

# Rappel sur les critères ergonomiques

Scapin (1990) : 8 critères

## Compatibilité

accord pouvant exister entre les caractéristiques de l'utilisateur (perceptions, mémoires, habitudes, etc.) et l'organisation des entrées, des sorties et du dialogue.

Exemples :

- organisation des infos affichées doit être conforme à l'organisation des infos à entrer
- le format des écrans doit être compatible avec celui des documents papier
- les procédures de dialogue doivent être compatibles avec l'ordre tel que se l'imagine l'opérateur ou celui dont il a l'habitude.

....

## Homogénéité – consistance

façon dont on effectue des choix similaires d'interface (codes, procédures, dénominations, etc.) dans des contextes identiques. Il s'applique à la localisation des objets; à leur format, leur dénomination ; mais aussi syntaxe des commandes.

Exemples :

- localisation similaire des fenêtres
- formats d'écrans similaires
- procédures similaires d'accès aux options de menus

...

## Guidage

ensemble des moyens mis en oeuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec le système. Quatre sous critères : prompting, distinction vs regroupement entre items, feed-back et clarté.

Exemples :

Prompting : indiquer les unités de mesure, indiquer toutes les informations d'état, guider les entrées de données par un format adéquat.

Groupements et distinctions entre items :

par la localisation : organiser des items, grouper des options de menus.

par le format : distinction entre des aires, des labels, des champs d'entrée.

Feed-back immédiat : toute action à l'initiative de l'utilisateur ou à l'initiative du système doit conduire à un résultat observable

Clarté : caractéristiques lexicales de présentation des informations sur l'écran (luminance des caractères, contraste caractères/fond, dimensions des lettres, espacement entre les mots, les lignes, paragraphes, longueurs des lignes...)

### Adaptabilité

capacité d'un système à réagir différemment selon le contexte, les besoins et les préférences de l'utilisateur. Il comprend deux sous-critères: flexibilité et prise en compte de l'expérience de l'utilisateur.

Exemples :

Flexibilité : capacité de l'interface à s'adapter à des actions variées de l'utilisateur

Prise en compte de l'expérience utilisateur : moyens mis en oeuvre pour permettre au système de respecter le niveau d'expertise de l'utilisateur.

### Charge de travail

ensemble des éléments de l'interface qui ont un rôle pour l'utilisateur, dans la réduction de sa charge mnésique et dans l'augmentation de l'efficacité du dialogue. Divisé en : brièveté et charge mentale.

Brièveté :

Concision : ne pas avoir à rentrer les zéros et les blancs devant un item ; si les codes sont > 4 caractères, utiliser des mnémoniques ou des abréviations ; entrées de données courtes

Actions minimales : minimiser le nombre d'étapes dans la sélection de menus ; ne pas entrer des données déjà introduites ; éviter les ponctuations pour les entrées de commandes

Charge mentale :

Limiter la densité de l'information sur l'écran en n'affichant que les infos nécessaires ; l'info ne doit pas nécessiter de traduction d'unités ; utiliser le minimum de quantificateurs

### Contrôle utilisateur

contrôle dont dispose l'utilisateur sur le dialogue ainsi qu'au caractère explicite de ses actions.

Actions explicites : l'interface doit exécuter seulement les opérations demandées par l'utilisateur

Contrôle utilisateur : permettre à l'utilisateur d'avoir toujours la main

### Gestion des erreurs

ts moyens mis en oeuvre pour éviter, réduire les erreurs et les corriger. Trois sous critères : protection contre les erreurs, qualité des messages d'erreur et correction des erreurs.

Protection contre les erreurs : moyens mis en place pour détecter des erreurs d'entrées de données ou de commandes. Par ex: un message qui informe sur le risque de perte de données.

Qualité des messages d'erreur : pertinence, exactitude de l'info données à l'utilisateur sur la nature de l'erreur commise et sur les actions à entreprendre pour la corriger.

Correction des erreurs : moyens mis à disposition de l'utilisateur pour lui permettre de corriger ses erreurs.

### Signifiante des codes

bonne définition, cad claire et univoque des divers codes ; labels des commandes, titres d'écrans, des options.

L'adéquation entre l'objet ou l'info affichée et son référent. Exemple : le titre doit véhiculer ce qu'il représente ; expliciter la ou les règles de contraction ou d'abréviation ; les labels doivent être distincts.

AFNOR, 1991 : 7 critères

AFNOR : Définition des critères ergonomiques de conception et évaluation des interfaces utilisateurs. Norme Z67-133-1.

### Compatibilité

capacité à s'intégrer dans l'activité des utilisateurs (organisation des fonctions conformes à la représentation de la tâche)

### Homogénéité

stabilité des choix de conception (présentation des informations, une même procédure conduit au même résultat).

### Guidage

moyens mis à la disposition des utilisateurs (structuration de l'affichage, message)

### Souplesse

capacité de l'interface à s'adapter (mode expert / mode débutant ; possibilité de paramétrer ou de chaîner des commandes fréquemment utilisées)

### Contrôle explicite

moyens permettant à l'utilisateur de maîtriser les effets des commandes (signaler les options actives dans les menus, pouvoir interrompre une impression)

#### Gestion des erreurs

guider l'utilisateur dans l'identification et la correction des erreurs (clarté des messages, commande undo)

#### Concision

réduction des activités de perception et de mémorisation (icônes, options par défaut)

Coutaz (5 critères fondamentaux)

Dans le chapitre 4 de son ouvrage, Joëlle Coutaz présente 5 points de référence fondamentaux. Ces 5 points sont classés en fonction de leur facilité de mise en application plus qu'en raison de leur impact sur la qualité de l'interaction : 1 cohérence, 2 concision, 3 feedback, 4 structuration des activités, 5 flexibilité.

#### Cohérence

la cohérence repose sur la caractère unitaire des constituants de l'interface (absence d'exceptions). Elle concerne le choix d'une métaphore d'interaction, la spécification du plan, l'exécution du plan.

#### dans le choix d'une métaphore d'interaction

La cohérence concerne tout d'abord le choix d'une métaphore d'interaction qui peut servir d'élément fédérateur à l'homogénéité de l'interface et à l'homogénéité de toutes les étapes de réalisation d'une tâche, depuis la spécification du plan, en passant par l'exécution, jusqu'à la perception et l'interprétation.

Qu'est-ce qu'une métaphore ? En linguistique, une métaphore est l'emploi d'un terme concret pour exprimer une notion abstraite par substitution analogique. Une métaphore permet à l'apprenant d'initialiser le modèle mental d'une notion mal connue, par exemple la notion de répertoire, avec le modèle d'une action connue, par ex celle de dossier.

Deux classes de métaphores :

Les métaphores qui s'inspirent du monde réel, comme la métaphore du 'Desktop' rendue célèbre par le Star de Xerox et reprise ensuite par les machines d'Apple.

Sur ces machines, l'interface tend à reproduire le monde du bureau. Un dossier électronique a la même présentation externe qu'un dossier réel : il comporte un chemise et un nom. Comme un dossier réel, on peut l'ouvrir, le reproduire, le ranger dans un autre dossier, etc. Sans oublier que toutes ces actions se font généralement avec la souris (ou tout autre dispositif de pointage) substitut électronique de la main.

Les métaphores qui parlent d'un mode abstrait. Ces interfaces montrent une image dans laquelle l'univers n'est plus représenté explicitement. L'interface n'est plus le monde miniaturisé mais devient un medium linguistique. L'utilisateur et le système sont engagés dans une conversation sur

un monde supposé. L'interface est alors un intermédiaire entre l'utilisateur et le monde dont il est question. Les langages de commande traditionnels, et les interfaces en langue naturelle entrent dans cette catégorie.

Ici l'utilisateur n'effectue pas l'action sur un objet visible mais décrit une action sur un objet supposé. L'utilisateur n'est pas en contact direct avec l'objet, mais manipule des structures linguistiques.

Quels que soient les avantages et inconvénients, une métaphore définit un style d'interaction. Elle constitue l'unité de cohérence générale pour l'image.

#### dans la spécification du plan

La cohérence concerne aussi la spécification du plan : d'une tâche à l'autre (au sein d'une application ou au entre applications) il faut que la suite des commandes à exécuter soit identique pour réaliser une sous-tâche commune.

exple : dans les environnements traitement de texte et courrier électronique, la réalisation de la sous-tâche " dupliquer-imprimer un texte " devrait utiliser la même suite de commandes.

#### dans l'exécution

La cohérence concerne aussi l'exécution. Avec ici 3 aspects : syntaxiques, lexicaux, pragmatiques.

##### au plan syntaxique

la définition de la structure syntaxique des commandes pose au moins deux questions :

-quel ordre choisir pour les arguments ? D'une commande à l'autre, l'ordre de spécification des arguments communs doit être identique (Barnard, 1981). Cette règle s'appliquerait aussi bien aux langages textuels qu'aux langages de type iconique.

-le préfixage est-il préférable à une notation postfixée ? Cela dépend du type de langage. Pour les langages textuels, Barnard aboutit à la conclusion que la forme Verbe-Objet est la structure qui convient. Pour les langages de type iconique, Foley suggère la forme inverse Objet-Verbe.

##### au plan lexical

deux problèmes : le choix d'une nomenclature, l'élimination des incohérences lexicales.

-choix d'une nomenclature : la terminologie du système est floue et ne correspond pas au vocabulaire de l'utilisateur. L'imprécision des termes d'un menu peut conduire l'utilisateur débutant à formuler une intention inappropriée. Exple : au restaurant si le menu, comporte 'plat du jour' on peut avoir recours aux compétences du serveur. Choisir des termes précis, didactiques et pertinents. Par exple " détruire " est plus précis qu'enlever. Les intitulés des champs d'entrée " heure "(hh mm ss) et " distance " (en km) précisent le format et l'unité de mesure. Ils sont davantage didactiques que les simples noms " heure " et " distance ".

-élimination des incohérences lexicales : veiller à ce qu'une fonction sémantique ou un concept soient toujours désignés par le même nom et ceci quelque soit le contexte. L'exemple de la

fonction “ terminer ” dans l’environnement UNIX est particulièrement évocateur : pour spécifier la fin de saisie d’un message électronique, la notation usuelle est “. ” : elle est “ q ” pour quitter la messagerie, et “ logout ” pour terminer la session.

au plan pragmatique

ici, cela concerne essentiellement la localisation et l’organisation spatiale des informations.

La cohérence spatiale aide l’utilisateur à acquérir une connaissance motrice qui permet d’anticiper les actions physiques. Ainsi, l’utilisateur sait par avance où diriger son regard, comment orienter les mouvements de la souris ou quel doigt actionner. Maintien d’une certaine forme de stabilité de l’écran et de l’utilisation des dispositifs physiques de contrôle. Stabilité n’est pas ici synonyme de permanence mais d’expectative : les informations doivent être là où l’utilisateur les attend et les dispositifs physiques doivent se conformer à cette attente. Exemples : souris et menus.

Souris : le bouton doit être lié en permanence à la même fonction (ou à la même classe de fonctions)

Menus : afficher les menus à la même place (ou bien l’emplacement désigné par la souris) et ordonner les options en conservant un ordre fixe. S’il n’y a pas de logique, alors utiliser la fréquence des options ou à défaut, l’ordre alphabétique.

Conclusion sur la cohérence : les règles sur la cohérence sont faciles à satisfaire avec un peu de motivation et de vigilance. Les incohérences lexicales sont généralement repérables sans trop de difficultés. Avec les boîtes à outils qui tendent à définir un style d’interaction homogène, avec l’organisation modulaire des logiciels qui facilite l’ajustement des éléments lexicaux, on peut dire que les outils actuels contribuent à cet objectif de cohérence.

Concision

La concision repose sur la combinaison du bref et de l’expressif.

Il s’agit ici de réduire le nombre d’actions physiques nécessaires à la spécification des expressions d’entrée.

(Le niveau de concision des expressions de sortie dépend de l’utilisateur et de la tâche en cours. Il est donc directement lié à la structuration et à la flexibilité du système).

Concision des expressions d’entrée : (1) abréviations ; (2) macrocommandes ; (3) couper-coller ; (4) valeurs par défaut ; (5) défaire-refaire

Abréviations

elles s’adressent essentiellement aux utilisateurs expérimentés. Elles sont utiles et doivent être compréhensibles, cad dérivables selon des règles précises.

Utiles : exple du MAC

-Un menu facilite l’étape de spécification d’un plan mais n’en favorise pas l’exécution. Il agit comme une extension de la mémoire à court terme : au lieu de reconstituer l’élément mental

recherché, l'utilisateur se contente de le reconnaître dans la liste des options proposées.

- Il prend du temps à s'afficher et à disparaître
- il impose des changements de contrôle, notamment lorsque dans la tâche, l'usage du clavier est dominant.

Le menu du Mac : l'abréviation concilie la facilité de spécification et la rapidité d'exécution.

Compréhensibles : 4 règles d'usage courant

- caractère spécial suivi de la lettre initiale de la commande (type Pomme C etc.)
- suppression des voyelles (détruire = dtrr)
- troncature minimale (tant qu'il n'y a pas d'ambiguïté ensemble : compiler, créer, détruire, définir 3 lettres, si pas définir, 2 lettres)
- troncature deux lettres non ambiguës (co, cr, de, df)

### Macrocommandes

la macrocommande est une technique qui répond aux besoins de spécificité et de généralité. Elles répond aux besoins spécifiques de l'utilisateur pour l'expression de ses intentions. On fournira à l'utilisateur un langage de haut niveau qui lui permet d'exprimer directement les structures de décomposition des problèmes les plus fréquents. Ce langage, qui permet d'exprimer les tâches usuelles, perd de la puissance de la généralité et rend difficile la spécification des tâches qui ne se décomposent pas dans ses termes. Ce conflit entre particulier et général est résolu dans les systèmes UNIX et environnements LISP grâce à l'extensibilité (cf. leurs notions de macrocommande et de fonction).

### Couper-coller

c'est la version électronique de la méthode naturelle qui consiste à découper et assembler des morceaux de documents. Il permet de :

- reproduire des informations
- suppléer au manque d'intégration des outils de production d'informations puisque qu'aujourd'hui, il est possible de construire un dessin, éditer une formule, et de recopier les informations produites dans le texte d'un document.

### Valeurs par défaut

évitent les répétitions de saisie et proposent des valeurs sémantiquement correctes. Elles contribuent ainsi à réduire les erreurs de typographie et de choix sémantique. Deux catégories de valeurs par défaut : les valeurs dynamiques et les valeurs semi-statiques.

Valeurs par défaut dynamiques : elles sont automatiquement réévaluées et mémorisées en cours de session. Ex : save as (nom du fichier en cours d'édition). Ce qui permet de faire une sauvegarde sous le même nom ou sous un nom voisin. Les modifications des valeurs par défaut dynamiques sont décidées par le système.

Valeurs par défaut semi-statiques : ne varient qu'à la demande explicite de l'utilisateur, comme les fichiers " profile " qui regroupent les options des applications.

#### Défaire-refaire

ces commandes désignent implicitement la dernière commande exécutée. Elles réduisent le nombre d'actions physiques. Elles interviennent directement dans la réparation des erreurs. La fonction défaire (Undo typing) est un moyen de correction immédiate, simple et directe. Elles permettent d'annuler l'effet d'une action involontaire ou erronée. Elle encourage par conséquent l'exploration et les tentatives incertaines.

#### Feed-back

Réaction du système aux actions de l'utilisateur.

L'expression de sortie permet d'évaluer la situation, de la comparer au but recherché, puis selon le cas, de poursuivre ou de modifier le plan d'actions. Le feedback a donc la responsabilité de refléter l'état du système ; non pas l'état interne au service de la logique du logiciel, mais l'état qui répond à l'attente de l'utilisateur, voire anticipe ses intentions.

L'attente de l'utilisateur est décrite ici de trois façons : (1) informer pour rassurer ; (2) informer pour réduire la charge cognitive ; (3) informer pour remédier aux erreurs.

#### informer pour rassurer

l'utilisateur doit être informé sur l'état interne du système : lorsque les commandes actives des fonctions du système dont le TR est long ( > qq's secondes). On peut notamment penser au cas d'opérations à risque comme le transfert d'un fichier sur un réseau local. Ainsi, dans le cas d'un transfert de fichier on indiquera : l'identité et la taille du fichier transféré ; pourcentage d'octets déjà transférés ; temps prévu pour achever le transfert ; indications des unités de mesure. La forme du suiveur, un sablier, indique que la situation d'attente est normale ; la progression de la barre de % et l'évolution du temps estimé permettent à l'utilisateur d'évaluer approximativement les conditions du transfert.

#### informer pour réduire la charge cognitive

rappeler à l'utilisateur son contexte de travail, l'aider à se situer dans son espace de travail et lui indiquer les options possibles. L'information peut concerner le rappel du contexte de travail proprement dit, la navigation, la présentation des options.

#### rappel du contexte de travail

Avec les systèmes organisés en états, la solution généralement utilisée est d'associer une forme de suiveur ou de curseur à chaque mode d'interprétation des actions d'entrée.

Pour les systèmes qui supportent le multifenêtrage et les activités multiples, aucune solution n'est vraiment satisfaisante. La fenêtre courante reçoit les actions d'entrée jusqu'à ce que l'utilisateur



désigne une autre fenêtre. La présentation de la fenêtre courante repose à peu près toujours sur la même technique qui consiste à la mettre en évidence par un cadre particulier. Les variations entre systèmes se manifestent essentiellement sur le changement de profondeur de la fenêtre. Pour le Macintosh, la fenêtre courante passe au premier plan. Pour d'autres elle conserve sa place dans l'empilement.

#### navigation

Les feedbacks peuvent aider l'utilisateur à se situer dans son espace de travail. Dans un espace linéaire (traitement de texte traditionnel) l'utilisateur n'a que deux degrés de liberté : en avant ou en arrière. Les barres de défilement n'ont pas toujours la qualité informative requise. Dans un espace plus vaste et plus complexe, comme un système hypertexte par exemple, l'utilisateur peut créer et modifier de manière interactive des liens entre des entités. Ces entités constituent les noeuds d'un réseau sémantique et servent à représenter n'importe quoi : image, idée, procédure de programme, morceau musical....L'idée directrice des hypertextes est de donner à l'utilisateur les moyens d'organiser et de communiquer ses pensées et ses connaissances.

#### présentation des options

C'est le résultat d'un compromis entre la visibilité, la validité et la concision. La visibilité définit la présence de l'option dans l'image pour que l'utilisateur en apprenne l'existence. Ex : présenter dans l'image toutes les options. La validité concerne la pertinence de l'option dans le contexte actuel pour que l'utilisateur sache s'il a le droit de la manipuler. Ex: présenter dans l'image toutes les options. La concision vise à réduire la surcharge cognitive sans perte d'information. Ex : on regroupe les options dans les menus ou des menus fantômes qui ont l'avantage de n'apparaître qu'à la demande.

#### informer pour remédier aux erreurs

Le système peut faciliter la détection, la détermination et la réparation des erreurs. La détection de l'erreur est déclenchée par le caractère immédiat du feedback. La détermination de la cause dépend de la qualité informative de l'image. Il faut choisir des messages qui contribuent à la collaboration entre l'utilisateur et le système. Le remède à l'erreur dépend des fonctions du système et notamment des fonctions Défaire et Refaire.

#### Structuration des activités

Structurer les activités consiste à organiser l'espace de travail de l'utilisateur en accord avec ses compétences.

Les "training wheels" (Caroll 83, 84, et Kay 85). On propose la méthode des "roulettes de sécurité" qui consiste à grouper les fonctions d'un système en niveaux de complexité. Cette organisation peut se pratiquer selon deux axes : le fond et la forme.

Dans la vie réelle, on fixe des roulettes de part et d'autre de la roue arrière des vélos afin que les enfants évoluent sans risque de chute. On retire les roulettes dès que l'enfant maîtrise d'équilibre.

L'utilisateur doit pouvoir réaliser dès le départ des tâches simples sans erreur. Carroll propose d'organiser les fonctions du système en niveaux de complexité croissante. Chaque niveau définit des garde-fous cognitifs.

Lorsque l'utilisateur a compris toutes les fonctions d'un niveau, il a droit au niveau supérieur, et ceci jusqu'au libre accès à toutes les fonctions du système.

### Structurer le fond

sur le fond, structurer l'interaction c'est définir une hiérarchie d'environnements. Chaque environnement est un contexte qui se caractérise en particulier par l'ensemble des fonctions accessibles.

Au sommet de la hiérarchie, l'environnement initial le plus simple possible.

Au bas de la hiérarchie les environnements les plus complexes. La difficulté est d'allier fréquence et simplicité. Il n'est pas certain que les fonctions fondamentales du système, c'est-à-dire les fonctions les plus fréquemment utilisées, soient du point de vue cognitif, les plus simples. Le concepteur doit alors user de compromis pour structurer l'espace des commandes ; cela consiste à définir la granularité des commandes.

Si le niveau d'abstraction est trop fin, l'utilisateur doit prendre en charge la composition des commandes pour accomplir les tâches élémentaires.

Si le niveau d'abstraction est trop élaboré, les tâches usuelles s'expriment directement mais l'utilisateur risque de rencontrer de graves difficultés pour l'expression des tâches qui ne se décomposent pas dans les termes du langage de commande.

### Structurer la forme

La structuration de la forme intervient partout où il est nécessaire d'organiser la complexité visuelle, de réduire la charge cognitive tout en restant informatif.

Elle concerne la présentation des fonctions, des menus et des messages.

#### Fonctions

L'image doit à la fois mettre en évidence les fonctions usuelles simples et voiler (sans toutefois les cacher) celles qui le sont moins. Lorsque la logique du système le permet, toutes doivent être assorties de valeurs par défaut raisonnables.

Dans ces conditions, l'utilisateur débutant est prompt à employer telles quelles les fonctions qui s'imposent dans l'image. Il se trouve capable de réaliser les tâches les plus courantes et ceci à moindre effort. L'expérience aidant, l'utilisateur va se définir des objectifs plus exigeants qui demanderont la recherche de fonctions adaptées, celles précisément que le système voilait.

#### Menus

les menus équilibrés permettent de meilleures performances.

Si un compromis doit être effectué entre la largeur et la profondeur, préférer la largeur. Les applications actuelles devraient pouvoir se satisfaire d'une profondeur 3.

## Messages

ils posent deux problèmes. Celui de la qualité informative et celui de la concision. La conciliation entre les deux peut se résoudre en organisant les messages en niveaux de raffinement et en donnant à l'utilisateur le moyen d'accéder au degré de détail voulu.

## Flexibilité

La flexibilité d'une interface désigne sa faculté d'ajustement aux variations de l'environnement, et notamment de l'utilisateur. Elle peut être automatique (interface adaptative ou intelligente) ou manuelle (interface adaptable). 4 éléments pour une interface adaptable, cad une interface qui est modifiable sur intervention explicite de l'utilisateur

### Réparations lexicales

Le lexique d'un système est flexible lorsqu'il est modifiable sans qu'il soit nécessaire d'altérer le logiciel source. On peut : adapter la terminologie ; corriger les incohérences ; reformuler les énoncés des messages ; redéfinir les raccourcis ; surcharger l'association fonctionnelle des touches de la souris et du clavier, etc.

### Valeurs par défaut

Les fichiers de profil permettent de personnaliser le comportement des logiciels

### Choix sur l'initiative du dialogue

Laisser l'initiative à l'utilisateur

### Représentations multiples

Associer plusieurs objets de présentation à un concept donné ; Produire des variations sur une forme de base unique.