

<h2>Critères ergonomiques</h2>	
Nicolas Roussel	INRIA
Michel Beaudouin-Lafon	Université Paris-Sud

<h2>Avant de commencer...</h2>	
Source des critères et recommandations <i>De l'ergonomie du logiciel au design des sites web</i> Jean-François Nogier, Dunod, 2002	
Avantages	
<ul style="list-style-type: none"><li>- ils permettent de corriger de nombreux problèmes d'utilisabilité</li><li>- ils sont faciles à retenir, et assez simples à appliquer</li><li>- ils peuvent servir à une évaluation simple et rapide d'un système existant</li></ul>	
Inconvénients	
<ul style="list-style-type: none"><li>- ce n'est pas une check-list</li><li>- il y a quelques redondances</li><li>- certains critères sont à manier avec précaution</li><li>- chaque règle peut avoir des exceptions</li></ul>	

<h2>Catégories</h2>	
1- Compatibilité	
2- Guidage	
3- Homogénéité	
4- Flexibilité	
5- Traitement des erreurs	
6- Concision	

<h2>1. Compatibilité</h2>	
---------------------------	--

## 1. Compatibilité

Capacité du système à s'intégrer dans l'activité des utilisateurs

Recommandations :

- parler le langage de l'utilisateur
  - exemple : système de PAO pour typographe vs. secrétaire
- présenter les informations de façon cohérente par rapport aux autres supports de travail
  - exemple : version électronique d'un formulaire papier
- l'accès aux commandes doit être compatible avec la tâche de l'utilisateur
  - exemple : commandes fréquentes au premier niveau des menus

Nécessité d'étudier et d'analyser les pratiques de travail des utilisateurs visés

## 2. Guidage

## 2. Guidage

Objectif : faciliter l'utilisation du système et son apprentissage

Mécanismes de retour (*feed-back* et *feed-forward*) pour

- réduire la charge cognitive
- prévenir des situations d'erreur
- rassurer (ex : progression d'une opération)

Quelques données sur la perception du temps de réponse :

- $t < 0.1$  sec : perçu comme instantané
- $t < 1$  sec : le délai est perçu, mais ne perturbe pas l'utilisateur
- $t > 10$  secondes : l'utilisateur va vouloir faire autre chose en attendant

Pour les longs délais : changement de curseur, barre de progression ou animation aléatoire

## 2. Guidage (suite)

Recommandations : incitation (*feed-forward*)

- griser les commandes non disponibles
- fournir la liste des saisies attendues
- donner le format de saisie des données
- donner des valeurs par défaut « intelligentes »

Recommandations : retour utilisateur (*feed-back*)

- à toute action de l'utilisateur doit répondre un changement de présentation de l'interface
- indiquer les modes de fonctionnement du système
- signaler les traitements longs par un message d'attente
- toujours faire apparaître les saisies de l'utilisateur
- rendre visibles les traitements réalisés par le système

## 2. Guidage (fin)

Recommandations : groupement/distinction (design graphique)

- regrouper les informations de même type par le format ou la position
- distinguer par une présentation différente les informations distinctes

Recommandations : lisibilité (typographie)

- utiliser une police droite
- employer des lettres sombres sur fond clair

## 3. Homogénéité

## 3. Homogénéité

Concerne la cohérence globale de l'interface

- interne : à l'intérieur d'une application
- externe : entre les applications, avec la métaphore du système

Principe : un système qui a l'air familier est perçu comme simple à utiliser

Intérêt : faciliter l'apprentissage et l'utilisation

Risque : freiner ou bloquer l'évolution (inertie des standards)

## 3. Homogénéité (suite)

Recommandations

- les fenêtres doivent suivre le même schéma d'agencement
  - exemple : boîte de recherche en haut à droite
- la sémantique des boutons de la souris doit être constante
- le même vocabulaire doit être utilisé pour désigner les commandes du logiciel
  - exemple : ouvrir / copier-coller / préférences / ...
- la syntaxe des commandes doit être cohérente sur l'ensemble de l'interface

La consistance ne concerne pas que l'apparence visuelle

Autres exemples : syntaxe et effets des commandes

### 3. Homogénéité (suite)

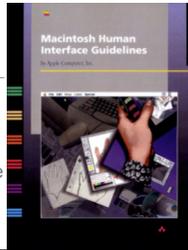
Guides de style :

- publiés par les concepteurs d'interfaces graphiques
- décrivent le "look and feel" d'une plateforme
- sont souvent vicieux : faciliter la vie de ceux qui les suivent et compliquer celle de ceux qui ne les suivent pas...

Exemples :

- OSF Motif
- IBM CUA
- Apple Human Interface Guidelines
- MS Windows

Bons sur le principe, mais peuvent être difficiles à suivre  
Implémentés (en partie) dans les boîtes à outils d'interface



### 3. Homogénéité (suite)

Motif style guide v1.1 : *MessageDialogs should be used to convey a message to the user. They must not interrupt the user's interaction with the application. They should include a message, and one of the following button arrangements.*

- |                    |                   |
|--------------------|-------------------|
| OK                 | Cancel            |
| OK Help            | Cancel Help       |
| OK Cancel          |                   |
| OK Cancel Help     |                   |
| Yes No             | Retry Cancel      |
| Yes No Help        | Retry Cancel Help |
| Yes No Cancel      |                   |
| Yes No Cancel Help |                   |

### 3. Homogénéité (fin)

Macintosh Human Interface Guidelines

Figure 5-11 Examples of correct and incorrect window titles

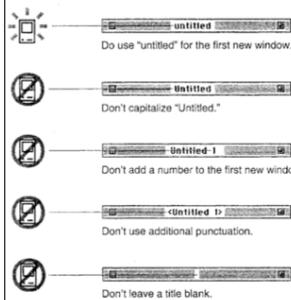


Table 10-1 Pointers

Pointer	Name	Used for
	Arrow	Scroll bar, other controls, size box, title bar, close box, zoom box, menu bar, desktop
	Crosshairs	Drawing, shrinking, or stretching graphic objects
	I-beam	Selecting and inserting text
	Plus sign	Selecting fields in an array
	Wristwatch	Showing that a lengthy operation is in progress

### 4. Flexibilité

#### 4. Flexibilité

Capacité de l'interface à s'adapter à différents contextes d'utilisation

Recommandations :

- permettre l'activation des commandes au clavier ou à la souris
- autoriser le déclenchement d'une commande fréquente depuis plusieurs fenêtres de l'application
- permettre à l'utilisateur de paramétrer le logiciel selon ses préférences
- fournir un moyen rapide d'accéder aux commandes des menus

Peut être contradictoire avec la concision

#### 4. Flexibilité (suite)

Les utilisateurs expérimentés doivent pouvoir se servir rapidement des opérations les plus courantes

Solutions :

- icônes
- abréviations (Ctrl-s)
- mnémoniques (Alt-f-e)
- touches spéciales (Xerox Star: move, copy, properties...)
- complétion automatique, frappe anticipée (*type ahead*)
- mécanismes de navigation, historiques
- *redo*

#### 4. Flexibilité (fin)

**Adaptabilité** : personnalisation explicite par l'utilisateur

- lexique
- préférences de présentation
- modalités du dialogue

Attention à l'interface de configuration des préférences : d'autant plus difficile que l'application est générale

Attention à ne pas transformer l'utilisateur en concepteur

**Adaptativité** : personnalisation dynamique sans action explicite de l'utilisateur

L'adaptativité est très controversée...

- non prédictibilité
- rupture du principe de cohérence

#### 5. Traitement des erreurs

## 5. Traitement des erreurs

Regroupe les différents moyens visant à protéger l'utilisateur des erreurs et à corriger celles-ci

Recommandations générales :

- éviter les erreurs (!)
- détecter les erreurs au plus tôt
- permettre une correction aisée des erreurs
- faciliter l'exploration et l'apprentissage du système

## 5. Traitement des erreurs (suite)

Les utilisateurs vont faire des erreurs (c'est certain)

Exemples :

- *habitude* : créer un nouveau document à partir d'un document existant, et l'enregistrer sans lui donner un nouveau nom
- *incompréhension* : croire que l'on déplace une copie d'un objet alors qu'on déplace l'objet
- *exploration* : essayer un bouton que l'on ne connaît pas
- *fausse manipulation* : cliquer sur la commande voisine de celle que l'on vise

Solutions (entre autres) :

- empêcher les manipulations qui produisent des erreurs
- faciliter la détection et la correction d'erreur par le feed-back et un historique (*défaire - refaire*)

## 5. Traitement des erreurs (suite)

Recommandations : protection contre les erreurs

- feed-forward:
  - griser les commandes non disponibles
  - fournir la liste des valeurs possibles, et de bonnes valeurs par défaut
- minimiser les saisies clavier
- détecter les erreurs au plus tôt
- limiter les pertes de données (sauvegarde automatique, journal)
- valider explicitement les commandes irréversibles

Recommandations : correction des erreurs

- placer le message d'erreur là où l'utilisateur est censé regarder
- mettre en évidence le champ erroné
- afficher des messages d'erreur explicites, brefs, non réprobateurs et auto-suffisants

## 5. Traitement des erreurs (suite)

L'utilisateur n'aime pas se sentir "piégé »

- Engagement prématuré : forcer à prendre une décision trop tôt
- Essayer de toujours lui laisser une porte de sortie clairement identifiée

Solutions :

- bouton Annuler (*cancel*)
- bouton Reset/Default pour annuler des changements
- commande Défaire (*undo*), avec historique et Refaire (*redo*)
- commande Interrompre (et/ou Pause)
- commande Quitter

## 5. Traitement des erreurs (suite)

On peut tirer partie de la formation des habitudes

- la répétition et l'entraînement développent des habitudes
- *The ideal humane interface would reduce the interface component of a user's work to benign habituation. Many of the problems that make products difficult and unpleasant to use are caused by human-machine design that fails to take into account the helpful and injurious properties of habit formation.* (J. Raskin)

Réduire le nombre de choix peut favoriser le développement d'habitudes

Les habitudes sont sources d'erreurs :

- valider sans réfléchir une boîte de confirmation

## 5. Traitement des erreurs (suite)

Les messages d'erreur sont importants



Le message doit être rédigé dans des termes compréhensibles par l'utilisateur, si possible reliés à la tâche qu'il essayait d'accomplir  
 Eviter de donner l'impression à l'utilisateur qu'il est stupide  
 Essayer d'inclure une façon de corriger l'erreur

- Exemple : manque de place -> vider la corbeille

## 5. Traitement des erreurs (suite)

Prévoir un **système d'aide**

Remarque : le système d'aide n'est pas là pour aider à faire passer une interface mal conçue

La plupart des utilisateurs ne lisent pas la documentation

Situation la plus courante : le « mode panique »

- besoin ponctuel, spécifique et urgent
- nécessite une documentation en ligne avec des mécanismes d'indexation

## 5. Traitement des erreurs (fin)

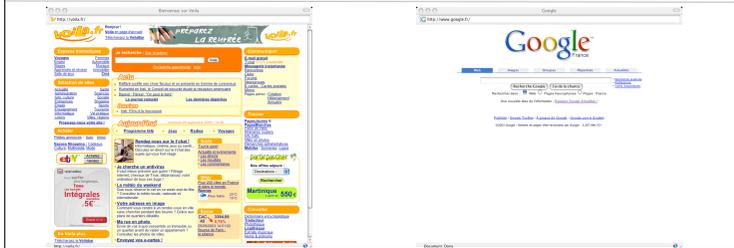
L'aide peut prendre différentes formes :

- tutoriels, démonstrations (« comment créer un tableau »)
- manuel de référence (table des matières, index, etc.)
- mémento (cartes de référence, *tip of the day*, *tooltips*, glossaire)
- aide contextuelle (*qu'est ce que c'est ?*)
- assistants

Exemples d'aide contextuelle :

- où suis-je ?
- où est ... ?
- et maintenant ?
- que s'est-il passé ?
- pourquoi est ce que ça ne ... ?

## 6. Concision



## 6. Concision

Ensemble des moyens visant à réduire la charge perceptive et mnésique de l'utilisateur

Recommandations :

- n'afficher que les informations pertinentes (mais cela dépend de ce que veut faire l'utilisateur)
- réduire le nombre d'actions élémentaires pour atteindre un objectif donné
- minimiser les saisies/lectures
- éviter les textes trop verbeux
- ne pas demander de saisir des informations qui peuvent être déduites
- éviter à l'utilisateur d'avoir à se souvenir d'informations
- ne pas l'obliger à faire des calculs qui peuvent être automatisés

## 6. Concision (suite)

Présenter uniquement les informations nécessaires

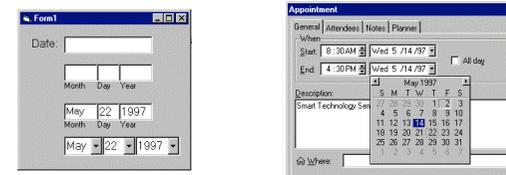
- *less is more* : limiter le nombre de widgets et de fenêtres qui introduisent des manipulations supplémentaires
- cacher ou supprimer les informations non essentielles (cf. *progressive disclosure* du Star)
- utiliser si possible l'ordre naturel des choses

Préférer les techniques de manipulation directe aux techniques nécessitant la saisie de commande ou d'arguments (*point and click instead of remember and type*)

## 6. Concision (fin)

Préférer un petit nombre de commandes génériques

Expliciter les contraintes existantes :



*Reconnaître* est plus facile que *se souvenir*, mais bien souvent, en terme de représentation graphique, *less is more...*