

# Concepteur-architecte informatique

UE : NSY110

## • Interaction homme-machine •

Copies et diffusion non autorisées sans accord écrit.  
Document provisoire.

Année 2011/12  
François Lacomme

# 1. Ergonomie des systèmes d'information

## 1.1. Généralités

Qu'est-ce qui fait le succès d'un produit ?

- ✓ La prouesse technique ?
- ✓ Sa capacité à répondre au besoin des utilisateurs ?
- ✓ Son utilisabilité (soit sa capacité à être facilement utilisé) ?

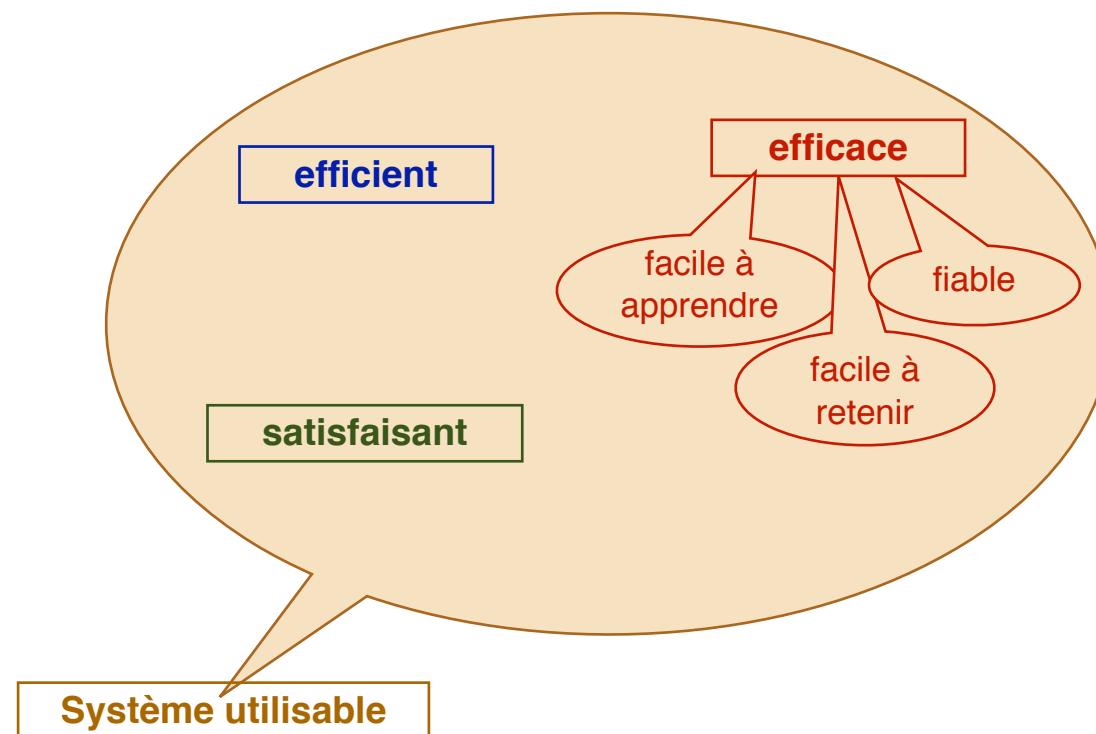
L'usage d'un produit ou d'un instrument est caractérisé par :

- ✓ **son utilité** ; soit sa capacité à faciliter la réalisation d'une activité ;
- ✓ **son utilisabilité** ; soit sa facilité d'emploi, ou encore son ergonomie.  
(*Usability* pour les ergonomes anglo-saxons).



La norme ISO 9241-11 définit l'utilisabilité comme :

« le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec **efficacité**, **efficiency** et **satisfaction**, dans un contexte d'utilisation spécifié »



L'efficience est la capacité d'atteindre le résultat avec le minimum d'effort, de dépense ou de temps.

L'**utilisabilité** est **essentielle**, car :

- ✓ elle détermine la performance des utilisateurs ;
- ✓ elle favorise la réalisation rapide des tâches, sans perte de temps, avec moins de stress ;
- ✓ elle conditionne la réussite commerciale du produit ou du projet.

La réussite d'un projet informatique, outre les aspects purement techniques, dépend donc de la prise en compte de l'utilisabilité, et cela tout au long du processus de développement.

L'évaluation de l'utilisabilité d'un logiciel se fait grâce à des **tests d'utilisabilité**.

- ✓ On place l'utilisateur final en situation effective d'utilisation du logiciel ;
- ✓ on mesure les performances de l'utilisateur ;
- ✓ on relève :
  - ▶ les problèmes qu'il rencontre,
  - ▶ les questions qu'il se pose et
  - ▶ les fonctionnalités qu'il apprécie ou non.

Nous allons faire, dans ce cours, la synthèse des **règles et recommandations** qui résultent des études et expérimentations réalisées depuis plusieurs années dans le domaine de l'ergonomie du logiciel.

Le domaine d'application couvre aussi bien le **développement de logiciel** que la **conception et le design de site web**. Nous appliquerons les conseils pratiques et les méthodes présentées à travers des exercices ou mini-projets.

## Quelques définitions...

- ✓ **Ergonomie** : Recherche d'une meilleure adaptation entre une fonction, un matériel et son utilisateur.
- ✓ **Interface** :
  - Dispositif (matériel et logiciel) grâce auquel s'effectuent les échanges d'informations entre deux systèmes.
  - Limite commune à deux systèmes.
- ✓ **Expérience utilisateur** : Ensemble des connaissances qu'une personne acquiert et des perceptions qu'elle développe, lorsqu'elle utilise des produits ou des ressources informatiques.
- ✓ **HCI** (Human Computer Interface, ou Human Computer Interaction) : Interface personne-système, ou interface homme-machine

[ Le cours est prévu suivant le plan suivant :

- ✓ Ce premier chapitre expose différents critères ergonomiques et donne des recommandations afin de mettre en œuvre ces critères.
- ✓ Des notions sur les sciences cognitives nous éclaireront sur l'être humain.
- ✓ Le chapitre suivant, L'écran et la page, introduit des notions liées à la PAO et la chaîne de l'édition de publications, avant de se focaliser sur l'affichage des informations à l'écran et sur le graphisme de l'interface.
- ✓ Un chapitre, Le dialogue homme-machine, indique comment organiser les contenus et composants (fenêtres, dialogues, menus, boutons, composants de sélection, champs de saisie, etc.) d'une interface ergonomique.
- ✓ Les erreurs et l'aide en ligne seront abordées ensuite.
- ✓ Nous surferons ensuite vers la conception de sites web, sur des vagues de la navigation web et des aspects graphiques spécifiques à internet.
- ✓ En guise de conclusion, nous aborderons la pratique de l'utilisabilité, pour comprendre les avantages d'une telle pratique pour le développement de logiciels.

]

Un premier schéma du **processeur humain** est proposé en figure 1.1 ; d'autres schémas seront proposés avec le chapitre 2.

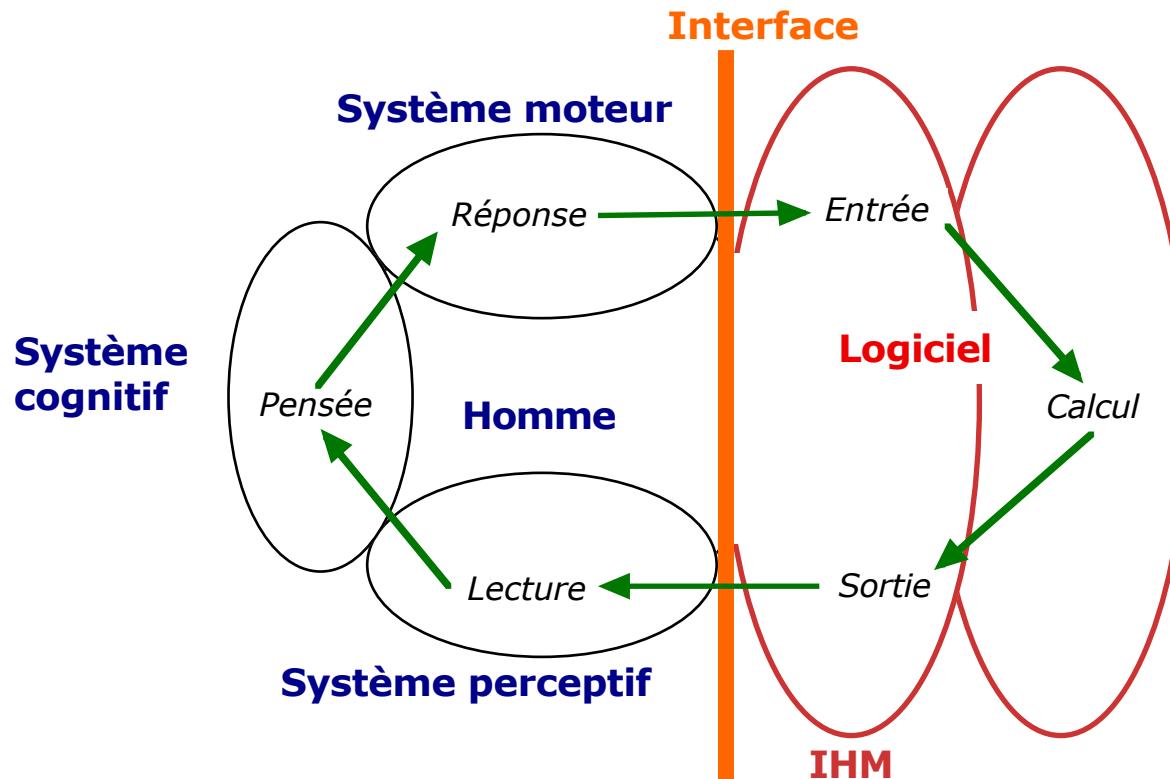


Figure 1.1 — Le processeur humain ; parallèle avec le logiciel

Plongeons maintenant dans les critères ergonomiques...

## 1.2.Compatibilité

La **compatibilité**, c'est la capacité du logiciel à s'intégrer dans l'activité des utilisateurs. L'objectif est de réduire le transfert de connaissance entre le **métier** et l'utilisation du logiciel.

La **logique d'utilisation** doit donc correspondre à la **logique de l'utilisateur**.

*Recommandations :*

- ✓ Parler le langage de l'utilisateur.
- ✓ Présenter les informations de façon cohérente par rapport aux autres supports de travail.
- ✓ Rendre l'accès aux commandes compatible avec la tâche de l'utilisateur.

### 1.3.Guidage

Le **guidage** est lié aux critères suivants :

- ✓ Incitation ;
- ✓ Groupement/Distinction ;
- ✓ Retour utilisateur ;
- ✓ Lisibilité.

L'objectif est de faciliter l'utilisation du logiciel et son apprentissage. L'utilisateur, lorsqu'il est guidé par le logiciel, comprend mieux le type d'interaction attendue et hésite moins. Il apprends d'autant plus vite que le **fonctionnement** du système lui semble **clair**.

- **Incitation**

Ce critère réunit les différents moyens visant à **conduire** l'utilisateur à effectuer des actions spécifiques. L'incitation aide l'utilisateur dans son interaction en fournissant les éléments nécessaires.

*Recommandations :*

- ✓ Griser les commandes non disponibles.
- ✓ Fournir la liste des saisies attendues.
- ✓ Donner le format de saisie des données.

- **Groupement/Distinction**

Lorsque l'on utilise un outil pour la première fois, on applique la règle de la **similarité** : ce qui se ressemble devrait fonctionner de la même manière.

Dans un logiciel, ce guidage consiste donc à **regrouper** (ou associer) les informations de même type et à **distinguer** (ou dissocier) les données particulières.

*Recommandations :*

- ✓ Regrouper les informations de même type par le format ou par la position.
- ✓ Distinguer par une représentation différente les informations distinctes.

- **Lisibilité**

Il s'agit de faciliter la **perception** des informations. Si la lecture est facile, l'utilisateur comprend mieux le fonctionnement du système.

Des pages web bien présentées, sobres, mais claires et structurées seront plus souvent visitées que des pages confuses, chargées et de lecture difficile.

*Recommandations :*

- ✓ Utiliser une police droite.
- ✓ Employer des lettres et symboles sombres sur un fond clair.
- ✓ Rester sobre et éviter les effets "sophistiqués" de style.

Ce point important sera approfondi avec le chapitre suivant.



- **Retour utilisateur**

Il s'agit des différents comportements de l'interface homme-machine qui visent à **montrer le fonctionnement** du système :

- ✓ en signalant ce que fait le système ;
- ✓ en fournissant un retour aux actions de l'utilisateur.

Ces comportements réactifs permettent à l'utilisateur d'établir clairement le lien entre ses propres actions et l'état résultant du système.

*Recommandations :*

- ✓ Le système doit répondre à toute action de l'utilisateur, par 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 utilisateur
- ✓ Rendre visible les traitements réalisés par le logiciel.

## 1.4.Homogénéité

Cela concerne la **cohérence** globale de l'interface homme-machine.

L'objectif est de respecter une logique d'utilisation constante aussi bien au niveau des procédures que de la représentation.

Le graphisme, l'emplacements des éléments d'information ou de présentation, le vocabulaire, le format et la syntaxe des données doivent être **similaire d'un écran à un autre**.

L'homogénéité rend le système prévisible et stable. L'utilisateur apprend vite à utiliser le logiciel, par généralisation.

*Recommandations :*

- ✓ Les fenêtres doivent suivre le même schéma d'agencement.
- ✓ La sémantique des boutons de la souris et des raccourcis-clavier doit être constante.
- ✓ Utiliser le même vocabulaire pour désigner les commandes du logiciel.
- ✓ La syntaxe des commandes doit être cohérente sur l'ensemble de l'interface.
- ✓ S'appuyer sur une charte graphique, les modèles et les styles ; s'aider de gabarits pour construire les interfaces.

## 1.5. Flexibilité

La flexibilité (ou **souplesse**) est la capacité de l'interface homme-machine à s'adapter à **differents contextes d'utilisation**.

Un logiciel est utilisé par **différentes personnes** qui ont chacune leur propre stratégie pour atteindre leur objectif selon les contraintes de la tâche et leur préférence.

Nous devons donc fournir plusieurs moyens pour atteindre un même objectif. Ces moyens sont complémentaires et adaptés à différents contextes d'utilisation ou à différents utilisateurs.

*Recommandations :*

- ✓ Permettre d'activer les commandes et à la souris et au clavier.
- ✓ Autoriser l'action d'une commande fréquente depuis plusieurs fenêtres (ou contextes) de l'application.
- ✓ Permettre à l'utilisateur de paramétrier le logiciel selon ses préférences.
- ✓ Fournir un moyen rapide d'accéder aux commandes des menus.

## 1.6. Contrôle utilisateur

Ce critère concerne le degré de contrôle de l'utilisateur sur les traitements réalisés par le système. L'objectif est que l'utilisateur **comPRENNE le fonctionnement** et qu'il puisse **prévoir l'effet des commandes**.

C'est un facteur important d'acceptation du système. Le système et son fonctionnement deviennent explicites et prévisibles.

L'utilisateur :

- ✓ devient autonome et performant dans son interaction,
- ✓ il garde la maîtrise du processus,
- ✓ il fait moins d'erreurs et
- ✓ il apprend plus facilement.

*Recommandations :*

- ✓ Demander explicitement confirmation pour les commandes importantes ou difficilement réversibles.
- ✓ Offrir la possibilité d'interrompre les traitements long.
- ✓ Autoriser les retour en arrière.  $\Rightarrow$  *Commande Undo.*

## 1.7.Traitement des erreurs

(Protection contre les erreurs ; correction des erreurs)

Quels sont les moyens visant à protéger l'utilisateur des erreurs et à les corriger ?

L'objectif est de :

- ✓ Éviter les erreurs ;
- ✓ détecter les erreurs au plus tôt ;
- ✓ permettre une correction aisée des erreurs ;
- ✓ faciliter l'exploration et l'apprentissage du système.

- **Protection contre les erreurs**

Les moyens de protections permettent d'**éviter les erreurs** provoquées par un utilisateur connaissant mal le logiciel, ou inexpérimenté.

*Recommandations :*

- ✓ Griser les commandes non disponibles.
- ✓ Fournir la liste des valeurs possibles.
- ✓ Déetecter les erreurs au plus tôt.
- ✓ Minimiser les saisies claviers
- ✓ Prévenir les risques de pertes de données.

- **Correction d'erreur**

Les moyens de correction d'erreur concourent à dédramatiser les erreurs.

*Recommandations :*

- ✓ Placer le message d'erreur là où regarde l'utilisateur.
- ✓ Afficher un message explicite, bref, non réprobateur et autosuffisant.
- ✓ Mettre en évidence la donnée erronée.

## 1.8.Concision

La concision concerne la réduction de la **charge perceptive et mnésique** de l'utilisateur.

L'objectif est d'allouer un minimum de ressources au système cognitif et de minimiser les risques d'erreurs en réduisant le nombres de stimuli.

Pour cela :

- ✓ N'afficher que les informations pertinentes ;
- ✓ réduire le nombre d'actions élémentaires pour atteindre un objectif donné.

*Recommandations :*

- ✓ Minimiser les saisies/lectures ; pas de zéro en tête d'un nombre,
- ✓ pas d'unité, etc.
- ✓ Éviter les textes trop verbeux.
- ✓ Ne pas demander d'informations qui peuvent être déduite par le système.
- ✓ Éviter à l'utilisateur d'avoir à se souvenir d'informations d'une fenêtre à une autre.
- ✓ Ne pas avoir à faire des calculs qui peuvent être effectués automatiquement.

## 1.9. Évaluation de l'utilisabilité d'un logiciel ou d'un site web

Voir le Tableau 1.1 de l'ouvrage : *De l'ergonomie du logiciel au design des sites web* (Jean-françois Nogier, Dunod).

## 2. Notions sur les sciences cognitives

### 2.1. Le modèle du processeur humain

Dans ce chapitre, nous allons nous intéresser à l'être humain en situation de travail. Les modèles présentés sont issus des travaux de S.K. Carl et de D.A. Norman.

Le modèle du processeur humain se rapproche de celui d'un système informatique.

- ✓ Il prends en entrée des données, les **stimuli perceptifs**,
- ✓ il effectue des **traitements** à partir de ces données,
- ✓ il produit des sorties, par des **actions motrices**.

La figure 1.1, déjà donnée au chapitre 1, illustre ce modèle :

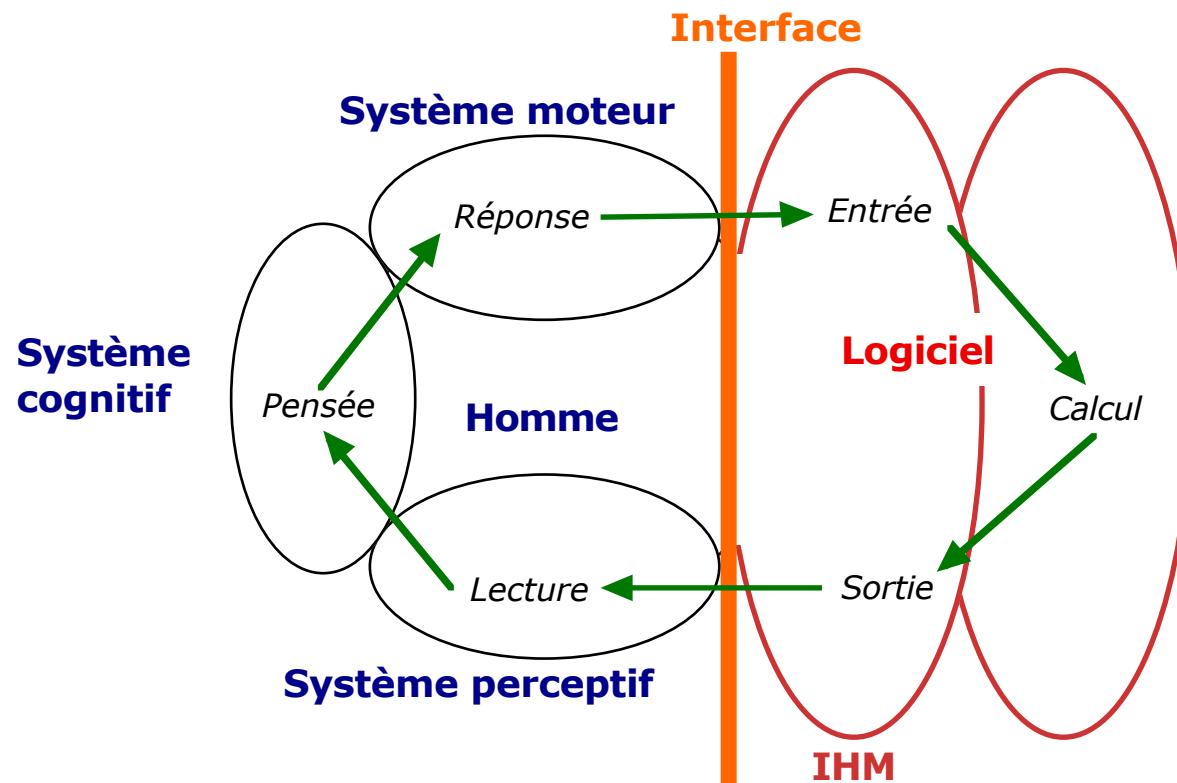


Figure 1.1 — Le processeur humain ; parallèle avec le logiciel



- ✓ Le système perceptif traite les informations reçues du monde extérieur, via des stimuli, mémorisés dans des registres sensoriels sous forme analogique. Chaque sens est alors indépendant.
- ✓ Le système cognitif réalise des opérations cognitives de traitements des connaissances à partir des données de différents registres sensoriels. Il recherche, au besoin, des informations en mémoire à long terme.
- ✓ L'action décidée, la réponse est traitée par le système moteur par des stimuli musculaires. Pour actionner :
  - ▶ le clavier
  - ▶ un dispositif de pointage ou de contrôle : **souris**, trackpad (pavé tactile), écran tactile, télécommande sans fil (ex.: wiimote de Nintendo...), système de reconnaissance de mouvement et d'image (ex.: Kinect de Microsoft)...

Les opérations de traitement de connaissances sont réalisées en **mémoire à court terme**. L'unité de traitement est appelé le **mnème**.

Ainsi, une suite telle que "96 52 34" peut, suivant les individus, être composé de :

- ✓ 6 mnèmes (9, 6, 5, 2, 3 et 4),
- ✓ 3 mnèmes (96, 52 et 34),
- ✓ un seul, s'il s'agit un code connu (comme un n° de téléphone par ex.).



La capacité de traitement de cette mémoire à court terme est de **5 à 9 mnèmes**. Des regroupements par 2 ou 3 mnèmes, reliés entre eux, sont souvent réalisés.

La mémoire à court terme peut donc être structurée comme suit :

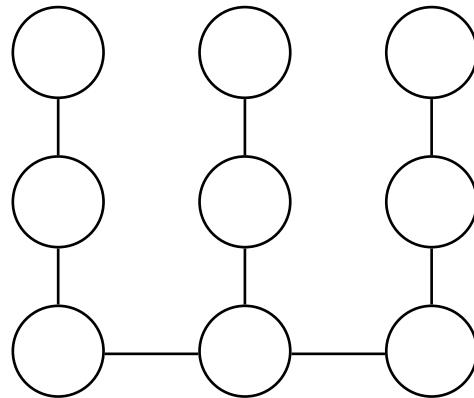


Figure 2.1 – Structure de la mémoire à court terme

On en déduit qu'il nous faut **minimiser** le nombre de données que l'utilisateur doit prendre en compte simultanément. De plus, la présentation et l'organisation des informations sur le support peuvent faciliter les **regroupements** opérés par le système perceptif de l'utilisateur.

Les processus de raisonnement, l'apprentissage et la communication orale utilisent beaucoup la mémoire à court terme.

La mémoire à long terme sert au **stockage** des connaissances.

## 2.2.Théorie de l'action

- **Le cycle action-évaluation**

Cette théorie, élaborée par D. A. Norman, va nous permettre de **mesurer la complexité d'utilisation** d'une interface homme-machine. Elle considère que toute action est un cycle itératif en deux temps :

- ✓ L'utilisateur exécute une commande
- ✓ Il évalue la modification induite par rapport au but fixé.

Plus précisément, ce cycle action-évaluation peut être décomposé suivant la figure 2.2.

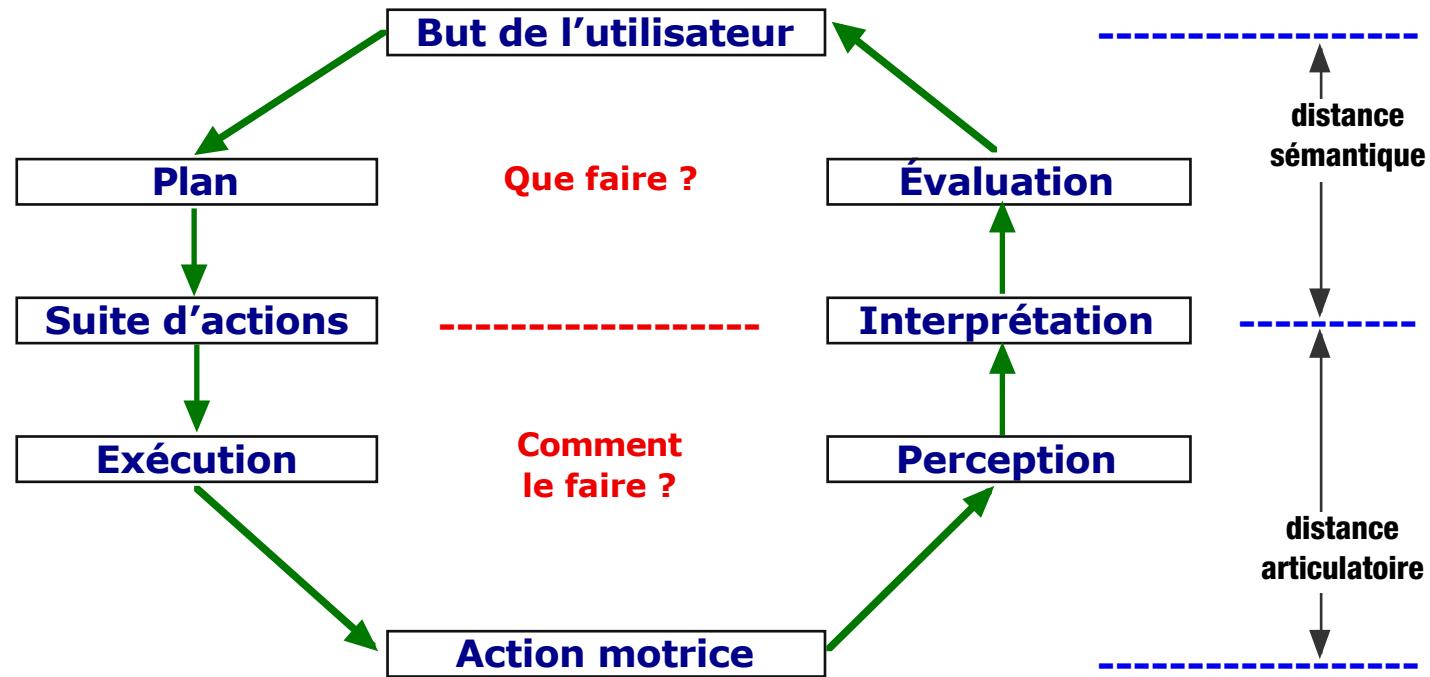


Figure 2.2 — Décomposition du cycle action-évaluation



- ✓ À partir du but de l'utilisateur, celui-ci élabore un plan ;
- ✓ Ce plan donne lieu à une suite d'actions ;
- ✓ Chaque action est exécutée physiquement ;
- ✓ L'état du système en est modifié, ce que perçoit l'utilisateur ;
- ✓ Il interprète ces changements ;
- ✓ Et il compare le résultat en fonction du but fixé.

La région "Que faire ?" est celle des processus liés à la compréhension du système. La région "Comment le faire ?" regroupe les opérations liées à la manipulation. L'épaisseur de ces régions représente une distance, mesurant une complexité d'utilisation :

- ✓ **La distance sémantique** symbolise la distance entre le but fixé et les actions à réaliser ; elle mesure la **difficulté de compréhension** du système pour la réalisation d'une tâche et celle de l'évaluation des modifications vis à vis du but fixé.
- ✓ **La distance articulatoire** symbolise la distance entre l'action et sa représentation physique ; elle mesure la **complexité de manipulation** du système ; elle rend compte de la quantité de ressources cognitives et motrices nécessaires aussi bien à la réalisation d'une commande qu'à la perception de l'état résultant du système.

Notre but est donc de construire des interfaces utilisateur contribuant à **réduire ces distances** sémantiques et articulatoires.

- **Tâche et activité**

**La tâche est l'objectif** que cherche à atteindre l'utilisateur. Pour la réalisation d'une tâche, l'être humain procède par décomposition en sous-buts, puis en actions élémentaires. Les informations relatives à une tâche sont conservées en mémoire jusqu'à ce que l'objectif initial soit atteint.

Dans ce cadre:

- ✓ **La tâche désigne ce qui doit être fait.** Or une tâche peut être réalisée de différentes manières, suivant le contexte. Une réalisation effective de la tâche est appelée *activité*.
- ✓ **L'activité représente ce qui est fait.**

Cette différence entre tâche et activité est importante, car la conception d'un logiciel s'appuie sur une **description des tâches** (consignée dans le cahier des charges) et non sur les activités. Mais l'utilisateur, lui, se servira du logiciel dans un **contexte d'activité**.

Le rôle des tests d'utilisabilité est alors de vérifier la **concordance** entre la tâche et l'activité, lors d'une mise en situation de l'utilisateur.

Pour être adapté à l'activité des utilisateurs, l'interface homme-machine doit tenir compte :

- ✓ de différents contextes d'utilisations,
- ✓ des diverses stratégies d'utilisation
- ✓ et de la diversité des utilisateurs ;

ce qui correspond bien au critère de **flexibilité** vu au premier chapitre.

De plus, l'utilisateur doit mener en parallèle deux tâches :

- ✓ La **tâche métier**, soit l'objectif qu'il cherche à atteindre,
- ✓ La **tâche système**, consistant en l'utilisation du logiciel proprement dit.

L'utilisation du logiciel ne doit pas perturber la réalisation de la tâche métier.

L'interface homme-machine doit **réduire** au mieux la **tâche système**, au profit d'un **guidage** dans la réalisation de la tâche métier.

### 3. L'écran et la page

#### 3.1. La création et la publication de documents

- **La chaîne de l'édition**

La chaîne traditionnelle de l'édition comporte neuf étapes :

- ✓ La conception : rédaction de texte **manuscrit** et ébauche de figure
- ✓ La préparation : **dactylographie** du texte ; traitement de texte ; dessin des figures ; photographies
- ✓ Lecture, par un **comité de lecture** et corrections
- ✓ **Typographie** et **photocomposition** (traitement et composition des textes destinés à l'impression)
- ✓ **Maquettage**
- ✓ **Photogravure**
- ✓ **Correction**
- ✓ Lecture d'épreuve et **bon à tirer**
- ✓ **Impression** sur un groupe de presse.

L'informatique et les techniques numériques apportent une évolution de cette chaîne :

- ✓ L'utilisateur peut lui même, à l'aide de son ordinateur personnel et son PAN (Personal Area Network), dactylographier son texte, préparer des figures, scanner des photos ou les récupérer depuis un appareil photo numérique.
- ✓ Le graphiste réalise des illustrations, scanne, prépare et corrige les images et les photos.
- ✓ La **mise en page** (le montage) est réalisée par un maquettiste.
- ✓ Le **flashage** consiste à convertir le **document d'exécution** pour le transposer en support imprimable. On obtient autant de films que de couleurs de base (soit 4 en quadrichromie : CMYK pour *Cyan, Magenta, Yellow et Black*).
- ✓ Une épreuve est réalisée (un *cromalin*) pour que le client puisse signer le **bon à tirer**.
- ✓ Les films sont utilisés pour une impression en offset, sur un groupe de presse couleur.

Les imprimeurs s'équipent maintenant de groupes de presse de type offset utilisant directement un document d'exécution, par exemple au format PDF.

On élimine alors la génération des films.



- **La présentation et la mise en page**

L'unité de base de la composition est la **page**. La page correspond à un **support papier**, à partir duquel on définit un **format**, puis un **cadre**.

Le format du document est la taille des feuilles du document final. S'il est inférieur à la taille du support, ce dernier sera massicoté après reproduction et assemblage.

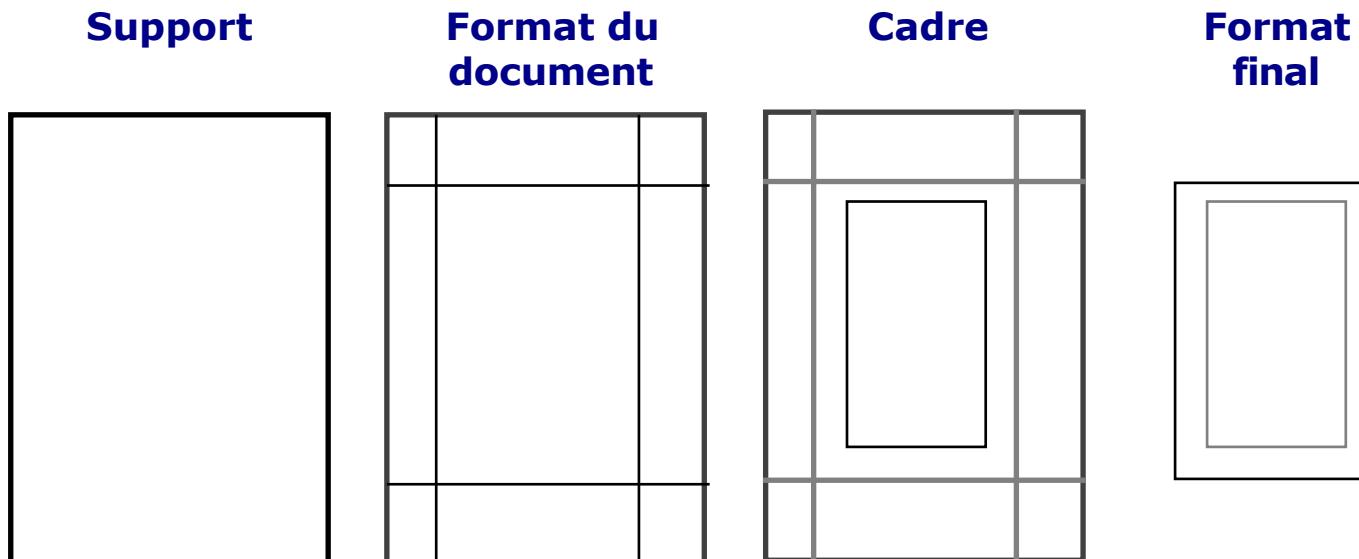


Figure 3.1 — Support, format et cadre



Le cadre, inférieur au format, correspond à l'encombrement du texte et des figures.

En Europe, le format A4 (210 \* 297 mm) est le standard le plus utilisé pour le support ; les autres formats standards sont définis en doublant la surface ou en la réduisant de moitié :

- ✓ A5 (148 \* 210 mm)
- ✓ A4 (210 \* 297 mm)
- ✓ A3 (297 \* 420 mm)

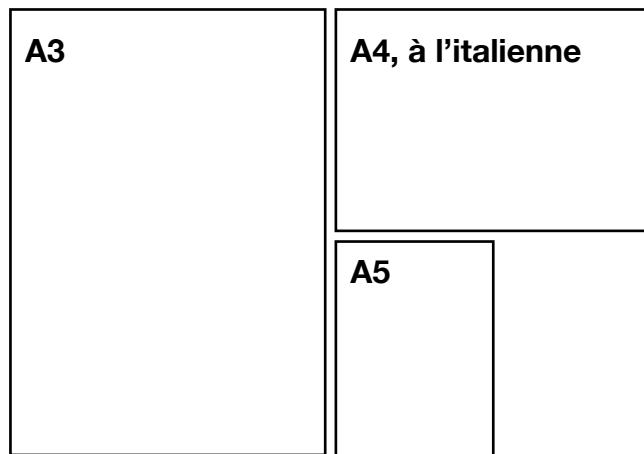


Figure 3.2 — Les formats standards européens

Les autres standards sont, en Europe :

- ✓ B5 (176 \* 250 mm)
- ✓ Tabloïd (279,4 \* 431,8 mm) : c'est un format standard des quotidiens ; c'est le double du format Lettre US

Et, en Amérique :

- ✓ Lettre US (215,9 \* 279,4 mm) : plus large et moins long que le A4
- ✓ Légal US (215,9 \* 355,6 mm)

Les éléments à considérer pour choisir un cadre sont :

- ✓ La **justification** : la longueur maximale d'une ligne de texte. Elle représente en général entre 2/3 et 3/4 de la largeur du format final du document.
- ✓ Les **marges** : elles déterminent la hauteur maximale du cadre et son emplacement dans la page.

Pour les documents recto-verso, les différentes marges sont, idéalement de la plus petite à la plus grande :

- ✓ la marge de petit fond (la **marge intérieure**, coté reliure)
- ✓ la marge de tête (la **marge haute**)
- ✓ la marge de grand fond (la **marge extérieure**)
- ✓ la marge de pied (la **marge basse**)

En dehors du cadre, en marges de tête et de pied, on porte la pagination et les références constantes de l'ouvrage, des parties ou des chapitres.

Pour un document recto-verso :

- ✓ Le **recto**, appelé "belle page" est à droite lorsque le document est ouvert. Elle porte toujours les numérotations **impaires**.
- ✓ Le **verso**, ou "fausse page" est la page de gauche ; elle porte les numérotations **paires**.

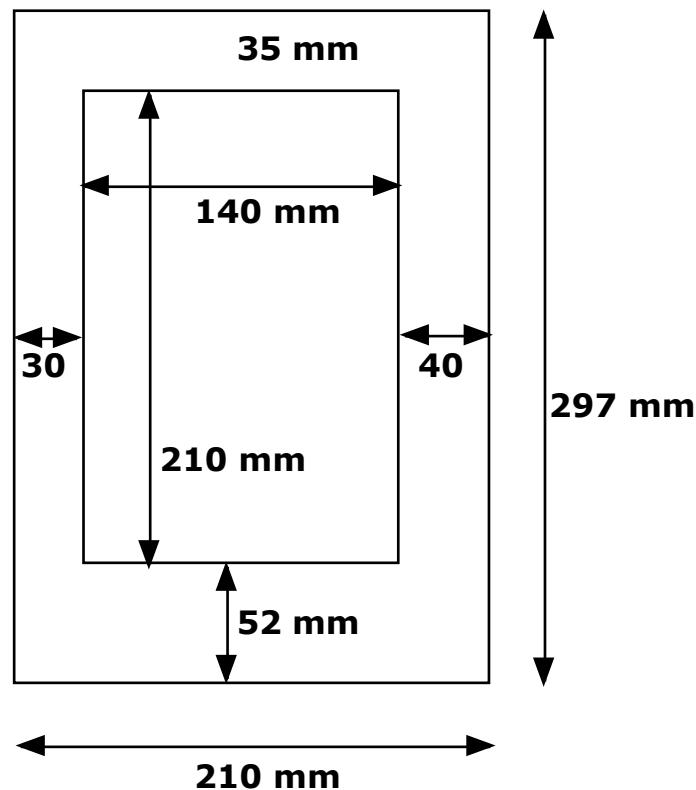


Figure 3.3 — Exemple de marges d'une page A4 recto

- **Notions de typographie**

Les éléments suivants font partie du jargon de l'imprimerie moderne.

- ✓ Le **caractère typographique** : c'est un signe gravé, tracé ou écrit. Cela désigne aussi bien le matériel servant à imprimer le signe que le signe lui-même.
- ✓ Le **bas de casse** : les minuscules.
- ✓ Les **capitales** : les majuscules
- ✓ Une **police de caractère**, ou **fonte** : c'est le type de caractère, indépendamment de sa taille ou de son style.

- ✓ Caractères avec empattement ou caractères **sérif** : les jambages des signes sont ornés. Les empattements guident et facilite la lecture dans le corps d'un texte.

Exemples de caractères serif :

**Times, Bodoni, Baskerville**

- ✓ Caractères sans empattement ou caractères **sans serif** : Ce sont les caractères dits antiques, linéales ou caractères bâtons.

Exemples de caractères sans serif :

**Helvetica, Futura, Lucida**

- ✓ Le **corps** : la hauteur d'un caractère, blancs de séparation compris, en points (1 point = 0,376 mm)
- ✓ La **chasse** : largeur visible d'un caractère.
- ✓ Un "i" chasse moins qu'un "m".
- ✓ L'**encombrement** : la chasse, plus les blancs de droite et de gauche.
- ✓ L'**approche** : le blanc entre deux lettres.

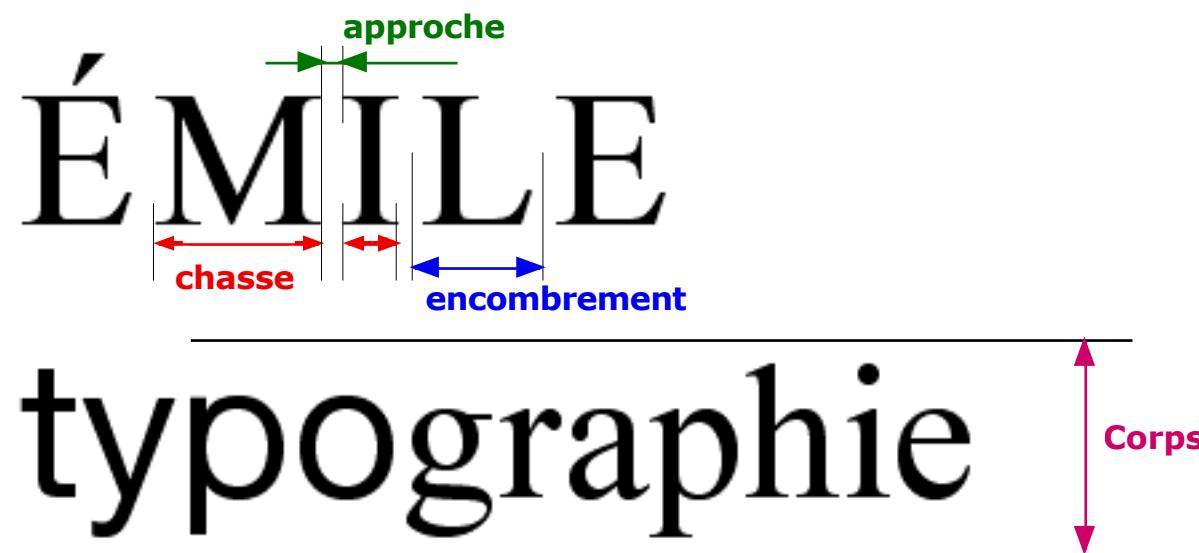


Figure 3.4 — Corps, chasse, approche et encombrement

- ✓ La **graisse** : C'est l'épaisseur du trait d'un caractère. Light, book, medium, bold ou gras, extra-bold ou extra-gras sont différentes graisses.
- ✓ Le gras est utilisé pour **mettre en valeur** un titre ou un mot dans le texte. Il est mieux perçu que l'italique, et plus lisible que tout autre effet.
- ✓ La **pente** : Un caractère est droit (romain) ou penché (italique).
- ✓ L'italique est réservée aux citations, aux titres d'ouvrages, aux mots étrangers ou aux termes techniques ou scientifiques.
- ✓ Le **style** : regroupe, en informatique, les possibilités d'effets spéciaux. Par exemple :

**gras, italique, souligné, double souligné,  
barré, relief, ombré,** exposant ou indice

Sauf pour le gras et l'italique, l'usage des effets spéciaux est à bannir, ou bien à manipuler avec précaution. Cela réduit la lisibilité du texte et brise l'harmonie graphique des mots.

Partisans ou non du souligné, évitez donc de souligner des phrases entières, car cela est pénible à lire (et c'est encore pire, du reste, pour le double souligné).

Voici par ailleurs quelques codes typographiques et quelques conseils :

- **La capitale**

Elle est employée :

- ✓ En début de phrase, après les points (ordinaires, d'exclamation, d'interrogation, de suspension).
- ✓ Pour la lettre initiale d'un nom propre tel qu'un nom de nationalité, une marque, un nom géographique, etc. Les noms de langue sont composés en bas de casse, ainsi que les noms géographiques lorsqu'ils deviennent adjetifs. Exemple :
  - ▶ les Allemands, le golfe de Gascogne, l'océan Arctique ;
  - ▶ la démocratie française, la reine anglaise ;
  - ▶ la plupart des Belges parlent trois langues : le français, le néerlandais et l'anglais.

Nota : Il est souvent nécessaire de consulter un ouvrage spécialisé pour vérifier l'emploi de la majuscule. Un simple dictionnaire ne suffit pas.

- ✓ Avec l'informatique, on retrouve l'usage de **capitales accentuées** et de capitales pour les cédilles. Cela permet de faciliter la lecture.

Que veut en effet dire le titre :

### **LE GENERAL CHAHUTE A L'ASSEMBLEE**

est-ce bien :

### **LE GÉNÉRAL CHAHUTÉ À L'ASSEMBLÉE**

ou alors est-ce le Général qui chahutait ?



- ✓ La lisibilité d'un texte en toute majuscule est médiocre par rapport au même texte composé en minuscule :

**OUTILS DE GESTION**

**outils de gestion**

**outils de gestion**

**outils de gestion**

Figure 3.5 — Lisibilité d'un texte en toute majuscule ou en bas de casse

- **La ponctuation**

- ✓ Une ponctuation différente change le sens de la phrase :
  - ▶ *Messieurs les Anglais, tirez les premiers !...*
  - ▶ *Messieurs ! les Anglais ! Tirez les premiers !*
- ✓ Pour la ponctuation finale des titres, on supprime le point, les deux points ou le point-virgule, mais on garde les autres ponctuations.
- ✓ Une espace (féminin en langage typographique) peut être sécable ou insécable.

✓ Règles de disposition des espaces avant et après les signes de ponctuation :

<b>Signe</b>		<b>Avant</b>	<b>Après</b>
le point	.	pas de blanc	une espace
la virgule	,	pas de blanc	une espace
le point-virgule	;	une esp. insécable	une espace
les deux points	:	une esp. insécable	une espace
le point d'interrogation	?	une esp. insécable	une espace
le point d'exclamation	!	une esp. insécable	une espace
la parenthèse ouvrante	(	une espace	pas de blanc
la parenthèse fermante	)	pas de blanc	une espace
un crochet ouvrant	[	une espace	pas de blanc
un crochet fermant	pas de blanc	une espace	
le guillemet français	«	une espace	une esp. insécable
le guillemet français	»	une esp. insécable	une espace
le guillemet ouvrant	“	une espace	pas de blanc
le guillemet fermant	”	pas de blanc	une espace
le tiret ouvrant	—	une espace	une espace
le tiret fermant	—	une espace	une espace
les points de suspension...			
- à la fin d'une phase...		pas de blanc	une espace
- à la fin d'un mot...		pas de blanc	une espace
- pour remplacer un mot		une espace	une espace

### • Les nombres, dates et heures

- ✓ Un âge, une durée et une fraction s'écrivent généralement **en toutes lettres** :
  - ▶ Julie a eu dix ans mercredi dernier.
  - ▶ J'ai patiné pendant deux heures trente.
  - ▶ Il a vidé plus des deux tiers du réservoir.
- ✓ Mais une durée de vie, une durée avec une idée de comparaison ou une valeur mathématique s'écrivent **en chiffres** :
  - ▶ Pendant les 5 premières années...
  - ▶ Le record fut battu de 12 secondes et 30 centièmes.
  - ▶ L'échelle de cette carte est 1/25 000.
- ✓ Une date s'écrit comme ci-dessous, **sans capitale** au jour ni au mois :
  - ▶ Elsa est née le 13 juin 1994.
  - ▶ Fait à Millau, le mercredi 19 mars 2003.
- ✓ Une heure s'écrit en lettres si elle exprime une durée, en chiffres sinon :
  - ▶ J'ai attendu deux heures. Notre rendez-vous était fixé à 15 h 30.
- ✓ La notation 10:30 n'est pas française, mais anglo-saxonne.
- ✓ Autres exemples :
  - ▶ Pas plus de trois centimètres.
  - ▶ Le courant est de cinq ampères pour une puissance de vingt watts.
  - ▶ 1 500 €      3 A      20 W      40 km      50 Hz      19,6 %

### 3.2. Agencement de l'écran

Un utilisateur de logiciel va passer de longues périodes de travail devant son écran : c'est le principal média de sortie de l'interface homme-machine. La vision est en effet la partie la plus développée du système perceptif des humains.

La façon de présenter les informations mérite donc une attention particulière.

Quelles règles allons-nous suivre pour organiser l'écran afin de :

- ✓ mettre en évidence les données importantes et
- ✓ faciliter les mouvements de la souris ?

Comment choisir les couleurs, les polices ou les icônes ?

Quel langage employer pour les éléments de l'interface et les dialogues ?

Ce paragraphe et les suivants vont répondre à ces questions.



- **Les stratégies d'exploration visuelle**

En découvrant pour **la première fois** un support d'information (une affiche de publicité, un écran...), l'usager adopte une stratégie d'**exploration en Z** : Le regard part du coin supérieur gauche, se dirige vers la droite, parcourt systématiquement la zone centrale et se termine dans le coin inférieur droit.

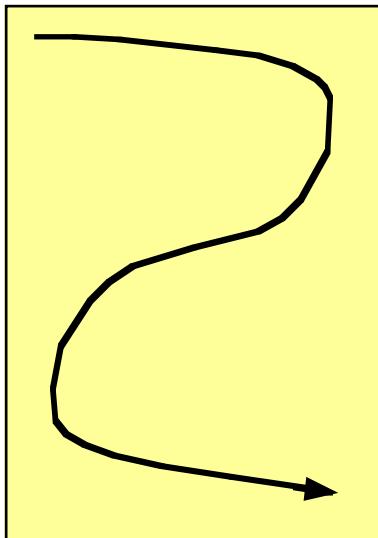


Figure 3.6 — Le parcours du regard pour la première exploration d'un support



Une fois qu'il connaît l'image, il effectue une **recherche sélective**, en cherchant une information à l'endroit où il pense pouvoir la trouver. Il adopte en fait une stratégie de fixation sélective sur les points qui lui semblent pertinents.

De plus, les symétries, les titres, les graphiques et les espaces viennent orienter l'exploration visuelle.

- **L'accessibilité à la souris**

Le déplacement du pointeur à l'aide de la souris est **plus facile dans la zone centrale** de l'écran. Le pointage d'un objet placé dans un coin de l'écran est particulièrement difficile, car le mouvement y est plus contraint que dans une zone dégagée.

- **Visibilité et accessibilité des zones de l'écran**

Le schéma suivant, figure 3.7, présente les caractéristiques standard de visibilité de d'accessibilité des zones d'écran. Il nous donne donc des indications pour faciliter le parcours visuel et les mouvements de la souris (ou d'autres dispositifs de pointage).

Très visible peu accessible	Très visible	Peu accessible
Très accessible	La plus visible et la plus accessible	Très accessible
Peu accessible	Peu visible et peu accessible	Peu accessible

Figure 3.7 — Visibilité et accessibilité de zones d'écran

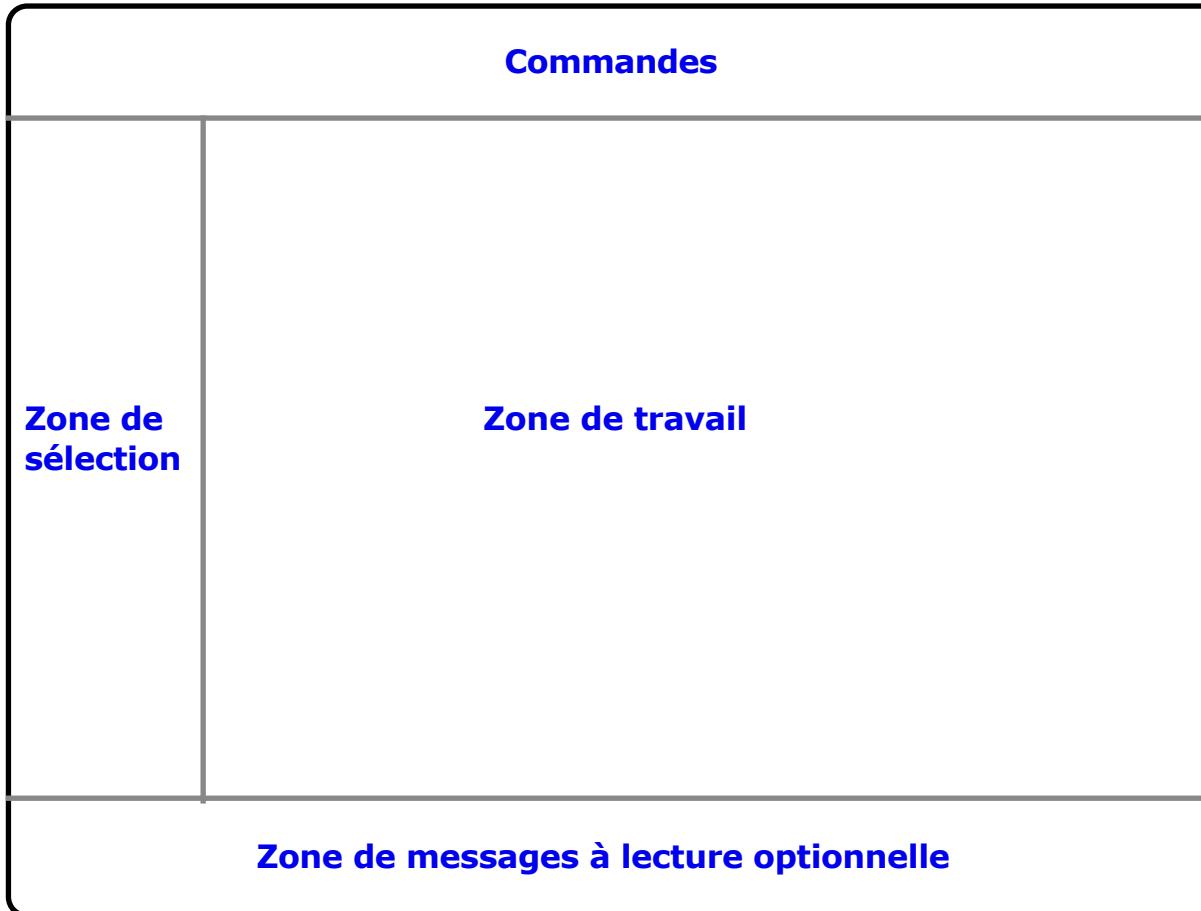


Figure 3.8 — Exemple de zones d'un tracé régulateur



- **Positionnement**

Pour exploiter ces caractéristiques et pour répondre au besoin de cohérence du positionnement des objets de l'interface, on utilise un **tracé régulateur**. Le tracé régulateur détermine une **disposition standard** des écrans, des fenêtres et des dialogues.

C'est un facteur d'homogénéité important, facilitant l'apprentissage du logiciel.

La figure 3.8 présente les principales zones d'un tracé régulateur.

Les zones de manipulations sont sujets à de nombreuses sélections à la souris : il s'agit

- ✓ des palettes d'outils ;
- ✓ des barres de menus ;
- ✓ des barres de boutons.

On les organise dans le sens de la lecture, avec :

- ✓ les commandes importantes placées en premier ;
- ✓ les objets fréquemment utilisés placés au centre.



Ainsi donc :

- ✓ Dans une barre de menu, l'espace central est plus facile à atteindre.
- ✓ Une palette d'outils est parcourue des yeux de l'utilisateur, qui s'aide souvent du pointeur lorsqu'il cherche une commande à utiliser. Il est donc bon que cette palette soit relativement compacte pour tout avoir sous les yeux et pour limiter les mouvements de la souris.
- ✓ Nous devons également regrouper des informations en relation entre elles.

### 3.3. Graphisme

Le graphisme d'un logiciel doit :

- ✓ permettre une lisibilité correcte,
- ✓ permettre une mise en évidence efficace,
- ✓ répondre à des critères esthétiques.

Nous allons donc en voir de toutes les couleurs, nous battre avec les polices et étudier la mise en évidence.



## 3.4. Couleur

### • Les composantes de la couleur

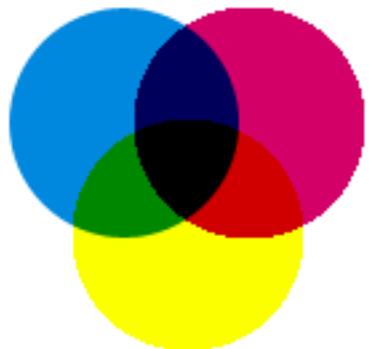
Les objets n'ont pas de couleur. Ils ont la propriété d'absorber ou de réfléchir les rayons lumineux. Une pomme rouge absorbe toute une gamme de radiations lumineuses, sauf celles situées dans une bande du spectre entre le magenta et le jaune.

Newton a découvert, en 1669, qu'une lumière blanche qui traverse un prisme se disperse et se décompose selon les couleurs de l'arc-en-ciel.

On rappelle que les longueurs d'onde du spectre visible sont comprises entre 380 nm (violet) et 780 nm (rouge).

Dans la pratique, nous utiliserons différents codages suivant les domaines d'application.

Synthèse soustractive



Synthèse additive

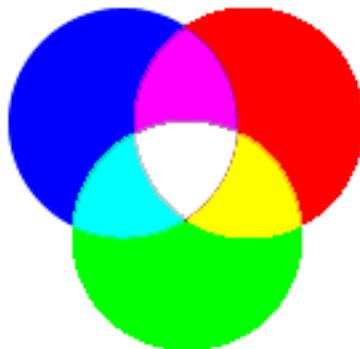


Figure 3.9 — Synthèses soustractive et additive des couleurs

- ✓ Le **codage HSV**, pour Hue, Saturation, Value, (soit **TSL**, pour Teinte, Saturation, Luminosité) correspond bien à la façon dont l'être humain perçoit la couleur.
- ✓ Le **codage RVB**, pour Rouge, Vert, Bleu, ou RGB *in english* est adapté aux couleurs affichées sur un écran. Ce codage indique la proportion de lumières rouge, vert et bleu projetées. Une couleur, addition de trois couleurs primaire, résulte d'une synthèse additive.
- ✓ Le **codage CMYK**, pour Cyan, Magenta, Yellow & Black, est celui des imprimeurs, qui réalisent une synthèse soustractive par un mélange d'encre cyan, magenta, jaune et noire.



Figure 3.10 — Spectres de couleurs saturées, désaturées, assombries

Couleur	Teinte HSV	R	V	B
Rouge	0°	255	0	0
Orange	30°	255	128	0
Jaune	60°	255	255	0
Vert-jaune	90°	128	255	0
Vert	120°	0	255	0
Vert bleuté	150°	0	255	128
Cyan	180°	0	255	255
Bleu-cyan	210°	0	128	255
Bleu	240°	0	0	255
Mauve	270°	128	0	255
Rouge violacé	300°	255	0	255
Magenta	330°	255	0	128
Rouge	360°	255	0	0



Voici les paramètres du codage HSV :

- ✓ La **teinte** correspond à la longueur d'onde principale ; c'est ce que l'on appelle couramment la couleur ou la **nuance**. Elle s'exprime en général en degrés, car on la représente sur une roue chromatique.
- ✓ La **saturation** correspond au spectre de la couleur, c'est à dire à la largeur de bande de longueurs d'onde occupée par la couleur.  
Elle varie entre 0 % (couleur achromatique, donc un gris) et 100 % (couleur saturé).  
Une bande étroite donne une couleur saturée. Une bande large caractérise une couleur désaturée (comme pour les pastels).
- ✓ La **luminosité** (ou encore intensité, brillance ou clarté) correspond au degré de réflexion de la lumière. Cela correspond au niveau de gris pour une couleur achromatique. Elle varie entre 0 % (noir) et 100 % (couleur pure)



### • La perception des couleurs

La couleur est perçue par la rétine de l'œil, par deux types de photorécepteurs :

- ✓ Les bâtonnets, en périphérie de la rétine, sont sensibles aux formes ;
- ✓ Les cônes, au centre de la rétine (la fovéa), sont sensibles aux couleurs.

Les cônes sont composés de trois types de pigments qui ne sont ni proportionnés ni réparties de façon homogène dans la fovéa :

- ✓ Les pigments rouges et verts sont fréquents au centre de la fovéa ;
- ✓ Les pigments bleus sont surtout répartis à la périphérie de la fovéa.

Il en résulte qu'un bleu périphérique améliore la vision centrale ; le bleu est donc recommandé pour les encadrements

Le bleu saturé est à éviter pour le texte ou des petits objets.

La sensibilité de l'œil dépend :

- ✓ de la longueur d'onde de la couleur ; l'œil est plus sensible au vert-jaune et moins sensible au violet ou au rouge.
- ✓ de la luminosité ; la nuit, cette sensibilité se décale vers le violet, et le rouge devient peu visible.



Pour favoriser la distinction et la reconnaissance des couleurs, il est recommandé de choisir

- ✓ des couleurs saturés,
- ✓ une gamme de couleurs réparties de façon homogène dans le spectre.

La *chaleur* d'une couleur influence sa perception.

- ✓ Une couleur rouge, orange ou jaune sera qualifiée de **couleur chaude** ;
- ✓ le vert, le bleu et le violet sont considérés comme des **couleurs froides**.

Un objet de couleur chaude semble occuper plus d'espace et être plus proche qu'un objet de même surface mais de couleur froide.

Une couleur n'est pas perçue de la même manière par tout le monde. Ainsi, le **daltonisme** touche environ 8 % d'hommes et 0,4 % de femmes.

Une grosse proportion de daltoniens confondent le rouge et le vert, car ces couleurs sont pour eux comme désaturées (grises). D'autres daltoniens confondent le bleu et le jaune. Les daltoniens restent sensibles à la clarté.

La couleur ne doit donc être qu'**un moyen supplémentaire** de mise en évidence d'objets ou d'informations. Il ne faut pas s'appuyer que sur ce moyen de présentation.

- **Les recommandations**

- ✓ Etablir un **code des couleurs**.

- ▶ Le codage couleur, ou code des couleurs, permet d'associer à chaque couleur de l'interface une signification précise et uniforme pour l'ensemble du logiciel ou du site web.

- ✓ **Minimiser** le nombre de couleurs, de 4 à 8.

- ▶ Le codage couleur doit être simple ;
    - ▶ au delà de 8 couleurs, on dépasse la capacité de la mémoire à court terme et le codage est difficile à assimiler.

- ✓ **Respecter** le codage couleur utilisé dans la profession des utilisateurs.

- ▶ Des pertes ou des dépenses sont codées en rouge pour les financiers ou les comptables...

✓ **Vérifier** les accords de couleur

- Ne pas hésiter à consulter un graphiste, ni à lui confier la création d'une charte graphique et d'une maquette.

✓ **Distinguer** les couleurs par **teinte et clarté** ;

- Ne pas utiliser que la couleur pour la mise en évidence.
- Les daltoniens ont alors plus de chance de distinguer les couleurs du codage
- et une mise en évidence sera efficace pour tous.



- ✓ Utiliser
  - des couleurs **très contrastées** pour exprimer une **différence** ;
  - des couleurs **peu contrastées** pour exprimer une **similitude**.
- ✓ Un jeu de couleurs complémentaires (rouge / cyan, bleu / jaune, etc.) et de clartés distinctes vont marquer les différences.
- ✓ Un regroupement sera déduit avec des objets de couleurs proches (rouge / rose, bleu / violet, etc.)
- ✓ **Éviter** le bleu saturé pour du texte ou des petits objets.
- ✓ Utiliser le bleu pour encadrer, car il améliore la vision centrale.
- ✓ Éviter les fonds marrons ou verts.
  - Le texte écrit sur de tels fonds sera peu lisible. Le choix des couleurs du texte ou des petits objets placés sur ces fonds seront difficiles.



- ✓ Préférer des couleurs **peu saturés** pour les **grandes** surfaces.
  - ▶ Une couleur vive pour un fond de fenêtre sera perçue comme agressive.
- ✓ Une couleur **neutre et claire** convient bien pour un **fond d'écran**.
  - ▶ Le gris est la couleur la plus neutre et il facilite une bonne perception des objets et de leurs contrastes.
  - ▶ Un gris clair diminue les réflexions, offre en général un meilleur équilibre de luminance entre l'écran et l'environnement visuel et diminue donc la fatigue visuelle.
- ✓ Attention à la taille des petits objets.
  - ▶ Leur taille semble être plus petite sur un fond coloré ou gris.
- ✓ Concevoir d'abord une interface en noir et blanc.
  - ▶ Classer les objets par ordre d'importance et déterminer ceux à mettre en évidence.
  - ▶ N'ajouter la couleur qu'après avoir appliqué les autres valorisations.

### 3.5. Polices

Deux couleurs sont plus facile à différencier que deux polices de caractère. Le choix des polices reste important pour tant pour le design d'un logiciel que celui d'un site web ou de toute autre publication.

Nous tiendrons compte des recommandations suivantes.

- ✓ Utiliser un jeu d'au **maximum 4 polices** de caractères.
  - ▶ Avec plus, l'interface sera confuse, la signalétique et l'esthétique de la typographie sera médiocre.
- ✓ Associer une police du jeu à chaque type d'objet de l'interface.
- ✓ Préférer l'usage des minuscules, avec une majuscule pour la première lettre d'un label, d'un titre ou d'une phrase.
  - ▶ Le dessin des lettres minuscules est plus riche que celui des lettres majuscules. Les minuscules sont toujours plus lisibles. Cependant, un mot en majuscule au sein d'un texte sera mis en évidence.

✓ Utiliser une police **droite et sans serif**.

- ▶ À l'écran, des polices comme Verdana, Arial, Helvetica, etc. facilitent la lecture, à cause de leur dessin simple, homogène et contrasté.
- ▶ Par contre, sur un support papier, qui supporte une résolution bien supérieure, les empattements d'une police serif servent de guide horizontal à l'œil.

✓ Écrire avec des **lettres sombres** sur un **fond clair**.

✓ Préférer quelques lignes longues plutôt que plusieurs lignes courtes.

- ▶ Une lecture sera plus rapide avec des lignes de 50 à 55 caractères composés en une seule colonne (et 30 à 35 en double colonnes).

✓ **Aérer** le texte.

- ▶ Des lignes d'espacement et les blancs sont à la page ce que la ponctuation est au texte : une respiration.

✓ **Organiser** et **hiérarchiser** les écrits.

- ▶ Les titres et intitulés sont lus plus souvent que le texte dans des écrits du type rapport, mémoire, article professionnel, page de magazine ou de quotidien. La hiérarchie donnée à l'écrit fournit un guide pour le lecteur.

### 3.6.Icônes

Les icônes sont des graphismes plus ou moins abstraits qui occupent moins de place que du texte et sont indépendantes de la langue.

Une icône est **reconnue** et non pas lue. Elle est interprétée plus rapidement que le texte.

La signification d'une icône n'est pas toujours évidente. La **compréhension des icônes** auprès des utilisateurs doit être vérifiée, à travers un test d'utilisabilité.

La création des icônes est une tâche délicate et demande de longues heures de conception. Pour obtenir une bonne homogénéité, elles doivent être dessinées ensemble, donc après l'identification exhaustive de toutes les commandes.

Les interfaces à icônes donnent un côté **ludique** au logiciel, mais elles nécessitent un **apprentissage**. Elles impliquent plus d'erreurs que lorsque les commandes sont identifiées par du texte.

Le tableau 2.3 de l'ouvrage *De l'ergonomie du logiciel au design des sites web* propose un classement par difficulté d'interprétation croissante. Il montre que la difficulté d'interprétation augmente avec l'abstraction de l'icône.



Voici quelques recommandations au sujet des icônes :

- ✓ Le lien entre une icône et sa signification doit être le plus direct possible.
- ✓ Attention aux différences culturelles.



Figure 3.11 — Que signifient ces icônes ?

- ✓ Utiliser les icônes pour les commandes ou objets fréquemment utilisées.



- ✓ Accompagner l'icône de son nom, ou bien donner une description de la commande ou de l'objet lors du pointage de l'icône.
  - ▶ Le mieux étant de permettre à l'utilisateur de choisir d'afficher ou non les textes associés aux icônes.



**Figure 3.12** — Donner le choix à l'utilisateur d'afficher ou non un texte avec les icônes

- ✓ Limiter le nombre d'icônes à 12 (à 20 grand maximum).
  - ▶ La mémorisation des icônes se dégrade au delà de 12 icônes.

### 3.7. Mise en évidence

Les moyens pour **attirer l'attention** de l'utilisateur sur certains objets sont :

- ✓ le clignotement ;
- ✓ la graisse ;
- ✓ l'inversion vidéo ;
- ✓ la taille ;
- ✓ la police de caractère ;
- ✓ le soulignage ;
- ✓ la forme ;
- ✓ les puces ;
- ✓ le rapprochement ;
- ✓ l'encadrement ;
- ✓ le son ;
- ✓ la couleur...



Comment utiliser ces mises en évidence à bon escient, de façon cohérente et avec parcimonie ?

- ✓ La mise en évidence doit être **exceptionnelle**.
  - Un emploi abusif disperse l'attention de l'utilisateur.
- ✓ **Limiter à 5** maximum le nombre de moyens de mise en évidence utilisés sur l'interface.
  - Ne pas utiliser plusieurs moyens de mise en évidence sur le même objet.
- ✓ La mise en évidence ne doit pas affecter la perception de cet objet.
  - C'est l'inconvénient de l'inversion vidéo, qui modifie la couleur de base de l'objet.
- ✓ Lorsque la mise en évidence reflète l'état d'un objet, elle doit disparaître lorsque l'utilisateur acquitte l'événement .

Les considérations et recommandations qui suivent dépendent du moyen de mise en évidence.

- **Le clignotement :**

Le système visuel est très sensible aux mouvements, perçus par la zone périphérique de la rétine. Si l'œil fixe l'écran alors qu'un objet se déplace ou clignote dans une autre région de l'écran, le cerveau détecte le mouvement et incite le regard à se tourner vers la source du mouvement.

- ✓ Le clignotement est un moyen très efficace pour attirer l'attention.
- ✓ Permettre à l'utilisateur **d'arrêter** le clignotement.
  - ▶ Étant très attractif, il devient vite désagréable.
- ✓ La **fréquence** du clignotement doit rester **faible** (autour du hertz).
- ✓ Régler une durée d'affichage supérieure à celle de l'effacement

- **L'inversion vidéo :**

Cela consiste à inverser la couleur du fond avec la couleur du texte ou de l'objet et, par extension, la technique de modification de couleur de fond.

- ✓ L'inversion vidéo est un moyen assez efficace pour attirer l'attention
- ✓ Mais il réduit la lisibilité du texte, notamment si le texte inversé est long.

- **La graisse :**

La graisse est moins efficace que l'inversion vidéo, mais cela dénature moins l'apparence graphique du texte. Il reste assez efficace pour un site web.

- **La taille :**

Ce mode de mise en évidence sert à représenter l'importance **relative** de différents objets graphiques.

Des comparaisons seront plus rapides d'après d'un diagramme en barres que d'après une table de données chiffrées.

- **La police de caractère :**

Ce codage est moins efficace que l'inversion vidéo ou la graisse, mais il peut être utilisé pour des textes longs.

- **Le soulignement :**

Il réduit la lisibilité et ne doit pas être utilisé pour des textes longs.

En association avec la couleur, ce mode de mise en évidence représente [les liens](#) dans les applications internet.

- **La forme :**

Elle permet d'identifier rapidement un type d'information.

Il convient de définir les règles d'association de formes aux types d'informations (règle de la cohérence).

- **Les puces**

Placées en début de lignes, elles contribuent à structurer la présentation d'un texte et à mettre en évidence les informations importantes.

- **Le rapprochement ou l'encadrement**

Les informations liées entre elles doivent être regroupées ou rapprochées.  
Un encadrement est un moyen efficace de regrouper de telles informations.

- **Le son**

Un signal sonore constitue une alerte efficace. Une urgence est indiquée avec un signal répétitif, que l'utilisateur doit pouvoir interrompre.

Ce mode de mise en évidence doit rester appliqué à des cas particuliers, pour ne pas agacer l'utilisateur (ni ses voisins).

Choisir un nombre de sons différents inférieur à 5, distincts en timbre et tonalité et offrir à l'utilisateur la possibilité de régler le niveau sonore.

- **La couleur**

Un objet de couleur différente par rapport au reste de l'interface sera mis en évidence, surtout si la couleur est saturée et s'il est accompagné d'un autre mode de mise en évidence .

### 3.8.Langage

Le langage de l'interface homme-machine est la base du dialogue entre l'utilisateur et l'application. Pour faciliter l'apprentissage et la mémorisation du logiciel, quelques règles et recommandations doivent être suivies.

- **Libellé de commandes**

Pour la désignation d'une commande :

- ✓ Un texte est plus explicite qu'une icône ; mais il prend plus de place et il est dépendant de la langue.
- ✓ Utiliser un seul libellé pour chaque commande.
- ✓ Utiliser une syntaxe homogène pour le libellé de commandes.
- ✓ Faciliter la distinction entre commandes, objets et états.
- ✓ Utiliser un verbe à l'infinitif dans le libellé. Ex. :
  - ▶ ***Imprimer la liste***
  - ▶ ***Supprimer les éléments sélectionnés***



- ✓ Éviter les verbes à la forme nominale (on les réservera à la description d'état).  
Ainsi, le terme "Affichage" sera-t-il lié au déclenchement d'un processus d'affichage ou à la présentation de paramètres d'affichage ?
- ✓ Éviter l'usage d'abréviations.

### • Messages

Ils sont utilisés soit pour questionner l'utilisateur sur les actions à venir, soit pour indiquer le résultat d'une action.

- ✓ Classer les différents types de messages suivant leur rôle et leur niveau :
  - ▶ Alertes (remarque intéressante, mise en garde, appel à la prudence et signalement d'erreur mineure, grave ou fatale) ;
  - ▶ Demandes de confirmation ;
  - ▶ Message d'autre zone de dialogue ;
  - ▶ Indications de progression ;
  - ▶ Indications d'état ou statut, etc.

- ✓ Définir le type de contenant approprié pour chaque type de message ; un message d'un type donné sera ainsi affiché dans un des conteneants suivants :
  - ▶ une zone de dialogue modale (déplaçable ou non) ;
  - ▶ une zone de dialogue non modale (déplaçable) ;
  - ▶ une zone de statut de la fenêtre courante...
- ✓ Utiliser une zone de dialogue comme contenant du message si dernier doit être lu.
- ✓ Utiliser la zone de statut (en bas de fenêtre) pour un message à lecture optionnelle.
- ✓ Sinon, afficher le message près de l'objet auquel il est lié.



- ✓ Employer une forme affirmative et active.

Par exemple, quels sont les messages préférables parmi les suivants ?

- ***Vous ne pouvez pas quitter avant de sauvegarder votre session.***
- ***Sauvegarder votre session avant de quitter.***

- ***La grille des programmes est affichée après avoir sélectionné la chaîne TV.***
- ***Pour afficher la grille des programmes, sélectionner la chaîne TV***

- ✓ Indiquer la marche à suivre :

- cela évitera les impasses ;
- cela participe à l'apprentissage du logiciel.



- ✓ Respecter l'ordre dans lequel les actions doivent être exécutées.
- ✓ Éviter les messages réprobateurs, critiques ou culpabilisants.
- ✓ Utiliser une terminologie précise, directe, concise et pertinente.
- ✓ Utiliser des messages adaptés au niveau de connaissances de l'utilisateur :
  - ▶ Pour un utilisateur expérimenté à l'usage du logiciel, un message d'erreur indiquant simplement la syntaxe de la commande suffira ;
  - ▶ Pour des débutants ou des utilisateurs occasionnels, des éléments sémantiques seront précisés (marche à suivre, alternatives...).
- ✓ Adopter un vocabulaire homogène et cohérent avec le reste de l'interface.
- ✓ Jouer sur la typographie pour faciliter la lecture des messages.



- ✓ Tenir compte des contraintes liées à la localisation de logiciels :
  - ▶ Prévoir une compatibilité mondiale dès le début du processus de développement ;
  - ▶ Ne pas inclure le libellé des messages directement dans le code.
  - ▶ Stocker toute information propre à une zone géographique dans des ressources indépendantes du reste du code ; utiliser autant que possible des références (ou indirections ou pointeurs) aux libellés, aux icônes et aux graphismes.
  - ▶ Tenir compte des longueurs différentes de texte suivant la langue, de sens de lectures différents, de conventions différentes sur l'usage des couleurs, des graphismes, des calendriers et des heures...

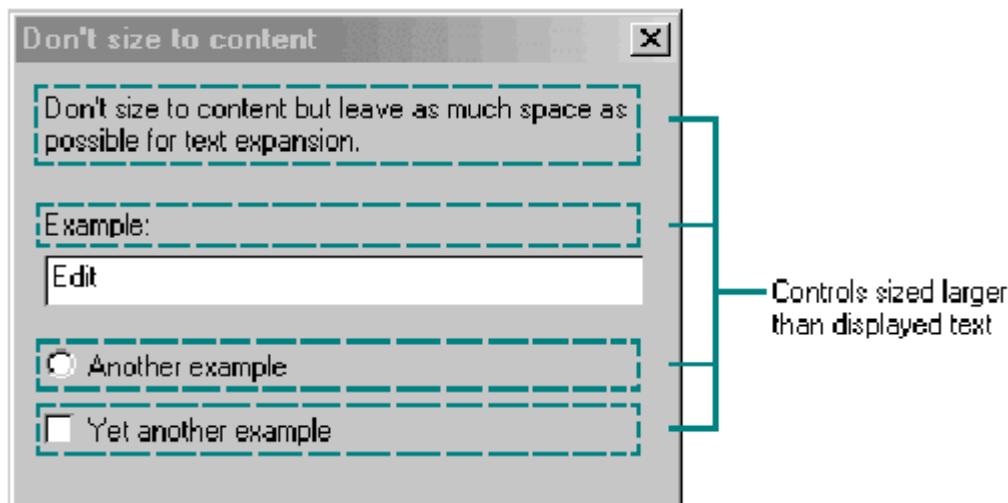


Figure 3.13 — Prévoir le maximum de place pour le texte

## 4. Le dialogue homme-machine

### 4.1. Organisation du dialogue

Le dialogue homme-machine correspond aux échanges entre l'utilisateur et le logiciel.

Les modes de dialogue correspondent à différentes formes que prend ce dialogue. Ex. :

- ▶ appui sur une combinaison de touches du clavier,
- ▶ utilisation du bouton droit de la souris
- ▶ ou manipulation de composants de l'interface.

Une interface homme-machine propose souvent plusieurs modes pour une même activité.

L'agencement du dialogue détermine l'utilisabilité du logiciel et repose sur trois principes, respectivement liés aux critères de compatibilité, de contrôle utilisateur et de guidage :



- **Adapter la structure du dialogue à l'activité de l'utilisateur :**

- ✓ Les commandes et les données doivent se présenter dans **l'ordre d'utilisation**.
  - ▶ L'utilisateur doit disposer des commandes dont il a besoin au cours du déroulement de son activité.
  - ▶ Permettre à l'utilisateur d'effectuer ce qu'il veut quand il le veut.
- ✓ **Regrouper** les informations dépendantes et liées à une même tâche dans une même fenêtre.
  - ▶ Chaque fenêtre forme alors un ensemble cohérent de données et de commandes.

- **Laisser à l'utilisateur l'initiative du dialogue :**

- ✓ Pour éviter les impasses, donner les possibilités de :
  - ▶ Revenir en arrière (**Undo** ; Annuler) ;
  - ▶ Abandonner un dialogue en cours (**Cancel** ; Annuler) ;
  - ▶ Interrompre un traitement en cours (**Stop**)



- **Fournir des moyens de guidage, afin que l'utilisateur puisse s'orienter :**
  - ✓ Permettre une **navigation simple** et logique au sein du logiciel ;
    - ▶ Laisser une trace du chemin parcouru (un fil d'ariane) en présentant les différents choix effectués (ou les différents niveaux traversés).

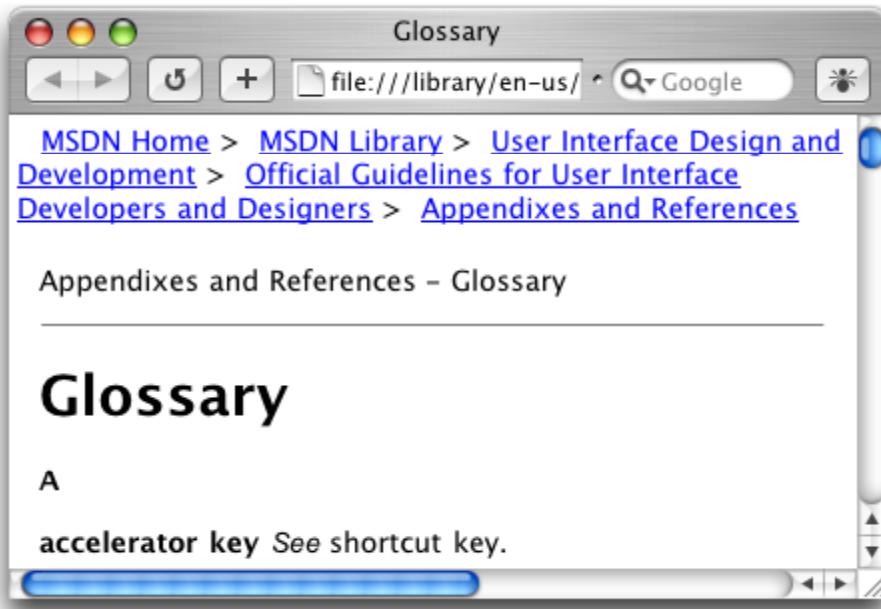


Figure 4.1 — Dans une page web, laisser une trace du chemin parcouru est usuel et pratique



## 4.2. Conception des fenêtres

- Nous considérerons différents types de fenêtre :

✓ Une fenêtre (*window*) principale d'un système de multi-fenêtrage ; on réserve le terme de fenêtres à ces zones gérées de manières indépendantes.

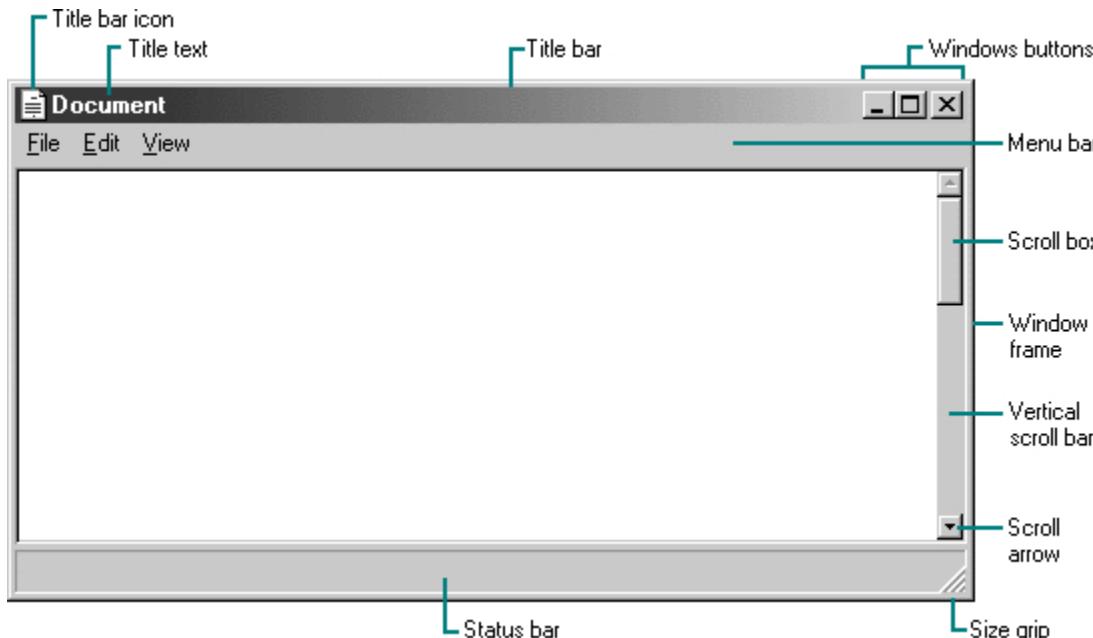


Figure 4.2 — Une fenêtre standard (Windows 98)



- ✓ Une fenêtre utilitaire (*utility window*) ; c'est une boîte d'outils (*toolbox*) ou une palette (*palette*) dont on ne peut faire défiler le contenu et qui propose un choix d'outils ou de commandes.
- ✓ Une zone de dialogue (*dialog box*) est une zone, ou une fenêtre particulière, qui dépend d'un contexte donné et qui contient, outre des données à consulter ou à modifier, des boutons qui servent à déclencher un traitement sur la base des paramètres définis dans la zone.
  - ▶ Une zone de dialogue permet soit de solliciter une information de l'utilisateur, soit de l'informer de l'attente d'exécution d'une procédure.
  - ▶ La conception de zones de dialogue fera l'objet du prochain paragraphe.

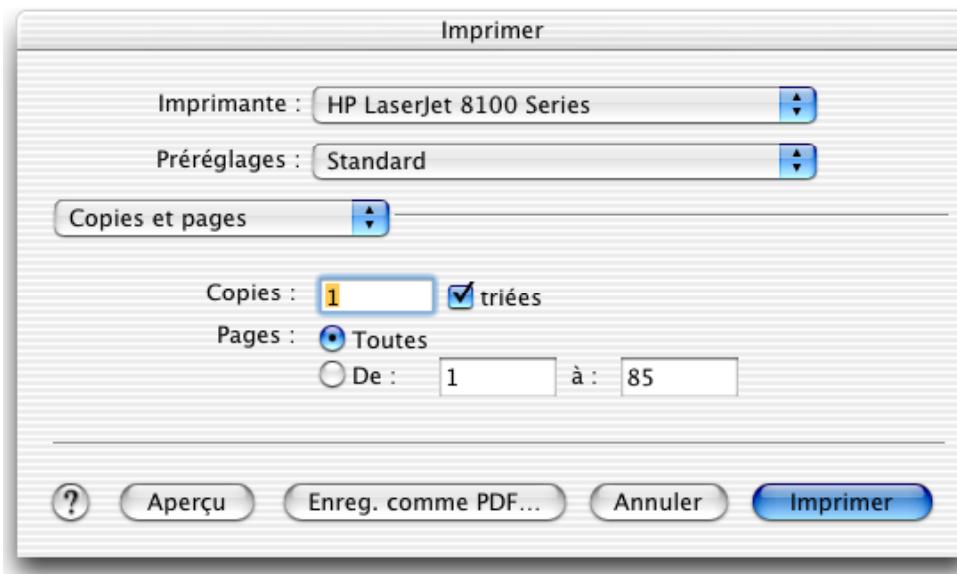


Figure 4.3 — Une zone de dialogue

### ● Multi-fenêtrage

Le multi-fenêtrage est une fonctionnalité connue des utilisateurs de la plupart des systèmes d'exploitation proposant une métaphore de bureau.

Lorsqu'elle est créée, une nouvelle fenêtre recouvre une partie du bureau et des fenêtres déjà affichées.

- ▶ L'utilisateur (averti) peut déplacer où il le souhaite, agrandir ou réduire la nouvelle fenêtre :
- ▶ il organise son espace de travail selon sa tâche et ses préférences.

Un certain apprentissage des commandes de manipulation des fenêtres reste nécessaire.

Les recommandations sont les suivantes :

- ✓ Simplifier la gestion des fenêtres afin de ne pas perturber la tâche métier.
- ✓ Rester cohérent avec la gestion des fenêtres du système d'exploitation.
- ✓ Faciliter l'ouverture et l'activation des fenêtres.
- ✓ Faciliter la mémorisation des commandes d'organisation des fenêtres.
- ✓ Mettre en évidence la fenêtre active.
- ✓ Automatiser ou autoriser la mémorisation d'un arrangement de fenêtres.
- ✓ La taille des fenêtres doit être compatible avec la taille de l'écran.
- ✓ Moins il y a de fenêtres, plus elles sont complexes et difficiles à utiliser.
- ✓ Permettre à un utilisateur expérimenté d'utiliser moins de fenêtres :
  - il peut préférer des fenêtres plus complexes mais moins de manipulations de fenêtres.
- ✓ Minimiser le nombre d'information à mémoriser d'une fenêtre à l'autre.

### 4.3. Conception des zones de dialogues

Suivant les modalités d'utilisation de zones de dialogues, on peut distinguer :

✓ La zone de dialogue **non modale** (*modeless dialog box*):

- ▶ Similaire à une fenêtre de document mais sans bande de défilement ni contrôle de taille, l'utilisateur peut déplacer, désactiver, réactiver ou refermer une telle zone.
- ▶ Une zone de dialogue non modale permet par exemple d'effectuer des modifications dans les documents d'après les paramètres saisis par l'utilisateur.
- ▶ Ex. : Zone de dialogue **Rechercher & remplacer** :

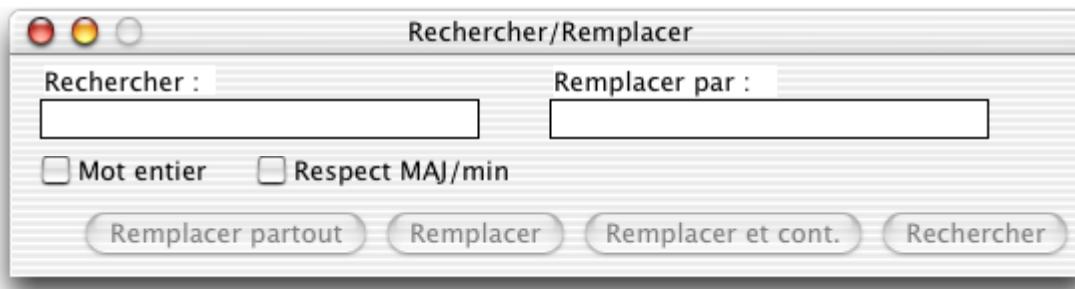


Figure 4.4 — Une zone de dialogue non modale



✓ La zone de dialogue **modale déplaçable** :

- ▶ Elle reste au premier plan tant que l'utilisateur ne la referme explicitement par un bouton de commande.
- ▶ Elle sert également à effectuer des modifications dans les documents ;
- ▶ si elle masque une partie d'une fenêtre en arrière plan que souhaite voir l'utilisateur, il la déplace.

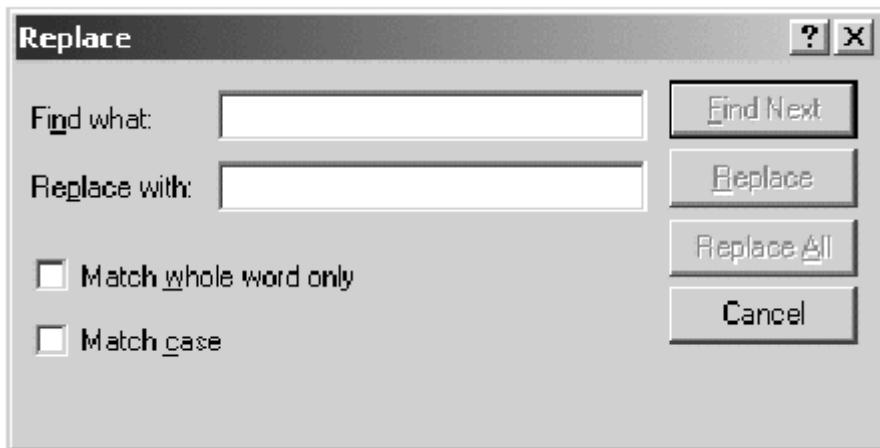


Figure 4.5 — Une zone de dialogue modale déplaçable



✓ La zone de dialogue **modale non déplaçable** :

- ▶ Elle reste fixe et au premier plan tant que l'utilisateur ne la referme explicitement.
- ▶ Elle **constraint** l'utilisateur à fournir des informations indispensables pour effectuer une opération en cours.

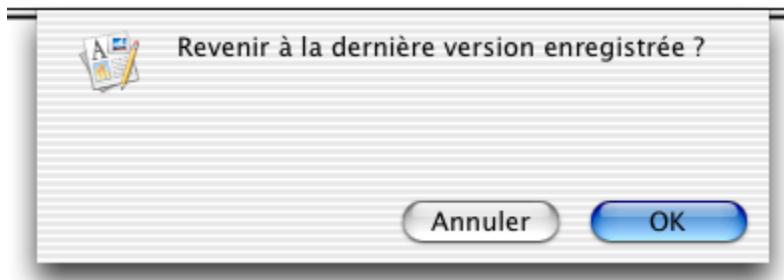


Figure 4.6 — Dialogue modale non déplaçable

✓ Le **message d'alerte** (*Alert box*):

- ▶ C'est un type de zone de dialogue modale qui sert à signaler à l'utilisateur une **situation d'erreur** et à **empêcher toute autre activité** tant que l'utilisateur ne rectifie pas la situation d'erreur.

Une zone de dialogue modale contraint l'utilisateur à ne travailler qu'à l'intérieur de la zone. La plupart des menus deviennent inaccessibles.

- ✓ Exemple type de zones de dialogue modales :
  - Une zone de dialogue d'ouverture de fichier
  - une zone de dialogue d'enregistrement de fichier.

Les règles à observer sont :

- ✓ Présenter les informations **pertinentes** par rapport à la **tâche**.
  - Le contenu de la zone de dialogue doit être adapté à l'attente de l'utilisateur.
- ✓ Afficher la zone de dialogue à un **emplacement pratique**, là où l'utilisateur regarde :
  - à l'intérieur de la fenêtre à laquelle cette zone est liée
  - ou à défaut, centrée par rapport à l'écran.
- ✓ Présenter les composants dans **l'ordre d'utilisation**.
- ✓ Minimiser les déplacements de la souris.
- ✓ Fournir un retour sur la position du focus (le composant actif) et, le cas échéant, du curseur (dans un champs de saisie).
- ✓ Faciliter l'accès aux composants les plus utilisés.



- ✓ Mettre en évidence les éléments les plus importants.
- ✓ Organiser la zone de dialogue à l'aide de séparateurs et en regroupant les composants liés entre eux.
- ✓ Guider l'utilisateur ; fournir une aide contextuelle ou donner le moyen d'afficher une aide.
- ✓ Permettre l'utilisation de raccourcis clavier :
  - Touche **Tabulation** pour passer au composant actif suivant ;
  - Touches **Majuscule+Tabulation** pour passer au composant actif précédent ;
  - Touche **Entrée (Enter)** pour la validation du dialogue ;
  - Touche **Échappement (Escape)** pour annuler le dialogue ;
  - Autres raccourcis en cohérence avec l'application et le système d'exploitation.
- ✓ Uniformiser la syntaxe employée dans tous les dialogues de l'application.

## 4.4.Les composants de zones de dialogues

Les composants d'une zone de dialogue sont :

- ✓ les cases de commande
- ✓ les boutons de commande,
- ✓ les libellés,
- ✓ les composants de sélection,
- ✓ les champs de saisie...

Voici les recommandations concernant ces composants :

- **Case de commande**

Également appelé **bouton de commande** (*command button*), la case de commande permet un accès rapide, par un simple clic, à une commande.

- ▶ L'aspect du bouton est modifié dès que le bouton de la souris est enfoncé alors que le pointeur est dans le bouton (état bouton pressé).
- ▶ Cependant, l'action associée au bouton n'est déclenchée que si le bouton de la souris est relâché alors que le pointeur est bien situé dans le bouton.

- ✓ Utiliser des cases de commande pour les commandes **fréquemment** utilisées.
- ✓ Placer l'intitulé de la commande en toute lettre dans la case.
- ✓ Griser les boutons inactifs.



- ✓ Parmi les boutons d'une zone de dialogue, un seul peut être le bouton par défaut.
  - Il représente l'action que l'utilisateur a le plus de chance d'effectuer, à condition que cette action n'est pas potentiellement dangereuse.
  - Ce bouton par défaut est reconnaissable par une mise en évidence spécifique, conformément au guide de l'interface utilisateur en vigueur.



Figure 4.7 — Boutons de commande par défaut pour Windows 98 et pour Mac OS X

- ✓ Toujours permettre à l'utilisateur de revenir en arrière, donc d'annuler l'action qui a provoqué l'affichage de la zone de dialogue.

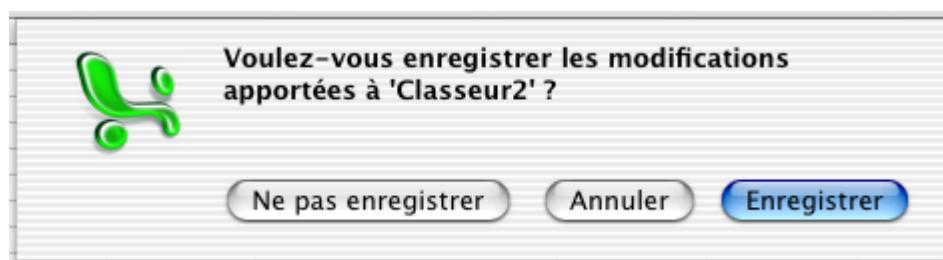


Figure 4.8 — Le bouton "Annuler" va annuler la commande "Quitter" ou "Fermer" qui a provoqué l'apparition de cette zone de dialogue modale.



## • Les composants de sélection

Il s'agit de différents composants comme :

- ✓ une liste de sélection,
  - ✓ une combo-box,
  - ✓ une série de boutons radio,
  - ✓ une série de cases à cocher...
- 
- ✓ Utiliser les boutons de sélection comme des boutons radio (*radio buttons*) ou des cases à cocher (*checkbox buttons*) pour des choix peu nombreux mais fréquents.
    - ▶ Pour des choix exclusifs, utiliser alors forcément les boutons radio.



Figure 4.9 — Cases à cocher et boutons radio



- ✓ La liste de sélection est proposée pour des choix peu fréquents ou lorsque la place est réduite.

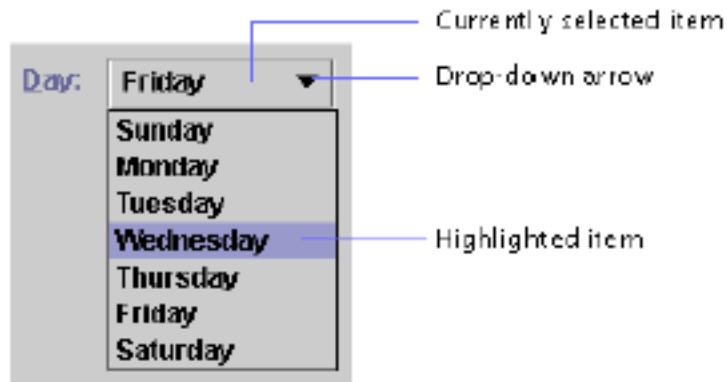


Figure 4.10 — Liste de sélection



- ✓ Un combo-box est une liste de sélection avec une barre de défilement et un champs de saisie éditable.
  - ▶ Il est donc utilisé lorsque le nombre de choix est important ou variable, ou lorsqu'un champs de saisie doit être associé.

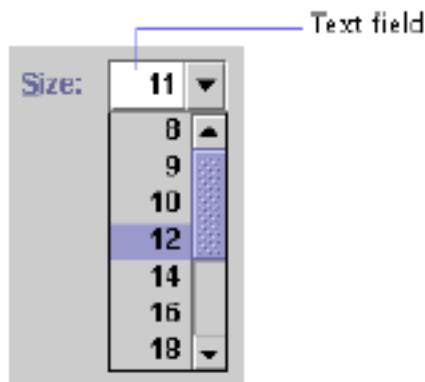


Figure 4.11 — Combo-box

- **Les champs de saisie**

Pour remplir un champs de saisie, l'utilisateur se focalise sur une région très réduite de l'écran. Cette petite région doit donc contenir tout le nécessaire à la saisie, sans surcharge inutile.

- ✓ Choisir des libellés concis mais très compréhensible.
- ✓ Rester homogène et cohérent au niveau de la présentation des libellés et des champs de saisie :
  - ▶ Utiliser des gabarits pour l'espacements des éléments
  - ▶ Respecter un alignement cohérent au sein des champs de saisie (textes alignés à gauche dans le champs et nombres alignés à droite).
- ✓ Proposer, si cela est possible,
  - ▶ une valeur par défaut,
  - ▶ l'accès aux dernières saisies,
  - ▶ ou permettre de compléter automatiquement la saisie.



- ✓ Prévenir les erreurs de saisie.
- ✓ Préciser le format de la saisie.
- ✓ Permettre à l'utilisateur de choisir les unités.
- ✓ Vérifier si un combo-box n'est pas préférable plutôt qu'un champs de saisie.
- ✓ Éviter les passages de minuscule à majuscule, les saisies de zéros inutiles...



## 4.5. Conception des menus

Les menus proviennent d'une barre de menu, d'un menu contextuel ou d'un menu déroulant.

- ✓ La **barre de menu** (*menu bar*) est un ensemble de menus.
- ✓ Un **menu**, (*drop-down menu*) déroulé à partir d'une barre de menu, d'un menu déroulant ou d'un menu contextuel est **composé** d'une suite d'**articles de menu** (*menu items*).

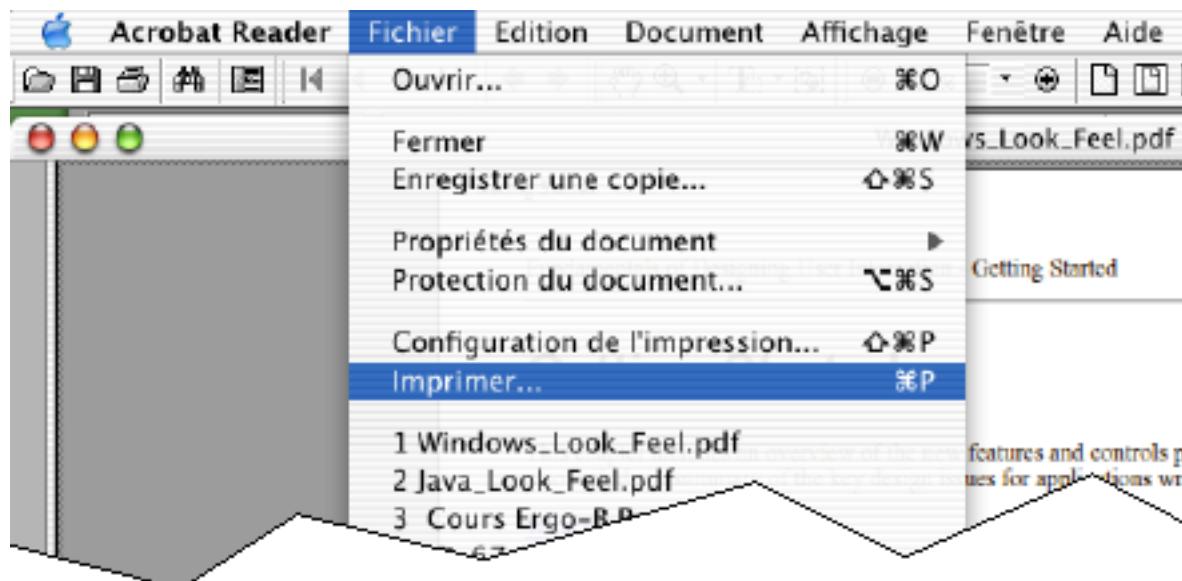


Figure 4.12 — Sélection d'un article d'un menu d'une barre de menu



- ✓ Il peut être **hiérarchique** et donc comporter un sous-menu (*submenu*) qui se déroule à partir d'un article de menu.

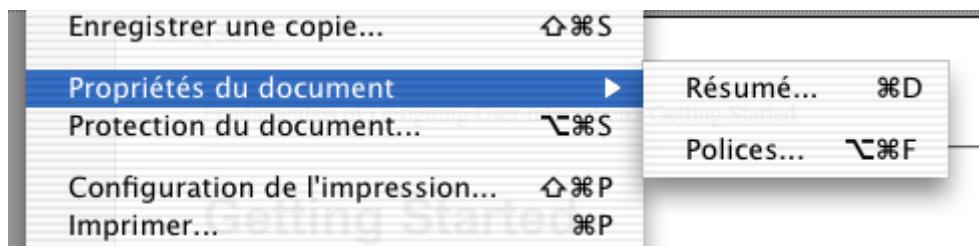


Figure 4.13 — Le sous-menu d'un article de menu

- ✓ Les menus sont faciles à apprendre :
  - ils présentent un ensemble de commandes disponibles dans un contexte de tâche donnée.
  - Ils répondent au principe de "voir et pointer", où un utilisateur retrouve toutes les options disponibles à un moment donné (et où les commandes indisponibles apparaissent en grisées).



✓ Mais les menus impliquent :

- un déplacement de la souris,
- un clic sur une barre de menu ou un clic droit sur un objet,
- l'affichage du menu,
- la sélection d'un article dans ce menu à l'aide de la souris.

Cela impose une certaine dextérité et quelques délais, surtout si le nombre d'articles se trouve important.

- ✓ Certaines contraintes de conception de barre d'outils (toolbar) sont proches de celles de la conception de menu.
- ✓ Un raccourci clavier est souvent associé à un article de menu.

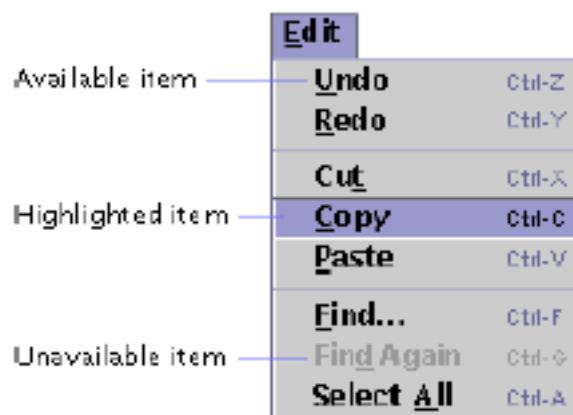


Figure 4.14 — Les articles d'un menu



- ✓ Certains articles fonctionnent en bascule. Par exemple, l'action dans un menu *Fenêtre* de l'article *Afficher les outils* affichera la palette d'outils et l'article sera remplacé par *Masquer les outils*.
- ✓ D'autres articles seront précédés d'une coche si l'attribut correspondant est activé.



Figure 4.15 — Usez des coches avec parcimonie

- **Recommandations générales**

- ✓ Les menus et articles de menu actifs dépendent du **contexte** de la tâche.
  - ▶ La structure des menus doivent être ajustée à l'activité et au contexte.
- ✓ Minimiser la taille des menus
  - ▶ **10 articles** conviendront à un débutant, alors qu'un utilisateur expérimenté assurera avec 20 articles. Au delà, cela sera fastidieux.
- ✓ Organiser un menu suivant :
  - ▶ l'ordre d'utilisation ;
  - ▶ la fréquence d'utilisation ;
  - ▶ l'importance des commandes.
- ✓ Utiliser des **séparateurs** pour mettre en évidence des **groupes** d'items.
- ✓ **Séparer** les commandes à effet destructeur.



- ✓ Pour le **nommage** d'articles de menus, il y a lieu de considérer les différentes catégories :
  - ▶ Si l'article est lié à une commande ou une action, alors préférer le **verbe** de l'action.  
Ex. : Article *Fermer* du menu *Fichier*.
  - ▶ Si l'article modifie l'attribut d'un objet sélectionné, il est judicieux d'utiliser un **adjectif** pour nommer cet article.  
Ex. Article *Gras* du menu *Style*.
- ✓ **Griser** les commandes non disponibles.
- ✓ Adjoindre un **raccourci clavier** aux commandes fréquentes.
- ✓ Placer des **points de suspension** à la suite d'un article de menu lorsque la commande exige des **informations complémentaires** avant d'être exécutée.
- ✓ Respecter le nommage, l'emplacement et le rôle des menus standards d'une barre de menu.

### ***Fichier, Édition, ... , Aide***

Se référer au guide de l'interface utilisateur (*guideline*) approprié.



- ✓ En général, le menu **Fichier** contient les éléments liés à la gestion de l'application et de ses documents :

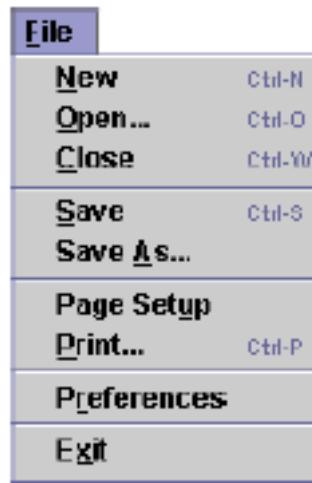


Figure 4.16 — Exemple de menu Fichier

- ▶ Nouveau ; un sous-menu peut préciser les types de document à créer.
- ▶ Ouvrir...
- ▶ Fermer ; s'applique à la fenêtre au premier plan (et au document qu'elle contient).
- ▶ Enregistrer
- ▶ Enregistrer sous...
- ▶ Version précédente
- ▶ Format d'impression... (ou Mise en page...)
- ▶ Imprimer...
- ▶ Préférences...
- ▶ Quitter



- ✓ Le menu ***Édition*** présente les articles permettant à l'utilisateur de changer et éditer le contenu de documents et d'autres éléments :



Figure 4.17 — Un exemple de menu Édition

- ▶ Annuler ; préciser ce que cela annule.
- ▶ Rétablir ; préciser ce que cela rétabli.
- ▶ Répéter ; préciser ce que cela répète.
- ▶ Couper
- ▶ Copier
- ▶ Coller
- ▶ Effacer (ou Supprimer)
- ▶ Tout sélectionner
- ▶ Les articles pour rechercher/remplacer...
- ▶ S'il y a lieu, les articles liés à l'insertion d'éléments.



- ✓ Un menu **Format** sera utile pour les commandes de formatage du document et des textes.
- ✓ Un menu **Affichage** sera utile pour les commandes liés au zoom, à la taille du document par rapport à la fenêtre, etc.
- ✓ Un menu **Fenêtre** peut présenter les éléments de gestion de fenêtres et de palettes.  
(Afficher/masquer, premier plan/arrière plan, gestion de barres d'outils, etc.)
- ✓ Le menu **Aide** présente l'aide pour l'application et les modules installés, l'accès à des éléments de support sur le web, etc.



### • Les menus contextuels ou déroulants

Un menu déroulant est une liste de sélection. Nous l'avons donc déjà décrit avec le paragraphe 4.4 - *Les composants de zones de dialogues*.

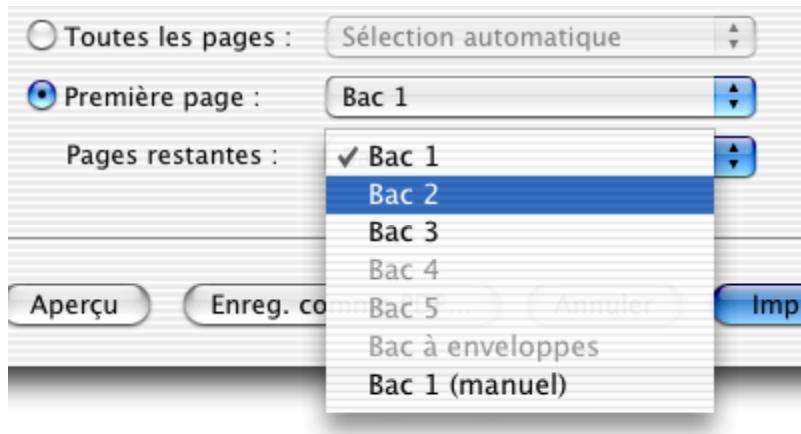


Figure 4.18 — Menus déroulants



Le menu contextuel (*contextual menu*) est généralement déclenché à l'aide du **bouton droit** de la souris alors que le pointeur est situé sur un composant concerné.

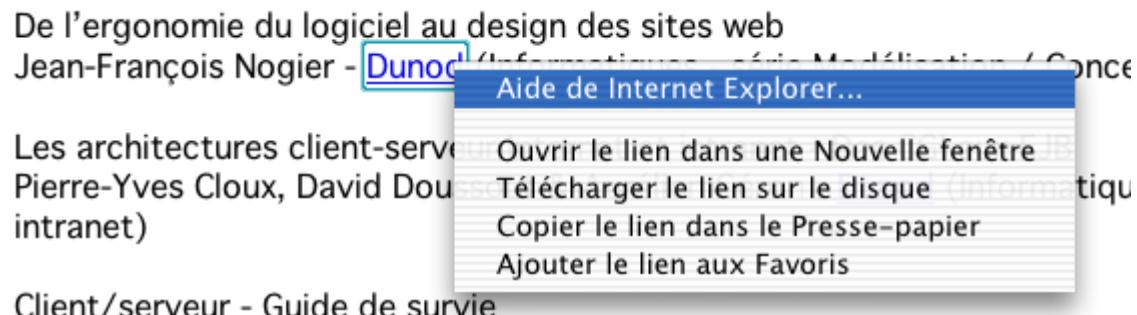


Figure 4.19 — Un menu contextuel

Son avantage est de minimiser le déplacement de la souris pour atteindre des commandes et actions possibles pour l'objet sélectionné ou pour les éléments liés. C'est également un moyen pour économiser de la place.

Cependant, l'apprentissage de l'usage des menus contextuels n'est pas immédiat.

- ✓ Permettre un **menu contextuel** pour des **commandes fréquentes** et rapides.
- ✓ Mettre en évidence les composants associés à un menu contextuel.
- ✓ Éviter les menus contextuels pour les commandes occasionnelles.

## 4.6.Raccourcis clavier et touches de fonctions

Un **raccourci clavier** ou équivalent clavier (*keyboard equivalent*) est une combinaison de touches du clavier utilisable pour **activer une commande** donnée. Un raccourci clavier est souvent associé à un article de menu, et éventuellement à des cases de commande de zone de dialogue.

L'apprentissage des raccourcis clavier est donc favorisé par l'usage fréquent de certains articles de menu.

- ✓ **Respecter les raccourcis réservés** et recommandés pour le système d'exploitation.
  - ▶ En particulier, ceux des commandes des menus *Fichier* et *Édition* sont bien connus par les utilisateurs.
- ✓ Éviter d'assigner des raccourcis clavier à des commandes occasionnelles.

Les **touches de fonction** (*function keys*) sont usuellement situées sur la rangée supérieure du clavier. Les touches les plus accessibles sont aux extrémités.

- ✓ Les utilisateurs expérimentés apprécient l'utilisation de touches de fonction, malgré l'effort d'apprentissage nécessaire.
- ✓ Utiliser les touches les plus accessibles pour les commandes fréquentes
- ✓ Éviter d'assigner des touches de fonction pour les commandes à risque, ou alors utiliser les touches d'accès difficiles (et permettre de défaire la commande).
- ✓ Utiliser une même touche pour les commandes à bascule (comme *Afficher/Masquer*).

## 4.7. Manipulation directe

- **Généralités**

C'est le mode de dialogue prédominant d'une interface utilisateur graphique (*GUI : Graphical User Interface*). La **souris** — ou d'autres dispositifs de pointage — permet à l'utilisateur d'agir directement sur les objets et de les **manipuler**.

La manipulation directe s'appuie sur des métaphores du monde réel et se base sur des enchaînements du type "**voir, pointer, agir**".

Cela respecte la forme courante des actions utilisateurs : **nom**, puis **verbe**.

- ✓ L'utilisateur peut ainsi **choisir un objet** qui l'intéresse (le nom) puis **choisir les actions** qu'il veut effectuer sur cet objet (le verbe). Le choix des actions (des **verbes**) disponibles se retrouvent dans les menus, que l'utilisateur peut parcourir pour choisir une action disponible sans avoir à se souvenir d'un nom de commande particulier.
- ✓ L'utilisateur peut faire **glisser un objet** (le nom) **sur un autre objet** auquel une action (le verbe) est associée.
  - Il ne choisit alors pas une action dans un menu, mais ce qui arrive à l'objet lorsqu'il est placé sur un autre objet est évident.
  - (Exemple : faire glisser une icône sur une corbeille signifie bien que l'on veut se débarrasser de ce que représente l'icône.)



Avec l'exemple de la métaphore du bureau (et de son environnement), il est évident pour l'utilisateur que :

- ▶ Un document informatique peut être rangé dans un dossier, qui visuellement ressemble à une chemise papier.
  - ▶ Mais, contrairement à la chemise papier, un dossier informatique peut stocker un nombre quasi-illimité de documents ou de sous-dossiers ; (on sera en fait limité par la capacité physique du disque).
- ✓ La manipulation directe donne l'impression à l'utilisateur de contrôler directement les objets affichés.
- ✓ Une action élémentaire est généralement réversible.
- ✓ L'impact de la commande réalisée par une action élémentaire est directement visible.

- **Avantages**

- ✓ C'est un mode d'interaction familier, **facile à mémoriser** et à apprendre :
  - ▶ L'utilisateur peut se focaliser sur le résultat qu'il cherche à obtenir et non sur la manière de déclencher la commande.
- ✓ Les erreurs sont peu nombreuses, car les objets manipulés sont visibles et changent de présentation en fonction des actions de l'utilisateur.
  - ▶ Le clavier étant moins utilisé, les erreurs de saisie sont plus rares.
- ✓ Chaque commande est **réversible**, notamment par l'action ou la manipulation inverse :
  - ▶ L'utilisateur peut rattraper ses erreurs, en défaisant sa commande, en l'annulant ou en revenant en arrière.
- ✓ Les utilisateurs occasionnels ou ayant peu de connaissance informatique mais connaissant le domaine applicatif resteront à l'aise avec la manipulation directe.

### ● Inconvénients

- ✓ Ce mode de dialogue **manque de guidage**.
  - ▶ Il ne fournit généralement pas d'indication sur la façon de s'en servir.
  - ▶ Il n'est pas auto-explicatif comme peut l'être une fenêtre de dialogue.
- ✓ La méconnaissance du domaine applicatif est alors un handicap.
- ✓ La manipulation de petits objets nécessite un effort de précision.
- ✓ Les utilisateurs expérimentés préféreront un dialogue clavier, pour gagner en rapidité.

### ● Conséquences

La manipulation directe sera bien adaptée au traitement de données géo-référencées, tel que :

- ✓ La surveillance de processus ;
- ✓ La simulation ;
- ✓ La CAO ;
- ✓ le traitement d'images vectorielles (ex. Adobe Illustrator), etc.

## • Recommandations

Elles visent à réduire les distances sémantiques et articulatoires (qui ont été définis au paragraphe 2.2 sur la théorie de l'action).

✓ **Réduire la distance sémantique** (celle qui sépare le but fixé par l'utilisateur et les objets représentés de l'interface). À cette fin, pour faciliter la compréhension du système :

- Adapter les fonctionnalités aux intentions de l'utilisateur.  
Les **fonctionnalités** proposées répondent alors aux **besoins** de l'utilisateur.
- Présenter les **résultats** sous forme directement **exploitables**.  
Cela va réduire également la distance sémantique, puisque les données affichées répondent aux questions de l'utilisateur.

✓ **Réduire les distances articulatoires**, c'est à dire minimiser les ressources perceptives et motrices nécessaires à l'utilisation du logiciel (et donc faciliter la manipulation et la lecture) :

- Transcrire les commandes sous forme physique.
- Exemple : pour déplacer un fichier, on déplace son icône.
- Afficher les informations de façon à **faciliter leur lecture** et en évitant les ambiguïtés. Une représentation graphique est plus simple à lire et plus rapide à interpréter qu'un texte.  
Exemple : une barre de progression d'un traitement en cours est plus explicite que l'affichage d'un pourcentage.

- ✓ Fournir un **retour visuel** d'une manipulation directe. Plusieurs moyens sont disponibles :
  - ▶ Modifier l'apparence de l'objet lié à la commande ;
  - ▶ Modifier l'apparence de l'objet manipulé ;
  - ▶ Modifier la forme du pointeur souris.
- ✓ **Aider** les utilisateurs débutants :
  - ▶ Fournir un guidage via la barre d'état de la fenêtre ;
  - ▶ Proposer des alternatives par des menus ou des dialogues ;
  - ▶ Afficher une aide contextuelle lors du pointage d'un objet ou d'une commande (au *survol souris*)
- ✓ Offrir aux utilisateurs expérimentés des **accès rapides**, à travers des modes de dialogues alternatifs : raccourcis clavier et touches de fonctions.
- ✓ Simplifier la sélection (et la désélection) de données ; permettre des sélections multiples.
- ✓ Faciliter les tâches répétitives :
  - ▶ Proposer la commande "Redo"
  - ▶ Permettre la création de scripts ou de macro-commandes.

- ✓ Améliorer la précision des saisies :
  - ▶ Permettre l'activation d'un effet magnétique (et sa désactivation)
  - ▶ Proposer une saisie alternative au clavier.
- ✓ Rester homogène quant au rôle des boutons et opérations de la souris :
  - ▶ **Bouton de gauche => sélection**, choix de menu déroulant, glisser-déposer :
    - **Simple clic** : dans une zone de texte pour fixer le point d'insertion, sinon pour la sélection d'un objet sous le pointeur.
    - **Double-clic** dans une zone de texte, pour sélectionner le mot sous le pointeur ; pour un autre composant le double-clic va l'ouvrir, si la métaphore est correcte. (Double-clic d'une icône pour l'ouverture du document ou du dossier, ou alors, pour un autre type d'objet, ouverture d'une zone de dialogue...)
    - **Triple-clic** dans une zone de texte, pour sélectionner une portion plus grande de texte (ligne ou paragraphe).
    - **Glisser** : soit pour sélectionner un ensemble de composants, soit pour déplacer un objet.
  - ▶ **Bouton droit => menu contextuel**
- ✓ Ne pas déclencher de fonctions importantes ou difficilement réversibles par un simple clic. Un clic accidentel est fréquent.

### • Le temps de réponse

Pour l'utilisateur, le temps de réponse est une **attente** ; cela mesure le temps entre l'envoi d'une commande par l'utilisateur et la réponse du système.

Ce qui intéresse l'utilisateur, c'est le temps global de réalisation de la tâche, qui inclus :

- ✓ le temps de réponse, le délai d'affichage du résultat,
- ✓ le temps nécessaire pour lire, comprendre et interpréter la réponse
- ✓ et le temps mis pour décider et mettre en œuvre les actions consécutives à cette réponse.

Si on ajoute à cela le temps perdu par erreur et le temps mis pour corriger ces erreurs, on est loin d'un simple temps de réponse « informatique » du système.

Le **temps de réponse** influe sur l'utilisabilité du logiciel :

- ✓ C'est un facteur de stress, surtout si ce temps de réponse s'allonge et qu'aucun retour ne l'informe des traitements en cours.
- ✓ Le temps de réponse alourdit la charge de travail, car cela oblige l'utilisateur à faire des efforts de mémorisation des informations nécessaires à l'échange. Au delà de 2 à 6 secondes, on dépasse les capacités de la mémoire à court terme et les mécanismes de mémorisation sont plus complexes. Certaines informations risquent alors d'être oubliées.

Lorsque les temps de réponse sont faibles et si les erreurs sont facilement corrigées, l'utilisateur peut adopter une stratégie du type *essai - erreur*, sans craintes et avec peu de stress.

Des temps de réponse importants ou des erreurs difficiles à corriger vont allonger les temps de réflexion de l'utilisateur qui va commettre moins d'erreurs, avec plus de stress.



D'où les conseils suivants :

- ✓ Minimiser les **variations** du temps de réponse.
- ✓ Rendre les sources de variations visibles. Un système peut par exemple afficher le nombre d'utilisateurs connectés si cela influence le temps de réponse.
- ✓ **Changer la présentation** de l'objet sur lequel porte l'action : l'utilisateur perçoit immédiatement que son action a été prise en compte ; il ne va donc pas réitérer son action inutilement.
- ✓ Afficher une icône ou un **curseur d'attente** (sablier...) pour des temps de réponse estimés entre 2 et 6 secondes.



Figure 4.21 — Exemples de curseurs d'attente



- ✓ Au delà de 6 secondes, afficher un **dialogue de progression** du traitement en cours.
  - ▶ La zone de dialogue de progression permet à l'utilisateur d'estimer le temps d'attente.
  - ▶ La barre d'avancement, ou jauge (*Progress bar*) doit être rafraîchie régulièrement pour bien montrer que le traitement est en cours.
- ✓ Un **message** qui précise le type d'activité en cours sera souhaitable pour les processus de plusieurs minutes.

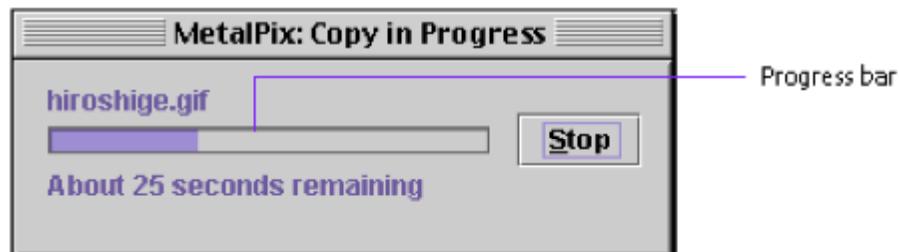


Figure 4.21 — Une zone de dialogue de progression

- ✓ **Laisser la main** à l'utilisateur :
  - ▶ il peut alors interagir avec le système à travers d'autres fenêtres que celle où une attente locale interdit l'interaction.



## 5. Les erreurs ; l'aide en ligne

### 5.1.Traitement des erreurs

Il est fondamental de mettre en œuvre les moyens nécessaires pour prévenir les erreurs de l'utilisateur et de lui fournir une aide adaptée. C'est d'ailleurs souvent lorsqu'une erreur survient que l'utilisateur consulte pour la première fois l'aide en ligne.

L'erreur est perçue par l'utilisateur comme un échec ; elle perturbe le déroulement de la tâche telle que l'utilisateur l'avait prévue ; elle augmente le stress et peut favoriser le rejet du logiciel.

Pour mieux prévenir et corriger les erreurs, nous allons, avec ce paragraphe, analyser les différents types d'erreur et fournir, pour chaque type d'erreur, des recommandations appropriées. Le paragraphe suivant expliquera comment concevoir une bonne aide en ligne.

- **Erreurs d'intention**

Une erreur d'intention est commise lorsque la manière de faire est correcte, mais que l'action ne permet pas d'atteindre le but fixé.

L'utilisateur **n'a pas utilisé** la commande ou l'action correcte, par exemple suite à une incompréhension du sens des commandes ou de l'enchaînement des opérations.  
L'utilisateur fait une interprétation incorrecte.

Pour éviter les erreurs d'intention, le fonctionnement du système doit être bien compris par l'utilisateur. Le logiciel est alors facile à apprendre et l'utilisateur commet moins d'erreur car il s'est construit un modèle mental du fonctionnement du système.

À cette fin :

- ✓ **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, en répondant aux questions du type :  
*Comment faire pour [...] ?*

- Erreurs d'exécution

Elles correspondent à des "défaillances" du processeur humain. Le modèle du processeur humain (figure 1.1) nous permet de distinguer les origines d'erreurs d'exécution.

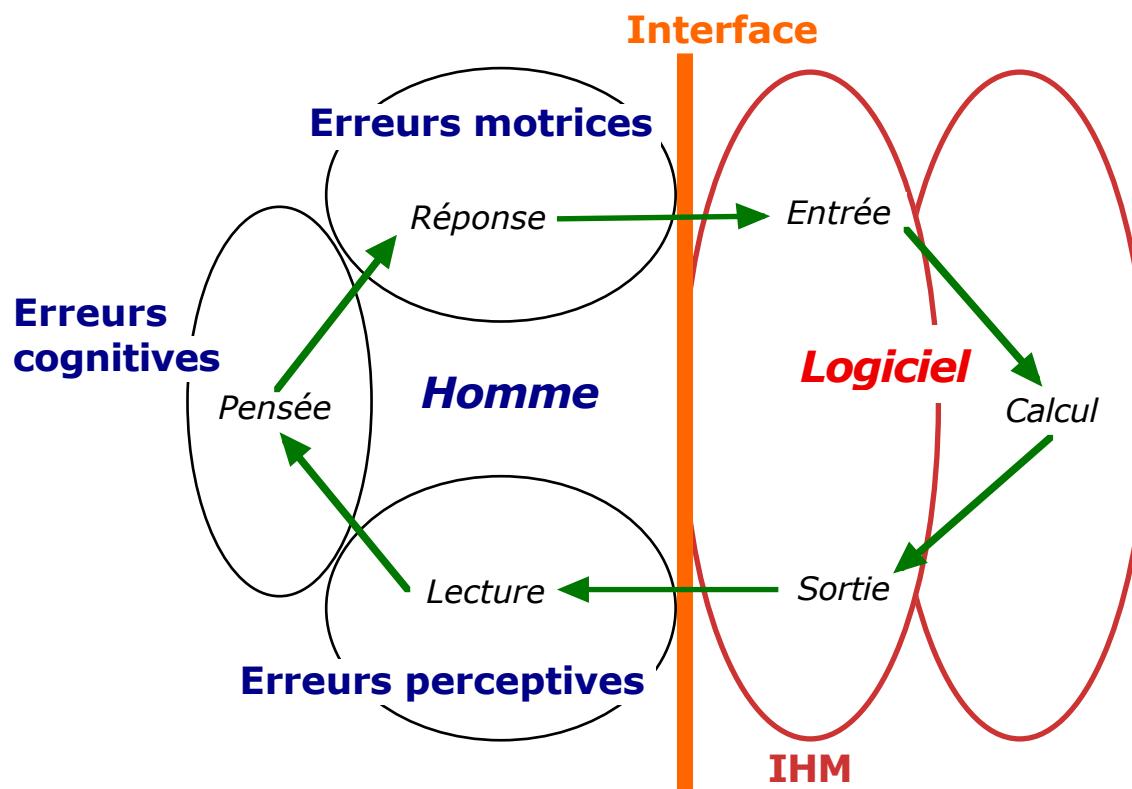


Figure 5.1 — Les différents sous-systèmes du processeur humain expliquent les différents types d'erreurs d'exécution

Ces trois types d'erreurs, **erreurs perceptives, cognitives et motrices** sont détaillées ci-après.

### • Erreurs perceptives

Il peut s'agir de confusion de lecture ; certains caractères présentent en effet un dessin assez proche :

Police	Séries lettre-chiffre sources de confusions					
Verdana	B	8	Z	2	I	1 I
Arial	B	8	Z	2	I	1 I
Century Gothic	B	8	Z	2	I	1 I
Futura condensed	B	8	Z	2	I	1 I
Textile	B	8	Z	2	I	1 I
Courier New	B	8	Z	2	I	1 1
	S	5	S	5	S	5
	O	0	O	0	O	0

Figure 5.2 — Séries de caractères sources de confusions



- ✓ Utiliser les techniques de mise en évidence.
  - ▶ Une mise en évidence inexiste ou inefficace est également une source de confusion ; une information importante risque de ne pas être vue si elle n'est pas correctement mise en évidence.
- ✓ Rendre bien visible les changements de mode et les états du système.
  - ▶ Par exemple, sur un éditeur de texte, le mode "Toute majuscule", activé par l'appui sur la touche *verrouillage majuscule* du clavier est une source fréquente d'erreur car ce changement d'état est peu visible.
- ✓ Si le retour-utilisateur est peu visible ou si le délai du retour est important, des erreurs perceptives seront commises.

- **Erreurs cognitives**

Il s'agit essentiellement d'erreurs liées à des **difficultés de mémorisation**.

- ✓ Minimiser le travail de mémorisation.
  - ▶ L'utilisateur doit-il se souvenir de certaines particularités d'emploi du logiciel (unité, format, syntaxe...) ?
  - ▶ Doit-il mémoriser une valeur d'une fenêtre à une autre ?
- ✓ Éviter de demander un calcul mental.
  - ▶ Un ordinateur ne peut-il pas réaliser des conversions d'unité ?
- ✓ Fournir des **aides mnémoniques**.
  - ▶ Par exemple, choisir des raccourcis clavier en cohérence avec le nom de la commande.
- ✓ Concevoir une présentation cohérente.
  - ▶ L'homogénéité facilite la mémorisation.
- ✓ Mettre en jeu la reconnaissance plus que le souvenir.
  - ▶ Un article de menu sera facilement retrouvé dans un menu. (Et son emplacement sera mémorisé si cet article est souvent utilisé.)

- **Erreurs motrices**

Ce sont les erreurs de **manipulation** ou de **saisie**.

- ✓ **Agrandir** les objets à sélectionner.

- ▶ La sélection d'un petit objet réclame une coordination œil-main difficile.

- ✓ Prévoir des appuis accidentels.

- ▶ Des clics involontaires peuvent survenir ; les erreurs de frappe au clavier sont fréquentes.
  - ▶ Si par exemple une saisie attendue est numérique, filtrer les caractères alphabétiques.

- ✓ Éviter le changement fréquent de mode d'interaction.

- ▶ Les erreurs motrices augmentent avec les difficultés de manipulation. Des alternances nombreuses entre le clavier et la souris sont ainsi sources d'erreurs.



### • Prévention des erreurs

- ✓ Tout risque de perte de données et toute commande difficilement réversible doivent provoquer une **demande de confirmation**.
  - Utiliser à cette fin une **zone de dialogue modale** pourvue d'un message bref et pertinent, et de boutons de commande non ambigus.
  - Cependant, les utilisateurs expérimentés apprécieront de pouvoir désactiver cette protection (ponctuellement ou non).

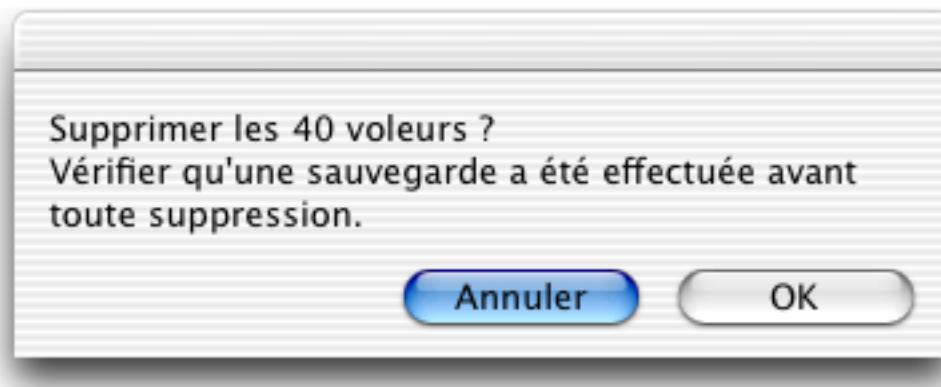


Figure 5.3 — OK pour supprimer ou OK pour vérifier ?

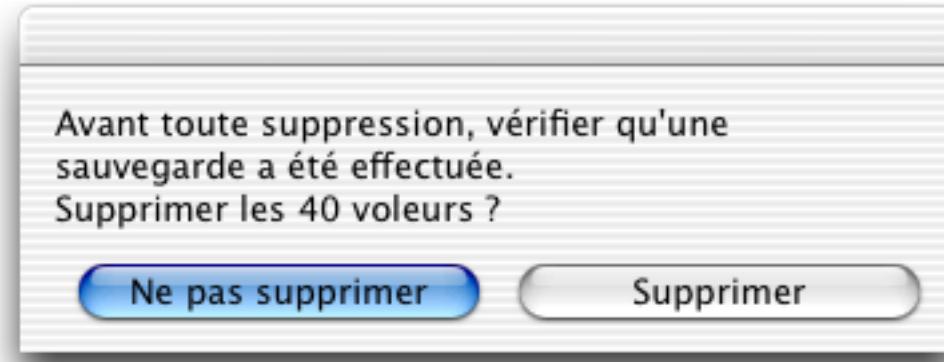


Figure 5.4 — Ce message est plus clair et moins ambigu

- ✓ Désactiver les commandes indisponibles.
  - Un guidage efficace contribue à une prévention des erreurs. Empêcher l'exécution d'actions non valides permet d'éviter des erreurs.



- ✓ Ne rendre actif un élément que si le contexte s'y prête.
  - ▶ Ainsi, un champs (et son label) ne devrait être affiché que lorsqu'il est nécessaire de le remplir.
  
- ✓ Réaliser des tests d'utilisabilité pour identifier les erreurs.
  - ▶ Des erreurs, notamment des erreurs d'intention, sont difficiles à prévoir. Les tests d'utilisabilité vont permettre d'en identifier un bon nombre.

- **Correction des erreurs**

Les erreurs perturbent la planification des tâches de l'utilisateur ; elles devront être signalées et corrigées le plus simplement possible.

✓ **Signaler** l'erreur au plus tôt.

- ▶ La description de l'erreur doit être brève, précise et suffisante. Indiquer le moyen de corriger l'erreur est un plus.

✓ Placer le message d'erreur dans le **champ de vision** de l'utilisateur.

- ▶ Par exemple, à proximité du champ erroné, sinon dans la fenêtre courante.

✓ Fournir la possibilité de corriger facilement les données erronées.

- ▶ **Mettre en évidence** les données à corriger.

✓ Permettre le retour en arrière.

✓ Autoriser l'interruption des traitements longs.

✓ Donner un accès aisément à l'aide en ligne.

## ● Messages d'erreurs

Le logiciel peut répondre différemment à une erreur de l'utilisateur :

- ✓ Le logiciel **bloque** l'utilisateur ; il affiche une zone de dialogue modale qui empêche l'utilisateur de continuer tant que l'erreur ne soit pas corrigée.
  - ▶ Cela ne doit être utilisé qu'exceptionnellement, car cela casse le rythme du dialogue.
- ✓ Le logiciel **met en garde** l'utilisateur, soit par un message, soit par un signal sonore ;
  - ▶ il n'empêche pas l'utilisateur de continuer, ce qui convient lorsque l'erreur n'est pas importante.
- ✓ Le logiciel **ne répond pas** à la commande erronée ;
  - ▶ cela convient aux erreurs sur commande ou action à manipulation directe, lorsque cette manipulation devrait modifier la présentation de l'objet au cas où elle est exécutée correctement.
  - ▶ Cela reste peu explicite, et il peut être préférable d'afficher un message.
- ✓ Le logiciel **corrige** automatiquement l'erreur, lorsque cela est possible.



Le message d'erreur doit répondre aux recommandations suivantes :

- ✓ Le message d'erreur doit préciser la nature du problème et **donner les moyens d'y remédier.**

▸ Un compromis doit être trouvé entre l'exhaustivité et la concision du message.

- ✓ Le message doit être **coopératif.**

▸ Il peut contribuer à mieux assimiler le fonctionnement du système.

▸ ***Erreur : syntaxe incorrecte***

▸ ***Erreur : le nom du dossier de destination manque***

- ✓ Le message doit fournir une description du problème.

▸ Il peut contribuer à comprendre la cause de l'erreur.

▸ ***Erreur d'écriture sur disque***

▸ ***Le disque est plein***



- ✓ Éviter les ambiguïtés :

- ***Document non trouvé***
- ***Le nom du document doit être suivi de son extension***

- ✓ Adapter le message au niveau de connaissance de l'utilisateur.
- ✓ Lorsque la commande est peu utilisée ou si l'utilisateur est débutant, des alternatives peuvent être indiquées :

- ***Erreur dans le nombre de billet***
- ***Le nombre de billets doit être inférieur à 10, sinon utiliser la commande 'Groupe'***

- ✓ Ne pas dramatiser ; ne pas culpabiliser.
- Dramatiser n'apporte aucune aide ! On évitera donc les mots trop négatifs ou trop noirs : *Désastreux, catastrophique, fatal, illégal, mauvais...*
- ***Erreur fatale : mauvaise saisie !!!***
- ***Erreur : cette valeur doit être un nombre entier***



- ✓ Un ordinateur ne cause pas comme un homme. (Éviter l'anthropomorphisme.)
  - Le logiciel est un outil pour aider l'être humain. Ce n'est pas une raison pour humaniser le logiciel.
    - ***Désolé, je ne peux accepter cette valeur***
    - ***Valeur non acceptée***

## 5.2. Aide en ligne

L'aide est considérée par les utilisateurs comme un outil utile et pratique. Mais son utilisation est une tâche en soi, dont le but est d'apprendre à se servir du logiciel. Or, l'objectif de l'utilisateur est plutôt d'**utiliser le logiciel**.

Lorsque les aides sont utilisées, c'est souvent en cas d'urgence ou de difficulté : L'utilisateur ne sait pas comment réaliser une tâche avec le logiciel (et il ne trouve personne pour l'aider). L'utilisateur se demande alors "**Comment faire pour (...) ?**"

L'aide est aussi utilisée lorsqu'il ne comprend pas comment fonctionne le logiciel : soit il souhaite connaître le **rôle d'une commande** ou d'un composant, soit c'est la dynamique du système qui est mal comprise.

Les aides sont donc en fait peu employées. Les utilisateurs préfèrent l'assistance d'une personne connaissant le logiciel, puis, si elle existe, l'assistance téléphonique : l'assistant tient compte du contexte, s'adapte aux actions de l'utilisateur et participe à l'apprentissage en expliquant le logiciel et en donnant des conseils d'utilisation.

Une bonne aide en ligne doit donc satisfaire les trois caractéristiques, **contextuelle**, **dynamique** et **formatrice** pour être plus facile à utiliser.

Les différentes formes d'aide sont :

✓ La barre d'état (*Status bar*)

- Elle permet d'afficher des messages contextuels :
- soit pour indiquer la **progression** d'un traitement en cours,
- soit pour répondre à "**Qu'est-ce que c'est ?**" lorsque l'utilisateur pointe sur un composant ou une commande.
- Le texte est court et facile à comprendre.

✓ Les bulles d'aide (*Tool tip*)

- Elles répondent à "**Qu'est-ce que c'est ?**", en expliquant le rôle d'un composant de l'interface.
- Le texte est court et facile à comprendre.

✓ Le manuel en ligne (*Help viewer*)

- Il répond à "**Qu'est-ce que c'est ?**", de façon plus détaillé
- et à "**Comment faire pour (...) ?**".
- Il s'adresse à des utilisateurs expérimentés, car la recherche dans ce type d'aide est plus complexe. C'est une aide statique : elle ne dépend pas du contexte. Ce manuel d'aide est composé de différents documents HTML ; il est affiché avec un navigateur spécifique (*Help viewer*).

✓ L'assistant (*Wizard*)

- ▶ Il répond à “**Comment faire pour (...) ?**”, en guidant l’utilisateur dans la réalisation d’une tâche.
- ▶ Il peut être très utile, bien qu'il offre peu de visibilité sur le fonctionnement du système et qu'il laisse l'utilisateur passif.

✓ Le didacticiel (*Tutorial*)

- ▶ C'est un outil **d'autoformation** ;
- ▶ il facilite la prise en main,
- ▶ il induit une attitude active et positive de l’utilisateur.
- ▶ Cependant, il est long à utiliser et peut parfois être trop général au niveau du contenu.



Les conseils et recommandations sont :

- ✓ Fournir un **accès simple et rapide** à l'aide.
  - ▶ L'utilisateur veut une assistance rapide pour vite revenir à sa tâche.
  - ▶ Un menu Aide (*Help menu*) peut accueillir les articles pour activer différentes formes d'aide.
  - ▶ Dans une zone de dialogue, une case de commande d'aide permet l'accès à une aide spécifique à cette zone.
  - ▶ Les *Guidelines* indiquent aux développeurs les us et coutumes liés au système d'exploitation ou à la plate-forme visés, notamment les raccourcis clavier préconisés pour accéder à l'aide.



Figure 5.5 — Le menu Aide d'une application en Java

- ✓ Structurer l'aide en fonction de la **tâche utilisateur**.
  - ▶ Les fonctionnalités du logiciel permettent d'accomplir des tâches métier, et c'est bien ces dernières qui intéressent l'utilisateur.
- ✓ L'aide doit participer à l'apprentissage du logiciel.
  - ▶ Une description précise du comportement du logiciel facilitera son apprentissage. L'utilisateur peut alors se construire un modèle correct du fonctionnement du système.
  - ▶ Pour favoriser la mémorisation par une mise en œuvre concrète, le pourquoi de certaines fonctionnalités pourra être expliqué en fonction d'éléments propres au métier.
- ✓ Utiliser le **vocabulaire de l'utilisateur**.
  - ▶ Les ambiguïtés sont évitées si le vocabulaire est celui du métier de l'utilisateur.
- ✓ Permettre l'exploration.
  - ▶ La rubrique **Voir aussi** invite l'utilisateur à explorer l'aide et lui montre les liens entre les différentes commandes.

- ✓ Donner des titres explicites.
  - ▶ Les titres permettent de structurer le contenu et aident l'utilisateur à se repérer dans l'aide en ligne.
- ✓ Utiliser des listes pour décrire les différentes étapes d'une procédure.
  - ▶ Un texte structuré est plus facile à mémoriser ; Une bonne structuration de l'aide facilitera la réalisation d'une procédure étape par étape.
- ✓ Faire une phrase pour chaque étape d'une procédure.
- ✓ Illustrer.
  - ▶ Un schéma vaut mieux qu'un long discours. Par exemple, un diagramme explique mieux qu'un texte la navigation dans un logiciel.
- ✓ Présenter les informations dans l'ordre dans lequel l'utilisateur doit les prendre en compte.
- ✓ Indiquer d'abord **le but**, puis **l'action** pour réaliser ce but.
  - ▶ L'utilisateur est alors guidé par les objectifs et le texte est plus accrocheur.

- ✓ Ne pas utiliser le futur.
  - ▶ L'utilisateur est censé effectuer la commande parallèlement à la lecture de l'aide.
  - ▶ Le **présent** ou l'**impératif** s'impose donc.
- ✓ **Rédiger clairement** ; à cette fin :
  - ▶ Utiliser des phrases courtes et simples.
  - ▶ Faire des paragraphes courts, séparés par une ligne blanche.
  - ▶ Faire des lignes de 50 à 55 caractères.
  - ▶ Éviter les coupures de mot.
  - ▶ Employer la forme active.
- ✓ Éviter les négations.
- ✓ **Help doesn't**
  - ▶ L'aide n'en est pas une. La première aide, c'est d'abord les différents moyens mise en œuvre dans l'interface pour guider l'utilisateur et lui éviter de faire des erreurs.
  - ▶ Le principal rôle d'une aide est d'expliquer le fonctionnement du logiciel.

## 6. Conception de sites web

### 6.1. Pourquoi faire simple ?

Internet est à la disposition d'une grande variété d'utilisateurs. Il offre à tous la possibilité d'accéder aux ressources informatiques réparties et hébergées par différents serveurs du réseau.

- ✓ Le grand public utilise des applications web domestiques à travers le réseau **Internet**.
- ✓ Dans le monde professionnel les utilisateurs utilisent :
  - ▶ Des réseaux **Intranet** (les applications web internes à une entreprise) ;
  - ▶ Des réseaux **Extranet** (les applications web offertes par des fournisseurs à leurs clients, ou vice-versa) ;
  - ▶ Divers outils de communication et de service Internet.

Sur le web, c'est l'utilisateur qui contrôle l'usage du logiciel ; les ordinateurs ne servent que d'interface entre les fournisseurs de contenu et les surfeurs.

L'interface utilisateur généralement offerte par les navigateurs web est relativement limitée et restreinte.

- ✓ L'information se présente sous forme de pages affichées dans une **fenêtre unique**.
- ✓ De fait, le web est souvent considéré par le grand public comme difficile à utiliser.

La fréquentation d'un site est très liée à son utilisabilité.

- ✓ Ainsi, les sites les plus populaires présentent peu de problèmes d'utilisation et ils se chargent rapidement.

Dans le domaine du commerce électronique, où tout devrait être fait pour que l'utilisateur devienne un client — de préférence un client fidèle —, des études montrent que, en 2006, 49 % des acheteurs ont abandonné au moins une fois leur caddie dans la cyber-boutique.

Les formulaires en ligne découragent aussi de nombreux surfeurs.



Un site facile à utiliser peut générer une affluence et donc des résultats conséquents. L'exemple d'IBM est significatif : en 1998, il a revu une partie de ses sites web, en mettant en pratique deux principes ergonomiques simples :

- ▶ Homogénéité de présentation ;
- ▶ Accès rapide aux pages les plus fréquentées.

Suite à la mise en service, le trafic a augmenté de 120 % sur leurs sites de commerce électronique, où les ventes ont augmentées de 400 %.



## 6.2. Définition et développement du site

Sous son apparence simple, la conception de sites web cache de multiples difficultés.

- ✓ cette technologie résulte d'une convergence entre l'informatique, les télécommunications et l'audiovisuel.
- ✓ Le développement de site est au carrefour de la communication, du marketing, de l'infographie, de l'ergonomie et de l'informatique.

Le succès d'Internet est lié à l'adoption de standards simples. Exemple :

- ✓ Web = HTML + HTTP + CGI
- ✓ E-mail = SMTP + POP3

Dans le cas du web, le serveur HTTP constitue un **middleware**, intermédiaire entre les navigateurs web et les applications et ressources distantes (Figure 6.1).

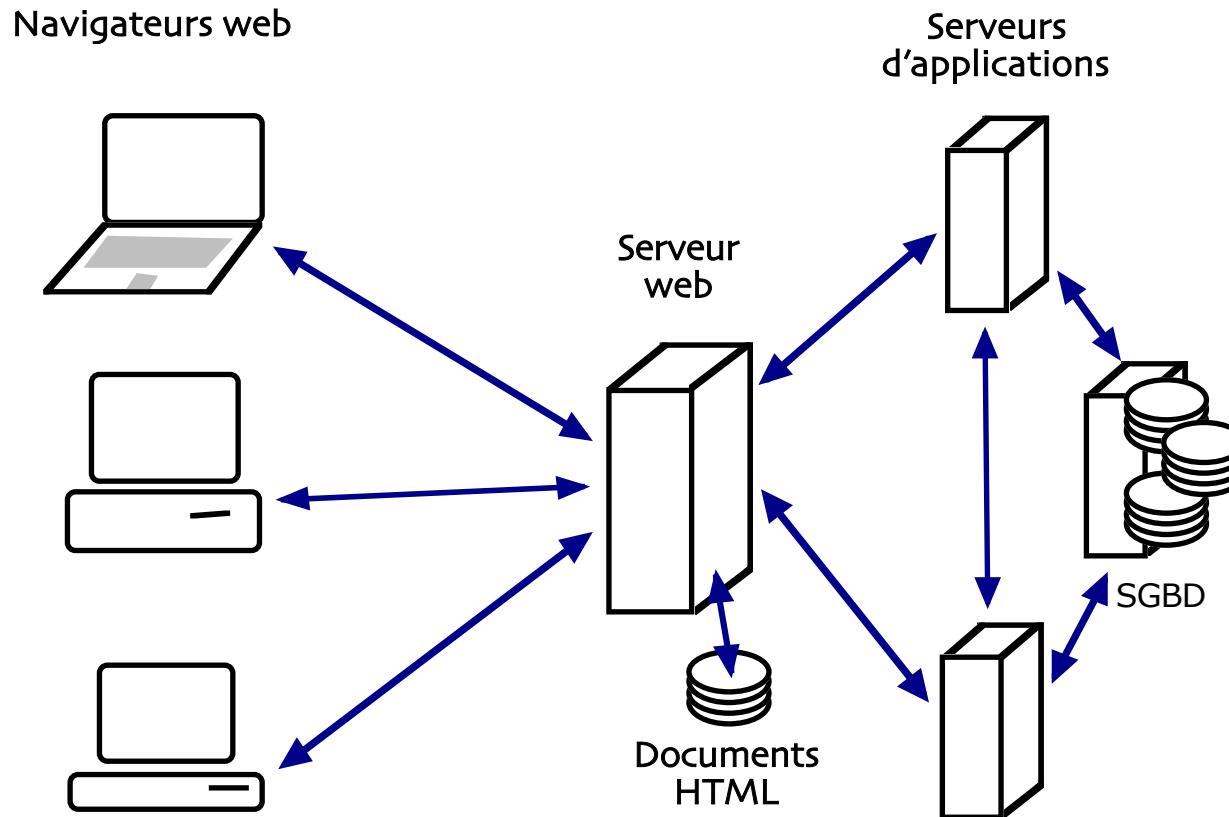


Figure 6.1 — Architecture du web :  
Clients - Serveur web - Serveurs d'applications

Le développement d'un site web implique la conjugaison de compétences et donc un **travail d'équipe**. Comme tout projet, on peut considérer le cycle de développement comme l'itération des étapes suivantes :

#### ✓ **Définition du site**

- **Analyse des besoins** : identifier les objectifs de l'entreprise, les services à mettre en ligne et le public visé.
- **Architecture de communication** : Description détaillée et justifiée du contenu du site et de son organisation.
- **Charte graphique et protocole de navigation** : Le design graphique s'appuie sur la charte existante de l'entreprise et sur le besoin de communication. Un protocole de navigation est proposé.
- **Prototypage** ; un prototype est testé avec les utilisateurs ; on vérifie avec le client que les objectifs de communication sont atteints.  
Des modifications de l'architecture de communication, d'éléments graphiques et de navigation sont apportées si nécessaire.

#### ✓ **Développement du site**

- **Développement** ; mise en page et assemblage du site ; réalisation des interfaces avec les applications et SGBD existants.
- **Mise en ligne et référencement**. Promotion du site.

✓ **Vie du site**

- Maintenance
- Analyse des statistiques
- Mises à jour

**L'équipe de développement** rassemble et fait interagir les acteurs nécessaires aux composantes et compétences suivantes :

- ✓ La composante communication/marketing ;
- ✓ la compétence utilisabilité ;
- ✓ la compétence graphique ;
- ✓ la compétence rédactionnelle ;
- ✓ la compétence développement.

## 6.3.La navigation web

Une principale tâche de l'internaute est la recherche d'information, ce qui le pousse à naviguer d'une page à une autre. Pour faciliter cela, un site doit permettre à l'internaute de **comprendre où il est et comment faire pour aller où il veut**.

La navigation est la source de deux importants problèmes d'utilisabilité :

✓ **La perte du référentiel** : l'internaute se perd ; il ne sait plus où il est.

- Le navigateur permet, certes, de revenir en arrière et garde une trace du chemin parcouru, mais il ne présente pas une carte global du site.

✓ **L'attente** : le temps de chargement des pages rallonge la tâche.

- Ce temps résulte du temps de connexion aux serveurs contenant les ressources à afficher et au temps de chargement de ces ressources.

La navigation, élément clé de l'utilisabilité du site, demande donc un soin particulier lors de la conception et le design du site ; des tests spécifiques vérifieront que la navigation reste facile. Il faut donc veiller aux points suivants :

- ✓ Les liens doivent répondre aux **objectifs de l'internaute**.
  - ▶ Les liens doivent être conçus en fonction de l'information qu'ils vont permettre de trouver ;
  - ▶ ils doivent répondre aux besoins et questions du visiteur.
- ✓ Le texte des liens doit être **facile à lire**, à comprendre et à apprendre.
  - ▶ Le libellé d'un lien ne doit pas être source d'ambiguïté ;
  - ▶ l'utilisateur doit pouvoir le mémoriser facilement lors de la lecture de la page.
- ✓ Les principales informations doivent être accessibles le plus rapidement possible.
  - ▶ Trois clics maximum doivent suffire pour atteindre des pages importantes, à condition que les pages se chargent rapidement.
  - ▶ Le chemin reste alors simple, les liens sont faciles à mémoriser et un retour en arrière aisément.



✓ **Guider** l'internaute dans sa navigation.

- ▶ Pour ne pas se perdre, l'internaute doit savoir où il est, par où il est passé et il doit comprendre où il peut aller.
- ▶ Une cohérence forte doit exister entre le **libellé d'un lien**, le **titre de la page** affichée après activation de ce lien (le titre de la fenêtre et le titre rappelé dans le contenu de la page).
- ▶ La présentation du lien "visité" change ; cela permet à l'internaute de mieux se repérer dans le site.
- ▶ Lorsque le lien fait parti d'un bandeau de liens également présent sur la nouvelle page, sa présentation change (ce n'est plus un lien et l'élément est mis en évidence).

✓ Fournir une vue globale du site.

- ▶ Une carte du site permet au visiteur de s'orienter aisément. Une vue d'ensemble peut être donné par le sommaire de la page d'accueil, ou bien une page dédiée peut afficher un plan du site décrivant précisément l'arborescence.

✓ Ne pas changer la couleur et le comportement par défaut des liens.

- ▶ L'internaute navigue d'un site à un autre. Il apprend et reconnaît les comportements standards, notamment ceux des liens.
- ▶ La cohérence de l'aspect des liens facilite le premier contact alors qu'un comportement non standard peut déconcerter le visiteur.

## 6.4. Architecture du site

La structure du site a été déterminée lors de la définition de site, en fonction des objectifs de communication. Le site se présente alors en différentes rubriques.

### • Généralités

Cette structure devrait satisfaire aux points suivants :

✓ **Regrouper** les informations.

- Une rubrique est composée d'un ensemble cohérent d'informations, présentées sur plusieurs pages.

✓ Présenter les informations dans l'ordre où elles seront utilisées.

✓ Préférer une arborescence régulière.

- Une topologie régulière et simple est plus facile à mémoriser et on réduit alors les risques de s'égarer (dans une branche isolée ou suite à un saut d'une branche à une autre).

- ✓ Organiser le site en largeur plutôt qu'en profondeur.
  - ▶ L'internaute visite alors le site à partir de la page d'accueil et ne risque pas de se perdre si le site se limite à trois niveaux de profondeur.
- ✓ **Hiérarchiser** les informations.
  - ▶ Une présentation structurée des informations minimise la charge cognitive.
  - ▶ Par exemple, les contenus très rédactionnels peuvent être structurés par une succession des éléments suivants :
    - Titre de l'article ;
    - Résumé de l'article ;
    - Un lien pour atteindre l'article détaillé.

**News****■ Voyages en ligne : la dernière minute fera l'affaire**

Les vacanciers tardent à réserver leurs séjours. Pourtant, les voyagistes en ligne ne s'en émeuvent pas. Ils attendent même de meilleurs résultats qu'en 2002.

<http://lettres.01net.com/u.asp?a=13659:784:4>

**■ HP privilégie les services pour PC aux PC**

Les HP Compaq d330 et d530 sont les premiers modèles professionnels depuis la fusion des deux constructeurs. Ils présentent peu d'innovations, mais une meilleure stabilité.

<http://lettres.01net.com/u.asp?a=13659:784:5>

**■ Le Silicon Sentier de nouveau sur la route**

Soutenu par la Ville de Paris, l'ex-lobby de la Net-économie fusionne avec l'association Nouvelles Entreprises & Territoire (NET). Objectif : transformer le Sentier et les arrondissements limitrophes en un vaste « quartier numérique ». Wi-Fi à l'appui.

<http://lettres.01net.com/u.asp?a=13659:784:6>

**■ Le CDMA n'obtient pas son visa d'entrée en France**

L'Autorité de régulation des télécoms et Bercy bloquent définitivement l'arrivée d'une norme rivale de l'UMTS.

<http://lettres.01net.com/u.asp?a=13659:784:7>

**Figure 6.2 —** Hiérarchisation d'informations : chaque article est composé de son titre, d'un résumé et du lien pour lire la suite.

- **La page d'accueil**

- ✓ **Alléger** la page d'accueil.

- ▶ Elle doit apparaître le plus rapidement possible, donc être légère. La moyenne actuellement constatée de 90 Ko est très importante pour les abonnés à des liaisons bas débit.
    - ▶ 40 Ko, c'est mieux ; qui peut battre Google et ses 12 Ko ?

- ✓ Informer dès la page d'accueil.

- ▶ Lorsque cela est possible, on fournit dès la première page les informations concrètes répondant aux questions que se posent les visiteurs.

- ✓ Donner à la page d'accueil une bonne image globale du site.
  - ▶ En la découvrant, un visiteur cherche à savoir si le site répond à ses attentes.
  - ▶ Cette première page doit préciser le contenu de façon claire et synthétique. Répond-elle à cette check-list ?
    - Que contient le site ?
    - À qui s'adresse-t-il ?
    - Comment fonctionne-t-il ?
    - Quel est son objectif ?
    - Qui en est le responsable ?
    - De quand date la dernière mise à jour ?
- ✓ Un **hall d'entrée** est préférable à une porte d'entrée.
  - ▶ Une première page cohérente avec les autres pages du site et qui propose une exploration immédiate du site est plus efficace qu'une porte d'entrée, intermédiaire retardant l'accès véritable au site.

- **La barre de navigation**

- ✓ Définir une **barre de navigation** pour toutes les pages du site.
  - ▶ La barre de navigation, un bandeau de liens ou une zone de navigation permet d'accéder aux différentes rubriques du site. On permet alors une navigation transversale du site et un retour rapide à la page d'accueil.
  - ▶ La position de cette barre ou de cette zone doit être cohérente sur l'ensemble du site.
- ✓ Fournir une **vue globale du site** avec la barre de navigation.
  - ▶ Lorsque le site comprend de nombreuses pages, on peut utiliser une deuxième barre de navigation spécifique à la rubrique courante.
- ✓ Placer la barre de navigation **en haut à gauche** de la page.
  - ▶ Elle est alors lu en premier car en tête du parcours visuel, et toujours visible.
- ✓ Répéter la barre navigation horizontale (ou une zone de navigation) en bas de la page si la hauteur de la page dépasse la hauteur de l'écran.



- ✓ La barre de navigation constitue un guidage.
  - ▶ La page courante est mise en évidence dans la barre de navigation et le titre de la page est cohérent avec le libellé mis en évidence.
- ✓ Organiser les rubriques de la barre de navigation selon leur utilisation.
  - ▶ L'ordre des rubriques résulte soit d'un ordre des tâches, soit de la fréquence de visite des rubriques, soit de leur importance.
  - ▶ Vérifier cet agencement des rubriques avec des tests d'utilisabilité.
- ✓ Utiliser un fil d'ariane (*Breadcrumbs*). C'est une trace du chemin parcouru, ou des différents niveaux traversés.

[High-tech](#) > [Informatique](#) > [Disques durs & Clé USB & Stockage optique](#) > [\*\*Disques durs externes\*\*](#)

Figure 6.3 — Exemple de fil d'ariane

### • Les liens

Le lien, composant de base de la navigation web, est lu systématiquement. C'est un élément de guidage important.

Lorsque le lien apparaît dans le texte, sa sélection implique une interruption de la lecture en cours et une modification de l'espace de référence lorsque la nouvelle page est affichée.

- ✓ Placer les liens sur les **mots clés** qui décrivent le contenu de la page liée.
- ✓ Dans une liste, ne mettre en évidence que les mots différents
- ✓ **Ne pas répéter** des liens similaires sur une même page.
  - En les répétant, l'internaute risquerait de croire qu'ils conduisent à des pages différentes.
  - Indiquer le lien sur la première occurrence.
  - On ne doit répéter le lien qu'en cas d'éloignement important d'une occurrence.

- ✓ Ne pas mettre de lien dans un titre.
- ✓ Utiliser un texte explicite pour les liens.
  - ▶ L'utilisateur doit comprendre, avant de cliquer, où il va aller. S'assurer également de la cohérence entre le **libellé du lien** et le **titre de la nouvelle page**.
- ✓ Différencier clairement les liens particuliers, comme ceux conduisant vers des documents volumineux.
  - ▶ S'il s'agit d'un lien pour télécharger un document, le libellé du lien doit être explicite ;
  - ▶ L'indication du type et de la taille du fichier sera souvent appréciée.
- ✓ Différencier clairement les **liens externes**.
- ✓ Plus un lien est grand, plus il sera facile à sélectionner.
  - ▶ Le pointage du lien est plus facile si son libellé est grand et s'il est placé en bordure de l'écran.
- ✓ Placer le lien **à la fin d'une phrase** ou d'un paragraphe.
  - ▶ Cela permet alors la lecture complète de la phrase ou du paragraphe.

- **Icônes ; boutons de navigation**

- ✓ Préférer du texte à des icônes, sauf si les utilisateurs sont expérimentés et que la place manque.
  - ▶ Une combinaison d'un graphisme et du texte évite les ambiguïtés.
- ✓ Utiliser l'attribut *a/t* des balises *img* des icônes.
  - ▶ Une bulle d'aide (*Tool tip*) sera affichée lorsque le pointeur reste un moment sur l'image.
- ✓ Ne pas supposer que les utilisateurs connaissent bien le fonctionnement du navigateur.

## 6.5.La page web

- **Agencement**

- ✓ Adopter un **tracé régulateur** homogène pour toutes les pages.
  - ▶ Cette cohérence renforcera le *look* du site et facilitera l'apprentissage du site.
  - ▶ Avec les éditeurs de sites — par exemple *Adobe Dreamweaver* — on définit un modèle de page qui servira pour la création de toutes les pages du site.
- ✓ Alléger.
  - ▶ Les sites les plus visités sont ceux pourvus de pages légères. On attend pas plus de 8 secondes avant de zapper sur un autre site.
- ✓ Concevoir les pages selon des résolutions d'écran types.  
Les résolutions d'écran de postes de travail sont :
  - ▶ 800 x 600 pour 10 %
  - ▶ 1024 x 768 pour 55 %
  - ▶ 1280 x 800 ou 1280 x 1024 pour 25 %
  - ▶ autre ou non précisé pour 10 %

- ✓ Définir une **zone protégée de 800 x 600** dans le coin supérieur gauche.
  - Cette zone est donc visible pour une grande majorité d'utilisateurs, et elle recouvre bien le parcours visuel.
  - Elle doit comprendre le titre de la page, une zone de navigation et les messages principaux à communiquer.
- ✓ Prévoir une version spécifique aux mobiles (**Services Internet mobiles**).
- ✓ Éviter les barres de défilement.
  - Leur utilisation implique l'interruption du processus de lecture et leur manipulation nécessite en général une bonne dextérité.

- ✓ **Bannir les cadres.** Les cadres (*frames*) posent de très nombreux problèmes d'utilisabilité :
  - ▶ La navigation devient parfois incohérente.
  - ▶ Le titre de la page n'évolue avec le contenu d'un cadre.
  - ▶ On n'imprime pas toujours ce que l'on veut.
  - ▶ La mémorisation d'un signet (ou favoris) n'est pas cohérente avec le contenu des cadres (c'est l'URL de la page parente qui est conservée).
- ✓ Ne concevoir des pages avec des cadres que pour des documentations en lignes, où un index, dans un cadre à droite, permet un accès rapide à un contenu affiché dans le grand cadre.
- ✓ Éviter les pages longues.
  - ▶ Une page idéale **ne dépasse pas 50 lignes** de texte. Au delà de deux écrans, un compromis doit être trouvé : doit-on découper une page longue en plusieurs ?

- ✓ Éviter les pages vides.
  - Une série de pages trop courtes impose une navigation fréquente.
  - **Les pages vides sont décevantes.** On tentera de réorganiser et de regrouper les contenus de faible densité.
- ✓ Permettre un retour rapide en haut de page.
  - Les déplacements verticaux au sein d'une page sont réalisés, pendant la lecture, avec la barre de défilement verticale, la molette de la souris, ou encore, sur le clavier, la barre d'espace ou la touche "Page suivante" (*Pg dw*).
  - La lecture faite, un lien ou un **bouton "Haut de page"** (*Top*) est apprécié.
- ✓ Fournir une **zone de navigation locale**.
  - Une liste horizontale de liens vers les rubriques ou sections de la page peut constituer un sommaire local et permet un accès rapide pour les habitués.
  - Cette zone sera également appréciée en bas des longues pages.

- **Graphisme**

- ✓ Définir une **charte graphique** pour l'ensemble du site.
  - ▶ Elle contribue au *look* du site et elle consolide sa cohérence.
- ✓ Elle consiste à construire un ou un faible nombre de **modèles** de pages :
  - ▶ Chaque modèle est composé de différentes zones.
  - ▶ Chaque zone a un rôle et un graphisme déterminés :
    - zone de navigation principale et secondaire
    - zone réservée au texte, etc.
  - ▶ Veiller à la cohérence des couleurs et du style des titres et textes pour chaque modèle et pour l'ensemble du site.
  - ▶ On minimisera le nombre de polices et de couleurs.



✓ Utiliser une **même feuille de style** pour tout le site.

- ▶ La définition des feuilles de styles (*CSS = Cascading Style Sheets*) fait partie de la conception de la charte graphique.
- ▶ Un document qui défini une feuille de style sera attaché aux documents HTML.  
Exemple :

```
<head>
  <!-- ... -->
  <link href="/mesStyles.css" rel="stylesheet" type="text/css">
  <!-- ... -->
</head>
```

✓ S'assurer du fonctionnement du site **sans feuille de style**.

- ▶ Certains utilisateurs (dont les malvoyants) les désactivent.
- ▶ Les navigateurs antérieurs à IE 4 ne gèrent pas les feuilles de styles.



- ✓ Utiliser des images dégradées ou des fonds clairs ou blancs pour garantir la **lisibilité du texte**.
  - Le texte sera plus lisible si sa couleur est noire sur un fond très clair ou blanc.
  - Un fond sombre sous un texte blanc ou très clair réduit la lisibilité des pages.
  - Un fond de couleur saturée est fatiguant ou même agressif.
- ✓ N'utiliser les images qu'à bon escient.
  - Le temps de chargement de la page dépend du poids des images, qui doivent **appuyer** le message.
- ✓ **Minimiser la taille des images**.
- ✓ Privilégier la **qualité de l'image** plutôt que sa taille.
  - Le poids de l'image est proportionnel à la surface de l'image.



✓ **Accélérer** le chargement des images.

- ▶ La taille des images sera indiquée dans le code HTML.
- ▶ Si cela est possible, certaines images seront pré-chargées.
- ▶ Les grandes images placées en bas de page pourront se charger pendant la lecture du haut de page.
- ▶ La réutilisation d'image est efficace car les images restent dans la mémoire cache du navigateur.

✓ Privilégier l'affichage progressif des grandes images

✓ Choisir le format d'image approprié.

- ▶ Les formats PNG, GIF entrelacé et JPEG progressif seront préférés.
- ▶ Le format GIF, limité à 256 couleurs, est adapté à des illustrations et des dessins contenant des plages de même couleur. Les couleurs intermédiaires sont simulées par diffusion (*dithering*). GIF accepte la transparence.
- ▶ Le format PNG est meilleur que GIF pour un usage similaire.
- ▶ Le format JPEG est adapté aux images photographiques. La qualité dépend de la compression appliquée.
- ▶ À voir aussi : MNG ou APNG pour les images animées et SVG pour les graphiques vectoriels.

- ✓ Utiliser les **animations avec parcimonie** ; prévoir la possibilité d'arrêter l'animation ; ne pas placer d'animation à coté de texte.
  - ▶ Les animations gênent la lecture des éléments proches et sont facilement agaçantes.
- ✓ Utiliser les images cliquables à bon escient.
  - ▶ Les images cliquables, ou **ImageMaps**, sont des images où des zones sensibles sont des liens.
  - ▶ L'image est donc souvent de grande taille et les zones cliquables ne sont visibles que lorsque l'on les pointe.
  - ▶ Il est donc nécessaire d'indiquer clairement les zones sensibles en figurant explicitement des boutons.
  - ▶ Des liens textuels restent indispensables pour les utilisateurs désactivant l'affichage d'images.
- ✓ Ne pas utiliser plus de 2 à 3 polices de caractères.
  - ▶ Au delà, la signalétique du site sera plus difficile à comprendre.

- ✓ Autoriser l'utilisateur à changer la taille des polices de caractères.
  - ▶ Choisir de préférence des tailles relatives.
- ✓ Vérifier sur divers navigateurs et plates-formes que les choix de polices et de tailles restent adaptés.
- ✓ Utiliser des listes à puces pour structurer le discours et accélérer la lecture.

### ● Contenu de la page

En général, les internautes **ne lisent pas** la page web ; ils la parcourent des yeux. Lors de ce balayage en diagonale, les titres et les principaux liens sont lus ; les textes moins importants sont sautés.

- ✓ Faire ressortir les éléments **clés** du texte.



- ✓ Utiliser la typographie pour créer les accroches :
  - ▶ On joue sur la **graisse** pour mettre en valeur un mot dans une phrase.
  - ▶ Une liste à puces fait ressortir une série de points clés.
  - ▶ La taille et la graisse font la titraille.
- ✓ Ne pas en faire trop.
  - ▶ Trop de mises en valeur vont rendre la page confuse.
- ✓ Placer l'information importante en **haut de la page**.
- ✓ Le choix du **texte des liens** est très important.
  - ▶ Les liens sont très souvent lus car ils servent de repère dans la navigation.
  - ▶ Un lien doit décrire explicitement la page désignée.
- ✓ Permettre à l'internaute d'**imprimer la page**.
  - ▶ Une version imprimable de l'article ou de la rubrique peut être prévue lorsque la présentation "écran" est mal adaptée à l'impression.

- ✓ Rédiger simplement.
  - ▶ Un texte rédigé clairement, avec des phrases simples, affirmatives, à la forme active et sans subordonnées est plus facile à comprendre. Développer une idée par paragraphe, en 10 lignes maximum. Une page doit comprendre 3 à 4 paragraphes maximum.
- ✓ Rester concis ; aller à l'essentiel.
- ✓ Être objectif ; énoncer des faits et du concret.
- ✓ Commencer par la conclusion.
  - ▶ Le **style journalistique** est approprié à la rédaction de pages web : l'internaute accède en premier à l'essentiel de l'information ; les détails suivent et seront lus si l'internaute le souhaite.
- ✓ **Éviter** la mention "En construction".
  - ▶ Préférer indiquer ce que la page contiendra, et la date prévu de mise à jour.

- ✓ Le contenu informatif de chaque page doit être suffisant.
- ✓ Le contenu doit répondre au check-list suivant :
  - ▶ À qui s'adresse la page ?
  - ▶ Qui l'a écrite ?
  - ▶ Que contient la page ?
  - ▶ Quand ? Date de mise à jour.
  - ▶ Comment fonctionne la page ? Cas d'une navigation particulière ou pour un formulaire, par exemple.
  - ▶ Pourquoi ? Objectif de la page.

## 7. Pratique de l'utilisabilité

### 7.1. Avantages ; place dans le cycle de développement

L'**utilisabilité** peut être définie comme l'ensemble des fonctions que l'utilisateur connaît et utilise correctement.

La **fonctionnalité** d'un système est l'ensemble des fonctions qu'il peut réaliser.

- ✓ La pratique de l'utilisabilité (*Usability Engineering*) consiste alors à **faire coïncider** la fonctionnalité d'un système et son utilisabilité.

L'utilisabilité est maintenant un critère de choix pour les clients :

- ▶ Elle détermine la productivité,
- ▶ ainsi que le temps et le budget de formation à l'usage du système informatique.

Dans le cadre d'un projet de développement, la pratique de l'utilisabilité permet la réduction des coûts :

- ▶ L'implication de l'utilisateur final dès la phase de conception permet une consolidation rapide de la compréhension des besoins du client.
- ▶ Cela évite des retours en arrière dus à des interprétations erronées du cahier des charges.

L'interface occupe en moyenne 50 % du code développé, voire 80 % pour les applications internet. D'où l'importance de faire participer l'utilisateur pendant la conception et à lui faire évaluer l'interface sur des prototypes.

Pour une ligne de produit logiciel, la pratique de l'utilisabilité s'inscrit dans un cycle d'améliorations de systèmes et de composants réutilisables.

Un **guide de style** est un document qui définit les principes ergonomiques adoptés pour le logiciel et la charte graphique de son interface. Ce guide, qui sert de base à la construction des composants de l'interface, est mis à jour en fonction des retours d'utilisation.

- **Place dans le cycle de développement**

La pratique de l'utilisabilité trouve sa place à différents niveaux du **cycle de développement** :

- ✓ En amont, **l'analyse de la tâche** consiste à identifier la tâche prévue (ce qui doit être fait) et à observer l'activité effectivement réalisée.
- ✓ On doit donc déterminer :
  - ▶ Les buts de l'utilisateur ;
  - ▶ Quelles informations lui sont nécessaires pour cela ;
  - ▶ Le mode opératoire ; ce que fait l'utilisateur pour cela.
  - ▶ Comment sont traités les cas exceptionnels et les urgences.
- ✓ L'analyse de la tâche permet de structurer l'interface homme-machine et de débuter la conception des fenêtres de l'application.

**L'analyse de la situation de travail** consiste étudier le contexte lié à l'utilisation du logiciel.

✓ On s'intéresse :

- ▶ aux connaissances informatiques des différents utilisateurs
- ▶ à leur expérience du domaine applicatif.

✓ Le but est de déterminer le **type de guidage** et les **modes de dialogue** à mettre en place au sein du logiciel.

- ▶ Ainsi, un logiciel utilisé de façon occasionnelle doit favoriser la facilité d'apprentissage (via des menus déroulants).
- ▶ Au contraire, pour les utilisateurs expérimentés et formés, on doit privilégier la rapidité d'exécution (grâce à des cases de commande, des raccourcis clavier et des touches de fonction).

Lors de la phase de spécification, la **rédaction du manuel utilisateur** et la construction de **prototypes** servent à consolider le design de l'interface.

- ✓ Écrire le manuel utilisateur dès la phase de spécification nous incite à **adopter le point de vue de l'utilisateur**.
- ✓ Ce manuel donne une base solide aux discussions entre le client et les concepteurs.
- ✓ Il sera revu à la fin du développement pour devenir le *vrai* **manuel utilisateur** du logiciel.

Les évaluations et tests présentés ci-dessous interviennent dès le développement de prototypes ; ils sont également menés suite à la réalisation du produit final.

Enfin, des **enquêtes** auprès des utilisateurs vont permettre de recueillir des retours concrets pour améliorer les prochaines versions.

## 7.2. L'évaluation ergonomique

Les principales méthodes d'évaluation ergonomiques sont l'évaluation par inspection, l'évaluation coopérative et le test d'utilisabilité.

- **Évaluation par inspection**

Cela consiste à passer en revue chacun des composants de l'interface pour vérifier qu'ils respectent une **grille de critères ergonomiques**.

- ✓ Une grille de critères ergonomiques dépend des spécificités de l'application et des utilisateurs.
- ✓ L'ensemble des critères définis au chapitre 1 doivent être pris en compte, et être adapté au produit et au métier.

Des grilles de critères ont ainsi été développés dans le domaine de la navigation aérienne.

Cette technique est très fine et détaillée :

- ▶ une analyse de chaque composant du logiciel est réalisée.
- ▶ Les tests d'utilisabilité restent cependant indispensables.

L'évaluation par inspection s'appuie également sur l'expertise d'un évaluateur, fort de son expérience et de sa connaissance de produits équivalents.

### • **Évaluation coopérative**

Cela consiste à observer l'utilisateur pendant qu'il utilise le logiciel et à identifier les processus cognitifs qu'il met en œuvre, en l'invitant à penser à voix haute.

L'évaluateur pose ainsi des questions, afin que l'utilisateur verbalise le but poursuivi et son mode opératoire :

- ✓ Que voulez-vous faire ?
- ✓ Comment faites-vous cela ?
- ✓ Que fait le système ?
- ✓ Que signifie ce message ?

L'évaluateur ne doit pas aider l'utilisateur ; en cas d'erreur, il observe les stratégies de récupération menées par ce dernier.

Ce type de tests peut être réalisé très tôt dans le développement, à l'aide de prototypes.



- **Test d'utilisabilité**

Considérons la définition suivante de l'utilisabilité [norme ISO 9241-11] :

✓ *Un système est utilisable lorsqu'il permet à l'utilisateur de réaliser sa tâche avec **efficacité**, **efficience** et **satisfaction** dans le contexte d'utilisation spécifié.*

Le test d'utilisabilité consiste à valider ces trois critères :

- ✓ **Efficacité** : On vérifie que les objectifs visés par l'utilisateur sont atteints.
- ✓ **Efficience** (rendement) : On mesure les ressources nécessaires pour atteindre ces objectifs ; le temps d'exécution de la tâche par exemple.
- ✓ **Satisfaction** : On détermine si le système est agréable à utiliser.



On notera également :

- ✓ les incompréhensions ;
- ✓ les erreurs commises
- ✓ la facilité avec laquelle l'utilisateur corrige ces erreurs.

La facilité d'apprentissage du logiciel peut aussi être évaluée.

Tout événement qui montre une difficulté d'utilisation fait l'objet, juste après le test, d'une analyse avec l'utilisateur. Cela permet de mieux comprendre la cause du problème et de trouver parfois une solution originale.

Pour de tels tests d'utilisabilité, on doit se placer dans un contexte réel d'utilisation.

- ✓ Un protocole doit être établi, et on doit préciser pour chaque test :
  - ▶ Les conditions matérielles ;
  - ▶ Le profil des utilisateurs ;
  - ▶ Les différents scénarios : les consignes données aux utilisateurs, les objectifs à valider et les mesures effectuées.



Les points suivants sont à considérer :

- ✓ Le test permet d'évaluer le logiciel, pas l'utilisateur.
  - ▶ Ce point doit être rappelé à l'utilisateur pour **le mettre en confiance** ;
  - ▶ L'objectif du test est bien d'identifier les problèmes d'utilisabilité, non de mesurer la capacité de l'utilisateur.
- ✓ Définir un objectif précis par séance de test.
  - ▶ Les résultats du test sont aisément exploitable si le scénario est bien ciblé et s'il répond à un objectif précis.
- ✓ Choisir un panel d'utilisateur représentatif.
  - ▶ Choisir ceux qui utiliseront effectivement le système ; choisir des utilisateurs novices, expérimentés et ceux qui connaissent bien le métier.
- ✓ Cinq utilisateurs suffisent.
  - ▶ Ils révèlent 80 % au moins des problèmes d'utilisabilité.



## 8. Annexe

### 8.1. Liens web sur l'ergonomie et l'interaction homme-machine

**Cette liste indicative est mise à jour en ligne :** <http://ihm.seancetenante.com/Signets-IHM.php>

#### Ergonomie - Accessibilité

<a href="http://www.ergolab.net/index.html">http://www.ergolab.net/index.html</a>	Ergonomie web et logiciel, architecture de l'information, utilisabilité
<a href="http://fr.wikipedia.org/wiki/Utilisabilit%C3%A9">http://fr.wikipedia.org/wiki/Utilisabilité</a>	Utilisabilité - Wikipédia
<a href="http://ergolab.net/articles/lexique-ergonomie-interfaces.html">http://ergolab.net/articles/lexique-ergonomie-interfaces.html</a>	Petit lexique de l'ergonomie des interfaces - Ergolab
<a href="http://www.la-grange.net/accessibilite/">http://www.la-grange.net/accessibilite/</a>	Plongez dans l'accessibilité
<a href="http://www.useit.com/alertbox/">http://www.useit.com/alertbox/</a>	Alertbox: Jakob Nielsen's Newsletter on Web Usability
<a href="http://www.useit.com/alertbox/9605.html">http://www.useit.com/alertbox/9605.html</a>	Top Ten Mistakes in Web Design (Jakob Nielsen's Alertbox)
<a href="http://www.recorvo.com/aix-provence-usabilite-sites-web.html">http://www.recorvo.com/aix-provence-usabilite-sites-web.html</a>	Conseil en usabilité des sites web
<a href="http://www.lergonome.org/pages/accueil.php">http://www.lergonome.org/pages/accueil.php</a>	Lergonome - Accueil
<a href="http://www.accessiweb.org/fr/guide_accessiweb/index.html">http://www.accessiweb.org/fr/guide_accessiweb/index.html</a>	Guide AccessiWeb - Accueil
<a href="http://www.w3.org/WAI/">http://www.w3.org/WAI/</a>	Web Accessibility Initiative (WAI) - home page
<a href="http://www.w3.org/WAI/References/QuickTips/Overview.php">http://www.w3.org/WAI/References/QuickTips/Overview.php</a>	WAI, Quick Tips to Make Accessible Web Sites
<a href="http://www.accessify.com/">http://www.accessify.com/</a>	Accessify - Web accessibility tools and resources
<a href="http://espace-accessibilite.com/sommaire.php3?id_rubrique=1">http://espace-accessibilite.com/sommaire.php3?id_rubrique=1</a>	[Espace Accessibilité] Accessibilité du Web -



## Standardisation

<http://www.microsoft.com/france/Accessibilite/>

L'accessibilité chez Microsoft, Accueil

<http://www.msdn.microsoft.com/accessibility>

Accessibility

<http://www.ergolab.net/articles/publicite-web-ergonomie.html> Publicité sur le web : que fait l'ergonomie ? - Ergolab

<http://listes.cru.fr/sympa/info/ergoihm>

ergoihm - Ergonomie des Interfaces Homme-Machine

<http://www.upassoc.org/>

UPA - The Usability Professionals' Association

<http://www.sqlagency.com/blogs/ergonomicgarden/index.php>

Ergonomic Garden - sql agency

## Ergonomie Web -

<http://www.ergologique.com/conseils/>

Conseils pratiques en ergonomie et conception centrée utilisateur

<http://www.ccdmd.qc.ca/ressources/?id=1223&onglet=description> Notions de design de l'interface utilisateur

[http://www2.netvibes.com/ergoihm#Ergo\\_IHM](http://www2.netvibes.com/ergoihm#Ergo_IHM)

Univers ErgoIHM

<http://www.ergophile.com/>

Ergophile - Jacinthe Busson

<http://www.utilisabilite.info/dotclear/index.php>

Utilisabilité.info - Les Nouvelles de l'Utilisabilité



## Web design

- <http://typetester.maratz.com/> Typetester – Compare fonts for the screen
- <http://www.fredcavazza.net/mes-ressources/mes-tutoriels/> FredCavazza.net » Mes tutoriels
- [http://www.fredcavazza.net/doc/tutoriels/formulaire/SVF\\_intro.htm](http://www.fredcavazza.net/doc/tutoriels/formulaire/SVF_intro.htm) Des formulaires plus simples (Introduction)
- <http://www.temesis.com/> Temesis - Qualité, accessibilité et conformité des sites Internet
- <http://www.opquast.com/> Opquast - Bonnes pratiques qualité pour les services en ligne
- <http://www.opquast.com/mon-opquast/> Mon-Opquast
- <http://css.alsacreations.com/> Alsacréations Cours et tutoriels HTML, CSS et standards Web
- <http://openweb.eu.org/css/> Openweb.eu.org - Les feuilles de style en cascade
- <http://meyerweb.com/eric/css/edge/index.html> css/edge
- <http://www.veblog.com/fr/2001/0506-flash-utilisabilite-0.html> veblog: flash et utilisabilité, le livre blanc
- <http://www.ergolab.net/articles/ergonomie-flash.html> Ergonomie et Flash - Ergolab
- <http://edu.ca.edu/typo/> Typographie sur le web
- <feed://www.sqlagency.com/blogs/ergonomicgarden/atom.php> Ergonomic Garden
- <http://www.journaldunet.com/solutions/moteur-referencement/conseil/14-conseils-pour-reussir-son-netlinking-interne/14-conseils-pour-reussir-son-netlinking-interne.shtml> Réussir son netlinking interne sur Journal du Net
- <http://www.simpleweb.fr/> SimpleWeb, simplifions l'internet
- <http://www.journaldunet.com/diaporama/0701-conseil-abandonpanier/index.shtml> 5 conseils pour éviter l'abandon de chariot



## Ajax

[http://www.auditweb.net/conseils/ajax\\_et\\_ergonomie.html](http://www.auditweb.net/conseils/ajax_et_ergonomie.html)

<http://www.scriptol.fr/ajax/ajax-garrett.php>

<http://www.scriptol.fr/ajax/>

<http://www.zebulon.fr/dossiers/84-dynamisez-site-internet.html>

Parole d'expert - Ajax et ergonomie

Ajax - La création du nom par Jesse James Garrett

Ajax - Scriptol.fr

DHTML, Ajax, Drag and Drop :

## Dynamisez votre site Internet !

<http://script.aculo.us/>

script.aculo.us - web 2.0 javascript

<http://ajaxjournal.free.fr/>

Ajax Journal

<https://developer.mozilla.org/Fr/AJAX>

AJAX - Mozilla Developer Center

<http://www.xul.fr/>

Ajax et XUL, tutoriels et documents avec JavaScript, DOM et XML

## Couleur

<http://www.vischeck.com/>

Vischeck: Home

<http://www.vischeck.com/daltonize/>

Vischeck: About Daltonize

<http://www.iamcal.com/toys/colors/>

Color Vision - by Cal Henderson

<http://www.ficml.org/jemimap/style/color/wheel.html>

4096 Color Wheel Version 2.1

<http://www.ficml.org/jemimap/style/color/couleur.html>

Palette de 4096 couleurs version 2.1

<http://www.yoyodesign.org/outils/ncolor/ncolor.php?langue=fr>

nColor v2.1 : Un sélecteur de couleurs permettant des comparaisons entre visions standard et daltoniennes



## Guidelines

<a href="http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html">http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html</a>	Apple Human Interface Guidelines: Introduction to Apple Human Interface Guidelines
<a href="http://msdn2.microsoft.com/en-us/library/aa286531.aspx">http://msdn2.microsoft.com/en-us/library/aa286531.aspx</a>	MS User Interface Design and Development
<a href="http://msdn2.microsoft.com/en-us/library/aa511440.aspx">http://msdn2.microsoft.com/en-us/library/aa511440.aspx</a>	Vista Guidelines section
<a href="http://library.gnome.org-devel/hig-book/stable/">http://library.gnome.org-devel/hig-book/stable/</a>	GNOME Human Interface Guidelines 2.2
<a href="http://usability.kde.org/hig/">http://usability.kde.org/hig/</a>	KDE Usability Project - KDE Human Interface Guidelines
<a href="https://www-306.ibm.com/software/ucd/">https://www-306.ibm.com/software/ucd/</a>	IBM Design

## Java

<a href="http://penserenjava.free.fr/">http://penserenjava.free.fr/</a>	Penser en Java
<a href="http://java.sun.com/docs/books/tutorial/uiswing/">http://java.sun.com/docs/books/tutorial/uiswing/</a>	Trail: Creating a GUI with JFC/Swing (The Java™ Tutorials)
<a href="http://java.sun.com/docs/books/tutorial/2d/images/index.html">http://java.sun.com/docs/books/tutorial/2d/images/index.html</a>	Lesson: Working with Images (The Java™ Tutorials &gt; 2D Graphics)
<a href="http://java.sun.com/docs/books/tutorial/information/download.html">http://java.sun.com/docs/books/tutorial/information/download.html</a>	Download The Java Tutorial
<a href="http://jmdoudoux.developpez.com/">http://jmdoudoux.developpez.com/</a>	Cours Java et Eclipse de J.M. Doudoux - Club d'entraide des développeurs francophones



## Divers

<http://www.dontclick.it/>

www.dontclick.it

<http://www.dsi.univ-paris5.fr/typo.html>

Typographie française

<http://www.sqlagency.com/blogs/ecreativegarden/index.php?2007/10/22/62-typographie-et-web>

ecreativegarden - sql agency - Design / Richmédia - Typographie et Web

[http://www.beonlineb.com/click\\_around.html](http://www.beonlineb.com/click_around.html)

The Arcade Fire presents

## NetBeans

<http://www.netbeans.org/kb/60/java/quickstart-gui-legend.html>

GUI Builder - NetBeans IDE 6.0 Guide

<http://www.netbeans.org/kb/60/java/>

NetBeans IDE 6.0: General Java Programming

<http://www.netbeans.org/kb/60/java/quickstart-gui.html>

Designing a Swing GUI in NetBeans IDE

## 8.2.Bibliographie sélective

- ✓ [Ergonomie du logiciel et design web : Le manuel des interfaces utilisateur](#)  
par Jean-François Nogier - Dunod
  
- ✓ Couleur - le guide le plus complet  
Tom Fraser, Adam Banks - Evergreen, Taschen