. In all cases computer responses must be provided, they should be fast, with appropriate and consistent timing for different types of transactions. In all cases, a fast response from the computer should be provided with information on the requested transaction and its result.

*Rationale(s) :*

Feedback quality and rapidity are two important factors for the establishment of user confidence and satisfaction as well as for the understanding of the dialogue. These factors allow the users to gain a better understanding of the system's functioning.
The absence of feedback or a delayed feedback can be disconcerting to the user. The users may suspect a system failure and may undertake actions that may be disruptive for the ongoing processes.

*Examples of guidelines:*

- All users' entries should be displayed except for secure entries. Even in this case, every keyed entry should produce a perceptible feedback (e.g., symbols such as stars).
- Following user interruption of data processing, display an advisory message assuring the user that the system has returned to its previous status.
- When computer processing is lengthy, information concerning the state of the processing should be provided to the user.

*Comments :*

• ***Immediate Feedback vs Prompting***

The distinction between *Prompting* and *Immediate Feedback* is sometimes subtle. For example, following a user action, a message can indicate that the command has been executed and also which action must be taken to continue the transaction. Thus a message can be a feedback for the previous action(s) and a prompting for the subsequent action. *Immediate Feedback* concerns messages that inform the user about the previous actions. *Prompting* concerns messages that either guide the users for future actions or suggest performing a specific action.

• ***Immediate Feedback vs Legibility***

When the quality of feedback is unsatisfactory, even from a lexical point of view, it is a matter of *Immediate Feedback*, not a matter of legibility. *Immediate Feedback* concerns all characteristics of the system's responses following

* Elementary criterion

users' actions (presence or absence of a feedback, quality of the feedback from a semantic and lexical point of view).

## 1.4. LEGIBILITY *

*Definition :*

*Legibility* concerns the lexical characteristics of the information presented on the screen that may hamper or facilitate the reading of this information (character brightness, contrast between the letter and the background, font size, interword spacing, line spacing, paragraphs spacing, line length, etc.). By definition, the criterion *Legibility* does not concern feedback or error messages.

*Rationale(s) :*

Performances are increased when the presentation of information on the screen takes into account the cognitive and perceptual characteristics of the users. A good legibility facilitates the reading of the information presented. For instance, dark letters on a light background are easier to read than the other way around; a text presented with mixed upper/lower case is read faster than a text presented with only upper-case letters.

*Examples of guidelines:*

- Titles should be centred.
- Labels should be displayed in upper case letters.
- Cursors should be distinguished from other displayed items.
- When space for text display is limited, display a few long lines of text rather than many short lines of text.
- Display continuous text in wide columns, containing at least 50 characters per line.
- Right justification should be employed if it can be achieved by variable spacing, maintaining constant proportional differences in spacing between and within words, and consistent spacing between words in a line.
- In display of textual material, keep words intact, with minimal breaking by hyphenation between lines.

*Comments :*

***Legibility* does not apply to feedback or error messages.**
Indeed, all aspects related to reading difficulties for the users, or more generally to the quality of messages concerning the feedback or the error messages should be assigned respectively to the criterion *Immediate Feedback* or to the criterion *Quality of Error Messages*.

• ***Legibility vs Immediate Feedback***

When the quality of feedback is unsatisfactory, even from a lexical point of view, it is a matter of *Immediate Feedback*, not a matter of legibility. *Immediate Feedback* concerns all characteristics of the system's responses following users' actions (presence or absence of a feedback, quality of the feedback from a semantic and lexical point of view).

* Elementary criterion

- ***Legibility vs Quality of Error Messages***

  When error messages are unsatisfactory, even from a lexical point of view, it is a matter of *Quality of Error Messages*, not a matter of *Legibility*. The criterion *Quality of Error Messages* concerns all characteristics of information related to the users' errors.

- ***Legibility vs Prompting***

The lexical aspects of the prompts (instructions, messages indicating to the users what needs to be done, etc.) which can have an effect on the ease of reading (e.g., font size, type of font, etc.), concern the criterion *Legibility*.

- ***Legibility vs Significance of Codes***

*Legibility* does not concern the semantic characteristics of information, their relevance or their significance. These characteristics are to be related to *Significance of Codes*.

- ***Legibility vs Compatibility***

The criterion *Legibility* does not concern the semantic characteristics of information, their relevance or their significance. When these characteristics are related to the task, then the criterion *Compatibility* should be considered.

# 2. WORKLOAD

*Definition :*

The criterion *Workload* concerns all interface elements that play a role in the reduction of the users' perceptual or cognitive load, and in the increase of the dialogue efficiency.
The criterion *Workload* is subdivided into two criteria: *Brevity* (which includes *Concision* and *Minimal Actions*), and *Information Density*.

*Rationale(s) :*

The higher the workload, the higher the probability of making errors. Also the less users are distracted by unnecessary information, the more they will be able to accomplish their task efficiently. Besides, the shorter the required actions, the faster the interactions.

## 2.1. BREVITY

*Definition :*

The criterion *Brevity* concerns the perceptual and cognitive workload both for individual inputs and outputs, and for sets of inputs (i.e., sets of actions needed to accomplish a goal or a task). *Brevity* corresponds to the goal of limiting the reading and input workload and the number of action steps. The criterion *Brevity* is subdivided into two criteria: *Concision* and *Minimal Actions*.

*Rationale(s) :*

Short term memory capacity is limited. Consequently, the shorter the entries, the smaller the probability of making errors. Besides, the more succinct the items, the shorter the reading times.

Also, the more numerous and complex the actions necessary to reach a goal, the more the workload will increase and consequently the greater the risk of making errors.

## 2.1.1. CONCISION *

*Definition :*

The criterion *Concision* concerns perceptual and cognitive workload for individual inputs or outputs.
By definition, *Concision* does not concern feedback or error messages.

*Rationale(s) :*

Short term memory capacity is limited. Consequently, the shorter the entries, the smaller the probability of making errors. Besides, the more succinct the items, the shorter the reading times.

*Examples of guidelines:*

- For numerical data, entry of leading zeros should not be necessary.
- If codes are longer than 4 or 5 characters, use mnemonics or abbreviations.
- Allow users short data entries.
- When a measurement unit is associated with a particular data field, include that unit as part of the field label rather than requiring a user to enter it.

*Comments :*

- ***Concision vs Prompting***

Good prompting, such as instructions, messages indicating to the user what needs to be done may be too long. This flaw in the prompting concerns the criterion *Concision*. In the same way, good prompting can indicate the way data must be entered, but at the same time a problem of *Concision* may be introduced if, in addition to the data, the user must enter measurement units.

- ***Concision vs Minimal Actions***

By *Minimal Actions* we refer here to procedures, or steps; when the length of items or data entries is concerned, then it is a matter of *Concision*.

- ***Concision vs Information Density***

The criterion *Concision* concerns the succinctness of individual items presented while *Information Density* concerns the density of the set(s) of information presented on the screen. Thus an item can be relevant but not presented in a sufficiently concise way. In this case the criterion *Concision* must be put forward. If items are superfluous then it is a matter of *Information Density*.

- ***Concision vs Quality of Error Messages***

The criterion *Concision* does not apply to error messages. When error messages are not sufficiently succinct, it is a matter of *Quality of Error Messages*.

---

* Elementary criterion

## 2.1.2. MINIMAL ACTIONS *

*Definition :*

The criterion *Minimal Actions* concern workload with respect to the number of actions necessary to accomplish a goal or a task. It is here a matter of limiting as much as possible the steps users must go through.

*Rationale(s) :*

The more numerous and complex the actions necessary to reach a goal, the more the workload will increase and consequently the more probable the risks of making errors.

*Examples of guidelines:*

- Minimise the number of steps required to make a selection in a menu.
- Do not require data entry by the user when the data can be derived by the computer.
- Avoid users' entries of commands that include punctuation.
- For data entry, display currently defined default values in their appropriate data fields.
- For long, multipage displays, it should be possible to request a particular page directly, without having to go through all intermediary pages.

*Comments :*

• ***Minimal Actions vs Concision***

By *Minimal Actions* we refer here to procedures, or steps; when the length of items or data entries is concerned, then it is a matter of *Concision*.

• ***Minimal Actions vs User Experience***

The criterion *Minimal Actions* concerns the length of transactions and procedures, whatever the level of user experience.
When the length of transactions and procedures is inadequate for a particular user group (e.g., no short cuts are available to the experienced users), it concerns the criterion *User Experience*.

• ***Minimal Actions vs Flexibility***

The criterion *Flexibility* concerns the possibility of accomplishing a task in different ways, that is with different procedures. These procedures may be minimal or not. This criterion also concerns the means available to the users for adapting the interface to their particular needs.

• ***Minimal Actions vs Error Correction***

Problems related to the criterion *Minimal Actions* can result from inadequate error correction mechanisms. When the number of steps necessary to correct an error may be reduced, it is a matter of *Error Correction*. The criterion

* Elementary criterion

*Minimal Actions* refers to procedures, steps or actions that are not related to the correction of errors.

- ***Minimal Actions vs Compatibility***

A lack of compatibility (e.g., inadequate match between the task and the dialogue steps) may increase the number of actions necessary to accomplish a task. In this case, the criterion *Compatibility* must be invoked, not *Minimal Actions*, which is the consequence of the problem rather than the cause.

## 2.2. INFORMATION DENSITY *

*Definition :*

The criterion *Information Density* concerns the users' workload from a perceptual and cognitive point of view with regard to the whole set of information presented to the users rather than each individual element or item.

*Rationale(s) :*

In most tasks, users' performances are worsened when information density is too high or too low: in these cases, errors become more likely. Items that are not related to the task should be removed.

Memory load on the user should be minimised. Users should not have to memorise long data lists or complicated procedures. They should not have to undertake complex cognitive activities when these are not required by the task at hand.

*Examples of guidelines:*

- Provide only necessary and immediately usable data for any transaction; do not overload displays with extraneous data.
- Data should not require unit translations.
- Query language should use the minimum of quantifiers in query formulation.
- Do not require users to remember data accurately from one display frame to another.
- Provide automatic computation of derived data, so that a user does not have to calculate and enter any number that can be derived from data already accessible to the computer.

*Comments :*

• ***Information Density vs Concision***

The criterion *Concision* concerns the succinctness of individual items presented while *Information Density* concerns the density of the set(s) of information presented on the screen. Thus an item can be relevant but not presented in a sufficiently concise way. In this case the criterion *Concision* must be put forward. If items are superfluous then it is a matter of *Information Density*.

* Elementary criterion

# 3. EXPLICIT CONTROL

*Definition :*

The criterion *Explicit Control* concerns both the system processing of explicit user actions, and the control users have on the processing of their actions by the system.

The criterion *Explicit Control* is subdivided into two criteria: *Explicit User Action,* and *User Control.*

*Rationale(s) :*

When the users explicitly define their inputs, and when these inputs are under their control, errors as well as ambiguities are limited. Moreover, the system will be better accepted by users if they have control over the dialogue.

## 3.1. EXPLICIT USER ACTION *

*Definition :*

The criterion *Explicit User Action* refers to the relationship between the computer processing and the actions of the users. This relationship must be explicit, i.e., the computer must process only those actions requested by the users and only when requested to do so.

*Rationale(s) :*

When computer processing results from explicit user actions, users learn and understand better the application functioning and fewer errors are observed.

*Examples of guidelines:*

- Always require a user to take an explicit ENTER action to initiate processing of entered data; do not initiate processing as a side effect (e.g., updating a file) of some other action (e.g., printing a file).
- If menu selection is accomplished by pointing, provide for dual activation, in which the first action (positions a cursor) designates the selected option, followed by a separate second action that makes an explicit control entry.
- Users' command entries should be completed with an ENTER action following editing facilities.

*Comments :*

- ***Explicit User Action vs User Control***

The criterion *Explicit User Action* should be distinguished from *User Control*. The first criterion refers to the explicit character of the actions requested by the users, while the second criterion refers to the control capabilities the users should have over the ongoing processing.

* Elementary criterion

## 3.2. USER CONTROL *

*Definition :*

The criterion *User Control* refers to the fact that the users should always be in control of the system processing (e.g., interrupt, cancel, pause and continue). Every possible action by a user should be anticipated and appropriate options should be provided.

*Rationale(s) :*

Control over the interactions favours learning and thus diminishes the probability of making errors. As a consequence, the computer becomes more predictable.

*Examples of guidelines:*

- Allow users to pace their data entry, rather than having the pace being controlled by computer processing or by external events.
- The cursor should not be automatically moved without users' control (except for stable and well known procedures, such as in form-filling).
- Users should have the control over the screen pages.
- Allow users to interrupt or cancel a current transaction or process.
- Provide a CANCEL option which will have the effect of erasing any changes just made by the user and restoring the current display to its previous version.

*Comments :*

• ***User Control vs Explicit User Action.***

The criterion *Explicit User Action* should be distinguished from *User Control*. The first criterion refers to the explicit character of the actions requested by the users, while the second criterion refers to the control capabilities the users should have over the ongoing processing.

* Elementary criterion

# 4. ADAPTABILITY

*Definition :*

The adaptability of a system refers to its capacity to behave contextually and according to the users' needs and preferences.
The criterion *Adaptability* is subdivided into two criteria: *Flexibility* and *User Experience.*

*Rationale(s) :*

The more diverse the ways of achieving a given task, the more likely a particular user will find a way suitable to him, a way that he will master in the course of learning. It follows that different procedures, options, and commands must be available to the users for them to achieve a given goal. In addition, a given interface cannot be suitable for all its potential users. To avoid negative effects on the users, the interface must adapt to them.

# 4.1. FLEXIBILITY *

*Definition :*

The criterion *Flexibility* refers to the means available to the users to customise the interface in order to take into account their working strategies and/or their habits, and the task requirements. *Flexibility* is reflected in the number of possible ways of achieving a given goal. In other words, it is the capacity of the interface to adapt to the users' particular needs.

*Rationale(s) :*

The more diverse the available means to perform a given task, the more probable the chances that users will choose and master one of them during learning.

*Examples of guidelines:*

- When user requirements are uncertain, provide users with some means to control display configuration.
- When interface designers cannot predict which default values will be helpful, permit users to define, change or remove default values for data entry.
- When some displays are unnecessary, the users should be able to remove them temporarily.
- Provide some means for users to change the data entry sequence to respect their preferred sequence.
- When text formats cannot be predicted in advance, allow users to specify and store for future use the formats that might be needed.
- Users should be able to assign names to data fields they have created.

*Comments :*

• ***Flexibility vs Minimal Actions***

The criterion *Flexibility* concerns the possibility of accomplishing a task in different ways, that is with different procedures. These procedures may be minimal or not. This criterion also concerns the means available to the users for adapting the interface to their particular needs.

• ***Flexibility vs User Experience***

A good flexibility should allow the general user population to adapt the interface to its particular needs. When an interface is used by several different types of users, or a particular type of users, and the interface allows users to perform a task in different ways that are tailored to their level of experience, then it is a matter of *User Experience.*
In other words, as soon as level of experience are invoked, the criterion *User Experience* is concerned.

* Elementary criterion

• ***Flexibility vs Compatibility***

Flexibility may be a means for attaining compatibility; however, there can be a good flexibility without compatibility, and vice versa. For example, a form-filling display may not match the source documents, in terms of item ordering and data grouping. In this case, a problem of *Compatibility* should be invoked even though the interface may allow (then there is also some Flexibility) or not (then there is no Flexibility) users to modify the data entry sequence.

## 4.2. USER EXPERIENCE *

*Definition :*

The criterion *User Experience* refers to the means available to take into account the level of user experience.

*Rationale(s) :*

Experienced and inexperienced users have different information needs. It may be desirable to provide inexperienced users with prompted modes of transactions allowing simple step-by-step actions. For experienced users, computer-initiated dialogues may be boring and slow down their interactions; short-cuts may allow them to access system functions more rapidly. Different levels of interaction should take users' experience into account.

However, most systems will have users with varying levels of experience. Users may become more expert with increased experience, or perhaps less expert after a long period of disuse. The interface should also be designed so as to accommodate the varying users' levels of experience.

*Examples of guidelines:*

- Allow experienced users to by-pass a series of menu selections and make an equivalent command entry or keyboard shortcuts directly.
- Allow experienced users to key in a series of commands at one time, and inexperienced users only step by step keying.
- Dialogue types must be designed to match the needs of different users.
- Different dialogue types should be provided as a function of the experience of the various users groups (e.g., provide prompting as an optional guidance feature that can be selected by novice users but can be omitted by experienced users.
- When techniques adopted for user guidance may slow an experienced user, provide alternative paths or modes permitting a user to by-pass standard guidance procedures.
- Following the output of an error message, permit users to request a more detailed explanation of the error that is adapted to their level of knowledge.

*Comments :*

• ***User Experience vs Minimal Actions***

The criterion *Minimal Actions* concerns the length of transactions and procedures, whatever the level of user experience.

When the length of transactions and procedures is inadequate for a particular user group (e.g., no short cuts are available to the experienced users), it concerns the criterion *User Experience.*

* Elementary criterion

• ***User Experience vs Flexibility***

A good flexibility should allow the general user population to adapt the interface to its particular needs. When an interface is used by several different types of users, or a particular type of users, and the interface allows users to perform a task in different ways that are tailored to their level of experience, then it is a matter of *User Experience.*
In other words, as soon as level of experience are invoked, the criterion *User Experience* is concerned.

# 5. ERROR MANAGEMENT

*Definition :*

The criterion *Error Management* refers to the means available to prevent or reduce errors and to recover from them when they occur. Errors are defined in this context as invalid data entry, invalid format for data entry, incorrect command syntax, etc.

The criterion *Error Management* is subdivided into three criteria: *Error Protection, Quality of Error Messages,* and *Error Correction.*

*Rationale(s) :*

System interruptions caused by users' errors have negative consequences on the users' activities. In general, these kinds of interruptions increase the number of interactions and disturb the organisation and the accomplishment of the task. By limiting the number of errors, the number of interruptions is also limited. The performance is thus better.

## 5.1. ERROR PROTECTION *

*Definition :*

The criterion *Error Protection* refers to the means available to detect and prevent data entry errors, command errors, or actions with destructive consequences.

*Rationale(s) :*

It is preferable to detect errors before validation rather than after: the detection is less disruptive.

*Examples of guidelines:*

- When a user requests LOG-OFF and if any pending transaction will not be completed, or if data will be lost, display an advisory message requesting user confirmation.
- Protect field labels from accidental change by users.
- Fields designed for information display should be protected: users should not be allowed to change the information contained in these fields.
- Ensure that user interface software will deal appropriately with all possible user errors, including accidental inputs.

*Comments :*

- ***Error Protection vs Prompting***

Error protection can be achieved by various means. Automatic mechanisms can be implemented that control the users' inputs. For instance, after users' entry, the computer will display an error message if the data entry format is not appropriate. This case would concern the criterion *Error Protection*. Another way of preventing errors would be to give users information on the type and format of the data to be entered. In this case *Prompting* is concerned. These two means of preventing errors can coexist.

* Elementary criterion

## 5.2. QUALITY OF ERROR MESSAGES *

*Definition :*

The criterion *Quality of Error Messages* refers to the phrasing and the content of error messages, that is: their relevance, readability, and specificity about the nature of the errors (syntax, format, etc.) and the actions needed to correct them.

*Rationale(s) :*

The quality of error messages promotes users' learning of systems by indicating to the users the reasons for their errors, their nature, and by teaching them ways to prevent or solve their errors.

*Examples of guidelines:*

- If the user selects an invalid function key, no system action should result except a message indicating the functions appropriate for that transaction step.
- For error messages, adopt task-oriented wording.
- Make the error messages as specific as possible.
- Make error messages brief but informative.
- Adopt neutral wording for error messages; do not imply blame on the part of the user, or personalise the computer, or attempt to make a message humorous.

*Comments :*

• ***Quality of Error Messages vs Prompting***

An error message can include prompting about the means to correct errors. It is indeed a case of *Quality of Error Messages*, not a case of *Prompting*. *Prompting* only refers to the guidance provided to the users in error-free situations.

• ***Quality of Error Messages vs Legibility***

When error messages are unsatisfactory, even from a lexical point of view, it is a matter of *Quality of Error Messages*, not a matter of *Legibility*. The criterion *Quality of Error Messages* concerns all characteristics of information related to the users' errors.

• ***Quality of Error Messages vs Concision***

The criterion *Concision* does not apply to error messages. When error messages are not sufficiently succinct, it is a matter of *Quality of Error Messages*.

* Elementary criterion

## 5.3. ERROR CORRECTION *

*Definition :*

The criterion *Error Correction* refers to the means available to the users to correct their errors.

*Rationale(s) :*

Errors are less disturbing when they are easily, and immediately corrected.

*Examples of guidelines:*

- Users should be allowed to edit an extended command during its composition before taking an explicit action to ENTER the command.
- Following error detection, require the user to re-enter only that portion of a data/command entry which is not correct.
- If a data entry transaction has been completed and errors detected, permit users to make corrections directly and immediately.

*Comments :*

• ***Error Correction vs Minimal Actions***

Problems related to the criterion *Minimal Actions* can result from inadequate error correction mechanisms. When the number of steps necessary to correct an error may be reduced, it is a matter of *Error Correction*. The criterion *Minimal Actions* refers to procedures, steps or actions that are not related to the correction of errors.

* Elementary criterion

# 6. CONSISTENCY *

*Definition :*

The criterion *Consistency* refers to the way interface design choices (codes, naming, formats, procedures, etc.) are maintained in similar contexts, and are different when applied to different contexts.

*Rationale(s) :*

Procedures, labels, commands, etc., will be better recalled, located, recognised, and used if their format, location, and syntax are stable from one screen to the other, from one session to the next. In these conditions the computer system is more predictable, learning and generalisation are facilitated, and the number of errors made is reduced. A lack of consistency can increase the search time considerably.
The lack of consistency is one important reason for users' rejection.

*Examples of guidelines:*

- Window titles should always be located in the same place.
- Use similar screen formats.
- Use similar procedures to access menu options.
- Consistent phrasing and punctuation should be used in all prompts.
- Prompts for data or command entry should be displayed in a standard location.
- Data entry fields should always be the same.

*Comments :*

• ***Consistency vs Prompting***

As long as it is a matter of comparison between several transactions, objects, etc., including instructions (prompting), it concerns the criterion *Consistency*. For instance, providing a single identification for each display in a consistent location at the top of the display frame concerns the criterion *Consistency*.

• ***Consistency vs Compatibility***

The criterion *Consistency* applies only within a given application. When consistency concerns external aspects (e.g., paper forms) or concerns other applications or environments, it is then a matter of *Compatibility*.

* Elementary criterion

# 7. SIGNIFICANCE OF CODES *

*Definition :*

The criterion *Significance of Codes* qualifies the relationship between a term and/or a sign and its reference. Codes and names are significant to the users when there is a strong semantic relationship between such codes and the items or actions they refer to.

*Rationale(s) :*

When codes are meaningful, their remembering and identification are easier. In addition, non significant codes or names may lead to inappropriate users' operations, and thus to errors.

*Examples of guidelines:*

- Titles should be distinct and meaningful.
- Make abbreviation rules explicit.
- Codes should be meaningful and familiar rather than arbitrary (e.g., M for Male, and F for Female rather than 1 and 2).

*Comments :*

• ***Significance of Codes vs Legibility***

*Legibility* does not concern the semantic characteristics of information, their relevance or their significance. These characteristics are to be related to *Significance of Codes.*

* Elementary criterion

# 8. COMPATIBILITY *

*Definition :*

The criterion *Compatibility* refers to the match between users' characteristics (memory, perceptions, customs, skills, age, expectations, etc.) and task characteristics on the one hand, and the organisation of the output, input, and dialogue for a given application, on the other hand.
The criterion *Compatibility* also concerns the coherence between environments and between applications.

*Rationale(s) :*

Information transfer from one context to another is faster and more efficient when the volume of information to recode by the users is limited.
Efficiency is increased when: the procedures designed to accomplish a task are compatible with the users' psychological characteristics; procedures and tasks are organised with respect to users' expectations and practices; translations, interpretations, or references to the documentation are minimised.
Performances are better when information is presented in a directly usable form.

*Examples of guidelines:*

- When data entry involves transcription from source documents, ensure that form-filling displays match those documents.
- Dialogues should reflect data structures or organisations which are perceived by users as being natural.
- Calendar formats should follow users' customs (American vs European calendar).
- Labels, prompts, and user guidance messages should be familiar to the users and task-oriented.
- Units of measurement should be familiar to the user.
- Displays of textual data, messages, or instructions, should follow design conventions for printed text.

*Comments :*

- ***Compatibility vs Prompting***

When the wording used for prompting differs from the users' language, it concerns the criterion *Compatibility.*

- ***Compatibility vs Legibility***

The criterion *Legibility* does not concern the semantic characteristics of information, their relevance or their significance. When these characteristics are related to the task, then the criterion *Compatibility* should be considered.

---

* Elementary criterion

• ***Compatibility vs Minimal Actions***

A lack of compatibility (e.g., inadequate match between the task and the dialogue steps) may increase the number of actions necessary to accomplish a task. In this case, the criterion *Compatibility* must be invoked, not *Minimal Actions*, which is the consequence of the problem rather than the cause.

• ***Compatibility vs Flexibility***

Flexibility may be a means for attaining compatibility; however, there can be a good flexibility without compatibility, and vice versa. For example, a form-filling display may not match the source documents, in terms of item ordering and data grouping. In this case, a problem of *Compatibility* should be invoked even though the interface may allow (then there is also some Flexibility) or not (then there is no Flexibility) users to modify the data entry sequence.

• ***Compatibility vs Consistency***

The criterion *Consistency* applies only within a given application. When consistency concerns external aspects (e.g., paper forms) or concerns other applications or environments, it is then a matter of *Compatibility*.

# III. The Ergonomic Criteria (French Version)

# 1. GUIDAGE

*Définition :*

Le *Guidage* est l'ensemble des moyens mis en oeuvre pour conseiller, orienter, informer, et conduire l'utilisateur lors de ses interactions avec l'ordinateur (messages, alarmes, labels, etc.), y compris dans ses aspects lexicaux.

Quatre sous-critères participent au *Guidage* : *Incitation, Groupement/Distinction entre Items, Feedback Immédiat* et *Lisibilité.*

*Justification(s) :*

Un bon guidage facilite l'apprentissage et l'utilisation du système en permettant à l'utilisateur : de savoir, à tout moment, où il se trouve dans une séquence d'interactions, ou dans l'accomplissement d'une tâche ; de connaître les actions permises ainsi que leurs conséquences ; et d'obtenir de l'information supplémentaire (éventuellement sur demande). La facilité d'apprentissage et d'utilisation qui s'ensuivent conduisent à de meilleures performances et occasionnent moins d'erreurs.

## 1.1. INCITATION *

*Définition :*

Le terme *Incitation* a ici une définition plus large que celle qu'on lui confère généralement. Ce critère recouvre les moyens mis en œuvre pour amener les utilisateurs à effectuer des actions spécifiques, qu'il s'agisse d'entrée de données ou autre. Ce critère englobe aussi tous les mécanismes ou moyens faisant connaître aux utilisateurs les alternatives, lorsque plusieurs actions sont possibles, selon les états ou contextes dans lesquels ils se trouvent. L'*Incitation* concerne également les informations permettant aux utilisateurs de savoir où ils en sont, d'identifier l'état ou contexte dans lequel ils se trouvent, de même que les outils d'aide et leur accessibilité.

*Justification(s) :*

Une bonne incitation guide les utilisateurs et leur évite par exemple d'avoir à apprendre une série de commandes. Elle permet aussi aux utilisateurs de savoir quel est le mode ou l'état en cours, où ils se trouvent dans le dialogue et ce qu'ils ont fait pour s'y trouver. Une bonne incitation facilite donc la navigation dans une application et permet d'éviter les erreurs.

*Exemples de recommandations :*

- Guider les entrées de données en indiquant le format adéquat et les valeurs acceptables; par exemple, fournir, au niveau du label, des indices supplémentaires sur le format d'entrée des données (ex.: date (jj/mm/aa) : _ _ / _ _ / _ _ ).
- Afficher les unités de mesure des données à saisir.
- Indiquer toutes les informations d'état (ex.: modes, valeurs, etc.).
- Pour chaque champ de données, fournir un label.
- Fournir des indices sur la longueur autorisée des entrées dans un champ.
- Donner un titre à chaque fenêtre.
- Fournir des aides accessibles en ligne.

*Commentaire(s) :*

• ***Incitation vs Groupement/Distinction par le Format.***

L'*Incitation* permet de préciser aux utilisateurs de manière directe les actions possibles. Ainsi, lorsque les utilisateurs ont la possibilité d'effectuer différentes actions (ex.: confirmer, annuler, copier, etc.), on peut présenter uniquement les options disponibles, ou bien indiquer ces actions dans un message.

D'autres moyens sont disponibles pour guider les utilisateurs, lesquels peuvent être moins directs. Ainsi, dans le cas ci-dessus, on peut, si toutes les options sont présentées, indiquer par des formats différents celles qui peuvent être utilisées.

Dans les cas où plusieurs choix ou actions sont possibles, il arrive par exemple, que le bouton de commande le plus fréquent ou le plus probable soit distingué par un contour surajouté auquel correspond un retour chariot.

---

* Critère élémentaire

Dans ce cas il s'agit du critère *Groupement/Distinction par le Format* et non du critère *Incitation* car l'information retour chariot n'est pas explicite. L'*Incitation* consiste à présenter directement les différentes actions possibles alors que le *Groupement/Distinction par le Format* pourra permettre de faire apparaître des caractéristiques particulières de ces actions possibles.

• ***Incitation vs Feedback Immédiat***

La distinction entre l'*Incitation* et le *Feedback Immédiat* est parfois ténue. Suite à une action par exemple, un message peut indiquer aux utilisateurs d'une part que leur demande a été exécutée et, d'autre part que pour continuer ils doivent faire telle ou telle autre action. Il peut donc y avoir un feedback sur les actions précédentes et une incitation pour les actions subséquentes. Si le message informe les utilisateurs sur les actions accomplies, il s'agit du critère *Feedback Immédiat*. Si le message guide les utilisateurs vers des actions ultérieures ou s'il demande à aux utilisateurs d'effectuer une action particulière, alors il s'agit du critère *Incitation*.

• ***Incitation vs Lisibilité***

Les caractéristiques lexicales qui peuvent avoir un effet sur la facilité de lecture (ex.: taille des caractères, type de police, etc.), dans le cas de l'incitation, concernent le critère *Lisibilité*.

• ***Incitation vs Concision***

Une bonne incitation, telle une consigne, ou un message indiquant aux utilisateurs ce qu'il faut faire peut être trop longue. Ce défaut dans un message d'incitation concerne le critère *Concision*. De la même façon on peut indiquer correctement, donc en utilisant une bonne incitation, la façon dont les données doivent être saisies mais induire un problème de *Concision* si, en plus des données, on demande par exemple aux utilisateurs d'indiquer des unités de mesure.

• ***Incitation vs Protection Contre les Erreurs***

Plusieurs façons d'offrir une protection contre les erreurs sont possibles. On peut par exemple mettre en place un mécanisme automatique de vérification des entrées. Ainsi, au moment de la validation des entrées, un message d'erreur apparaît si le format d'entrée n'est pas conforme à ce qui est attendu. Il s'agit bien dans ce cas-ci du critère *Protection Contre les Erreurs*. Une autre façon consiste à fournir une information renseignant les utilisateurs sur le type de données attendues ou encore sur le format de ces entrées : il s'agit alors du critère *Incitation*. Ces deux mécanismes peuvent aussi coexister.

• ***Incitation vs Qualité des Messages d'Erreur***

Un message d'erreur, peut comporter des incitations sur la manière de corriger une erreur. Il s'agit là du critère *Qualité des Messages d'Erreur* et non

pas du critère *Incitation*. L'*Incitation* concerne uniquement le guidage en situation normale.

• ***Incitation vs Homogénéité/Cohérence***

Dès qu'il s'agit de considérer ou de comparer plusieurs interactions, plusieurs objets, etc., y compris avec un objectif d'incitation, il s'agit en fait du critère *Homogénéité/Cohérence*. Par exemple, présenter les titres des fenêtres dans des positions identiques concerne le critère *Homogénéité/Cohérence*.

• ***Incitation vs Compatibilité***

Lorsque les termes utilisés lors de l'incitation ne sont pas ceux couramment employés par les utilisateurs, on est en présence d'un problème de *Compatibilité*.

## 1.2. GROUPEMENT/DISTINCTION ENTRE ITEMS

*Définition :*

Le critère *Groupement/Distinction entre Items* concerne l'organisation visuelle des items d'information les uns par rapport aux autres. Ce critère prend en compte la topologie (localisation) et certaines caractéristiques graphiques (format) afin d'illustrer les relations entre les divers items affichés, leur appartenance ou non-appartenance à une même classe, ou encore dans le but de montrer la distinction entre différentes classes d'items. Ce critère concerne aussi l'organisation des items à l'intérieur d'une même classe. Deux sous-critères participent au *Groupement/Distinction entre Items* : *Groupement/Distinction par la Localisation* et *Groupement/Distinction par le Format*.

*Justification(s) :*

La compréhension d'un écran dépend, entre autres choses, de l'arrangement, du positionnement et de la distinction des objets (images, textes, commandes, etc.) qui y sont présentés. Les utilisateurs auront plus de facilité à repérer les items et/ou groupes d'items et à connaître leurs liens si ils sont présentés de façon organisée (ex.: alphabétique, fréquence d'utilisation, etc.) d'une part, et si ces items ou groupes d'items sont présentés dans des formats ou codages qui permettent d'illustrer leurs similitudes ou leurs différences, d'autre part. De même, les utilisateurs pourront mieux les apprendre et s'en rappeler. Le groupement/distinction entre items entraîne un meilleur guidage.

## 1.2.1. GROUPEMENT/DISTINCTION PAR LA LOCALISATION *

*Définition :*

Le critère *Groupement/Distinction par la Localisation* concerne le positionnement des items les uns par rapport aux autres dans le but d'indiquer leur appartenance ou non-appartenance à une même classe, ou encore dans le but de montrer la distinction entre différentes classes. Ce critère concerne aussi l'organisation des items dans une même classe.

*Justification(s) :*

La compréhension d'un écran dépend, entre autres choses, de l'arrangement des objets (images, textes, commandes, etc.) qui y sont présentés. Les utilisateurs auront plus de facilité à repérer les différents items si ils sont présentés de façon organisée (ex.: alphabétique, fréquence d'utilisation, etc.). De même, ils pourront mieux les apprendre et s'en rappeler. Le critère groupement/distinction par la localisation entraîne un meilleur guidage.

*Exemples de recommandations :*

- Organiser les items en listes hiérarchiques.
- Grouper les options de menus en fonction des objets sur lesquels elles s'appliquent.
- Lorsque plusieurs options sont présentées, l'organisation de ces dernières doit être logique, i.e. représenter une organisation fonctionnelle pertinente ou significative (arrangement alphabétique, fonctionnel, fréquence d'utilisation, etc.).

*Commentaire(s) :*

- ***Groupement/Distinction par la Localisation vs Groupement/Distinction par le Format***

Le groupement et/ou la distinction entre items peut être obtenu par le format et/ou par la localisation. La localisation et le format correspondent à différents moyens d'affichage (topologie vs caractéristiques graphiques surajoutées). Par exemple, des options de menu peuvent être distinguées par leur localisation (ex.: les options les plus fréquentes vers le haut, les moins fréquentes vers le bas) et/ou par leur format (ex.: une ligne de séparation entre un ensemble d'options liées à la présentation d'un texte et un ensemble d'options concernant les types de caractères).

* Critère élémentaire

## 1.2.2. GROUPEMENT/DISTINCTION PAR LE FORMAT *

*Définition :*

Le critère *Groupement/Distinction par le Format* concerne plus particulièrement les caractéristiques graphiques (format, couleur, etc.) permettant de faire apparaître l'appartenance ou la non-appartenance d'items à une même classe, ou encore permettant d'indiquer des distinctions entre classes ou bien encore des distinctions entre items d'une même classe.

*Justification(s) :*

Les utilisateurs auront plus de facilité à connaître les liens entre items ou classes d'items si des formats, ou codages permettent d'illustrer leurs similitudes ou leurs différences. De tels liens seront ainsi mieux appris et mieux rappelés. Un bon groupement/distinction par le format entraîne un meilleur guidage.

*Exemples de recommandations :*

- Établir une distinction visuelle entre des aires ayant des fonctions différentes (commande, message, etc.).
- Établir une distinction visuelle entre les labels et les champs d'entrée.

*Commentaire(s) :*

- ***Groupement/Distinction par le Format vs Incitation***

L'*Incitation* permet de préciser aux utilisateurs de manière directe les actions possibles. Ainsi, lorsque les utilisateurs ont la possibilité d'effectuer différentes actions (ex.: confirmer, annuler, copier, etc.), on peut présenter uniquement les options disponibles, ou bien indiquer ces actions dans un message.
D'autres moyens sont disponibles pour guider les utilisateurs, lesquels peuvent être moins directs. Ainsi, dans le cas ci-dessus, on peut, si toutes les options sont présentées, indiquer par des formats différents celles qui peuvent être utilisées.
Dans les cas où plusieurs choix ou actions sont possibles, il arrive par exemple, que le bouton de commande le plus fréquent ou le plus probable soit distingué par un contour surajouté auquel correspond un retour chariot. Dans ce cas il s'agit du critère *Groupement/Distinction par le Format* et non du critère *Incitation* car l'information retour chariot n'est pas explicite.
L'*Incitation* consiste à présenter directement les différentes actions possibles alors que le *Groupement/Distinction par le Format* pourra permettre de faire apparaître des caractéristiques particulières de ces actions possibles.

- ***Groupement/Distinction par le Format vs Groupement/Distinction par la Localisation***

Le groupement et/ou la distinction entre items peut être obtenu par le format et/ou par la localisation. La localisation et le format correspondent à

* Critère élémentaire

différents moyens d'affichage (topologie vs caractéristiques graphiques surajoutées). Par exemple, des options de menu peuvent être distinguées par leur localisation (ex.: les options les plus fréquentes vers le haut, les moins fréquentes vers le bas) et/ou par leur format (ex.: une ligne de séparation entre un ensemble d'options liées à la présentation d'un texte et un ensemble d'options concernant les types de caractères).

## 1.3. FEEDBACK IMMÉDIAT *

*Définition :*

Le *Feedback Immédiat* concerne les réponses de l'ordinateur consécutives aux actions des utilisateurs, lesquelles peuvent être le simple appui sur une touche ou l'entrée d'une séquence de commandes. Dans tous les cas, l'ordinateur doit répondre, dans les plus brefs délais, avec un délai de réponse approprié et homogène selon les types de transactions. Dans tous les cas, une réponse aussi immédiate que possible doit être fournie à l'utilisateur le renseignant sur l'action accomplie et sur son résultat.

*Justification(s) :*

La qualité et la rapidité du feedback sont deux facteurs importants pour l'établissement de la confiance et de la satisfaction des utilisateurs ainsi que pour leur compréhension du dialogue. Ces facteurs permettent aux utilisateurs de se faire une bonne représentation du fonctionnement du système.
L'absence de feedback ou des délais trop importants entre les actions utilisateur et le feedback, peuvent déconcerter les utilisateurs, ce qui augmente les chances que les utilisateurs entreprennent des actions qui risquent d'entraver les transactions en cours.

*Exemples de recommandations :*

- Toujours faire apparaître sur l'écran les entrées effectuées par les utilisateurs, sauf pour les entrées confidentielles. Cependant, même dans ce cas, un feedback perceptible doit être fourni (ex.: l'affichage d'astérisques correspondant aux entrées clavier).
- Suite à l'interruption d'un traitement par l'utilisateur, le système devrait fournir un message indiquant le retour à l'état précédent.
- Dans les cas où les traitements sont longs, une information indiquant aux utilisateurs que les traitements sont en cours devrait leur être fournie.

*Commentaire(s) :*

• ***Feedback Immédiat vs Incitation***

La distinction entre l'*Incitation* et le *Feedback Immédiat* est parfois ténue. Suite à une action par exemple, un message peut indiquer aux utilisateurs d'une part que leur demande a été exécutée et, d'autre part que pour continuer ils doivent faire telle ou telle autre action. Il peut donc y avoir un feedback sur les actions précédentes et une incitation pour les actions subséquentes. Si le message informe les utilisateurs sur les actions accomplies, il s'agit du critère *Feedback Immédiat*. Si le message guide les utilisateurs vers des actions ultérieures ou s'il demande à aux utilisateurs d'effectuer une action particulière, alors il s'agit du critère *Incitation*.

---

* Critère élémentaire

• ***Feedback Immédiat vs Lisibilité***

Quand la qualité du feedback n'est pas satisfaisante, même du point de vue lexical, il s'agit du critère *Feedback Immédiat*, et non du critère *Lisibilité*. Le *Feedback Immédiat* concerne toutes les caractéristiques des réponses consécutives aux actions des utilisateurs (présence ou absence de feedback, qualité du feedback du point de vue sémantique et lexical).

## 1.4. LISIBILITÉ *

*Définition :*

Le critère *Lisibilité* concerne les caractéristiques lexicales de présentation des informations sur l'écran pouvant entraver ou faciliter la lecture de ces informations (luminance des caractères, contraste caractères fond, dimension des lettres, espacement entre les mots, espacement entre les lignes, espacement entre les paragraphes, longueur des lignes, etc.).
Par convention, le critère *Lisibilité* ne concerne ni le feedback ni les messages d'erreurs.

*Justification(s) :*

La performance est accrue lorsque la présentation des informations à l'écran tient compte des caractéristiques cognitives et perceptives des utilisateurs. Une bonne lisibilité facilite la lecture des informations présentées. Ainsi par exemple, les lettres sombres sur fond clair sont plus faciles à lire que l'inverse; le texte présenté en lettres majuscules et minuscules est lu plus rapidement que le texte présenté seulement en lettres majuscules.

*Exemples de recommandations :*

- Les titres doivent être centrés.
- Les labels doivent être en majuscule.
- Le curseur doit être facilement repérable.
- Il est préférable de présenter un texte avec quelques lignes longues plutôt que de nombreuse lignes courtes.
- Les lignes de texte continu doivent être d'au moins 50 caractères.
- La justification à droite des textes ne devrait être utilisée qu'avec un espacement variable, de sorte qu'un espacement proportionnel constant entre les lettres et les mots soit respecté.
- Dans les affichages de textes, utiliser le moins possible la césure des mots.

*Commentaire(s) :*

**Le critère *Lisibilité* ne s'applique ni au feedback, ni aux messages d'erreurs.**
En effet, tous les aspects liés à des difficultés de lecture par les utilisateurs, ou plus généralement à la qualité des messages concernant le feedback ou les messages d'erreurs sont affectés respectivement au critère *Feedback Immédiat* et au critère *Qualité des Messages d'Erreurs*.

- ***Lisibilité vs Feedback Immédiat***

Quand la qualité du feedback n'est pas satisfaisante, même du point de vue lexical, il s'agit du critère *Feedback Immédiat*, et non du critère *Lisibilité*. Le *Feedback Immédiat* concerne toutes les caractéristiques des réponses consécutives aux actions des utilisateurs (présence ou absence de feedback, qualité du feedback du point de vue sémantique et lexical).

---

* Critère élémentaire

• ***Lisibilité vs Qualité des Messages d'Erreur***

Quand les messages d'erreur ne sont pas adéquats, même du point de vue lexical, il s'agit du critère *Qualité des Messages d'Erreur*, et non du critère *Lisibilité*. Le critère *Qualité des Messages d'Erreur* concerne toutes les caractéristiques des informations relatives aux erreurs des utilisateurs.

• ***Lisibilité vs Incitation***

Les caractéristiques lexicales qui peuvent avoir un effet sur la facilité de lecture (ex.: taille des caractères, type de police, etc.), dans le cas de l'incitation, concernent le critère *Lisibilité*.

• ***Lisibilité vs Signifiance des Codes et Dénominations***

Le critère *Lisibilité* ne concerne pas les aspects sémantiques des informations, leur pertinence ou leur signification. Ceux-ci sont davantage du ressort du critère *Signifiance des Codes et Dénominations*.

• ***Lisibilité vs Compatibilité***

Le critère *Lisibilité* ne concerne pas les aspects sémantiques des informations, leur pertinence ou leur signification. Quand ces aspects sont liés aux tâches, le critère à considérer est la *Compatibilité*.

# 2. CHARGE DE TRAVAIL

*Définition :*

Le critère *Charge de Travail* concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs et dans l'augmentation de l'efficacité du dialogue.
Deux sous-critères participent au critère *Charge de Travail* : *Brièveté* (qui inclut les critères *Concision* et *Actions Minimales*), et *Densité Informationnelle.*

*Justification(s) :*

Plus la charge de travail est élevée, plus grands sont les risques d'erreurs. De même, moins l'utilisateur sera distrait par des informations non pertinentes, plus il pourra effectuer sa tâche efficacement. Par ailleurs, plus les actions requises seront courtes, plus rapides seront les interactions.

## 2.1. BRIÈVETÉ

*Définition :*

Le critère *Brièveté* concerne la charge de travail au niveau perceptif et mnésique à la fois pour les éléments individuels d'entrée ou de sortie et les séquences d'entrées (i.e., les suites d'actions nécessaires à l'atteinte d'un but, à l'accomplissement d'une tâche). Il s'agit ici de limiter autant que possible le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les utilisateurs.

Deux sous-critères participent au critère *Brièveté* : *Concision* et *Actions Minimales.*

*Justification(s) :*

Les capacités de la mémoire à court terme sont limitées. Par conséquent, plus courtes sont les entrées, plus limités sont les risques d'erreurs. Par ailleurs, plus succincts sont les items, plus court est le temps de lecture. Aussi, plus les actions nécessaires à l'atteinte d'un but sont nombreuses et compliquées, plus la charge de travail augmente et par conséquent plus les risques d'erreurs sont élevés.

## 2.1.1. CONCISION *

*Définition :*

Le critère *Concision* concerne la charge de travail au niveau perceptif et mnésique pour ce qui est des éléments individuels d'entrée ou de sortie. Par convention, la *Concision* ne concerne pas le feedback ni les messages d'erreurs.

*Justification(s) :*

Les capacités de la mémoire à court terme sont limitées. Par conséquent, plus courtes sont les entrées, plus limités sont les risques d'erreurs. Par ailleurs, plus succincts sont les items, plus court est le temps de lecture.

*Exemples de recommandations :*

- Pour les données numériques, la saisie des zéros précédant les nombres ne devrait pas être nécessaire.
- Si des codes sont supérieurs à 4 ou 5 caractères, utiliser des mnémotechniques ou abréviations.
- Permettre aux utilisateurs des entrées de données courtes.
- Lorsqu'une unité de mesure est associée à un champ de donnée, celle-ci doit faire partie du label du champ plutôt qu'être saisie par les utilisateurs.

*Commentaire(s) :*

• ***Concision vs Incitation***

Une bonne incitation, telle une consigne, ou un message indiquant aux utilisateurs ce qu'il faut faire peut être trop longue. Ce défaut dans un message d'incitation concerne le critère *Concision*. De la même façon on peut indiquer correctement, donc en utilisant une bonne incitation, la façon dont les données doivent être saisies mais induire un problème de *Concision* si, en plus des données, on demande par exemple aux utilisateurs d'indiquer des unités de mesure.

• ***Concision vs Actions Minimales***

Le critère *Actions Minimales* concerne les procédures ou étapes; quand il s'agit de la longueur des items à saisir, le critère concerné est plutôt la *Concision*.

• ***Concision vs Densité Informationnelle***

Le critère *Concision* concerne le caractère succinct de la présentation des items individuels alors que le critère *Densité Informationnelle* concerne la densité de l'ensemble des items affichés à l'écran. Ainsi un item peut être pertinent tout en n'étant pas présenté de façon suffisamment succincte. On invoquera alors le critère *Concision*. Si des items sont superflus il s'agit alors du critère *Densité Informationnelle*.

---

* Critère élémentaire

• ***Concision vs Qualité des messages***

Le critère *Concision* ne s'applique pas aux messages d'erreurs. Lorsque ces derniers ne sont pas suffisamment succincts, il s'agit du critère *Qualité des Messages d'Erreur*.

## 2.1.2. ACTIONS MINIMALES *

*Définition :*

Le critère *Actions Minimales* concerne la charge de travail quant aux actions nécessaires à l'atteinte d'un but, à l'accomplissement d'une tâche. Il s'agit ici de limiter autant que possible les étapes par lesquelles doivent passer les utilisateurs.

*Justification(s) :*

Plus les actions nécessaires à l'atteinte d'un but sont nombreuses et compliquées, plus la charge de travail augmente et par conséquent plus les risques d'erreurs sont élevés.

*Exemples de recommandations :*

- Minimiser le nombre d'étapes dans la sélection de menus.
- Ne pas demander aux utilisateurs d'entrer des données qui peuvent être déduites par l'ordinateur.
- Éviter les ponctuations pour les entrées de commandes.
- Pour la saisie de données, afficher dans les champs appropriés, les valeurs par défaut.
- Pour des documents contenant plusieurs pages, il devrait être possible d'atteindre une page donnée sans avoir à parcourir les pages intermédiaires une à une.

*Commentaire(s) :*

• ***Actions Minimales vs Concision***

Le critère *Actions Minimales* concerne les procédures ou étapes; quand il s'agit de la longueur des items à saisir, le critère concerné est plutôt la *Concision*.

• ***Actions Minimales vs Prise en Compte de l'Expérience de l'Utilisateur***

Le critère *Actions Minimales* concerne la longueur des transactions et des procédures quel que soit le niveau d'expérience des utilisateurs.
Quand la longueur des transactions et des procédures apparaît inadaptée pour un groupe particulier d'utilisateurs (ex.: qu'il n'existe pas de raccourcis clavier pour les utilisateurs expérimentés), il s'agit alors du critère *Prise en Compte de l'Expérience de l'Utilisateur*.

• ***Actions Minimales vs Flexibilité***

Le critère *Flexibilité* concerne l'existence de procédures différentes, quelles soient ou non minimales, permettant d'accomplir une même tâche, ou encore des moyens permettant d'adapter l'interface aux besoins particuliers d'un utilisateur.

---

* Critère élémentaire

• ***Actions Minimales vs Correction des Erreurs***

Les problèmes liés au critère *Actions Minimales* peuvent résulter de mécanismes de correction des erreurs inadéquats. Lorsque le nombre d'actions nécessaires à la correction des erreurs peut être réduit, il s'agit d'un problème de *Correction des Erreurs*. Le critère *Actions Minimales* concerne donc les procédures autres que celles liées à la correction des erreurs.

• ***Actions Minimales vs Compatibilité***

Un manque de compatibilité (ex.: un manque d'adéquation des procédures avec les tâches) peut conduire à une augmentation des actions nécessaires à l'accomplissement d'une tâche. Dans ce cas, il faut invoquer le critère *Compatibilité* et non pas *Actions Minimales* qui en est la conséquence, pas la cause.

## 2.2. DENSITÉ INFORMATIONNELLE *

*Définition :*

Le critère *Densité Informationnelle* concerne la charge de travail du point de vue perceptif et mnésique, pour des ensembles d'éléments et non pour des items.

*Justification(s) :*

Dans la plupart des tâches, la performance des utilisateurs est influencée négativement quand la charge informationnelle est trop élevée ou trop faible. La probabilité d'erreur augmente. Il faut donc supprimer les éléments sans lien avec le contenu de la tâche en cours.

Il faut aussi éviter d'imposer à l'utilisateur la mémorisation de longues et nombreuses informations ou procédures (la mémoire à court terme est limitée), ou toute activité nécessitant de sa part la mise en oeuvre d'activités cognitives complexes lorsque la tâche ne le requiert pas.

*Exemples de recommandations :*

- Limiter la densité informationnelle de l'écran, en affichant seulement les informations nécessaires.
- L'information ne doit pas nécessiter des traductions d'unités.
- Utiliser le minimum de quantificateurs, notamment dans les langages de requêtes.
- Éviter à l'utilisateur d'avoir à se rappeler des données d'une page écran à une autre.
- Les données qui peuvent être calculées à partir de celles saisies par l'utilisateur doivent l'être automatiquement. On ne doit pas exiger de l'opérateur d'effectuer des calculs qui peuvent être faits automatiquement.

*Commentaire(s) :*

• ***Densité Informationnelle vs Concision***

Le critère *Concision* concerne le caractère succinct de la présentation des items individuels alors que le critère *Densité Informationnelle* concerne la densité de l'ensemble des items affichés à l'écran. Ainsi un item peut être pertinent tout en n'étant pas présenté de façon suffisamment succincte. On invoquera alors le critère *Concision*. Si des items sont superflus il s'agit alors du critère *Densité Informationnelle*.

* Critère élémentaire

# 3. CONTRÔLE EXPLICITE

*Définition :*

Le critère *Contrôle Explicite* concerne à la fois la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions.
Deux sous-critères participent au *Contrôle Explicite* : *Actions Explicites* et *Contrôle Utilisateur.*

*Justification(s) :*

Quand les entrées des utilisateurs sont explicitement définies par eux-mêmes et sous leur contrôle, les ambiguïtés et les erreurs sont limitées. De plus, le contrôle qu'ont les utilisateurs sur le dialogue est un facteur d'acceptation du système.

## 3.1. ACTIONS EXPLICITES *

*Définition :*

Le critère *Actions Explicites* concerne la relation pouvant exister entre le fonctionnement de l'application et les actions des utilisateurs. Cette relation doit être explicite, c'est-à-dire que le système doit exécuter seulement les opérations demandées par l'utilisateur et pas d'autres et ce, au moment où il les demande.

*Justification(s) :*

Quand les opérations du système résultent des actions des utilisateurs, on observe moins d'erreurs et la compréhension du fonctionnement de l'application est facilitée.

*Exemples de recommandations :*

- Le système doit requérir une action explicite de validation par l'utilisateur (ex.: Entrée, Validation, OK) suite à une entrée de données; aucun traitement (ex.: sauvegarde d'un fichier) ne devrait être la conséquence d'une autre action (ex.: une demande d'impression).
- Lors d'une sélection d'options de menu par pointage, prévoir une action explicite de validation.
- L'entrée de commandes doit se terminer par une indication de fin (ex.: ENTER, OK) à laquelle des possibilités d'édition doivent être préalables.

*Commentaire(s) :*

• ***Actions Explicites vs Contrôle Utilisateur***

Le critère *Actions Explicites* se distingue du critère *Contrôle Utilisateur* par le fait que le premier concerne les traitements explicitement requis par l'utilisateur, alors que le second concerne le contrôle que l'utilisateur doit pouvoir exercer sur les traitements en cours.

* Critère élémentaire

## 3.2. CONTRÔLE UTILISATEUR *

*Définition :*

Par *Contrôle Utilisateur* on entend ici le fait que l'utilisateur doit toujours avoir la main, pouvoir contrôler le déroulement (ex.: interrompre, reprendre) des traitements informatiques en cours. Ses actions devraient être anticipées et des options appropriées fournies pour chaque cas.

*Justification(s) :*

Quand l'utilisateur a le contrôle du dialogue, les réactions de ce dernier sont prévisibles. L'apprentissage s'en trouve facilité et le risque d'erreurs diminué.

*Exemples de recommandations :*

- Autoriser les utilisateurs à contrôler le rythme de leurs entrées plutôt que de faire dépendre ce rythme des traitements du système ou d'événements extérieurs.
- Le curseur ne doit pas être déplacé sans contrôle des utilisateurs (sauf s'il s'agit de procédures stables et bien connues, ex., remplissage de formulaires).
- Ne pas passer d'une page écran à une autre sans contrôle de l'utilisateur.
- Autoriser l'utilisateur à interrompre à tout moment une action ou un traitement en cours (ex.: annuler).
- Fournir la possibilité de retour arrière conduisant à l'annulation des modifications en cours et au retour à la version immédiatement précédente.

*Commentaire(s) :*

• ***Contrôle Utilisateur vs Actions Explicites.***

Le critère *Actions Explicites* se distingue du critère *Contrôle Utilisateur* par le fait que le premier concerne les traitements explicitement requis par l'utilisateur, alors que le second concerne le contrôle que l'utilisateur doit pouvoir exercer sur les traitements en cours.

* Critère élémentaire

# 4. ADAPTABILITÉ

*Définition :*

L'adaptabilité d'un système concerne sa capacité à réagir selon le contexte, et selon les besoins et préférences des utilisateurs.
Deux sous-critères participent au critère *Adaptabilité* : *Flexibilité* et *Prise en Compte de l'Expérience de l'Utilisateur.*

*Justification(s) :*

Plus les façons d'effectuer une même tâche sont diverses, plus les chances que l'utilisateur puisse choisir et maîtriser l'une d'entre elles, au cours de ses apprentissages, sont importantes. Il faut donc fournir à l'utilisateur des procédures, options, et commandes différentes leur permettant d'atteindre un même objectif. Par ailleurs, une interface ne peut convenir à la fois à tous ses utilisateurs potentiels. Pour qu'elle n'ait pas d'effets négatifs sur l'utilisateur, cette interface doit, selon les contextes, s'adapter à l'utilisateur.

## 4.1. FLEXIBILITÉ *

*Définition :*

Le critère *Flexibilité* concerne les moyens mis à la disposition des utilisateurs pour personnaliser l'interface afin de rendre compte de leurs stratégies ou habitudes de travail et des exigences de la tâche. Le critère *Flexibilité* correspond aussi au nombre de façons différentes mises à la disposition des utilisateurs pour atteindre un objectif donné. Il s'agit en d'autres termes de la capacité de l'interface à s'adapter à des actions variées des utilisateurs.

*Justification(s) :*

Plus les façons d'effectuer une même tâche sont diverses, plus les chances que l'utilisateur puisse choisir et maîtriser l'une d'entre elles, au cours de ses apprentissages, sont importantes.

*Exemples de recommandations :*

- Quand les exigences utilisateurs sont imprécises, fournir aux utilisateurs une certaine latitude dans les possibilités de contrôle des affichages.
- Pour l'entrée de données, lorsque des valeurs par défaut ne sont pas connues à l'avance, le système doit permettre aux utilisateurs de définir, changer ou supprimer ces valeurs.
- Quand certains affichages sont inutiles, les utilisateurs doivent pouvoir les désactiver temporairement.
- La séquence des entrées de données doit pouvoir être modifiée pour s'adapter à l'ordre souhaité par les utilisateurs.
- Lorsqu'on ne peut spécifier à l'avance le format d'un document, on doit permettre aux utilisateurs d'en définir un et de le sauvegarder pour une utilisation ultérieure.
- Les utilisateurs devrait pouvoir assigner eux-mêmes le nom des champs de données qu'ils créent.

*Commentaire(s) :*

• ***Flexibilité vs Actions Minimales***

Le critère *Flexibilité* concerne l'existence de procédures différentes, quelles soient ou non minimales, permettant d'accomplir une même tâche, ou encore des moyens permettant d'adapter l'interface aux besoins particuliers d'un utilisateur.

• ***Flexibilité vs Prise en Compte de l'Expérience de l'Utilisateur***

Une bonne flexibilité doit permettre à la population générale des utilisateurs d'adapter l'interface à ses besoins particuliers. Quand une interface concerne plusieurs types d'utilisateurs ou un type particulier d'utilisateurs et que l'interface permet d'effectuer une tâche de plusieurs façons, en fonction du degré d'expérience des utilisateurs, alors il s'agit du critère *Prise en Compte de l'Expérience de l'Utilisateur.*

---

* Critère élémentaire

En d'autres termes, dès que l'expérience est invoquée, en rapport avec l'interface, le critère *Prise en Compte de l'Expérience de l'Utilisateur* est impliqué.

• ***Flexibilité vs Compatibilité***

La flexibilité peut être un moyen d'assurer une certaine compatibilité. Cependant la flexibilité peut être satisfaite sans que la compatibilité le soit. Ainsi, dans un dialogue par formulaire, l'ordre et le groupement des informations peut ne pas correspondre aux documents papier. Dans ce cas, il s'agit d'un problème de *Compatibilité*, qu'il soit possible (alors il y a une certaine flexibilité) ou non (alors il n'y a pas de flexibilité) de modifier la séquence d'entrée de données.

## 4.2. PRISE EN COMPTE DE L'EXPÉRIENCE DE L'UTILISATEUR *

*Définition :*

Le critère *Prise en Compte de l'Expérience de l'Utilisateur* concerne les moyens mis en oeuvre pour respecter le niveau d'expérience de l'utilisateur.

*Justification(s) :*

Des utilisateurs expérimentés n'ont pas toujours les mêmes besoins informationnels que les novices. Il peut être souhaitable de fournir aux utilisateurs inexpérimentés des transactions très guidées, au pas-à-pas. Pour des utilisateurs expérimentés, des dialogues à la seule initiative de l'ordinateur peuvent les ennuyer et ralentir leurs interactions; par contre, des raccourcis peuvent leur permettre d'accéder plus rapidement aux fonctions du système. Des moyens différenciés doivent donc être prévus pour tenir compte de ces différences d'expérience.

Cependant, l'expérience des utilisateurs peut varier. Les utilisateurs peuvent devenir plus experts à force d'utilisation, ou moins experts après de longues périodes de non-utilisation. L'interface doit aussi être conçue afin de tenir compte de ces variations du niveau d'expérience.

*Exemples de recommandations :*

- Autoriser les utilisateurs expérimentés à contourner une série de sélections par menu en formulant directement des commandes ou par des raccourcis clavier.
- Prévoir des choix d'entrées pas-à-pas ou multiples selon l'expérience des utilisateurs.
- Autoriser différents modes de dialogue correspondant aux différents groupes d'utilisateurs (ex.: permettre une incitation adaptée au niveau d'expérience des utilisateurs).
- Permettre la saisie de plusieurs commandes avant confirmation pour les expérimentés.
- Lorsque les techniques de guidage ralentissent les utilisateurs expérimentés, fournir des moyens à ces utilisateurs de contourner ce guidage.
- Les utilisateurs devraient pouvoir demander un niveau de détail des messages d'erreurs qui soit fonction de leur niveau de connaissance.

*Commentaire(s) :*

• ***Prise en Compte de l'Expérience de l'Utilisateur vs Actions Minimales***

Le critère *Actions Minimales* concerne la longueur des transactions et des procédures quel que soit le niveau d'expérience des utilisateurs.

Quand la longueur des transactions et des procédures apparaît inadaptée pour un groupe particulier d'utilisateurs (ex.: qu'il n'existe pas de raccourcis clavier pour les utilisateurs expérimentés), il s'agit alors du critère *Prise en Compte de l'Expérience de l'Utilisateur.*

---

* Critère élémentaire

• ***Prise en Compte de l'Expérience de l'Utilisateur vs Flexibilité***

Une bonne flexibilité doit permettre à la population générale des utilisateurs d'adapter l'interface à ses besoins particuliers. Quand une interface concerne plusieurs types d'utilisateurs ou un type particulier d'utilisateurs et que l'interface permet d'effectuer une tâche de plusieurs façons, en fonction du degré d'expérience des utilisateurs, alors il s'agit du critère *Prise en Compte de l'Expérience de l'Utilisateur.*
En d'autres termes, dès que l'expérience est invoquée, en rapport avec l'interface, le critère *Prise en Compte de l'Expérience de l'Utilisateur* est impliqué.

# 5. GESTION DES ERREURS

*Définition :*

Le critère *Gestion des Erreurs* concerne tous les moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent. Les erreurs sont ici considérées comme des saisies de données incorrectes, des saisies dans des formats inadéquats, des saisies de commandes avec une syntaxe incorrecte, etc.

Trois sous-critères participent à la *Gestion des Erreurs* : *Protection Contre les Erreurs, Qualité des Messages d'Erreurs* et *Correction des Erreurs.*

*Justification(s) :*

Les interruptions provoquées par les erreurs ont des conséquences négatives sur l'activité des utilisateurs. De manière générale, elles rallongent les transactions et perturbent la planification. Plus les erreurs sont limitées, moins il y a d'interruptions au cours de la réalisation d'une tâche et meilleure est la performance.

# 5.1. PROTECTION CONTRE LES ERREURS *

*Définition :*

Le critère *Protection Contre les Erreurs* concerne les moyens mis en place pour détecter et prévenir les erreurs d'entrées de données ou de commandes ou les actions aux conséquences néfastes.

*Justification(s) :*

Il est préférable de détecter les erreurs lors de la saisie plutôt que lors de la validation : ceci évite de perturber la planification.

*Exemples de recommandations :*

- Quand les utilisateurs terminent une session et qu'il y a un risque de perte de données, il doit y avoir un message le signalant et demandant confirmation de fin de session.
- Les labels de champs doivent être protégés.
- Les aires d'affichage doivent être protégés : les utilisateurs ne doivent pas pouvoir changer les informations contenues dans ces champs.
- Toutes les actions possibles sur une interface doivent être envisagées et plus particulièrement les appuis accidentels des touches du clavier afin que les entrées non attendues soient détectées.

*Commentaire(s) :*

• ***Protection Contre les Erreurs vs Incitation***

Plusieurs façons d'offrir une protection contre les erreurs sont possibles. On peut par exemple mettre en place un mécanisme automatique de vérification des entrées. Ainsi, au moment de la validation des entrées, un message d'erreur apparaît si le format d'entrée n'est pas conforme à ce qui est attendu. Il s'agit bien dans ce cas-ci du critère *Protection Contre les Erreurs*. Une autre façon consiste à fournir une information renseignant les utilisateurs sur le type de données attendues ou encore sur le format de ces entrées : il s'agit alors du critère *Incitation*. Ces deux mécanismes peuvent aussi coexister.

* Critère élémentaire

## 5.2. QUALITÉ DES MESSAGES D'ERREUR *

*Définition :*

Le critère *Qualité des Messages d'Erreur* concerne la pertinence, la facilité de lecture et l'exactitude de l'information donnée aux utilisateurs sur la nature des erreurs commises (syntaxe, format, etc.) et sur les actions à entreprendre pour les corriger.

*Justification(s) :*

La qualité des messages favorise l'apprentissage du système en indiquant aux utilisateurs les raisons ou la nature de leurs erreurs et en leur indiquant ce qu'il faut ou ce qu'ils auraient dû faire.

*Exemples de recommandations :*

- Si l'utilisateur sélectionne une touche fonction non valide, aucune action ne doit en résulter, si ce n'est un message indiquant les fonctions appropriées à cette étape de la transaction.
- Fournir des messages d'erreurs orientés tâches.
- Utiliser des termes aussi spécifiques que possibles pour les messages d'erreur.
- Utiliser des messages d'erreur aussi brefs que possible.
- Adopter un vocabulaire neutre, non personnalisé, non réprobateur dans les messages d'erreur; éviter l'humour.

*Commentaire(s) :*

• ***Qualité des messages d'erreur vs Incitation***

Un message d'erreur, peut comporter des incitations sur la manière de corriger une erreur. Il s'agit là du critère *Qualité des Messages d'Erreur* et non pas du critère *Incitation*. L'*Incitation* concerne uniquement le guidage en situation normale.

• ***Qualité des Messages d'Erreur vs Lisibilité***

Quand les messages d'erreur ne sont pas adéquats, même du point de vue lexical, il s'agit du critère *Qualité des Messages d'Erreur*, et non du critère *Lisibilité*. Le critère *Qualité des Messages d'Erreur* concerne toutes les caractéristiques des informations relatives aux erreurs des utilisateurs.

• ***Qualité des Messages d'Erreur vs Concision***

Le critère *Concision* ne s'applique pas aux messages d'erreurs. Lorsque ces derniers ne sont pas suffisamment succincts, il s'agit du critère *Qualité des Messages d'Erreur*.

---

* Critère élémentaire

## 5.3. CORRECTION DES ERREURS *

*Définition :*

Le critère *Correction des Erreurs* concerne les moyens mis à la disposition des utilisateurs pour leur permettre de corriger leurs erreurs.

*Justification(s) :*

Les erreurs sont d'autant moins perturbatrices qu'elles sont faciles à corriger.

*Exemples de recommandations :*

- Fournir la possibilité de modifier les commandes lors de leur saisie.
- Suite à une erreur de saisie d'une commande ou de données, donner aux utilisateurs la possibilité de corriger seulement la portion de données ou de commande qui est erronée.
- Si les utilisateurs se rendent compte qu'ils ont commis une erreur d'entrée de données, leur donner la possibilité d'effectuer, au moment de leur détection d'erreur, les corrections souhaitées.

*Commentaire(s) :*

• ***Correction des Erreurs vs Actions Minimales***

Les problèmes liés au critère *Actions Minimales* peuvent résulter de mécanismes de correction des erreurs inadéquats. Lorsque le nombre d'actions nécessaires à la correction des erreurs peut être réduit, il s'agit d'un problème de *Correction des Erreurs*. Le critère *Actions Minimales* concerne donc les procédures autres que celles liées à la correction des erreurs.

* Critère élémentaire

# 6. HOMOGÉNÉITÉ/COHÉRENCE [3] *

*Définition :*

Le critère *Homogénéité/Cohérence* se réfère à la façon avec laquelle les choix de conception de l'interface (codes, dénominations, formats, procédures, etc.) sont conservés pour des contextes identiques, et sont différents pour des contextes différents.

*Justification(s) :*

Les procédures, labels, commandes, etc., sont d'autant mieux reconnus, localisés et utilisés, que leur format, localisation, ou syntaxe sont stables d'un écran à l'autre, d'une session à l'autre. Dans ces conditions le système est davantage prévisible et les apprentissages plus généralisables; les erreurs sont réduites. Le manque d'homogénéité peut augmenter considérablement le temps de recherche.
Le manque d'homogénéité est aussi une raison importante du refus d'utilisation.

*Exemples de recommandations :*

- Localisation similaire des titres des fenêtres.
- Formats d'écrans similaires.
- Procédures similaires d'accès aux options de menus.
- Lors du guidage, toujours utiliser les mêmes ponctuations et les mêmes constructions de phrases.
- Afficher à la même position les "prompts" pour la saisie des données ou des commandes.
- Le format des champs d'entrée de données doit toujours être le même.

*Commentaire(s) :*

• ***Homogénéité/Cohérence vs Incitation***

Dès qu'il s'agit de considérer ou de comparer plusieurs interactions, plusieurs objets, etc., y compris avec un objectif d'incitation, il s'agit en fait du critère *Homogénéité/Cohérence*. Par exemple, présenter les titres des fenêtres dans des positions identiques concerne le critère *Homogénéité/Cohérence*.

• ***Homogénéité/Cohérence vs Compatibilité***

Le critère *Homogénéité/Cohérence* s'applique au sein d'une interface donnée. Lorsque la cohérence concerne des aspects externes à l'application (ex.: formulaires papier) ou concerne d'autres applications ou environnements, on parle alors de *Compatibilité*.

---

[3] (ou CONSISTENCE)

* Critère élémentaire

# 7. SIGNIFIANCE DES CODES ET DÉNOMINATIONS *

*Définition :*

Le critère *Signifiance des Codes et Dénominations* concerne l'adéquation entre l'objet ou l'information affichée ou entrée, et son référent. Des codes et dénominations "signifiants" disposent d'une relation sémantique forte avec leur référent.

*Justification(s) :*

Lorsque le codage est signifiant, le rappel et la reconnaissance sont meilleurs. De plus, des codes et dénominations non significatifs pour les utilisateurs peuvent leur suggérer des opérations inappropriées et ainsi conduire à des erreurs.

*Exemples de recommandations :*

- Les titres doivent véhiculer ce qu'ils représentent, et être distincts.
- Rendre les règles d'abréviation explicites.
- Utiliser des codes et dénominations significatifs et familiers plutôt que des codes et dénominations arbitraires (ex.: M pour masculin et F pour féminin plutôt que 1 et 2).

*Commentaire(s) :*

- ***Signifiance des Codes et Dénominations vs Lisibilité***

Le critère *Lisibilité* ne concerne pas les aspects sémantiques des informations, leur pertinence ou leur signification. Ceux-ci sont davantage du ressort du critère *Signifiance des Codes et Dénominations.*

* Critère élémentaire

# 8. COMPATIBILITÉ *

*Définition :*

Le critère *Compatibilité* se réfère à l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétences, âge, attentes, etc.) et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part.
De plus, la *Compatibilité* concerne également le degré de similitude entre divers environnements ou applications.

*Justification(s) :*

Le transfert d'information d'un contexte à un autre est d'autant plus rapide et efficace que le volume d'information à recoder par l'utilisateur est réduit.
L'efficacité est accrue lorsque : les procédures nécessaires à l'accomplissement de la tâche sont compatibles avec les caractéristiques psychologiques des utilisateurs ; les procédures et les tâches sont organisées de manière à respecter les attentes, ou habitudes des utilisateurs ; les traductions, les transpositions, les interprétations, ou références à la documentation sont minimisées.
Les performances sont meilleures lorsque l'information est présentée sous une forme directement utilisable.

*Exemples de recommandations :*

- L'organisation des informations affichées doit être conforme à l'organisation des données à entrer.
- Les procédures de dialogue doivent être compatibles avec l'ordre tel que se l'imagine l'utilisateur ou celui dont il a l'habitude.
- Le format de date en français est "jour/mois/année". En anglais, ce format devient "mois/jour/année".
- Les termes employés doivent être familiers aux utilisateurs, et relatifs à la tâche à réaliser.
- Les unités de mesure doivent être celles qui sont normalement utilisées.
- L'affichage de texte à l'écran doit être conforme aux conventions utilisées pour la présentation des textes sur papier.

*Commentaire(s) :*

- ***Compatibilité vs Incitation***

Lorsque les termes utilisés lors de l'incitation ne sont pas ceux couramment employés par les utilisateurs, on est en présence d'un problème de *Compatibilité*.

* Critère élémentaire

• ***Compatibilité vs Lisibilité***

Le critère *Lisibilité* ne concerne pas les aspects sémantiques des informations, leur pertinence ou leur signification. Quand ces aspects sont liés aux tâches, le critère à considérer est la *Compatibilité.*

• ***Compatibilité vs Actions Minimales***

Un manque de compatibilité (ex.: un manque d'adéquation des procédures avec les tâches) peut conduire à une augmentation des actions nécessaires à l'accomplissement d'une tâche. Dans ce cas, il faut invoquer le critère *Compatibilité* et non pas *Actions Minimales* qui en est la conséquence, pas la cause.

• ***Compatibilité vs Flexibilité***

La flexibilité peut être un moyen d'assurer une certaine compatibilité. Cependant la flexibilité peut être satisfaite sans que la compatibilité le soit. Ainsi, dans un dialogue par formulaire, l'ordre et le groupement des informations peut ne pas correspondre aux documents papier. Dans ce cas, il s'agit d'un problème de *Compatibilité*, qu'il soit possible (alors il y a une certaine flexibilité) ou non (alors il n'y a pas de flexibilité) de modifier la séquence d'entrée de données.

• ***Compatibilité vs Homogénéité/Cohérence***

Le critère *Homogénéité/Cohérence* s'applique au sein d'une interface donnée. Lorsque la cohérence concerne des aspects externes à l'application (ex.: formulaires papier) ou concerne d'autres applications ou environnements, on parle alors de *Compatibilité.*

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)



Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399

