

Les Sélections

Techniques de l'Ingénieur

DOSSIER

TECHNIQUES DE L'INGÉNIEUR

l'expertise technique et scientifique de référence

h7215

Interfaces graphiques - Fondements cognitifs

Par :

Jocelyne NANARD
Université de Montpellier-, LIRMM

Marc NANARD
*Conservatoire national des arts et métiers (CNAM) Laboratoire d'informatique, de robotique et de
microélectronique de Montpellier (LIRMM), UMR CNRS/université de Montpellier-II*

Ce dossier fait partie de la base documentaire
Représentation et traitement des documents numériques
dans le thème **Documents numériques Gestion de contenu**
et dans l'univers **Technologies de l'information**

Document délivré le **13/09/2012**

Pour le compte

7200083408 - bibliotheque universitaire universite j fourier inpg // 193.48.255.141

Pour toute question :

Service Relation Clientèle • Éditions Techniques de l'Ingénieur • 249, rue de Crimée
75019 Paris – France

par mail : infos.clients@teching.com ou au téléphone : 00 33 (0)1 53 35 20 20



Interfaces graphiques

Fondements cognitifs

par **Marc NANARD**

Conservatoire national des arts et métiers (CNAM)
Laboratoire d'informatique, de robotique et de microélectronique de Montpellier
(LIRMM), UMR CNRS/université de Montpellier-II

et **Jocelyne NANARD**

Université de Montpellier-II, LIRMM

1. Définitions et principes	H 7 215 - 2
1.1 Spécificité de l'IHM.....	— 2
1.2 Principe général de la communication.....	— 2
1.3 Vocabulaire.....	— 2
1.4 Rappels sur les relations entre langage et interaction.....	— 3
1.5 Conséquence sur la conception d'interfaces.....	— 3
2. Transfert métaphorique	— 4
2.1 Rôle et intérêt.....	— 4
2.2 Principe.....	— 4
2.3 Exemples.....	— 4
3. Styles d'interfaces, modalités et métaphore d'interaction	— 4
3.1 Langages de commande textuels.....	— 5
3.2 Désignation directe et manipulation directe.....	— 5
4. Annexe : manipulation directe, boucle de Norman	— 7
Pour en savoir plus	Doc. H 7 218

Les interfaces graphiques, en anglais GUI (« Graphical User Interface »), sont les plus répandues à l'heure actuelle. Elles supposent d'une part des représentations visuelles à l'écran, aussi bien des objets de la tâche que réalise l'utilisateur que des objets de dialogue qui servent de support à l'interaction, et d'autre part un dispositif de désignation des objets à l'écran qui sert à initier la chaîne d'interaction.

Paradoxalement, la caractéristique principale des interfaces graphiques concerne moins l'aspect graphique que la métaphore d'interaction qu'elles utilisent. Toutes reposent sur la métaphore dite d'« interaction à désignation directe », par opposition à la métaphore de la « conversation » qui caractérise les interfaces à base de langages de commande textuels tels que Shell Unix.

En effet, depuis de nombreuses années, pratiquement depuis le milieu des années 1980, tous les ordinateurs sont équipés d'écrans graphiques. Même le texte y est dessiné, ce qui permet par exemple de disposer d'un nombre quelconque de polices de caractères. L'aspect graphique d'une interface n'est donc plus un critère discriminant. Même l'éditeur Vi (prononcer « vihai ») d'Unix utilise quelques possibilités des écrans graphiques. La vraie distinction repose donc essentiellement sur la façon d'interagir avec la machine.

En tenant compte de ce fait, nous présentons dans ce premier article les mécanismes cognitifs sur lesquels sont fondées les interactions dans les interfaces graphiques, puis dans un deuxième article [H 7 216], nous décrivons les principaux interacteurs des interfaces graphiques en expliquant les raisons de leur organisation et de leur architecture. Enfin, la dernière partie [H 7 217] montre comment sont implémentées ces interfaces et quels sont les principaux environnements de développement associés.

1. Définitions et principes

Le développement récent de l'informatique pour le grand public a été rendu possible non seulement par la baisse de coût des matériels mais surtout grâce à la maturation des techniques d'interaction homme-machine (IHM). L'usage de l'ordinateur est à la portée de tous, alors qu'il semblait, jusqu'aux années 1990, être réservé à des spécialistes. Se servir d'une machine devient de plus en plus simple. Mais cette simplicité n'est pas innée et ne relève pas uniquement des progrès du matériel. Elle a été obtenue comme l'aboutissement de longs efforts expérimentaux, et au prix d'une compréhension profonde de l'ensemble des mécanismes mis en jeu dans la communication, en particulier des mécanismes cognitifs associés. C'est au concepteur d'application de tenir compte des « facteurs humains » qui sont responsables du « confort » de l'utilisateur : la communication doit demander le moins d'effort possible à l'humain, au prix d'un travail supplémentaire pour la machine. Les interfaces graphiques sont un exemple de cette approche de la communication. L'interaction homme-machine a pris une importance capitale, ainsi que les aspects cognitifs, au sein de la discipline informatique.

1.1 Spécificité de l'IHM

La spécificité de l'interaction homme-machine est de faire intervenir deux partenaires, l'homme et la machine, qui ont des caractéristiques très différentes. L'humain est intuitif, associatif, subtil et il peut prendre en considération simultanément un grand nombre d'éléments. En revanche, ses capacités de raisonnement et sa mémoire à court terme sont limitées. À l'opposé, la machine est logique, rapide, peut mettre en jeu d'énormes capacités de mémoire de travail, mais ne dispose d'aucune possibilité d'intuition, ni de faculté d'association. L'interface a un rôle d'adaptation entre eux afin de tirer le meilleur de chaque partenaire.

1.2 Principe général de la communication

Toute communication, aussi bien entre humains qu'entre machines, suppose que les deux partenaires partagent :

- un support d'échange qui permet de véhiculer les *signifiants* ;
- un ensemble de conventions qui permettent de donner sens aux signifiants et de leur associer des *signifiés*.

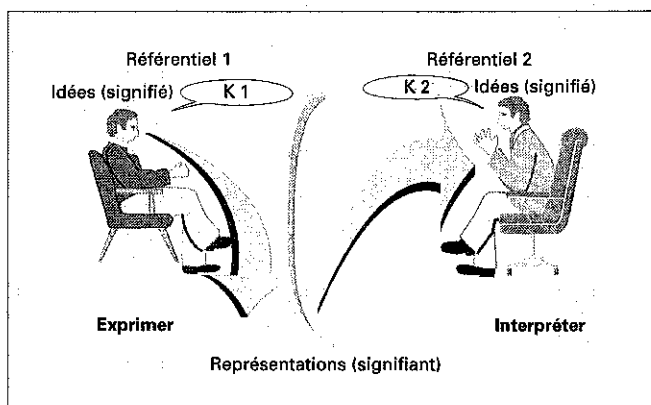


Figure 1 - Toute communication passe par l'élaboration et l'interprétation de signifiants

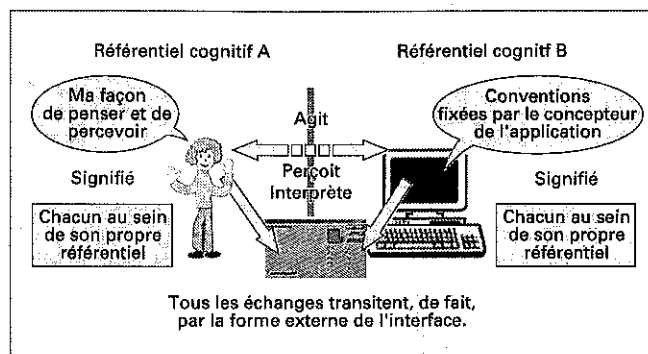


Figure 2 - Les principes de communication entre humain et ordinateur sont les mêmes qu'entre humains

L'un des partenaires, A, élabore une représentation externe du message qu'il veut communiquer. Cette élaboration se fait en fonction de conventions qui n'ont d'existence que dans le référentiel d'interprétation de A, mais sont supposées par lui être partagées par son partenaire B. Ce dernier observe la représentation élaborée et l'interprète comme un signifiant en fonction des conventions qui ont cours dans son référentiel d'interprétation propre. Il n'y a aucune raison pour que le sens initial donné par A à cette représentation soit identique à celui reconstruit par B, car les deux référentiels d'interprétation sont disjoints (figure 1).

Les problèmes de communication homme-machine (figure 2) ou de communication entre humains (figure 1) ont les mêmes causes fondamentales et relèvent, entre autres, de la **psychologie cognitive**. Pour que l'échange puisse avoir lieu avec le moins de risques de mauvaise interprétation, il faut respecter au moins l'un des principes suivants :

- A et B apprennent ensemble les conventions de représentation, ce qui s'avère en pratique quasi impossible, ou au moins extrêmement coûteux, comme la formation d'astronautes, de pilotes de chasse, etc. ;
- les conventions doivent être les plus « intuitives » possible pour que A et B les utilisent de façon naturelle sans les avoir apprises.

En quarante ans d'informatique, on est passé d'une approche basée sur l'apprentissage de conventions (se former à l'emploi de tel logiciel) à une approche basée sur des interfaces « intuitives », en particulier graphiques et multimédias. Rappelons la célèbre phrase du P.-D.G. d'Apple au début des années 1980 : « *c'est à la machine d'apprendre l'homme et non à l'homme d'apprendre la machine* ». Elle résume très bien l'évolution des interfaces homme-machine depuis cette époque.

1.3 Vocabulaire

1.3.1 Interaction et interface

Il y a souvent confusion entre les termes « interface homme-machine » et « interaction homme-machine ».

■ Interaction homme-machine

L'interaction homme-machine est l'ensemble des phénomènes physiques, logiques et cognitifs mis en jeu par un usager dans la réalisation de tâches intellectuelles avec le concours d'une machine, dans notre cas un ordinateur.

■ Interface homme-machine

L'interface homme-machine désigne l'ensemble des composants logiciels et matériels qui permet l'accomplissement de tâches avec le concours d'un ordinateur.

1.3.2 Signifiant et signifié

Les termes de « signifiant » et de « signifié » sont très souvent utilisés dans les interfaces. Par exemple, une icône en tant que dessin est un signifiant. Le sens qu'on lui accorde est le signifié. Ces mots sont souvent utilisés pour désigner les notions d'information et de sens et les opposer.

■ Signifiant (information)

Un signifiant (ou information) est une propriété de l'observation d'un phénomène physique. Il y a information s'il est possible d'observer des états distincts au sein du phénomène. L'information n'est pas une caractéristique intrinsèque du phénomène considéré, elle dépend toujours de l'observation qui en est faite.

■ Signifié (sens)

Un signifié (ou sens) est l'interprétation intellectuelle qu'un humain fait de l'observation d'un fait physique. On l'appelle aussi signifié.

Exemple : ainsi, l'image de la figure 3 représente un panneau de signalisation routière octogonal dont une tempête de neige a masqué la couleur et le message. Cette forme est, par elle seule, un *signifiant* : elle véhicule une *information*. Son sens, le *signifié*, est STOP. Il est reconnaissable par la seule forme de ce panneau. Le sens ne prend existence que dans l'interprétation intellectuelle qu'en fait l'humain, en fonction de ses conventions d'interprétation propres. Il n'est en aucun cas contenu ou présent dans le phénomène observé.

1.3.3 Référentiels d'interprétation

Une information prend sens du fait d'une convention d'interprétation. Par suite, une même information peut avoir des interprétations différentes, selon le référentiel dans lequel elle est interprétée.

On appelle **référentiel d'interprétation** l'ensemble des éléments d'une convention d'interprétation associant un sens à une information.

Dans la pratique des logiciels, un même signifiant peut avoir des interprétations différentes selon les usagers, lorsque ceux-ci se trouvent dans des référentiels d'interprétation différents.

Exemple : l'emploi d'une icône qui a un sens dans un logiciel et un autre sens dans un autre logiciel, peut causer une mauvaise interprétation si l'utilisateur avait l'habitude de l'autre logiciel.

Le signifiant, par nature, ne contient pas le signifié, le signifié est toujours reconstruit par interprétation par l'utilisateur dans son propre référentiel d'interprétation.

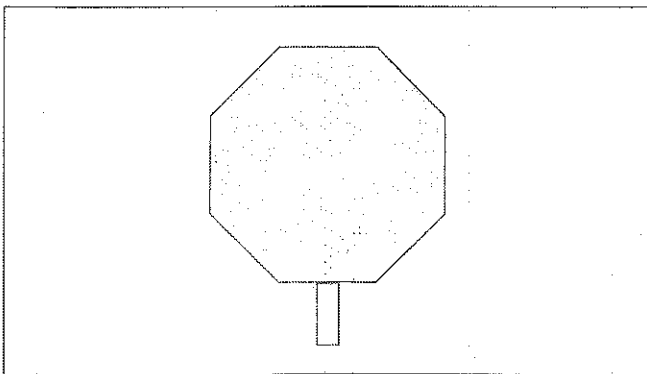


Figure 3 - Exemple de signifiant

1.4 Rappels sur les relations entre langage et interaction

Toute interaction repose sur l'emploi d'un langage, qui n'est pas nécessairement textuel. Plusieurs niveaux sont à prendre en considération dans un langage, dont en particulier :

- le niveau lexical ;
- le niveau syntaxique ;
- le niveau sémantique.

■ Niveau lexical

On appelle niveau lexical l'aspect d'un langage qui concerne la représentation des symboles formant l'alphabet du langage (ou lexèmes).

Exemple : les applications informatiques sont de plus en plus « localisées », c'est-à-dire que leurs messages sont convertis dans la langue du pays d'usage. Le remplacement de la forme externe des symboles du langage, par exemple « yes » par « oui », ne modifie que l'aspect lexical du langage, en aucun cas sa syntaxe ou sa sémantique. De même, en modifiant l'aspect des icônes, on ne modifie que l'aspect lexical du langage.

Les langages ne sont pas nécessairement textuels. Il existe des langages gestuels, par exemple le langage des signes. En informatique, l'interaction avec les interfaces graphiques repose sur un langage gestuel, c'est-à-dire dont les lexèmes sont des gestes. Par exemple, l'action de cliquer puis de déplacer la souris en maintenant le bouton pressé est un lexème connu sous le nom *drag*.

■ Niveau syntaxique

On appelle niveau syntaxique l'aspect d'un langage qui concerne les règles d'assemblage des symboles formant l'alphabet du langage pour en construire les mots valides.

Exemple : avec une interface graphique, pour détruire un ou plusieurs fichiers, « on les sélectionne, on clique sur la sélection, on fait glisser jusqu'à la poubelle ». L'enchaînement des opérations à réaliser est défini par la syntaxe de l'opération *détruire*.

La localisation des applications induit parfois des modifications dans le niveau syntaxique, par exemple en modifiant l'ordre de certains lexèmes. Ainsi, la date est représentée selon la syntaxe « jj-mm-aa » en français et selon la syntaxe « mm-dd-yy » en convention anglo-saxonne.

■ Niveau sémantique

On appelle niveau sémantique l'aspect d'un langage qui concerne la signification des mots valides, en particulier qui explicite comment construire le sens d'un mot du langage à partir des sens des lexèmes qui le constituent et des règles syntaxiques mises en jeu.

Exemple : l'action de cliquer sur l'icône d'un fichier, puis de la déplacer et de la relâcher sur l'icône de l'imprimante a pour sémantique l'impression de ce fichier.

1.5 Conséquence sur la conception d'interfaces

De façon très générale, pour concevoir des interfaces homme-machine efficaces, il est nécessaire de comprendre comment construire des signifiants et comment choisir des conventions d'interprétation telles qu'un humain ait spontanément le maximum de chances d'élaborer le signifié adéquat lorsqu'il interagit avec ce signifiant. Les niveaux lexicaux, syntaxiques et sémantiques permettent de spécifier les différents aspects de l'interaction et de sa signification.

2. Transfert métaphorique

2.1 Rôle et intérêt

Le développement d'applications grand public impose de concevoir des interfaces « intuitives » car il n'est ni possible, ni souhaitable de devoir former les utilisateurs. Pour cela, le dialogue s'appuie sur le mécanisme dit de transfert métaphorique pour suggérer le mode d'emploi aux usagers, en s'appuyant sur des analogies à des choses connues.

Le mot métaphore vient du latin *metaphora* qui signifie transposition. En linguistique, une métaphore est définie comme le procédé de langage qui consiste en un transfert de sens par substitution analogique.

Exemple : on peut « dévorer » du regard, parler d'une « passerelle » entre ordinateurs, de l'« alimentation » d'une machine, du « pied » de l'écran, d'un « menu ». Ce sont autant de mots qui ont été choisis pour évoquer, par analogie, des fonctions homologues de fonctions connues.

2.2 Principe

Une métaphore est caractérisée par la description d'un domaine appelé *domaine cible* en termes d'un autre domaine appelé *domaine source*, afin de transposer les mécanismes connus de raisonnement du domaine source vers le domaine cible.

Exemple : « mettre un fichier à la poubelle » est une métaphore. Elle suggère qu'il arrive au fichier (dans le domaine cible) une chose similaire (transposition) à ce qui arrive à un objet de la vie courante lorsqu'il ne sert plus (domaine source).

Au domaine source est associé un ensemble d'inférences supposées connues de l'utilisateur, lui permettant de raisonner sur ce domaine. La métaphore projette l'ensemble d'inférences du domaine source sur le domaine cible. Cela induit un transfert de relations structurelles et étend les mécanismes de raisonnement du domaine source au domaine cible.

L'utilisation des métaphores en communication homme-machine s'appuie sur les constatations suivantes :

- le raisonnement par analogie est fréquent dans le raisonnement humain ;
- les métaphores jouent un rôle essentiel dans les mécanismes mentaux pour faciliter le transfert de connaissances.

Les métaphores interviennent à trois niveaux dans la communication homme-machine :

- le mode général d'interaction ;
- l'organisation spatiale de l'interface ;
- les représentations proposées à l'utilisateur et les techniques d'interaction.

2.3 Exemples

■ Techniques d'interaction

- « Pointer » : en déplaçant un objet matériel (une souris) dans le monde source, l'utilisateur désigne en fait des éléments du monde cible (par exemple, des fichiers, des items de menu, etc.).

- « Déplacer » : bien que les objets informatiques n'aient pas de localisation spatiale, il est pratique d'effectuer ce transfert métaphorique pour représenter leur structure (par exemple, un dossier est dans un autre, sur le bureau, en bas, etc.).

■ Organisation de l'espace d'interaction

- « Bureau » : l'écran est organisé comme un bureau sur lequel les entrées informationnelles sont représentées comme les objets usuels que l'on trouve sur un bureau. L'interaction avec ces objets transpose les interactions usuelles : on ouvre par exemple un dossier.

- « Immersion » : dans la réalité virtuelle, l'utilisateur est situé au centre des représentations qu'on lui propose, il devient ainsi isolé du monde réel.

- « Réalité augmentée » : certains dispositifs mélangent les objets du monde physique et les représentations informatiques, par exemple en projetant des images sur des objets réels. Ainsi, lors d'une session de travail collaboratif, l'ordinateur peut projeter sur un document réel les annotations de participants distants.

■ Métaphores générales d'interaction

- « Désigner un objet en pointant sur sa représentation à l'écran » ou « élaborer une commande concernant cet objet » constituent deux exemples de métaphores générales d'interaction. L'usage de ces métaphores a de fortes implications sur le style de l'interface. Elles sont développées dans le paragraphe 3.

3. Styles d'interfaces, modalités et métaphore d'interaction

D'un point de vue formel, toute interaction repose sur l'emploi d'un langage, c'est-à-dire de règles (conventions) définissant comment sont organisées des suites de symboles véhiculant le sens donné à un échange. Ce langage peut prendre diverses formes. Chez les humains, le langage parlé, les formes écrites du langage, incluant les formes idéographiques telles que le chinois, et les formes gestuelles telles que le langage des signes chez les sourds sont autant d'exemples de formes de langage.

Dans l'interaction homme-machine, les langages, qui combinent des expressions en entrée et en sortie, peuvent utiliser différents modes parmi lesquels :

- le mode textuel ;
- le mode graphique. Les représentations graphiques peuvent être le support à la désignation directe ;
- le mode vocal.

Bien qu'il n'y ait pas de relation stricte entre les modes d'expression utilisés par les langages et les styles d'interface, on constate que, historiquement, les premières interfaces ont utilisé des langages écrits pour expliciter les échanges entre l'utilisateur et les applications, donnant lieu aux *interfaces à langage de commande textuel*. L'arrivée des techniques graphiques et de la souris comme dispositif de pointage a permis de gérer des modalités gestuelles et graphiques. Elle a conduit au style des *interfaces à désignation directe*. Le développement des techniques de reconnaissance vocale et de reconnaissance du geste a ouvert la voie à l'usage combiné de modalités aussi bien en entrée qu'en sortie, donnant lieu à l'interaction multimodale.

Nota : par la suite, l'exposé sera centré sur les interfaces graphiques. Pour une classification détaillée des modes et une caractérisation des interfaces multimodales, le lecteur est invité à consulter [1].

De façon plus générale, le style d'une interface est lié à l'usage d'une métaphore générale d'interaction.

3.1 Langages de commande textuels

La forme d'interaction la plus simple avec une application informatique repose sur un langage de commande textuel. De tels langages, dans la quasi-totalité des cas actuels, présentent une forme grammaticale très simple et surtout beaucoup plus rigide que celle de la langue naturelle. Ce sont des langages opératifs qui se limitent à transcrire des ordres simples donnés à la machine.

■ Métaphore de la conversation

On appelle métaphore de la conversation un style d'interaction dans lequel les deux partenaires communiquent par verbalisation des actions et des données. Le langage associé a le plus souvent une représentation textuelle.

La métaphore de la conversation est toujours liée à une verbalisation, qu'elle soit écrite, orale ou autre : l'utilisateur décrit ce qu'il veut faire, on n'applique pas des opérations sur des objets. Cette métaphore n'implique pas un langage textuel, même si, majoritairement, les interfaces à langages de commande sont basées sur le texte écrit.

La métaphore de la conversation est utilisée aussi bien dans les applications informatiques spécialisées que dans certaines applications grand public simples.

Exemples : typiquement, l'accès à un moteur de recherche sur le Web utilise la métaphore de la conversation dans les *browsers* courants. L'utilisateur doit « expliquer » ce qu'il cherche, les résultats étant fournis sous forme textuelle en retour. Au contraire, en naviguant sur le Web, l'utilisateur utilise la métaphore de la désignation directe ; il n'a plus à verbaliser ce qu'il veut consulter, il le « choisit » : il agit.

L'interaction avec le Minitel repose sur la métaphore de la conversation, chacun des partenaires verbalise à son tour.

Les *formulaire*s reposent également très souvent sur la métaphore de la conversation, l'organisation du dialogue étant alors imposée par le partenaire informatique.

Les langages de commande constituent une forme simple qui accorde un minimum de liberté à l'utilisateur, mais lui demandent dans tous les cas de verbaliser ses intentions.

■ Langages à lignes de commande

C'est la forme la plus archaïque des langages de commande. Leur intérêt est le faible coût de l'implémentation qui constituait une contrainte majeure quand les machines manquaient de puissance. Ces langages reposent sur des transactions simples (ou lignes) que l'utilisateur spécifie complètement avant que la machine ne les interprète.

Dans ce type d'interaction, l'interactivité est faible car la machine ne dispose pas des éléments de la commande avant sa soumission complète.

Les *lignes de commande* sont le plus souvent organisées sous forme de phrases, constituées d'un « verbe » représentant l'action à effectuer, et de « compléments », les paramètres. Le Shell Unix utilise typiquement ce type de langage, bien qu'il ne soit ni le seul, ni le premier.

Une interaction utilisant un langage de commande textuel n'est efficace que si l'utilisateur connaît le langage. Toute difficulté rencontrée dans le souvenir des éléments d'une commande ou de ses paramètres conduit aux inconvénients suivants :

- ralentissement de l'activité, causé par le fait que l'utilisateur est contraint à se reporter à un manuel de référence ;
- risque d'erreur, dont les conséquences peuvent éventuellement être graves.

Pour ces raisons, cette forme d'échange est de plus en plus réservée à des spécialistes d'informatique, par exemple pour piloter les logiciels systèmes, ou à des utilisateurs formés spécifiquement à l'emploi de logiciels spécialisés, par exemple la réservation de billets d'avion. Elle disparaît de plus en plus des autres applications informatiques où elle tend à être remplacée par des interfaces graphiques à désignation directe.

3.2 Désignation directe et manipulation directe

L'informatique grand public, c'est-à-dire celle qui est devenue accessible aux novices, n'a existé que lorsqu'elle a pu proposer d'autres alternatives à la métaphore de la conversation. La **manipulation directe** est le style d'interface actuellement le plus répandu. Elle consiste à donner l'impression à l'utilisateur qu'il agit plus qu'il ne donne des ordres.

La forme la plus pauvre de la manipulation directe est la **désignation directe**, dans laquelle la seule action de l'utilisateur est le choix parmi des propositions (par exemple, dans les menus).

La forme la plus riche de la manipulation directe utilise la **métaphore du monde réel** [2]. Elle est censée donner à l'utilisateur l'impression d'interagir avec des objets informatiques comme il le ferait avec des objets familiers. La représentation externe (graphique) des objets informatiques est empruntée à des objets du monde réel. Force est de constater que l'interaction avec ces représentations est souvent loin de l'interaction avec des objets du monde réel.

Les difficultés de conception d'interface respectant pleinement le principe de manipulation directe conduisent actuellement à l'émergence du concept d'**interaction instrumentale** qui consiste à déplacer et à manipuler des outils (par exemple, une palette graphique) directement sur les objets (par exemple, un cercle) avec lesquels l'utilisateur interagit (par exemple, pour changer sa couleur).

3.2.1 Métaphore d'interaction à désignation directe

On appelle désignation directe un type d'interaction dans lequel les seules actions que peut faire l'utilisateur sont :

- choisir parmi des propositions que fait le système ;
- guider le système pour qu'il fasse les propositions adéquates.

Exemple : ainsi, par désignation directe, pour ouvrir un document, il suffit de cliquer sur son icône dans l'explorateur de fichiers. L'utilisateur *désigne*. Pour lancer une impression, il suffit de sélectionner cette commande dans un menu. Au contraire, dans la métaphore de la conversation, l'utilisateur aurait élaboré une commande textuelle indiquant d'une part l'action à effectuer, par exemple « imprimer », et d'autre part l'opérande de cette action, ici le chemin d'accès au fichier, puis validé cette commande.

3.2.2 Métaphore du monde réel

On appelle métaphore du monde réel la transposition suggérée à l'utilisateur du monde réel dans le monde informatique et consistant à représenter des objets informatiques sous une forme évoquant des objets du monde connu de l'utilisateur.

Exemple : le monde réel simulé n'a pas nécessairement besoin d'être notre *vrai* monde réel, mais seulement une représentation que l'utilisateur considère comme telle. Considérons, par exemple, un logiciel de dessin moléculaire en 3D stéréoscopique. L'utilisateur peut essayer, avec des gants tactiles et un mécanisme à retour d'effort, de faire entrer en contact le site actif de la molécule d'un médicament avec un autre site actif, afin de sentir l'ampleur de l'effort de déformation de la molécule nécessaire à cet assemblage. Bien que l'utilisateur n'ait jamais touché de molécules de cette façon, ce logiciel à manipulation directe lui donne l'impression d'agir comme il le ferait dans le monde réel.

Naturellement, la métaphore du monde réel s'applique à des interactions beaucoup plus simples, par exemple, mettre à l'heure un pendule en tournant les aiguilles, mettre un document « à la poubelle », ranger un document dans un dossier, etc.

3.2.3 Manipulation directe

On appelle manipulation directe un style d'interaction reposant sur la désignation directe et l'utilisation de la métaphore du monde réel engendrant une perception de l'environnement informatique comme un monde sur lequel l'utilisateur a prise.

Du point de vue informatique, la machine simule un monde sur lequel l'utilisateur peut « agir » avec un mode opératoire dans lequel des actions élémentaires simples de l'utilisateur ont un effet immédiatement visible sur l'état du système [3]. Il n'est pas nécessaire de disposer d'appareillages complexes pour effectuer des interactions. Par exemple, des déplacements de la souris, des clics, des déplacements « bouton appuyé », ou le simple fait de toucher un écran tactile sont les lexèmes les plus fréquents.

Exemple : la manipulation directe doit donner à l'utilisateur l'impression d'agir, d'observer et de contrôler ses actions. Ainsi, un logiciel de retouche de photos permet par exemple de superposer une partie de l'image d'un endroit à un autre par la métaphore du *frottage*. L'utilisateur désigne ce qu'il veut prendre, puis où il veut le poser, puis il frotte pour effectuer le transfert proportionnellement au temps passé à frotter : il a l'impression d'agir et non de verbaliser ce qui doit être fait. Comme dans la vie courante, il observe le résultat de son action pour savoir s'il doit continuer, s'arrêter ou éventuellement défaire ce qu'il vient de faire. Il y a engagement de sa part dans l'action.

L'annexe sur la manipulation directe (5 4) rappelle les mécanismes mis en jeu dans la boucle d'interaction, du côté utilisateur et du côté système. Les mécanismes cognitifs impliqués dans cette boucle ont été initialement décrits par Donald Norman [2], c'est pourquoi, elle est par la suite référencée sous le terme de *boucle de Norman*. Dans [4], la manipulation directe a été décrite non comme un style d'interface, comme cela a été introduit par Shneiderman, mais comme un paradigme d'interaction.

3.2.4 Engagement

On appelle engagement une impression d'initiative que l'utilisateur éprouve, lorsqu'il perçoit son interaction comme une action qui modifie son environnement perceptible, comme cela est le cas dans son environnement habituel. Il est « engagé » dans l'action, un peu comme un joueur de jeu vidéo réaliste se prend au jeu et finit par en oublier l'aspect virtuel.

L'impression d'engagement est une des caractéristiques que devraient procurer les interfaces à manipulation directe [4]. L'engagement s'oppose à l'effort de conceptualisation que doit faire l'utilisateur concernant l'effet des actions, lorsqu'il utilise un langage de commande écrit, car l'effet de celles-ci n'est pas immédiatement perceptible.

La désignation directe est parfois abusivement confondue par certains auteurs avec la manipulation directe, bien que les nuances importantes que nous venons de signaler distinguent ces deux notions.

3.2.5 Interaction à désignation directe

La désignation directe constituant la base des mécanismes d'interaction utilisés dans les interfaces graphiques, elle est présentée plus en détail ici.

3.2.5.1 Principe

Le principe de la désignation directe repose sur les caractéristiques suivantes :

— l'utilisateur utilise un dispositif de pointage lui permettant d'exprimer un choix parmi un ensemble de valeurs. La souris en est un

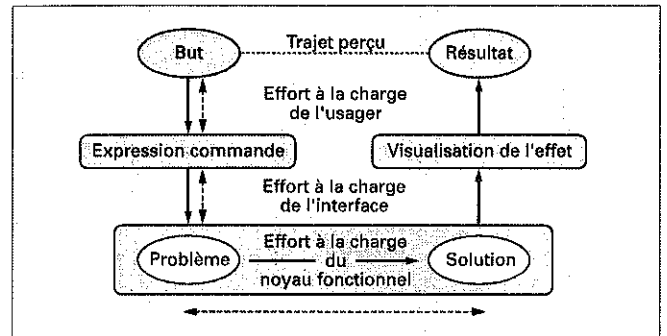


Figure 4 - Efforts demandés à un utilisateur lors d'une interaction

exemple, mais elle n'est pas le seul. Un écran tactile ou même un volant de jeu vidéo sont des dispositifs de pointage. À l'aide d'un tel dispositif, l'utilisateur guide le système en exprimant ses choix exclusivement parmi ceux qu'il propose ;

— symétriquement, le système utilise un dispositif de *restitution* pour proposer ses choix. L'écran en est un exemple, mais il n'est pas le seul. Par exemple, la proposition des choix peut se faire de façon vocale : de nombreux serveurs vocaux utilisent la métaphore de la désignation ;

— les deux dispositifs doivent être couplés car la désignation s'effectue au moyen du premier dans le second. Ce type d'interaction nécessite un temps de cycle de la boucle de Norman très faible, souvent inférieur au dixième de seconde pour le cycle élémentaire. L'écran, en particulier graphique, est le périphérique le plus adapté à ce type d'interaction.

Du point de vue de l'interaction (figure 4), la frontière entre le monde de l'utilisateur et celui de la machine a été décalée. En simulant, sur les organes de sortie un monde connu de l'utilisateur, on place la frontière virtuellement au-delà de l'écran, en réduisant la distance sémantique d'évaluation. Pareillement, en autorisant l'utilisateur à interagir avec ce qu'il voit à l'écran, on réduit la distance sémantique d'exécution (5 4).

3.2.5.2 Avantages

Ils sont surtout d'ordre suggestif :

— la désignation directe est très **intuitive**. Ce type d'interaction est mieux adapté à un novice car il ne nécessite presque pas d'apprentissage préalable ;

— la désignation directe entraîne une **impression d'action (ou engagement)** si les effets des commandes de l'utilisateur sont immédiatement rendus visibles ;

— la désignation directe augmente l'**impression de sécurité**. Le système ne propose à l'utilisateur que des actions possibles dans le contexte, d'où l'absence d'erreurs.

Le symbolisme des représentations d'actions et d'effets a pour vocation d'être suggestif et de rappeler des situations ou des objets du monde réel. Par exemple, une icône d'imprimante ou de poubelle, ou encore l'action de déplacer, sont des représentations symboliques évoquant des actions dont la dénomination dans un langage de commande écrit est parfois peu évocateur : dl, prn, mv...

L'utilisateur perçoit un « monde » constitué de propositions du système qu'il peut choisir. Cette approche concentre son activité et lui donne l'impression qu'il agit lorsqu'il exprime un choix. Le résultat du choix doit être immédiatement rendu perceptible pour l'utilisateur par le système, renforçant ainsi l'impression d'action.

Le principal avantage d'une interaction par désignation directe par rapport à une interaction à base de langage de commande est principalement psychologique : l'utilisateur a l'impression d'avoir l'initiative alors que son initiative réelle est plus réduite que dans un langage de commande. L'engagement est ici une (fausse) impression d'initiative.

La réduction effective de l'initiative de l'utilisateur permet au système de limiter les choix à des valeurs compatibles avec l'état courant du système et par suite d'éviter toute erreur de la part de l'utilisateur. Celui-ci se sent plus sécurisé. L'initiative réelle revient en fait au système. De ce fait, il peut guider implicitement l'utilisateur. Si la rhétorique de l'interaction est bien gérée, l'utilisateur ne constate pas qu'il a perdu l'initiative réelle, et préfère cette métaphore, en raison de l'impression d'engagement.

3.2.5.3 Inconvénients

L'interaction à désignation directe fait apparaître trois inconvénients principaux :

- les **temps d'interaction** par désignation directe sont en général plus longs qu'avec un langage de commande écrit ;
- les **abstractions** sont plus difficilement manipulables par désignation directe que des entités ayant une représentation naturelle ;
- les interactions par désignation directe sont **difficilement automatisables**.

La lenteur des interactions par désignation directe est due au mouvement nécessaire à la désignation, qui est plus lent qu'une simple frappe. Ainsi, atteindre l'item « copier » du menu « edit » est bien plus long que frapper la commande CTRL C. Les déplacements du pointeur en direction du point d'intérêt sont régis par la loi de Fitts (caractéristique de l'appareil sensori-moteur humain) [5]. Par contre, les interactions étant très naturelles, elles sont très vite « atomisées » (perçues par l'utilisateur comme une action simple), ce qui réduit la distance articulaire perçue. L'utilisateur perçoit très vite l'accès à une commande du menu comme une action simple et unitaire, donc plus facile à effectuer que de construire la commande correspondante.

Exemple : il est plus simple (atomique) de faire glisser un fichier jusqu'à l'icône de l'imprimante choisie, que de devoir élaborer une commande indiquant le nom de la commande "imprimer", le nom de l'imprimante et le chemin d'accès du fichier concerné.

La désignation suppose que l'entité à désigner ait une représentation. Plus une entité est abstraite ou complexe, plus il est difficile de la représenter et donc de permettre sa désignation.

Exemple : il est facile de symboliser un surlignage de couleur (instance), mais beaucoup moins le remplacement automatique d'une balise typographique <h3> par un style typographique « Titre de niveau 3 » (abstraction).

Ce manque de possibilité d'abstraction a une conséquence directe sur les possibilités d'automatisation. L'impossibilité de distinguer, lors d'une désignation, entre une variable et sa valeur rend difficile la définition de traitements génériques. Lorsqu'un usager désigne un fichier, désigne-t-il « ce fichier » ou « un fichier » ? Les langages de commande basés sur la métaphore de la conversation utilisent des macrocommandes paramétrables dans lesquelles la notion de variable permet de définir des traitements génériques. Au contraire, l'interaction par manipulation directe apparaît comme une suite d'actions effectives, mais dont aucune ne peut tenir lieu de variable. On peut faire répéter une suite d'actions « à l'identique » (par exemple, composer le numéro de téléphone d'un correspondant), mais pas lui donner un caractère générique.

4. Annexe : manipulation directe, boucle de Norman

Les aspects cognitifs de la démarche qu'effectue un humain qui interagit avec son environnement ont été modélisés par divers auteurs, en particulier par Norman en 1986 [2]. Ses travaux sont connus sous le nom de **théorie de l'action**. Ils fournissent le cadre de référence qui permet d'interpréter la plupart des phénomènes intervenant dans l'interaction homme-machine. Ce modèle introduit en particulier les notions de distances que perçoit l'utilisateur lorsqu'il interagit avec une machine : la **distance sémantique**, la **distance articulaire** et la **distance opératoire** (figure 5).

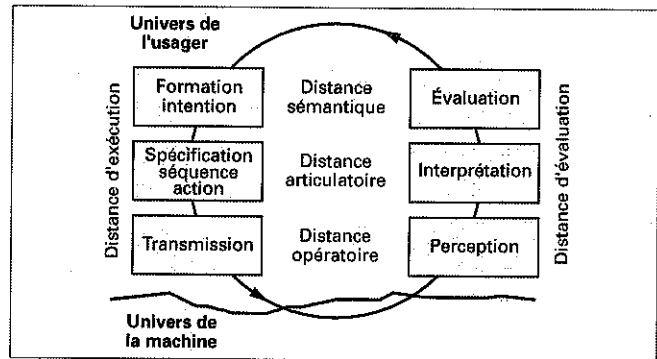


Figure 5 - Boucle de Norman

La théorie de l'action de Norman met en évidence un fait extrêmement important : l'élaboration de toute action met en jeu un cycle de rétroaction dans lequel l'humain évalue l'effet de l'action en le comparant au but escompté pour ajuster l'expression de l'action.

Exemple : pour conduire une voiture, il est indispensable de voir la route afin de corriger les déviations de trajectoire. Pour remplir une tasse sans la faire déborder, il faut observer son niveau. Pour conduire le curseur de la souris jusqu'à un bouton, il faut que ceux-ci soient visibles.

En absence de retour informationnel, ou s'il y a perturbation de celui-ci, il ne peut y avoir d'élaboration correcte d'action. Ce phénomène est observable aussi bien dans les activités de la vie quotidienne que dans les interactions avec les machines.

La réalisation d'une action intentionnelle met en jeu un cycle à quatre phases. Pour atteindre un *but*, un usager élabore une *commande*, l'*effectue*, perçoit son *effet*, et compare celui-ci au but visé. Les phases sont :

- l'élaboration de la commande ;
- l'exécution de la commande ;
- la perception de l'effet ;
- l'évaluation de l'effet.

On trouve plus de détails sur la théorie de l'action dans [2] et [4].

D'un point de vue informatique, la boucle de Norman a une très grande importance car elle permet de distinguer différentes phases dans les phénomènes qui interviennent dans une interaction.

On y distingue **deux référentiels**, propres à chacun des partenaires de l'interaction :

- celui de l'homme dans lequel l'utilisateur élabore des buts ;
- celui de la machine, capable d'effectuer des actions.

On y distingue **deux représentations** :

- celle de l'expression des commandes de l'homme vers la machine ;
- celle de l'effet des actions de la machine.

De cette organisation apparaissent les quatre phases du cycle :

- le passage du but à l'expression de la commande, à la charge de l'utilisateur. Elle s'appelle la **distance d'exécution** ;
- le passage de l'expression de la commande à l'action, à la charge de la machine. C'est le travail de l'**interface en entrée** ;
- le passage de l'action à l'expression des résultats, à la charge de la machine. C'est le travail de l'**interface en sortie** ;
- l'interprétation de l'expression des résultats et leur confrontation au but, à la charge de l'utilisateur. Elle s'appelle la **distance d'évaluation**.

On voit donc que l'effort à la charge de l'utilisateur est exprimé par la somme de la distance d'exécution et de la distance d'évaluation. Concevoir une bonne interface consiste donc à réduire ces deux distances au prix de l'augmentation du travail de l'interface en entrée et en sortie.

■ **Distance d'exécution**

On appelle distance d'exécution l'effort, à la charge de l'utilisateur, nécessaire au passage du but à l'expression de la commande.

Exemple : pour commander un objet dans un catalogue de vente par correspondance, il faut retranscrire sur un bordereau de commande le numéro d'article et son libellé figurant sur le catalogue. Cet effort est la distance d'exception de cette interaction. Sur un site Web, il suffit de cliquer sur l'icône « ajouter cet objet à mon panier ». Dans cette interface, la distance d'exécution est réduite pour l'utilisateur.

■ **Distance d'évaluation**

On appelle distance d'évaluation l'effort, à la charge de l'utilisateur, nécessaire à l'interprétation de l'expression des résultats et leur confrontation au but.

Exemples : lors de certains examens médicaux, une multitude de mesures est effectuée par une machine sur le corps du patient, par exemple une scintigraphie. Plutôt que de lister ces mesures en tant que telles, celles-ci sont retranscrites sous la forme d'une image dont les couleurs traduisent la valeur de la mesure. Elles sont donc plus facilement interprétables par le praticien, au prix d'un important travail de mise en forme des données transmises par l'interface. Cela réduit la distance d'évaluation pour l'utilisateur.

De même, dans le pilotage d'un avion, pour faciliter l'approche de la piste avec le bon angle, un dispositif délivre au pilote des signaux sonores différents selon qu'il est trop haut, trop bas, trop à droite ou trop à gauche. Le pilote n'a que faire des informations numériques sur sa trajectoire, l'interface les lui traduit sous une forme facile à évaluer. À noter le cycle de Norman : c'est à partir de l'évaluation par rapport au but (être sur la bonne trajectoire) que le pilote commande les actions nécessaires pour que l'avion suive la trajectoire voulue.

Pour affiner la théorie des interfaces, il convient d'analyser les constituants des distances d'exécution et des distances d'évaluation. Ces deux distances se décomposent de la même façon en distance sémantique, distance articuloire et distance opératoire. Ces niveaux sont à mettre en correspondance avec les niveaux lexicaux, syntaxiques et sémantiques mis en jeu dans l'implémentation des interfaces.

■ **Distance sémantique**

La distance sémantique dénote la différence entre le niveau d'abstraction offert par le langage d'interface et le niveau de conceptualisation de sa tâche par l'utilisateur.

La réduction de la distance sémantique vise à épargner un effort intellectuel à l'utilisateur en utilisant des notions qui lui sont familières, aussi bien pour lui permettre de s'exprimer, que pour lui présenter les informations.

Exemple : lorsqu'un logiciel de traitement de texte dispose de notes en bas de page, il réduit la distance sémantique en prenant en compte une abstraction du référentiel de l'utilisateur, qui, sans cela, aurait dû lui-même reconvertir cette notion en éléments de cadrage (marge inférieure, zone de pied de page, etc.), et par suite aurait eu un effort plus important à faire pour manipuler les concepts dont il a l'habitude.

La distance sémantique est souvent négligée ou méprisée, sous prétexte que chercher à la réduire revient à sous-estimer les capacités intellectuelles de l'utilisateur. C'est un faux problème car le rôle d'une interface n'est pas de tester les capacités intellectuelles mais d'épargner un effort.

■ **Distance articuloire**

La distance articuloire représente l'effort nécessaire à l'élaboration de la relation entre la signification de l'expression échangée avec le système et sa forme.

La réduction de la distance articuloire vise à épargner l'effort de planification dans l'élaboration des commandes ou dans l'interprétation des résultats. Ainsi, réduire la distance articuloire consiste, par exemple, à permettre à l'utilisateur d'exprimer par un seul geste, quelque chose qui lui aurait demandé plusieurs gestes autrement.

Exemples : l'exemple le plus célèbre est celui du mitigeur à un seul levier [4]. D'une main, on règle à la fois la température et le débit, alors que dans un mélangeur à deux robinets, il faut tourner séparément le robinet d'eau chaude et celui d'eau froide pour modifier le débit à température constante.

On remarque que cet exemple comporte aussi une réduction de la distance sémantique, car il utilise deux notions significatives pour l'utilisateur : « température » et « débit total » en remplacement de notions fonctionnelles « débit eau chaude » et « débit eau froide ». Ces deux notions sont mises en correspondance directe avec l'expression de la commande sur le plan articuloire : température par rotation, et débit par basculement.

Certaines interfaces ont une distance articuloire élevée. Ainsi, une montre dispose de boutons pour sa mise à l'heure. Pour fixer l'heure de la sonnerie (abstraction simple dans le référentiel de l'utilisateur, et supportée par la montre au titre de la distance sémantique), une succession de pressions dans un ordre et un nombre bien précis, sur divers boutons, est nécessaire. L'effort que l'utilisateur fait pour convertir l'expression mentale de sa commande – se réveiller à 6h30 – en une succession de pressions relève typiquement de la distance articuloire.

D'autres dispositifs, par exemple les réveils, ont des interfaces permettant directement de faire progresser ou régresser les heures et les minutes. Ils raccourcissent ainsi la distance articuloire.

Dans les applications informatiques, l'allongement de la distance articuloire peut induire des erreurs d'interprétation ou de commande. Le même phénomène se produit dans l'interprétation des documents, et donc de la perception des interfaces graphiques.

Exemple : les élections à la présidence des États-Unis en 2000 resteront célèbres en fournissant un très bel exemple d'usage inattendu de la distance articuloire. L'alignement entre les cases à cocher et les noms des candidats était tel qu'il pouvait facilement entraîner des erreurs d'interprétation chez des citoyens inattentifs.

■ **Distance opératoire**

La distance opératoire représente l'effort nécessaire à la transmission de la commande et à la perception du résultat. La distance opératoire relève le plus souvent de l'aspect matériel.

Exemples : la montre citée dans l'exemple relatif à la distance articuloire dispose de très petits boutons, protégés par un renforcement et surtout difficiles à enfoncer pour éviter d'être activés intempestivement. Pour les actionner, l'utilisateur doit se procurer un petit objet pointu et dur, et appuyer fort. Il ne peut le faire que très difficilement avec ses doigts. Cela relève typiquement de la distance opératoire.

Un logiciel de dessin permet de sélectionner les objets dessinés et de les déplacer. Pour sélectionner un objet, il suffit de positionner le curseur sur un des pixels de l'objet. La sélection d'objets filiformes devient très délicate sur le plan opératoire. Par exemple, il est très difficile de cliquer exactement sur un pixel dessiné d'un cercle. C'est pour cela que les logiciels de dessin implémentent cette sélection plutôt par le calcul de distance minimum entre le point visé et un point de l'objet. L'utilisateur ne voit pas la différence et peut sélectionner facilement un cercle ou une courbe. C'est un exemple pratique de réduction de la distance opératoire.