

# **RECOMMANDATIONS ERGONOMIQUES**

# LA DISPOSITION DES ELEMENTS DANS LES FENETRES

**1.** Pour regrouper les données (regroupement au sein de rubriques) et définir leur enchaînement (chronologie de présentation des données affichées ou à saisir), tenir compte des habitudes de travail de l'utilisateur :

- fréquence d'utilisation,
- séquence d'utilisation dans la tâche en cours,
- importance dans le contexte (saisies obligatoires, optionnelles).

Présenter les groupes de données les plus importants selon le critère privilégié (par exemple, fréquence d'utilisation) vers le haut de la fenêtre.

Remarque : si le travail de l'utilisateur s'appuie sur un formulaire papier, et si ce formulaire est bien construit, faire en sorte que la disposition des champs dans la fenêtre soit proche de la structure du formulaire.

**2.** Aligner verticalement les champs de saisie et/ou d'affichage avec leurs libellés ; la justification peut se faire de trois manières :

- cas 1 : si les libellés sont de dimension sensiblement identique, justifier les champs et les libellés à gauche ;
- cas 2 : si les libellés sont de dimensions très variables, justifier ces libellés à droite et les champs à gauche ;
- cas 3 : il est également possible de disposer les libellés au dessus des champs de saisie ; il faut alors les justifier à gauche.

Nom :	<input type="text"/>	Profession :	<input type="text"/>
Prénom :	<input type="text"/>	Ancienneté :	<input type="text"/>
Age :	<input type="text"/>		

Figure 1 : Cas 1 (libellés de dimension sensiblement identique)

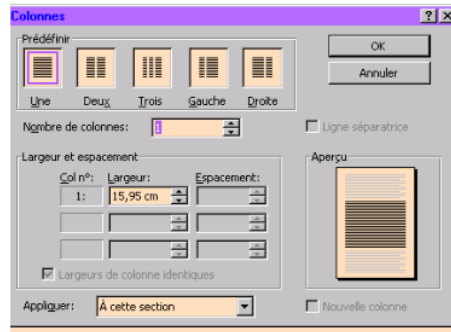
Nom :	<input type="text"/>	Profession :	<input type="text"/>
Prénom :	<input type="text"/>	Profession du conjoint :	<input type="text"/>
Adresse personnelle (domicile) :	<input type="text"/>	Nombre d'enfants à charge :	<input type="text"/>

Figure 2 : Cas 2 (libellés de dimension différente)

Nom :	<input type="text"/>	Nombre d'enfants à charge :	<input type="text"/>
Prénom :	<input type="text"/>	Profession :	<input type="text"/>
Adresse personnelle (domicile) :	<input type="text"/>	Profession du conjoint :	<input type="text"/>

Figure 3 : Cas 3 (libellés positionnés au dessus des champs de saisie)

- 3.** Attribuer un libellé à chaque donnée et à chaque groupe de données.
- 4.** Séparer les différentes catégories d'informations par des espaces et éventuellement par des boîtes de groupe (= encadrements).



**Figure 4 : Utilisation des boîtes de groupe (Word)**

- 5.** Placer l'en-tête de groupe2 au dessus de la boîte de groupe3, et en incrustation dans le cadre s'il y en a un.
- 6.** Ne pas utiliser de boîte de groupe pour les boutons de commande.
- 7.** Ne pas utiliser systématiquement des boîtes de groupe quand il y a un en-tête : trop de cadres peuvent gêner la lisibilité globale de la fenêtre. Utiliser alors des en-têtes de groupe sans représenter une boîte de groupe.

## LA PRESENTATION DES LISTES ET DES TABLEAUX

**1.** Attribuer systématiquement un titre aux listes et aux tableaux ; ces titres permettront à l'utilisateur de se repérer facilement et rapidement.

**2.** Si une liste ou un tableau est muni d'un ascenseur (car sa taille/format ne permet pas de voir l'intégralité de son contenu), alors :

- ascenseur vertical : les en-têtes de lignes de données doivent défiler mais les en-têtes de colonnes doivent toujours rester visibles ;
- ascenseur horizontal : les en-têtes de colonnes doivent défiler mais les en-têtes de lignes de données doivent toujours rester visibles.

**3.** Respecter les alignements standards des traitements de texte :

- libellés en général à gauche,
- numériques à droite.

Eviter les alignements centrés (effets de vagues verticales).

**4.** Les colonnes doivent être séparées par un double espace (1 caractère blanc x 2). Si cela ne peut pas être fait (par manque de place), séparer les colonnes par des traits (lignes verticales).

**5.** Comment organiser les éléments à l'intérieur d'un tableau/d'une liste (c'est-à-dire sur quoi se baser pour établir la chronologie des données) ?

- adopter l'ordre respectant les habitudes de travail de l'utilisateur ;
- le cas échéant, utiliser l'ordre alphabétique ou numérique.

## LES ELEMENTS TEXTUELS

- 1.** Utiliser le vocabulaire appartenant au domaine d'activité des utilisateurs.
- 2.** Utiliser des termes explicites et non ambigus. Faire des phrases (ou expressions) simples, c'est-à-dire facilement compréhensibles par l'opérateur ; préférer les formes affirmatives.
- 3.** Indiquer les unités de mesure utilisées (en abrégé si elles sont connues de l'utilisateur).
- 4.** Eviter d'utiliser des abréviations, sinon, fournir chaque fois que c'est possible des abréviations bien choisies :
  - utiliser uniquement celles qui sont le plus largement adoptées par les utilisateurs car ces derniers pourront avoir rencontré certaines abréviations qui signifient autre chose ;
  - préférer la contraction (omission des lettres internes) pour les mots courts ;
  - préférer la troncature pour les mots longs (omission des lettres finales).
- 5.** Par souci d'homogénéité, ne pas utiliser de synonymes : désigner toujours un même objet ou une même action par le même libellé.
- 6.** Lors de la création de listes d'éléments, on peut être confronté à des éléments de texte trop longs pour contenir dans l'espace d'une ligne de la liste. Dans ce cas, supprimer des parties de texte dans le milieu du libellé et y insérer des points de suspension, conservant ainsi le début et la fin du libellé de l'élément.  
Attention toutefois : dans certains cas, il sera plus judicieux de conserver le début du libellé et de faire suivre ce dernier par des points de suspension.
- 7.** Repérer chaque champ, liste, tableau, colonne ou groupe de données par un libellé. On distingue trois grands types de libellés :
  - le libellé de champ, à utiliser pour identifier une donnée affichée ou à saisir ;
  - l'en-tête de colonne, à utiliser pour identifier la colonne d'un tableau ou d'une liste ;
  - l'en-tête de groupe, à utiliser pour identifier un ensemble de données rassemblées sous la forme d'un groupe.

# LA TYPOGRAPHIE

**1.** Concernant la taille des caractères :

- taille minimum : 8 points ; des caractères de plus petite taille sont quasiment illisibles ;
- taille maximum : 16 points ; l'utilisation de caractères de plus grande taille gêne la lisibilité.

**2.** Choisir la police de caractères en fonction de critères de lisibilité (éviter l'italique).

**3.** Eviter d'utiliser plus de trois polices de caractères différentes dans une même fenêtre ou sur plusieurs fenêtres affichées simultanément.

**4.** De manière générale, utiliser une seule police de caractères dans un champ. Toutefois, il est possible d'utiliser des codages particuliers (police différente, couleur, italique, gras, souligné...) pour différencier certains textes tels que des mots clés, des liens, des libellés de champs de saisie obligatoire, etc.

**5.** Pour tous les libellés (champs de saisie et champs d'affichage, option de menu, boutons d'option, titre, etc.) utiliser une majuscule à l'initiale. Ne mettre donc en majuscule que la première lettre du premier mot du libellé et aucun autre mot de ce libellé.

# LA COULEUR

La couleur peut être employée pour coder visuellement l'information : différenciation et identification des informations affichées.

**1.** La couleur ne doit pas être le seul moyen utilisé pour communiquer de l'information et ne devrait pas être le seul élément qui distingue deux objets ; il doit y avoir d'autres indicateurs (libellés, forme, disposition) ; appliquer ce principe surtout pour tenir compte des utilisateurs daltoniens ou achromates mais aussi pour les écrans monochromes (même si ces derniers sont « rares » !)

**2.** Choisir les couleurs en gardant à l'esprit que l'objectif est de faciliter la lisibilité des informations affichées.  
C'est pourquoi certaines associations de couleurs doivent être évitées. Ainsi, essayer de ne jamais utiliser ensemble :

- le rouge avec le bleu,
- le jaune avec le violet,
- le jaune avec le vert.

Pour le fond des fenêtres, éviter les teintes de rouge, de jaune et de ses dérivés (vert, orange, ...). Privilégier plutôt le gris pâle, sinon du bleu foncé.  
Pour de l'information que l'utilisateur doit lire, éviter le bleu pâle (et les couleurs pâles en général).

**3.** Homogénéité : au sein d'une même application et entre différentes applications destinées à un même groupe d'utilisateurs, toujours choisir les mêmes " codages couleur ".

**4.** Respecter les règles d'association conventionnelles entre la couleur et sa signification :

- vert : signifie que tout se passe correctement,
- jaune ou orange : attention, vigilance,
- rouge : alerte, danger, interruption.

**5.** Eviter d'utiliser plus de 5 ou 6 couleurs différentes dans une même fenêtre (mais également au sein d'une même application). Au delà de 5 ou 6, cela entraîne une surcharge visuelle et l'utilisateur éprouve des difficultés à saisir le rôle joué par la couleur au sein de l'application.

**6.** Concernant les états sur les imprimantes noir et blanc : s'assurer que chaque couleur reste visible sur les impressions (il doit y avoir correspondance entre les couleurs qui apparaissent à l'écran et les niveaux de gris des sorties papier ; si le nombre de niveaux de gris disponible est insuffisant, prévoir des hachures).

## LES ICONES

Les icônes peuvent avoir deux fonctions :

- aider à identifier la nature d'une information, sous forme d'icône,
  - représenter une action par une icône associée à un bouton de commande.
- « Exemple : l'icône loupe (ou jumelles) est associée au bouton de commande <Rechercher>.

**1.** Attention, les icônes utilisées doivent être représentatives de l'action ou du concept que l'on souhaite représenter, autrement dit elles doivent être explicites. Il paraît néanmoins « prudent » de doubler les icônes d'un libellé.

**2.** Homogénéité : quand une icône est utilisée pour représenter quelque chose, la conserver pour l'ensemble de l'application. Autrement dit, ne pas utiliser deux ou plusieurs icônes différentes pour désigner une seule et même action ou un seul et même concept.



## L'UTILISATION DES ONGLETS

- 1.** Utiliser les onglets pour éviter les successions de boîtes de dialogue ou un trop grand nombre de boutons.
- 2.** Eviter d'utiliser plus d'une rangée d'onglets, car la manipulation de 2 ou 3 rangées d'onglets est complexe.
- 3.** Quand le principe des onglets est utilisé, si ces derniers sont liés (dépendants), les boutons <OK> et <Annuler> doivent être placés en dehors de l'onglet. Si les onglets sont indépendants, ces boutons doivent être placés sur chacun des onglets : le bouton <Annuler> devient <Fermer> et la demande de confirmation devient facultative (<OK> facultatif).
- 4.** Centrer les libellés des onglets et les afficher sur une seule ligne.
- 5.** Aligner les languettes des onglets à gauche et les accoler les unes aux autres.

# LA NAVIGATION INTRA-APPLICATION (CONVERSATIONS DE FENETRES)

La navigation intra-application fait référence au cheminement de l'utilisateur dans l'application. On peut aussi parler d'enchaînement d'écrans/de fenêtres.

## LA CONVERSATION LIBRE ET CONVERSATION GUIDEE

### Conversation libre

L'utilisateur bénéficie d'une grande liberté dans son cheminement à l'intérieur de l'application.

- **Avantages de la conversation libre :**

- l'utilisateur organise son travail comme il le souhaite (pas de contrainte « chronologique » dans la réalisation des tâches) ; pour l'organiser, il s'appuie essentiellement sur son expérience, mais aussi sur ses habitudes de travail ;
- il a une vision globale de l'ensemble des possibilités offertes par l'application ;

- **Inconvénients de la conversation libre :**

- toutes les tâches sont « équivalentes » (du point de vue de leur importance), les priorités entre les différentes tâches ne sont pas gérées par l'application ; du point de vue « développement », ceci implique d'utiliser des contrôles de cohérence importants ;
- si le nombre de fenêtres simultanément affichées à l'écran est important, l'utilisateur risque de s'y perdre.

### Conversation guidée

C'est exclusivement l'application qui pilote l'interface :

- **Avantages de la conversation guidée :**

- l'utilisateur est complètement guidé/assisté par le logiciel ; il peut ainsi gagner en efficacité, surtout s'il s'agit d'un utilisateur débutant ;
- la chronologie de présentation et de réalisation des tâches est pré-définie ; du point de vue « développement », ceci permet de réduire au maximum les contrôles de cohérence de l'application.

- **Inconvénients de la conversation guidée :**

- l'utilisateur ne peut prendre aucune initiative dans la gestion de ses tâches, il ne peut pas profiter de sa propre expérience pour organiser son travail comme il l'entend ;
- du fait de la rigidité de pilotage des tâches, l'évolution de l'application est rendue difficile ;
- l'utilisateur peut éprouver des difficultés à avoir une vision globale des possibilités de l'application.

### Quel type de conversation adopter ?

- Si l'utilisateur se trouve dans un contexte où la « rentabilité / productivité » est primordiale (exemple : saisie « de masse ») => choisir plutôt la conversation guidée
- Si l'utilisateur doit impérativement respecter une procédure dans la réalisation de sa tâche => choisir plutôt la conversation guidée.
- Si l'on souhaite proposer une application souple d'utilisation (pas de procédure imposée) => choisir plutôt la conversation libre.

– Si l'on souhaite que l'utilisateur ait une vue d'ensemble des fonctionnalités proposées par l'application => choisir plutôt la conversation libre.

Remarque : afin de donner davantage de liberté à l'utilisateur, on peut envisager de lui donner la possibilité de choisir son « mode » de conversation – libre ou guidé – en fonction de son degré de « maîtrise » de l'application (novice/expert) mais aussi en fonction de son expérience dans son travail (débutant/expérimenté).

## LE NOMBRE DE FENETRES

**1.** Comment définir le nombre de fenêtres à l'intérieur de l'application et la densité d'information affichée dans chaque fenêtre ? Pour chaque fenêtre, il faut trouver un équilibre/compromis entre :

- le désir de présenter toutes les informations nécessaires à l'utilisateur,
- la nécessité de présenter une fenêtre lisible (si trop d'informations sont affichées, l'utilisateur aura du mal à se repérer).

**2.** Dans la mesure du possible, la profondeur de l'application doit être limitée à trois niveaux : au delà, l'utilisateur risque de « se perdre » dans l'application et aura également des difficultés à la manipuler.

# LES CHAMPS DE SAISIE

## Aspect des champs de saisie

1. Dans les champs de saisie, cadrer les caractères saisis à gauche (lettres ou chiffres).
2. Dans la mesure du possible, la taille des champs doit être proportionnelle au nombre de caractères qui peuvent être saisis.
3. Lors de la transformation d'une fenêtre de saisie en fenêtre de consultation, modifier l'aspect des champs de saisie qui deviennent alors des champs d'affichage.
4. Dans le cas où la taille des données à saisir dépasse la largeur du champ de saisie, proposer un champ de type "explose". Ce type de champ s'agrandit lorsque le curseur est positionné dessus ; la totalité du champ devient donc lisible en cachant une partie de la fenêtre. Eviter d'utiliser le défilement ; si celui-ci est employé, préférer le défilement vertical plutôt qu'horizontal (mais ne pas utiliser le défilement horizontal pour la saisie d'une donnée de moins de 10 caractères).

## Initialisation des champs de saisie avec une valeur par défaut

1. Initialiser les champs de saisie avec une valeur par défaut chaque fois que cela est possible ; cette valeur doit correspondre :
  - à la valeur ayant la plus grande probabilité d'être choisie par l'utilisateur,
  - ou à la valeur précédemment choisie (ou saisie) par l'utilisateur dans le même contexte.

## Caractères à saisir

1. Ne pas forcer la saisie de caractères en minuscules ou en majuscules.
2. Quand des unités de mesure sont associées à un champ de saisie, les faire apparaître dans le libellé du champ de saisie (l'utilisateur ne doit pas avoir à saisir l'unité de mesure) ; ces unités doivent bien entendu être familières aux utilisateurs.
3. Quand le curseur est positionné sur un champ en saisie, il doit avoir un aspect différent (forme ) pour indiquer à l'utilisateur la possibilité de saisir des données.
4. Pour les champs qui nécessitent un format particulier (exemple : champ date), proposer un préformatage des données à saisir ou alors formater a posteriori (suite à la saisie de l'information par l'utilisateur).  
« Exemple : caractères de séparation dans une saisie de date .././.... »

## **Saisie obligatoire et saisie optionnelle**

1. Différencier les champs dont la saisie est obligatoire des champs dont la saisie est facultative. Solutions proposées (par ordre de préférence) :
  - libellés des champs dont la saisie est obligatoire en gras, et en maigre pour ceux dont la saisie est facultative ;
  - puce ou triangle de couleur devant les champs dont la saisie est obligatoire et pas de marque distinctive devant les champs dont la saisie est facultative ;
  - fond des champs à saisir en couleur (discrète) ;
  - encadrement gras pour les champs dont la saisie est obligatoire et encadrement fin pour ceux dont la saisie est facultative.

## **Validation de la saisie**

1. Lorsque l'utilisateur entre des données dans un champ de saisie, cela ne doit jamais entraîner de modification irréversible de l'environnement de l'application ; autrement dit, il ne doit pas y avoir de création, de mise à jour, ni destruction de données sans une validation explicite de l'utilisateur (message de confirmation).

## **LA SAISIE DANS UN TABLEAU**

1. L'utilisateur doit pouvoir se déplacer entre les cellules du tableau à l'aide de la touche < Tabulation > du clavier :
  - la combinaison de touches < Maj + Tabulation > doit positionner de nouveau le curseur sur la cellule précédente,
  - la combinaison des touches < Control + Tabulation > doit permettre de sortir du tableau et de passer au champ suivant dans la fenêtre.
2. La touche < Tabulation > doit valider le format de saisie courante et afficher correctement la donnée saisie (alignement, centrage).
3. Centrer les chiffres ou les cadrer à droite.

## **LA SELECTION D'UN NOMBRE LIMITE D'OPTIONS (RADIO)**

1. Utiliser des boutons d'option pour des choix exclusifs.
2. Proposer le choix entre deux options au moins, excepté dans le cas où le choix entre les différentes options n'est pas significatif (griser alors toutes les options).
3. Il doit toujours y avoir une valeur par défaut. Si un utilisateur peut choisir de ne sélectionner aucun des choix, ajouter une option « Aucun » ou l'équivalent.
4. Rassembler les boutons d'option dans des boîtes de groupe.

**5.** Ecrire le libellé à droite du bouton d'option.

## **Les options non exclusives : cases à cocher**

**1.** Eviter d'utiliser les cases à cocher pour les choix importants au niveau fonctionnel (exemple : la sauvegarde).

**2.** Rassembler les cases à cocher dans des boîtes de groupe.

**3.** Ecrire le libellé à droite de la case à cocher.

**4.** Afficher les cases à cocher pertinentes dans le contexte courant, griser le cadre et le libellé de celles qui ne sont pas utilisables.

## **LES DONNEES SAISIES PAR SELECTION DANS UNE LISTE**

**1.** L'utilisateur doit pouvoir désélectionner un élément aussi simplement et par le même moyen qu'il peut le sélectionner.

**2.** Mettre en évidence l'élément sélectionné de façon immédiate.

**3.** La sélection ne doit jamais entraîner une action. La désélection d'un élément ne doit pas avoir d'incidence sur la tâche en cours (pas de retour au début de la tâche).

**4.** Utiliser les listes pour afficher une suite d'éléments ; la sélection pourra être unique ou multiple (sélection d'un ou plusieurs éléments).

**5.** Afficher les éléments dans un ordre s'appuyant sur la logique de l'utilisateur ou la fréquence d'utilisation. Le cas échéant, utiliser l'ordre alphabétique.

**6.** Ne pas permettre au curseur de boucler sur la liste : lorsque l'extrémité est atteinte, laisser le curseur dessus.

**7.** Dans le cas des listes longues :

- permettre un positionnement direct basé sur la frappe du premier caractère de l'élément voulu,
- mettre l'initiale de chaque élément en majuscules pour favoriser la lisibilité.

**8.** Indiquer le nombre d'éléments sélectionnés.

**9.** Si la saisie est optionnelle, une boîte à liste déroulante doit contenir obligatoirement un élément “ Aucun ” positionné en première ligne qui sera sélectionné par défaut.

**10.** La sélection multiple doit être mise en œuvre quand l'utilisateur doit sélectionner des éléments séparés de la liste (c'est-à-dire situés non à la suite les uns des autres).

**11.** La sélection étendue doit être mise en œuvre quand l'utilisateur doit sélectionner plusieurs éléments consécutifs dans la liste.

## **LES MENUS**

Les menus présentent la liste des commandes disponibles à l'utilisateur. Il existe plusieurs types de menus, tels que les menus déroulants, les menus en cascade, les menus contextuels.

### **LA BARRE DE MENUS**

**1.** Proposer au moins deux menus (sans compter le menu système) mais pas plus de dix dans la barre de menus.

Pour cela, grouper ou hiérarchiser les actions et utiliser les menus déroulants.

**2.** Associer un raccourci clavier à tous les menus.

**3.** Les menus de la barre de menus ne doivent pas être des actions directes : ils doivent donner accès à des menus déroulants.

**4.** Dans la barre de menus, utiliser si possible un seul mot pour le libellé de chaque menu. L'initiale du mot doit être en majuscule.

**5.** Si un menu n'est jamais proposé à l'utilisateur (parce que le profil de l'utilisateur ne le justifie pas par exemple), ne pas afficher ce menu et ne pas laisser son emplacement vide. Le cas échéant (si cette recommandation est « techniquement » impossible à appliquer), griser le menu.

**6.** L'utilisateur doit aussi pouvoir accéder aux menus déroulants par le clavier en utilisant les touches flèche gauche et droite (déplacement en bouche sur la barre de menus).

## **La barre d'outils**

**1.** Pour représenter les outils, utiliser des icônes explicites.

**2.** Lorsque l'utilisateur choisit un outil, il est possible d'attribuer au curseur la forme de l'outil afin de permettre la visualisation directe de l'action en cours.



- 3.** Le contenu de la barre d'outil est invariable mais il peut être contextuel (estompé, inactif).
- 4.** Quand le curseur est placé dans une zone de la fenêtre extérieure au champ d'application de l'outil, modifier la forme du curseur pour montrer qu'il n'est pas utilisable. Par exemple, lui donner la forme de " stationnement interdit ".
- 5.** Quand le curseur se déplace sur la barre d'outils, lui donner la forme d'une flèche de sélection.
- 6.** L'utilisateur ne doit pouvoir sélectionner qu'un seul outil à la fois.
- 7.** Donner à l'utilisateur le choix d'afficher ou de ne pas afficher la barre d'outils. Ces boutons correspondront uniquement aux options de menu les plus fréquemment utilisées.
- 8.** Placer les boutons d'action de la barre d'outils dans l'ordre des menus. Grouper les options d'un même menu et les séparer des autres par six pixels.
- 9.** Donner à l'utilisateur la possibilité de positionner la barre d'outils aux quatre coins de la fenêtre : horizontalement en haut ou en bas de la fenêtre ou verticalement à droite ou à gauche.