

Source : dreamstyle.com

# V- Les critères ergonomiques

## 2- Des applications pour le web

### 2.4 – Communiquer avec l'utilisateur

iutenligne

I.U.T  
MONTPELLIER - SETE

## 2.4- Communiquer avec l'utilisateur

- ▣ 2.4.1- Le langage de l'interface
- ▣ 2.4.2- Internationalisation des interfaces
- ▣ 2.4.3- Traiter les erreurs
- ▣ 2.4.4- Aider l'utilisateur

## 2.4.1- Le langage de l'interface

- ▣ Des textes simples, non ambigus, facilitent l'apprentissage et la mémorisation
- ▣ La simplicité et l'ambiguïté n'est pas intuitive pour un concepteur, elle peuvent ne pas être les mêmes pour un utilisateur
- ▣ Il faut donc **employer le langage de l'utilisateur** et tester la compréhension de ce langage (interview par exemple pour identifier un certain vocabulaire – voir chapitre démarche)
- ▣ Le langage passe par
  - ▣ les libellés des commandes,
  - ▣ les messages
  - ▣ l'écriture web

## 2.4.1- Le langage de l'interface

### Les libellés des commandes

- ❑ Préférer des textes à des icônes
- ❑ Un seul et unique libellé pour chaque commande
- ❑ Utiliser une syntaxe homogène pour le libellé
- ❑ Eviter les verbes à la forme nominale : préférer Sauvegarder (action) plutôt que Sauvegarde (état)
- ❑ Eviter les abréviations : augmentent la charge cognitive – la mémorisation
  - ❑ Si obligatoires : rester homogène dans la construction des abréviations (choisir une règle - trois premières lettres, initiales du mot, ... - et l'utiliser à chaque fois) – et vérifier la compréhension par les utilisateurs (tests)
  - ❑ Plus l'abréviation est lisible et sa taille est courte, mieux elle est mémorisée

## 2.4.1- Le langage de l'interface

### Les messages

- ▣ Permettent au système de communiquer explicitement avec l'utilisateur soit pour le questionner afin de déterminer les actions à venir, soit pour expliquer un fonctionnement
- ▣ Définir le rôle et le contenu des différents types de messages de façon homogène
  - ▣ Définir une typologie pour les types de messages : messages d'erreur, message de demande de confirmation, message d'information, ... et associer à chaque type un gabarit de fenêtre et un format de texte.
- ▣ Employer une fenêtre de message lorsqu'on veut que le message soit lu
- ▣ Afficher le message près de l'objet auquel il est lié

## 2.4.1- Le langage de l'interface

### La rédaction des messages

- ❑ Les messages doivent être brefs, concis, pertinents, clairs, affirmatifs, adaptés au niveau de connaissance de l'utilisateur
- ❑ Les messages ne doivent pas être critiques mais constructifs :
  - ❑ Critique : « le nom de fichier n'est pas valide »
  - ❑ Constructif : « un nom de fichier ne peut pas contenir les caractères suivants /: »?<> »
- ❑ Eviter les impasses, expliquer la marche à suivre
  - ❑ Impasse « Saisissez votre code » / « Saisissez votre code puis appuyez sur *Validation* »
- ❑ Respecter l'ordre dans lequel les actions doivent être exécutées
- ❑ Homogénéiser le vocabulaire

## 2.4.1- Le langage de l'interface

### La rédaction des messages

Rester cohérent avec le reste de l'interface : adéquation entre le terme utilisé et l'action utilisateur attendue

- L'action *Tapez* peut être ambiguë :

Message	Action attendue
<b>Saisissez</b> votre code	Saisie clavier
<b>Appuyez</b> sur <i>Enter</i>	Appui sur une touche
<b>Tapez</b> <i>Enter</i>	Appui ou saisie ?

- La typographie adaptée facilite la lecture des messages :
  - ▣ **Déconseillé** : « Appuyer sur F1 pour afficher l'aide et sur F5 pour rafraichir l'image »
  - ▣ **Préférable** : « Appuyer sur **F1** pour afficher l'aide et sur **F5** pour rafraichir l'image »

## 2.4.1- Le langage de l'interface

### L'écriture web

- ❑ L'internaute se sert du web pour gagner du temps, il ne lit pas les pages mot à mot, il les « balaye » en diagonale à la recherche de ce qui l'intéresse.
- ❑ La lecture sur écran est plus lente que sur du papier : le texte doit être rédigé différemment
- ❑ **Les points-clés d'une rédaction efficace :**
  - ❑ Aller à l'essentiel : être concis
  - ❑ Rédiger simplement
  - ❑ Être objectif : énoncer des faits, du concret
  - ❑ Structurer l'accès aux informations
  - ❑ Le contenu de chaque page doit être auto-suffisant
  - ❑ Publier des informations utiles
  - ❑ Commencer par la conclusion : **principe de pyramide inversée**



## 2.4.1- Le langage de l'interface

### *L'écriture web : le principe de pyramide inversée*

- ❑ Le titre doit décrire explicitement le contenu de la page ou du paragraphe :
  - ❑ Élément essentiel de l'utilisabilité d'un site, permet d'identifier le contenu et d'aider l'utilisateur à vérifier qu'il est sur la page qu'il souhaite
  - ❑ Repère visuel important : doit être vu dès l'ouverture, à placer en haut à gauche – utiliser une forme graphique spécifique pour le repérer facilement
- ❑ Faire ressortir les éléments clés du texte
  - ❑ La typographie doit accrocher l'œil : mettre en gras, utiliser des puces
- ❑ Le texte du lien doit être cohérent avec le titre de la page
- ❑ Les liens doivent pointer vers les éléments complémentaires
- ❑ Justifier les textes à gauche
- ❑ Ne pas souligner les textes
- ❑ Permettre à l'utilisateur d'imprimer le texte

## 2.4.1- Le langage de l'interface

### *Vérifier l'efficacité rédactionnelle du contenu d'une page*

- ▣ Qui a écrit la page ? A qui s'adresse-t-elle ?
- ▣ Que contient la page ?
- ▣ Où ? Information de lieu, si nécessaire.
- ▣ Quand ? Date de mise à jour ou autre information de temps, si nécessaire
- ▣ Comment ? Comment fonctionne la page ? (ex : Mode d'emploi d'un service en ligne)
- ▣ Pourquoi ? Quel est l'objectif de la page ?

## 2.4.2- Internationalisation des interfaces

Comment traduire l'interface pour qu'elle soit compréhensible, sans ambiguïté par un utilisateur étranger ?

### La page d'accueil

- ▣ Diriger l'internaute vers la page d'accueil qui est dans sa langue : processus de détection
  - ▣ Par le biais de l'adresse IP
  - ▣ Si la zone détectée ne correspond pas à la langue du navigateur, le site affiche une fenêtre de sélection de la langue (ex : touriste chinois en France)
  - ▣ Si la position géographique ne peut pas être détectée ou qu'une ambiguïté existe (ex : en Belgique, entre le français et le flamand), préférable de demander à l'internaute de choisir la langue qu'il préfère

## 2.4.2- Internationalisation des interfaces

### Choisir sa langue

- ▣ Permettre à l'utilisateur de changer facilement sa langue
- ▣ Employer la langue cible ou un drapeau pour le lien vers la version/ langue correspondant (ex : pour la version française : écrire Français et non French)

### Traduction

- ▣ Il est préférable de ne pas traduire plutôt que de prendre le risque d'une mauvaise traduction
- ▣ Employer les unités locales (longueurs, poids, températures, monnaies, codes postaux, formats de numéros de téléphone, taille des vêtements, chiffres (1,000 signifie mille aux Etats-Unis et 1 en France))
- ▣ Conserver la même architecture pour les différentes langues du site
- ▣ Préférable de traduire tout le site plutôt que certaines parties
- ▣ Ne pas utiliser de texte dans les images pour pouvoir le modifier partout

## 2.4.2- Internationalisation des interfaces

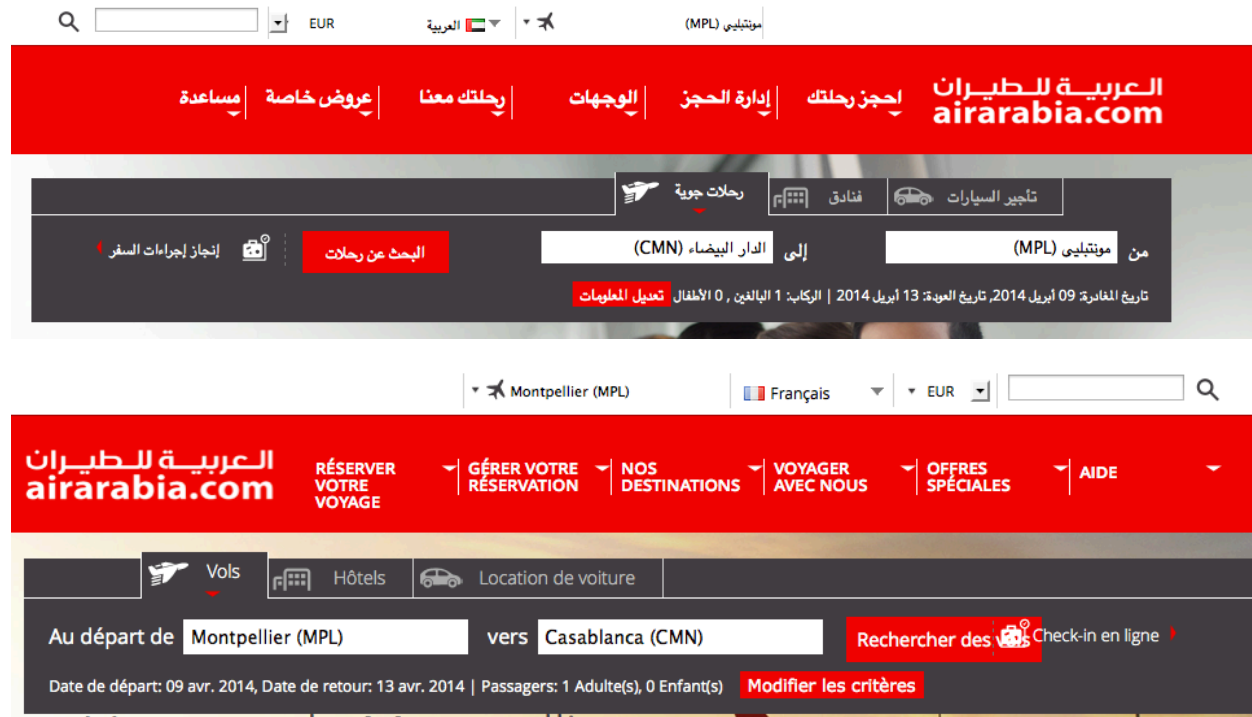
### Localiser l'interface

- ❑ Informer au plus tôt l'utilisateur sur les services offerts pour son pays : conditions de livraison ...
- ❑ Adapter le contenu du site à la localisation de l'internaute : législation du pays, visualisation de certaines vidéos impossible dans la langue ...
- ❑ Adapter le formulaire pour chaque pays : informations postales (ex : code postal), syntaxe de l'adresse (ex : numéro de rue placé à des endroits différents), saisie des dates (ex : mois avant ou après le jour), des horaires (ex : sur 24h ou AM/PM)

## 2.4.2- Internationalisation des interfaces

### Polices de caractères

- Adapter l'agencement de la page au sens de lecture



Version arabe du site airarabia.com

Logo à droite / moteur de recherche à gauche

Version française du site airarabia.com

Logo à gauche / moteur de recherche à droite

## 2.4.2- Internationalisation des interfaces

### Polices de caractères

- ▣ La taille du texte doit prendre en compte les caractères de l'écriture utilisée
- ▣ Utiliser l'écriture du pays
- ▣ En règle générale, l'écriture horizontale est préférée sur les interfaces
- ▣ Utiliser des formes adaptées (ex : souligner un texte en japonais peut être cause d'incompréhension)

### Spécificités culturelles

- ▣ Certaines images peuvent être interprétées différemment selon les cultures (les choisir avec précaution et les tester)
- ▣ Les icônes de l'interface doivent s'appuyer sur une symbolique universelle
- ▣ Prendre en compte la manière dont les couleurs sont interprétées (ex : le rouge est associé au mariage en Asie, au danger en Occident / le blanc est associé au mariage en Occident, au deuil en Asie)

**Méthodologie :** La majeure partie des choix doit être testée : conduire une étude de terrain avec des intervenants locaux, les utilisateurs des différents pays

## 2.4.3- Traiter les erreurs

Les erreurs sont mal acceptées par les utilisateurs car elles perturbent la façon dont ils ont prévu de mener la tâche et rallongent le temps de réalisation.

**Minimiser le risque d'erreur améliore l'utilisabilité de l'interface.**

Mais « l'erreur est humaine » :

- ❑ **Erreur d'intention** : la manière de faire est correcte mais l'action ne permet pas d'atteindre le but visé (l'utilisateur n'a pas utilisé la commande correcte par incompréhension du sens des commandes ou de l'enchaînement des opérations)
- ❑ **Erreur d'exécution** : l'intention est correcte mais la façon de faire n'est pas bonne (l'utilisateur a tapé par inadvertance sur une touche autre que celle souhaitée)



## 2.4.3.1- Les types d'erreurs

Pour **éviter les erreurs d'intention**, l'utilisateur doit comprendre le système.

- ▣ Faciliter l'apprentissage permet de diminuer ce type d'erreur : permet à l'utilisateur de se construire un modèle mental (voir Partie III : L'utilisateur) correct du fonctionnement du système et commet moins d'erreur
- ▣ Quelques recommandations :
  - ▣ Guider l'utilisateur,
  - ▣ Fournir une visibilité sur le fonctionnement du système,
  - ▣ Fournir un retour aux actions de l'utilisateur,
  - ▣ Rendre possible l'exploration du logiciel,
  - ▣ Fournir une aide en ligne adaptée à la tâche (« Comment faire pour ? »)

## 2.4.3.1- Les types d'erreurs

**Les erreurs d'exécution** correspondent à un « mauvais fonctionnement » du processus humaine, elles trouvent leur origine dans chacun des sous-systèmes : **perceptif**, **cognitif** et **moteur**

- ▣ **Les erreurs perceptives** sont souvent dues à une présentation similaire des objets (ex : confusions entre lettres et chiffres B8 / Z2 / I1 / S5 / O0)
  - ▣ Utiliser des techniques de mise en évidence
  - ▣ Rendre clairement visibles les changements de mode et les états du système (le mode *Majuscule* – touche *verrouillage des majuscules* - ne se voit pas suffisamment et provoque des erreurs – seule une diode sur le clavier indique que la touche est activée)
  - ▣ Fournir un feed-back immédiat aux actions de l'utilisateur (il arrive qu'on appuie plusieurs fois sur une touche en l'absence de retour visuel et/ auditif)

## 2.4.3.1- Les types d'erreurs, les erreurs d'exécution

- ▣ **Les erreurs cognitives** sont liées à des difficultés de traitement de l'information (de compréhension et de mémorisation)
  - ▣ Diminuer le travail de mémorisation (voir Partie III : L'utilisateur) : devoir se souvenir d'une valeur d'une fenêtre à une autre présente un risque d'erreur
  - ▣ Minimiser le calcul mental : l'utilisateur ne doit pas avoir à calculer une valeur qui peut l'être par le système
  - ▣ Fournir des aides mnémoniques : insérer des info-bulles pour rappeler à l'utilisateur ce que signifie la rubrique d'un formulaire
  - ▣ Concevoir un mode de dialogue et une présentation cohérente : source d'erreur si les mêmes commandes se retrouvent à des positions différentes selon le menu ou la fenêtre
  - ▣ Mettre en jeu la reconnaissance plutôt que le souvenir

## 2.4.3.1- Les types d'erreurs, les erreurs d'exécution

- ❑ **Les erreurs motrices** sont des erreurs de manipulation, plus les actions doivent être exécutées rapidement, plus le risque d'erreur augmente.
  - ❑ Agrandir les objets à sélectionner
  - ❑ Prévoir les appuis accidentels : mettre en évidence le bouton d'action le plus probable, l'associer à un dialogue de confirmation « Etes-vous sûr de vouloir supprimer ce document ? »
  - ❑ Eviter de changer trop fréquemment de moyen d'interaction : source d'erreur si différents modes de dialogue : le clavier puis la souris, puis à nouveau le clavier
  - ❑ Minimiser les saisies clavier
  - ❑ Homogénéiser les séquences de manipulation : respecter une logique d'utilisation
    - ❑ Les commandes doivent toujours être ordonnées de la même manière, au même emplacement
    - ❑ Lorsque certaines commandes sont inactives, il est préférable de les rendre inactives (les griser) mais présentes, plutôt que de les supprimer ce qui demande de réordonner l'ensemble et crée une incohérence

## 2.4.3.2- Les messages d'erreurs (1/3)

Plusieurs type de problèmes peuvent survenir :

- ❑ *L'interface bloque l'utilisateur* et l'empêche de continuer tant qu'il n'a pas corrigé l'erreur (ex : pas de possibilité de revenir sur une fenêtre en arrière plan). Convient aux erreurs grave pour lesquelles il y a un risque d'action définitive comme la destruction de données. Ne doit être utilisé qu'à titre exceptionnel car nuit au dialogue.
- ❑ *Le logiciel met en garde l'utilisateur par un message* (ou signal sonore) mais permet à l'utilisateur de continuer. Adapté aux erreurs de moindre importance.
- ❑ *Le logiciel ne fait rien*, ne répond pas à la commande erronée : erreurs sur des commandes à manipulation directe – peu explicite, il est préférable d'afficher un message en complément.
- ❑ Lorsque cela est possible, **le logiciel doit corriger automatiquement l'erreur** mais c'est difficile à mettre en œuvre comme la correction automatique de fautes de frappe.

## 2.4.3.2- Les messages d'erreurs (2/3)

- ▣ Le message d'erreur doit préciser la nature du problème et donner le moyen d'y remédier
- ▣ Le message doit être **coopératif** – il permet à l'utilisateur de mieux comprendre le système
  - ▣ Ex : préférer « Erreur : le répertoire de destination manque » à « Erreur : syntaxe incorrecte »
- ▣ Le message doit fournir une **description du problème**
  - ▣ Ex : préférer « Disque plein » à « Erreur d'écriture sur le disque »
- ▣ Eviter les sources **d'ambiguïté**
  - ▣ Ex : préférer « Il manque l'extension du fichier » à « Fichier non trouvé »

## 2.4.3.2- Les messages d'erreurs (3/3)

- ▣ Adapter le message au **niveau de connaissances** de l'utilisateur
  - ▣ Ex : préférer « Le nombre de billets doit être inférieur à 10 sinon utilisez la commande Billets groupés » à « Erreur dans le nombre de billets »
- ▣ **Ne pas culpabiliser** l'utilisateur, ni dramatiser
  - ▣ Ex : préférer « Erreur : valeur non reconnue » à « Erreur fatale : Mauvaise saisie !!! »
  - ▣ Eviter les termes : désastreux, catastrophique, fatal, illégal, mauvais, impossible ...
- ▣ Eviter l'anthropomorphisme : l'utilisateur doit **conserver le sentiment de maîtrise** du logiciel
  - ▣ Ex : préférer « Commande non acceptée » à « Désolé, je ne peux pas accepter cette commande »
  - ▣ **Le logiciel reste une machine.**

## 2.4.3.3- La prévention des erreurs

- Lorsqu'il y a un risque de perte de données ou que l'utilisateur active une commande difficilement réversible, un message doit signaler ce fait et **demander confirmation**
  - Chaque commande doit être validée par une action physique de l'utilisateur
  - Eviter qu'une commande importante ne soit déclenchée par inattention
  - Il est possible d'offrir la possibilité de désactiver cette protection pour les utilisateurs expérimentés
- **Rendre inactives** les commandes indisponibles
- **Rendre actifs les champs** en fonction du contexte : le champ d'un formulaire ne s'affiche que s'il est nécessaire de le remplir (Ex : le trajet retour d'un billet si la case Aller-Retour a été cochée)
- **Tester pour identifier les erreurs** : les erreurs d'intention sont difficiles à prévoir mais elles apparaissent dès les premiers tests d'utilisabilité



## 2.4.3.4- La correction des erreurs

Une erreur est une perte de temps pour l'utilisateur. Pour perturber le moins possible la planification de la tâche :

- ▣ Signaler l'erreur **au plus tôt**
- ▣ **Placer le message** là où l'utilisateur est censé regarder lorsque l'erreur se produit (dans la fenêtre courante ou près du champ erroné)
- ▣ Fournir la **possibilité de modifier** facilement les données saisies, une fois l'erreur signalée
- ▣ **Permettre le retour** en arrière : la faculté d'annuler améliore la performance de l'utilisateur car il travaille plus rapidement craignant moins les erreurs.
- ▣ **Autoriser les interruptions** pour les commandes longues

## 2.4.4- Aider l'utilisateur

L'aide est généralement utilisée en cas de problème et souvent dans un contexte d'urgence.

**Peu employée** car souvent conçue dans une optique de formation et répond rarement à l'erreur rencontrée de façon immédiate.

- ▣ Permettre l'accès à l'aide en ligne
- ▣ Structurer l'aide en s'appuyant sur la tâche de l'utilisateur et non sur les fonctionnalités de l'application
- ▣ L'aide doit participer à l'apprentissage du logiciel pour que l'utilisateur se construise un modèle correct de son fonctionnement

## 2.4.4- Aider : Comment ?

- ▣ Les **bulles d'aide** répondent à la question « Qu'est-ce que ? » : elles expliquent le rôle d'un objet dans l'interface. Le texte doit être court et facile à comprendre.
- ▣ Le **manuel en ligne** répond aux questions « Qu'est-ce que ? » et « Comment faire pour ? ». Structuré pour guider l'utilisateur dans sa recherche avec des mots-clés qui lui permettent de cibler les solutions proposées. Doit être détaillé pour retrouver le même contexte que la tâche que l'utilisateur est en train de faire.
- ▣ Le **didacticiel** est un outil de formation, il facilite la prise en main et induit une attitude positive de la part de l'utilisateur. Il est long à utiliser et pas toujours adapté à des problèmes spécifiques.
- ▣ **Aide directe par un conseiller** dans une fenêtre de discussion : personnalisation de la réponse pour l'utilisateur.

## 2.4.4- Aider l'utilisateur : utilisabilité

La recherche de l'aide est une tâche supplémentaire pour l'utilisateur, il faut **minimiser l'interruption de la tâche** en cours car l'accès à l'aide modifie la continuité de l'activité de l'utilisateur :

- ❑ Fournir un **accès rapide** à l'aide en ligne : bouton visible en permanence
- ❑ Décrire l'utilisation du système en s'appuyant sur le **vocabulaire métier** de l'utilisateur : adaptation à la tâche
- ❑ Permettre **l'exploration** (Ex : rubrique « Voir aussi »)
- ❑ Les titres doivent être **explicites** : aident au repérage et la compréhension de la structure
- ❑ **Illustrer** les descriptions avec des exemples : utiliser des diagrammes
- ❑ Utiliser des **listes** pour organiser les procédures : un texte structuré est plus facile à mémoriser, la numérotation donne une indication sur les étapes de la procédure

## 2.4.4- Rédiger une aide efficace

- ❑ Utiliser des phrases **courtes et simples**, éviter les subordonnées et les relatives
- ❑ Employer la **forme active**
- ❑ **Eviter** les négations
- ❑ Utiliser un **vocabulaire** connu de l'utilisateur
- ❑ Faire des paragraphes courts, Séparer les paragraphes par une ligne blanche
- ❑ Faire des lignes entre 26 et 80 caractères
- ❑ Eviter les coupures de mot
- ❑ **Ne pas justifier à droite** : augmente le temps de lecture si l'espacement n'est pas régulier
- ❑ **Espacer les lignes** de la hauteur d'un caractère, utiliser « double interligne ».
- ❑ Présenter les informations dans l'**ordre** dans lequel l'utilisateur doit les prendre en compte
- ❑ **Indiquer le but** en premier, puis l'action pour réaliser ce but
- ❑ A chaque étape de la procédure, correspond une phrase

## 2.4.4- Rédiger une aide efficace

- ▣ L'aide n'en est pas une (Nielsen, 1993 ; voir Partie IV-Démarche) :
  - ▣ Ce n'est pas l'aide en ligne qui aide l'utilisateur, ce sont les différents moyens mis en œuvre dans l'interface afin de le guider et lui permettre d'éviter des erreurs.
  - ▣ L'aide en ligne ne résout pas les problèmes d'utilisabilité.
  - ▣ C'est par une conception de l'interface guidée par l'utilisation que l'on évite les problèmes et non par l'aide en ligne qui a uniquement pour rôle d'expliquer le fonctionnement du logiciel.