

2 Quelques idées sur l'interface utilisateur

Un regard différent sur les logiciels que vous utilisez...

2.1 Introduction

2.1.1 Interface. Interface utilisateur

Le mot *interface*, créé dans le domaine de l'informatique, est passé dans le langage courant (et pédant) avec un sens assez vague. Explicitons la définition que nous utilisons ici :

Une interface est l'ensemble des règles communes de communication entre deux acteurs.

Dans un contexte purement technique, le mot pourra s'appliquer aux communications entre une unité centrale et un périphérique, entre deux ordinateurs, entre deux réseaux, entre deux logiciels.

L'expression *interface utilisateur* désigne quant à elle l'ensemble des règles de communication entre l'utilisateur et l'ordinateur.

L'interface utilisateur présente des spécificités liées à la présence de l'acteur humain. Comparé à la machine, ce dernier présente deux caractéristiques propres : il est *adaptable*, mais dispose de son *libre arbitre*. Cette adaptabilité a permis aux informaticiens, dans un premier temps, d'ignorer le problème. Cependant, la généralisation de l'informatique a imposé la prise en compte de cette interface dans le développement des logiciels.

2.1.2 Pourquoi se préoccupe-t-on de l'interface utilisateur ?

On ne se préoccupe guère de "l'interface utilisateur" d'une machine outil ou d'un appareil électroménager, si ce n'est sous les aspects *d'ergonomie* et de *sécurité*. Mais, contrairement à l'ordinateur, ces machines sont de simples systèmes mécanistes, dans lesquels il n'y a pas de hiatus entre le monde réel et la perception que l'on en a. Les actions à effectuer pour atteindre un objectif et l'évaluation de l'effet de ces actions sont immédiatement perceptibles.

La situation est tout autre pour la machine polyvalente qu'est l'ordinateur, qui est souvent utilisé pour simuler une situation concrète.

L'objectif que nous poursuivons ici n'est pas d'assimiler des règles ou des principes en vue de réaliser une interface utilisateur, mais plutôt d'analyser l'interaction homme-machine et surtout de mieux comprendre le maniement des ordinateurs. Une interface utilisateur bien faite a essentiellement pour but "d'apprendre l'homme à l'ordinateur"...mais restons réalistes : nous devons aussi faire une partie du chemin.

2.2 Éléments à considérer dans une modélisation de l'utilisateur

Il n'est de bonnes interfaces utilisateurs que par la prise en considération de ...l'utilisateur.

Cette phrase semblera être une évidence pour beaucoup et pourtant, beaucoup d'informaticiens pensent que l'ergonomie est évidente, ou simple affaire de bon sens. Leur premier défaut est de croire que ce qui leur paraît agréable est bien pour l'utilisateur. Leur deuxième défaut est de mieux comprendre les limitations de l'ordinateur que celles des hommes. C'est ainsi que, dans un travail de thèse intitulée "Interface utilisateur des systèmes interactifs complexes" et soutenue (avec succès) en 1992, l'utilisateur n'est pris en compte qu'au travers d'un paragraphe de 12 lignes intitulé "Psychologie cognitive et interface utilisateur" et d'un paragraphe de 17 lignes intitulé "modèle de l'utilisateur"!

2.2.1 Modéliser l'utilisateur

Pour tenter de modéliser l'utilisateur, il faut tenir compte de certaines caractéristiques du comportement humain.

L'homme perceptuel. L'utilisateur dispose d'une mémoire sensorielle qui stocke très temporairement les stimuli reçus par les organes des sens (vision et audition), d'une mémoire à court terme de capacité limitée (en moyenne 7 items ou mnèmes) dans laquelle l'oubli intervient au bout de 30 secondes en moyenne, et d'une mémoire à long terme

L'homme cognitif. L'utilisateur se crée un modèle mental du système qu'il utilise. Une bonne interface-utilisateur doit lui permettre de se créer un modèle mental correct, c'est-à-dire proche de celui du concepteur. L'interface-utilisateur doit également utiliser notre capacité à raisonner par analogie, en s'appuyant sur des *métaphores*.

L'homme social. Il faut également veiller à la cohérence des modèles mentaux des utilisateurs faisant partie d'un même groupe et participant à un même projet.

Le problème de l'interface utilisateur se pose donc à deux niveaux :

- L'interface perceptuelle : l'écran doit être clair, les menus ne doivent contenir qu'un nombre limité de commandes, ...
- L'interface conceptuelle : il faut veiller à l'adéquation entre le modèle conceptuel (imaginé par le concepteur) et le modèle que s'en fait l'utilisateur.

2.2.2 Typologies d'utilisateurs

De plus, les utilisateurs n'ont pas tous les mêmes besoins ni les mêmes stratégies d'apprentissage. Ainsi, les utilisateurs peuvent être classés selon une typologie d'expérience.

- *L'utilisateur naïf.* Il doit pouvoir utiliser un système sans apprentissage préalable. C'est, typiquement, l'utilisateur du Minitel ou de bornes de guidage. Pour cet usage, l'interface doit être totalement auto-descriptive, évitant toute nécessité d'apprentissage.
- *L'utilisateur occasionnel.* Il s'agit, typiquement, du cadre dans une entreprise, qui n'est pas spécialisé dans une tâche donnée, mais doit pouvoir utiliser un grand nombre d'applications différentes. Ce type d'utilisateur recherchera une interface intuitive, une cohérence entre les interfaces proposées par les différentes applications qu'il utilise.
- *L'utilisateur professionnel intensif.* Il ne s'agit pas ici de l'informaticien professionnel, mais plutôt de l'employé de bureau qui utilise quotidiennement une application donnée, telle qu'un traitement de textes ou un tableur. Ce dernier préférera une interface proposant des raccourcis-clavier évitant le passage par des boîtes de dialogue successives ou le maniement de la souris.

En raison de l'évolution rapide des matériels et des logiciels, tout utilisateur se retrouve fréquemment dans une situation d'apprentissage de l'outil utilisé. Or, les utilisateurs diffèrent également par les stratégies d'apprentissage qu'ils mettent en œuvre. On peut ainsi construire une typologie cognitive, basée sur des facteurs cognitifs et des facteurs de personnalité.

- *Apprentissage par opérations.* Les utilisateurs de ce type vont privilégier un apprentissage par l'action, de type essais/erreurs. Une interface adaptée doit leur proposer des messages d'erreur et de guidage et disposer d'une bonne *intégrité*, évitant la destruction involontaire de leur travail.
- *Apprentissage par la réflexion.* Ces utilisateurs vont souhaiter lire, comprendre et assimiler le manuel avant tout essai sur machine. Il importe donc de leur fournir une documentation claire et complète de l'outil mis à leur disposition.
- *Apprentissage par la connaissance.* Ce type d'utilisateur construit sa connaissance en s'appuyant sur ses savoirs antérieurs. Pour ce type d'utilisateur, une métaphore bien conduite sera la bienvenue.

2.3 Quelques éléments utilisés dans l'élaboration de modèles d'interfaces utilisateur

Tout le problème vient du hiatus qui apparaît entre la logique de l'utilisateur qui s'exprime en termes d'intentions et de buts de haut niveau d'une part et d'autre part des limites de la machine qui imposent à l'utilisateur de passer, pour atteindre ses buts, par une succession de tâches de bas niveau.

Cette remarque fondamentale explique notamment pourquoi les documentations et guides d'utilisation des logiciels sont parfaitement illisibles, et ne peuvent guère être utilisés qu'en interagissant avec la machine.

Pour tenter de remédier à ce problème, on tente de remonter le niveau de dialogue par les méthodes suivantes.

2.3.1 Utilisation de métaphores

Ainsi, les interfaces graphiques et les fenêtres constituent une *métaphore du bureau* sur lequel sont empilés des documents, des dossiers, à coté duquel se trouve une corbeille...

Les icônes symbolisent des objets du monde réel : documents, lettres, imprimante, tandis que les "boutons" et autres "boîtes à cocher" se comportent comme leurs équivalents mécaniques rencontrés sur les objets technologiques "grand public" : télévision, radio, voiture...

Ainsi, l'interface proposée sur la figure 1 est celle d'un programme de lecture de CD audio. Elle réalise une métaphore parfaite des touches de commande d'un lecteur de CD. Il faut toutefois noter que le logiciel est une émulation de l'objet concret correspondant. On pourra cependant la comparer à celle de la figure 2 qui, du point de vue informatique, est strictement "utilitaire", bien que les deux logiciels aient les mêmes fonctionnalités.

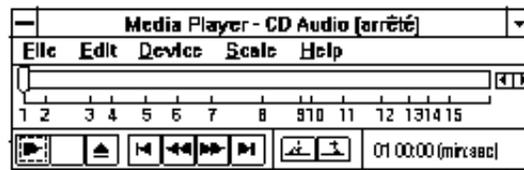


Figure 1: Lecture de CD audio. Logiciel 1

Une bonne métaphore doit évidemment être fondée sur une base de connaissances antérieures de l'utilisateur. Un traitement de textes, par exemple, est en partie fondé sur une métaphore de la machine à écrire ; mais cette métaphore constitue-t-elle une aide pour l'utilisateur qui n'a jamais utilisé la version mécanique de l'objet ?



Figure 2: Lecture de CD audio. Logiciel 2

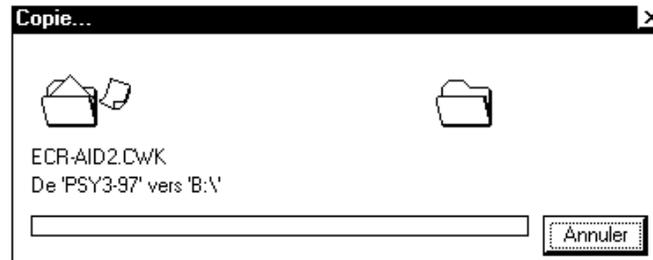


Figure 3: Copie de fichiers

2.3.2 Manipulation directe

On s'efforce de concevoir les commandes de façon que leur syntaxe soit en correspondance directe avec leur sémantique. Ainsi, pour copier, *mettre* un document sur une disquette, la commande consistera à faire glisser l'icône symbolisant le document sur l'icône symbolisant la disquette; pour imprimer un document, on fera glisser son icône sur celle de l'imprimante.

2.3.3 WYSIWYG

Cette expression est un acronyme de *what you see is what you get*. Il s'agit de faire en sorte qu'il y ait un couplage étroit entre les actions de l'utilisateur en manipulation directe, ce qui est vu à l'écran et le niveau des buts qui sont attendus. Exemples :

- Un traitement de textes affiche sur l'écran une image proche de ce que sera le document imprimé.
- Pour des tâches demandant un temps de traitement important, on avertit l'utilisateur par une symbolique adaptée: dans sa version la plus simple, il s'agit d'un curseur en forme de montre ou de sablier, mais on peut aussi symboliser la sortie des feuilles d'une imprimante sur une animation... Pour des tâches de copie de fichiers, on avertit l'utilisateur de l'état d'avancement de la tâche à l'aide d'un symbolisme du type "thermomètre" et une animation à l'écran lui suggère un modèle mental de la tâche qui est en train de s'accomplir (cf. figure 3).

2.3.4 Limiter la charge mentale de l'utilisateur

Il s'agit de limiter la charge mentale de l'utilisateur induite par l'exécution de tâches de bas niveau. On réalise cet objectif en fournissant à l'utilisateur des commandes pré-rédigées, des listes de choix possibles... Par exemple, la fenêtre de dialogue "Enregistrer sous" fournit la liste des formats d'enregistrement possibles, la liste des fichiers déjà existants (pour prévenir d'éventuels conflits de noms de fichiers), etc.

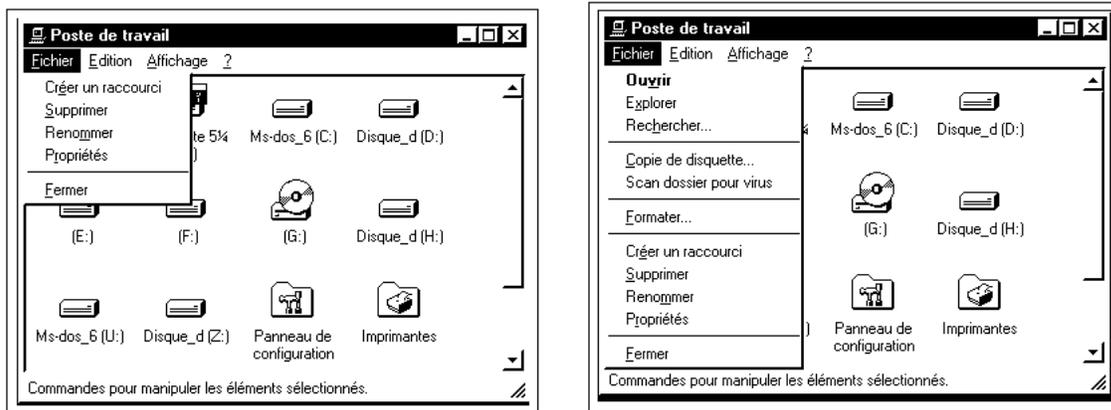


Figure 4: Explorateur de Windows NT.

2.4 Qualités des interfaces utilisateur

Les qualités attendues d'une interface utilisateur sont-elles compatibles entre elles? Une interface bien adaptée à un utilisateur débutant sera-t-elle également adaptée à un "professionnel intensif" qui recherche avant tout l'efficacité, mais doit quand même avoir droit à l'erreur...

Les paragraphes suivants proposent un certain nombre de critères, essentiellement qualitatifs, qui, pour la plupart, ne peuvent s'apprécier qu'en fonction d'un utilisateur.

2.4.1 Visibilité et transparence

Visibilité. L'interface utilisateur doit rendre visible ce qui est faisable sur le poste de travail de l'utilisateur. Ce souci de visibilité a conduit les concepteurs à proposer des métaphores, des menus déroulants rassemblant les commandes disponibles,...

Mais cette qualité semble bien absente d'un logiciel tel que l'explorateur de Windows NT, dont les menus dépendent étroitement de l'objet sélectionné (cf figure 4: à gauche, aucun objet n'est sélectionné dans la fenêtre d'arrière plan, alors qu'à droite, l'objet *Disquette A:* est sélectionné). Le débutant risque de chercher longtemps la commande permettant de formater une disquette, alors que l'acquisition de réflexes conduira l'utilisateur intensif à de fréquentes erreurs car les commandes occupent dans les menus des positions variables selon le contexte.

Transparence. L'utilisateur doit voir s'effectuer son propre travail (au sens de l'image mentale qu'il s'en fait). Par exemple, lorsque le temps de réponse est long, on doit lui permettre de voir progresser la tâche en cours. On peut aussi lui proposer un symbolisme adapté lorsqu'il accède à des ressources distantes via un réseau.

2.4.2 Intuitivité et cohérence

Intuitivité. Un système doit réagir de la façon prévue par l'utilisateur. L'intuitivité s'appuie sur le savoir antérieur des utilisateurs, l'analyse de la tâche manuelle, ... L'importance du choix et de la qualité des métaphores est ici essentielle.

Prévisibilité. A défaut d'être intuitif, un système peut au moins être prévisible, et notamment donner toujours le même type de retour pour le même type de commande.

Cohérence, consistance. Une action donnée doit toujours avoir les mêmes conséquences, quel que soit le contexte ou l'application.

Exemples :

- Un double-clic sur un objet déclenche généralement l'action la plus courante parmi celles qui mettent en jeu cet objet.
- La sélection d'un objet, suivie de l'appui sur la touche *Suppr* a généralement comme effet de supprimer cet objet.
- La touche *F1* donne généralement accès à l'aide, si elle existe.

On peut également trouver de nombreux exemples de logiciels dans lesquels cette qualité n'est pas respectée. Il s'agit notamment des systèmes dans lesquels l'effet d'une commande dépend du *mode* dans lequel on se trouve (mode *commande*, mode *saisie*, etc).

Exemples :

- Dans le module de dessin vectoriel de Clarisworks, le menu *Tout sélectionner* a un effet différent selon l'objet ou l'outil actif dans la palette.
- Sous Windows 3.1 et sous Windows 95/NT, la case de fermeture d'une fenêtre n'est pas placée au même endroit. Cette incohérence est la source de nombreuses fausses manœuvres pour les utilisateurs qui doivent travailler sur les deux systèmes.

2.4.3 Résilience et intégrité

Résilience. L'interface doit admettre des différences syntaxiques ou lexicales par rapport à la commande correcte, sans pour autant prendre de décision en cas d'ambiguïté. Ainsi, des mesures pourront aussi bien être indiquées en cm qu'en mm, un logiciel utilisant un langage de commande acceptera des abréviations, dans la mesure où celles-ci ne sont pas équivoques, etc.

Intégrité. L'intégrité est une qualité essentielle. Il s'agit de préserver les données et les résultats acquis par l'utilisateur, en évitant les conséquences d'une action intempestive de sa part. Cette qualité est obtenue notamment en introduisant des dialogues de confirmation pour les opérations destructrices, et une commande permettant d'annuler l'effet de la dernière opération réalisée.

Cohérence et intégrité sont toutes deux mises en jeu dans les situations où l'utilisateur croit exécuter une tâche donnée alors que le système entreprend une autre action. Par exemple, sous Windows NT, une action élémentaire telle que faire glisser une icône de fichier d'une zone de l'écran vers une autre zone de l'écran a un effet qui dépend à la fois des "positions relatives" de la source et de la cible et de la nature du fichier symbolisé par l'icône :

- il s'agit du même répertoire - l'action sera alors (sans doute) une duplication de fichier ;
- il s'agit de deux répertoires du même disque logique - l'action sera alors (sans doute) un déplacement de fichiers ;
- il s'agit de deux répertoires situés sur deux disques logiques différents - l'action sera alors une copie de fichiers ;
- l'icône sélectionnée est celle d'un fichier exécutable - l'action sera alors la création d'un raccourci vers ce programme.

Un manque d'intégrité, lié à un défaut de cohérence entre applications différentes, peut également être observé dans une situation telle que la suivante :

- Je travaille avec mon logiciel préféré (tableur, traitement de textes, ...); j'ai la possibilité - pratiquement illimitée sur les logiciels récents - d'annuler une commande dont le résultat ne me satisfait pas.
- Je quitte le logiciel, reviens au niveau du système d'exploitation et fais une copie de fichiers. Si je réponds *oui* à un message du type "Voulez-vous remplacer le fichier existant", il sera impossible d'annuler a posteriori l'opération effectuée, ni de retrouver la version précédente de mon fichier dans une quelconque corbeille...

Il y a des explications "informatiques" à ce genre de comportement. Dans le premier cas, on travaille en mémoire centrale, dans le second cas, sur des mémoires auxiliaires. Mais l'utilisateur en est-il conscient? Se fait-il un modèle correct (c'est-à-dire en accord avec celui du concepteur) de la machine qu'il utilise? L'émergence d'un tel modèle n'est-elle pas justement l'un des buts essentiels d'une formation à l'informatique?

2.4.4 Guidage et contrôle

Guidage. En plus de l'aide en ligne traditionnelle, on peut prévoir une aide contextuelle, des bulles d'aide... On peut aussi, de manière beaucoup plus élémentaire, guider l'utilisateur en faisant varier la forme du curseur en fonction de sa position sur l'écran.

Contrôle. Il y a là un équilibre difficile à trouver. Il faut faire en sorte que l'utilisateur se sente suffisamment contrôlé, qu'il n'ait pas à se demander en permanence s'il n'a pas commis une erreur, tout en lui laissant sentir qu'il a le contrôle sur la machine.

2.4.5 Adaptation

Adaptabilité — adaptivité. Une interface adaptable comporte des commandes permettant à l'utilisateur final de définir son propre style d'interface. Une interface adaptative s'adapte automatiquement en fonction de ce qu'elle perçoit de l'utilisateur. Mais de telles interfaces sont encore pour l'essentiel au stade de la recherche. Mais, la dernière version des logiciels de bureautique Microsoft, par exemple, propose des barres d'outils adaptatives: un bouton qui n'est jamais sollicité par l'utilisateur disparaît de la barre d'outils...

Automaticité. Il est intéressant que l'interface offre la possibilité d'automatiser certaines tâches répétitives à l'aide d'un langage de macro-commandes.

2.5 Analyse d'un exemple

Dans un système d'exploitation tel que Windows 98 ou Windows NT, un disque, ou plus généralement un support de stockage, est représenté par des icônes différentes selon qu'il s'agit d'un disque dur local, d'un volume réseau, d'un CD-ROM, d'une disquette, ... (cf. figure ci-dessous).



1) Ces icônes représentent-elles correctement les objets qu'elles symbolisent? Quels éléments guident l'intuition de l'utilisateur?

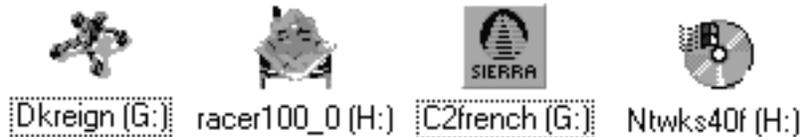
Dans un souci de visibilité, on indique à l'utilisateur les différents supports de stockage auxquels il a accès. Certains systèmes d'exploitation (par exemple, MS-DOS), ne le font pas : l'utilisateur doit alors trouver un autre moyen (essais-erreurs par exemple) pour identifier les supports disponibles. Les icônes diffèrent selon la nature du support : disquette, disque dur, CD-ROM, volume réseau. Comme toute représentation symbolique, celles-ci sont très schématisées. Cependant, deux caractéristiques des supports ont été particulièrement soulignées : le caractère amovible ou non du support, et le fait qu'il s'agisse d'un support local ou distant (volume réseau).

2) Ce choix de la part des concepteurs est-il motivé seulement par des préoccupations esthétiques ? Autrement dit, est-il important pour un utilisateur de connaître la nature des supports de stockage connectés à l'unité centrale qu'il utilise ?

Ces supports de natures différentes ont des propriétés différentes, qui jouent un rôle essentiel dans l'usage que l'on peut en faire. Les disquettes, supports amovibles, permettent facilement le transfert de fichiers d'un appareil à un autre. En revanche, il s'agit de supports lents et de très faible capacité. Le disque dur local conviendra pour des fichiers qui n'auront pas à être utilisés sur un autre appareil. Le CD-ROM, plus lent qu'un disque dur, est un support amovible. Mais il n'est accessible qu'en lecture, et ne peut servir à enregistrer les fichiers de l'utilisateur. Le volume réseau convient pour les fichiers que l'on souhaite exploiter sur différentes machines d'un même domaine. Encore faut-il disposer de droits d'accès en lecture/écriture sur ce volume.

Ce sont donc essentiellement les contraintes liées à l'utilisation d'un support d'un type donné qui sont importantes pour l'utilisateur, et non la nature proprement dite du support. Des métaphores symbolisant ces propriétés auraient pu tout aussi bien convenir. Sur un Macintosh, par exemple, la visualisation des droits de lecture/écriture est immédiate.

3) Quel jugement peut-on porter sur des icônes "personnalisées" telles que les suivantes ?



Ces icônes sont des graphismes personnalisés qui se substituent à l'icône par défaut du CD-ROM lorsque le support correspondant est inséré dans le lecteur. Elles semblent à première vue relever de la plus haute fantaisie. Elles induisent cependant chez l'utilisateur un raccourci dans le modèle qu'il se fait de sa machine. Pour l'utilisateur, par exemple, la première d'entre elles "est" le jeu Dark Reign et non le CD-ROM support des différents fichiers composant ce jeu. La pertinence du modèle ainsi suggéré est plus que discutable.

2.6 Conclusion

Il reste une qualité que nous n'avons pas citée : la *concision*. Mais cette qualité, fondamentale, semble bien absente de certains logiciels actuels...

Les problèmes posés par la réalisation d'interfaces utilisateur associent des problèmes d'*ergonomie* (commandes d'accès aisé, offrant peu de risques d'erreurs d'inattention, interface perceptuelle correcte...) et des problèmes liés à l'*apprentissage*. Ces derniers se posent souvent à deux niveaux :

- celui du logiciel et de son interface ;
- celui de la maîtrise du domaine représenté par l'application.

Ces problèmes sont l'objet de recherches. Comme dans d'autres secteurs de l'informatique, le transfert de la recherche vers la production industrielle est très rapide. Mais les principes de base qui guident la recherche sont souvent caricaturés de manière telle que les systèmes commercialisés s'en écartent notablement. Il reste encore beaucoup de chemin à parcourir vers une modélisation correcte de l'interface utilisateur...