

En 1945, Vannevar Bush, dans son célèbre article "As We May Think" décrivait un système électronique (imaginaire) permettant le stockage et la recherche d'informations, et inventait les concepts de l'hypertexte : navigation, indexation, annotation. Au début des années 60, alors que les systèmes à temps partagé font leurs débuts, Ivan Sutherland crée le système SketchPad (1963), qui utilise comme écran un oscilloscope. SketchPad est un outil de dessin qui permet de créer des schémas par instanciation de modèles prédéfinis (à l'instar du modèle classe instance des langages à objets - Simula date de la même époque). SketchPad permet la construction de structures hiérarchiques et utilise des contraintes pour définir les objets. L'interaction utilise un crayon optique. A la fin des années 60, Douglas Engelbart développe le système NLS-Augment. En 1968, il présente une démonstration à San Francisco où l'on voit un système de traitement de texte hiérarchique, multimédia (texte et graphique) et hypertexte. C'est un système permettant le travail coopératif : on peut partager des fichiers, annoter des documents, utiliser une messagerie, et même un système de visio-conférence avec partage d'écrans informatiques et télépointeurs. Engelbart a inventé la souris (et a déposé un brevet qui ne lui rapportera rien puisque les souris qui seront commercialisées utiliseront un principe différent de son brevet), ainsi que le clavier à une main, les fenêtres, etc.

En 1969, Alan Kay, alors chez Xerox et aujourd'hui Apple Fellow, présente sa vision d'un ordinateur de poche qui préfigure les "PDAs" (Personal Digital Assistants). Dans les années 70 et 80 (en encore aujourd'hui), les laboratoires de Xerox PARC (Xerox Palo Alto Research Center) sont un creuset sans équivalent pour le développement de systèmes interactifs novateurs. Pendant les années 70, le projet Smalltalk, qui lancera les langages à objets, conduit à la réalisation de la première station de travail à écran graphique et souris, l'Alto, qui dispose également d'un réseau local (Ethernet a été inventé aussi à Xerox PARC). L'interface contient déjà des fenêtres, menus, barres de défilement et utilise la sélection directe à la souris. En 1981, une nouvelle machine, le Star, utilise la métaphore du bureau avec des icônes représentant le système de fichiers et un certain nombre de commandes génériques. Le Star invente aussi la présentation WYSIWYG (What You See Is What You Get) : les documents apparaissent à l'écran sous la même forme que lorsqu'ils sont imprimés. Le Star sera un échec commercial (comme beaucoup de tentatives de Xerox de commercialiser ses inventions dans le domaine de l'informatique) mais inspirera Apple : en 1983, l'Apple Lisa est une copie du Star mais, trop cher et trop étriqué, c'est aussi un échec commercial. En 1984, le Macintosh, une évolution du Lisa vendu à un prix plus bas est un succès commercial sans précédent. Il faudra plus de 10 ans pour que Microsoft intègre à son interface graphique Windows des fonctionnalités comme l'interaction iconique ou le cliquer-tirer.

Dans la seconde moitié des années 80, au Massachusetts Institute of Technology (MIT) à Boston, un grand projet financé notamment par Digital Equipment Corp. (DEC), IBM et Tektronix a pour objectif de fournir à tous les étudiants du campus des postes de travail accessibles librement. C'est le projet Athena, qui donnera naissance notamment au X Window System, qui deviendra l'interface graphique standard des stations Unix. A la fin des années 80, on voit apparaître les premiers systèmes de réalité virtuelle qui immergent l'utilisateur dans un monde synthétique. Au début des années 90, les physiciens du CERN inventent un système hypertexte qui s'appellera plus tard le World Wide Web et envahira la planète.

Malgré cette profusion d'inventions, qu'il faudrait compléter par tous les travaux, moins spectaculaires mais indispensables, de nombreux chercheurs et praticiens, il faut reconnaître que les systèmes interactifs actuels sont souvent loin des attentes des utilisateurs. Des études ont montré que l'utilisation de l'informatique n'a pas augmenté la productivité des "cols blancs", et il n'est pas besoin d'étude pour constater que l'informatique n'a pas réduit la quantité de papier que l'on est amené à manipuler, mais au contraire l'a fait augmenter. C'est la preuve que la conception de systèmes interactifs de qualité est encore un domaine mal maîtrisé et souvent sous-estimé. On pourrait espérer que cette situation soit en train de changer et que la qualité de l'interface devienne un argument de vente de plus en plus important, grâce au développement de la concurrence sur un marché en pleine expansion. Malheureusement, lorsque la norme est la médiocrité et lorsque le marché est dominé par des acteurs qui ont la mainmise sur les technologies-clés de l'interaction, on ne peut être trop optimiste.

Historique des styles d'interaction

On peut classer la majeure partie des systèmes interactifs en trois catégories selon le style d'interaction qu'ils emploient.

Le terme de "style d'interaction" recouvre l'ensemble des méthodes, techniques et règles d'interaction qui sont employées dans un système.

1. Systèmes conversationnels

Historiquement, le style conversationnel est le plus ancien : il correspond à un mode d'interaction langagière inspiré de la conversation entre humains. Le langage humain a également inspiré les langages de programmation, il n'est donc pas surprenant de le retrouver dans l'interaction.

Dans un système conversationnel, les commandes sont fournies en tapant un texte au clavier, généralement une ligne terminée par un retour-chariot. Un feed-back permet à l'utilisateur de voir ce qu'il tape (et de le corriger), mais la commande n'est exécutée que lors de sa validation. A ce moment-là, la commande est interprétée et transformée en opérations sur les objets internes du système. Le système fournit

alors une réponse qui est soit un message d'erreur, soit le résultat, éventuellement vide, de la commande. L'utilisateur peut alors entrer une nouvelle commande, etc. Les shells de commandes comme ceux d'Unix sont des exemples de systèmes conversationnels.

L'inconvénient principal d'un système conversationnel est que l'utilisateur doit apprendre le langage de commande et que les réponses du système sont parfois difficiles à interpréter. Il en résulte une distance d'exécution et une distance d'évaluation importantes (voir la Théorie de l'Action). Par contre, pour des utilisateurs experts, ce mode d'interaction peut être plus efficace que tout autre, notamment par la possibilité d'étendre le système en programmant ses propres commandes (comme par exemple les shell-scripts sous Unix). D'autre part, les systèmes conversationnels impliquent un séquençement strict entre actions de l'utilisateur et réponses du système (à l'exception du fait que l'utilisateur peut interrompre le système si celui-ci tarde à répondre). Il est donc difficile de mener plusieurs tâches simultanément, alors que c'est une activité courante dans la vie quotidienne.

2. Menus, formulaires

Les systèmes à base de menus et de formulaires sont une extension du modèle conversationnel destinée à faciliter l'interaction en rendant le langage de commande explicite. Ainsi, la sélection d'une commande dans un menu nécessite de reconnaître cette commande dans une liste, alors qu'une interface conversationnelle oblige à se souvenir du nom de la commande. Les menus hiérarchiques permettent d'accéder à un grand nombre de commandes, au prix de commandes de navigation qui permettent de passer d'un menu à l'autre.

La syntaxe disponible avec un simple système de menus est très primitive. C'est pourquoi ce style d'interaction est complété par l'usage de formulaires, similaires dans leur principe aux formulaires papier : lorsque le formulaire lui est présenté à l'écran, l'utilisateur doit remplir ses champs (dans un ordre quelconque), puis valider le formulaire. C'est la validation qui déclenche l'exécution d'une commande par le système, et sa réponse sous forme d'un message d'erreur, d'un nouveau formulaire, d'un menu ou d'un document. Dans ce dernier cas, des commandes de navigation permettent de parcourir un document qui ne peut être totalement affiché à l'écran (dans le cas d'interfaces graphiques). La plupart des services Minitel sont des systèmes à base de menus et de formulaires, de même que beaucoup d'applications de gestion (au sens large du terme : voir par exemple les systèmes de réservation d'avions ou de trains).

Les systèmes à base de menus et de formulaires imposent, comme les systèmes conversationnels, un dialogue strict entre l'utilisateur et le système. Cependant, ces systèmes assistent l'utilisateur en lui fournissant des informations permettant de guider ses actions : liste des commandes disponibles dans les menus, noms des

champs dans les formulaires, etc. De plus l'interaction avec un formulaire n'impose pas d'ordre de remplissage et en cas d'erreur, il n'est en général pas nécessaire de tout re-saisir. Le fait de présenter les objets informatiques sous une forme graphique et la possibilité d'agir directement sur ces objets (cliquer dans un champ pour le remplir, sur un bouton OK pour valider) renforce la sensation d'engagement de l'utilisateur. L'inconvénient de ce style d'interaction est qu'il ne peut convenir qu'à des tâches prédéfinies. Comme pour le style conversationnel, il n'est en général pas possible de traiter plusieurs tâches en parallèle ou même hiérarchiquement. Par exemple, si l'un des champs à saisir est un numéro de client et que l'on ne connaît que son nom, il faudrait pouvoir utiliser le formulaire d'accès à la base de données des clients, sans avoir à abandonner le formulaire en cours. Enfin, les systèmes à base de menus et de formulaires ne sont en général pas programmables par l'utilisateur : si l'on utilise souvent le même formulaire avec les mêmes valeurs de champs, il est probable que l'on doive les entrer à chaque fois.

3. Navigation (hypertexte)

Un système hypertexte permet la navigation dans un ensemble de *nœuds*, reliés entre eux par un ensemble de *liens*. Un nœud se présente typiquement comme une page à l'écran. Les liens peuvent apparaître dans le corps de la page (mots soulignés ou encadrés) et/ou dans des menus/palettes. L'interaction consiste simplement à sélectionner des liens. D'autres primitives de navigation complètent les liens : navigation avant et arrière dans les nœuds visités récemment, navigation directe vers un nœud par son nom, etc.

Des exemples de systèmes de navigation sont les navigateurs du World Wide Web et le système d'information d'Emacs. On trouve également des systèmes de navigation comme sous-systèmes d'autres systèmes interactifs, notamment les systèmes d'aide en ligne.

4. Manipulation directe

L'avènement des stations de travail à écran graphique a amené les concepteurs d'interfaces à rapprocher l'interaction avec les objets informatiques de l'interaction avec les objets physiques. Ainsi, on a cherché à présenter les documents à l'écran sous une forme comparable à leur forme imprimée, tout en permettant de modifier ces documents en agissant directement dessus. Le développement de la métaphore du bureau procède de la même logique : en représentant des objets informatiques tels que fichiers et dossiers par des images évocatrices (les icônes), et en permettant l'action sur ces icônes par l'intermédiaire de la souris, on espère rendre les commandes possibles suffisamment intuitives pour éviter à l'utilisateur un apprentissage long et fastidieux.

Le terme de manipulation directe a été inventé par Ben Shneiderman en 1983, inspiré par certains logiciels de l'époque, notamment les premiers tableurs et les

jeux vidéo. Shneiderman caractérise les applications à manipulation directe par 4 principes :

- affichage permanent des objets d'intérêt (les objets d'intérêt sont les objets qui ont un sens pour l'utilisateur, qui appartiennent à son modèle mental : par exemple les mots, lignes et paragraphes d'un texte) ;
- action directe (via un dispositif de désignation comme la souris) sur les objets d'intérêt plutôt que syntaxe complexe ;
- actions incrémentales et réversibles dont l'effet sur les objets d'intérêt est immédiatement visible ;
- structuration de l'interface en couches afin de faciliter l'apprentissage.

Comparé aux styles d'interaction précédents, la manipulation directe permet de traiter des tâches non prédéfinies, en particulier les tâches créatives : édition de texte, de dessins, de schémas, de partitions, etc. L'invention des systèmes de multi-fenêtrage, qui a accompagné celle de la manipulation directe, permet de mener de front plusieurs tâches, de façon entrelacée. La manipulation directe permet non seulement d'interagir avec plusieurs applications, mais aussi de transférer des données entre applications (par cliquer-tirer ou couper-coller, notamment). L'inconvénient principal de la manipulation directe est de ne pas permettre aisément la programmation par l'utilisateur : si l'on doit copier tous les fichiers qui ont un nom se terminant par ".c", il est probable que l'on doive faire cette opération "à la main".

Un autre problème, qui n'est pas inhérent au style d'interaction lui-même, est que de nombreuses applications n'exploitent pas la manipulation directe correctement : beaucoup de commandes sont déclenchées par la combinaison d'une commande de menu et d'une (ou plusieurs) boîtes de dialogue. Il n'y a alors ni action directe sur les objets d'intérêt (la manipulation se fait dans la boîte de dialogue), ni effet immédiat de l'action (il faut valider pour voir l'effet). On peut alors parler de manipulation indirecte (proche en réalité du style menus/formulaires), et non de manipulation directe. Nous verrons que la raison de cet état de fait tient en grande partie aux outils de développement aujourd'hui disponibles, qui simplifient le développement des parties de l'interface utilisant la manipulation indirecte sans apporter de solution satisfaisante pour mettre en oeuvre les principes de la manipulation directe.

Deux catégories très répandues d'interfaces à manipulation directe sont l'édition de documents et l'interaction iconique.

4.1. Edition de documents

Le principe du WYSIWYG (What You See Is What You Get) est de présenter un document à l'écran sous une forme la plus proche possible de sa forme imprimée :

texte, schéma, partition musicale, etc. Ainsi, dans un système de traitement de texte WYSIWIG, l'équation

$$d = \sqrt{b^2 - 4ac}$$

apparaît telle quelle, tandis que dans un système non WYSIWYG, elle aurait par exemple la forme

$$\$d \text{ \equal } \sqrt{b^2 - 4 a c}\$$$

L'interaction avec un document WYSIWYG utilise en général

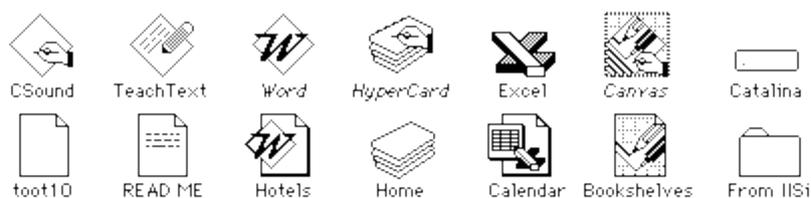
- des techniques de manipulation directe ;
- des techniques de manipulation "indirecte" (boîtes de dialogues, menus, palettes)

Pour mettre en oeuvre les techniques de manipulation directe, il peut être nécessaire de violer le principe de WYSIWYG. Par exemple, dans un logiciel comme Word, on peut visualiser le texte en mode pleine page ou en mode défilement, on peut afficher les caractères normalement invisibles (tabulations, espaces, retour-chariot, etc.).

Cependant, la notion de WYSIWYG est indépendante de celle de manipulation directe. On peut faire de la manipulation directe de représentations non WYSIWYG, et on peut éditer un document WYSIWYG par un langage de commande...

4.2. Interaction iconique

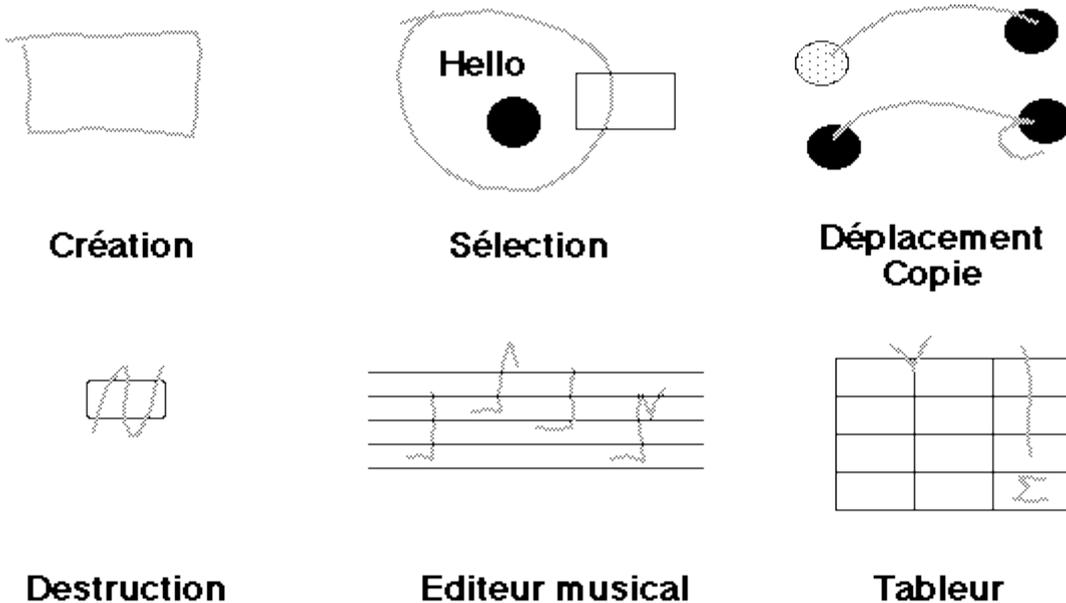
Un icône est une représentation graphique d'un objet ou d'un concept. Les icônes doivent être conçus pour être aisément reconnaissables, mémorisables, et différenciables les uns des autres. La stratégie la plus efficace consiste à créer des familles d'icônes, comme par exemple sur le Macintosh : la forme générale de l'icône indique s'il s'agit d'un document ou d'une application, le contenu indique de quel type de document il s'agit (texte, graphique...) :



L'interaction iconique consiste à effectuer des commandes par manipulation directe des icônes (sélection, déplacement, éventuellement vers un icône) et les techniques usuelles de manipulation indirecte (menus, boîtes de dialogues).

5. Interaction par reconnaissance de traces

Ce style d'interaction consiste à reconnaître les mouvements du périphérique de localisation par rapport à un vocabulaire gestuel prédéfini. Un geste définit une commande et éventuellement certains de ses paramètres (taille, portée du geste, point de départ et/ou d'arrivée). La position du geste détermine son objet. L'interaction est donc concise, mais invisible : il est difficile de savoir quel est le vocabulaire.



Exemple : de nombreux PDAs (Newton, Pilot, etc.)

6. Autres styles

D'autres styles d'interaction existent ou émergent. On peut notamment citer la réalité virtuelle, la réalité augmentée, l'interaction multimodale, le collecticiel.

La réalité virtuelle immerge l'utilisateur dans un monde synthétique : tout ce que l'utilisateur perçoit (vue, ouïe et, idéalement, toucher) est produit par le système et inversement toutes ses actions (actions physiques comme parole) sont interprétées par le système. Issue des travaux des militaires dans le domaine des simulateurs de vols, la réalité virtuelle trouve aujourd'hui des applications dans des domaines très spécialisés : médecine, téléopération en milieu hostile, etc.

La réalité augmentée procède d'une approche opposée à la réalité virtuelle : au lieu d'immerger l'utilisateur dans un monde synthétique, on intègre l'interface du système informatique dans les objets et l'environnement quotidiens. Par exemple, une feuille de papier peut servir d'interface : une caméra ou une tablette graphique permet de capturer ce que l'on écrit dessus, et un projecteur permet d'afficher des informations sur la feuille. La frontière entre les mondes physique et informatique s'efface, rendant l'interface invisible.

L'interaction multimodale cherche à exploiter au mieux les capacités d'action et de perception de l'être humain, en s'intéressant notamment aux combinaisons entre modes d'interaction. Par exemple, en entrée la combinaison du geste et de la parole est plus puissante que chacune des modalités prises indépendamment : je peux dire "mets ça ici" en désignant du doigt le "ça" et le "ici" alors qu'il est difficile d'exprimer la même commande seulement par la voix ou par le geste. En sortie, la combinaison de l'image et du son permet par exemple d'attirer l'attention (grâce au son) et de communiquer une information sous forme graphique.

Le collecticiel est une extension de la notion de système interactif dans laquelle plusieurs utilisateurs interagissent avec le système afin de communiquer entre eux via le système. Un exemple d'application collecticielle est un éditeur partagé qui permet à plusieurs personnes de modifier, simultanément, le même document, et de voir, en temps réel, les modifications effectuées par les autres personnes. L'intérêt croissant pour le collecticiel vient de la constatation que la plupart de nos activités sont des activités de groupe, alors que l'usage de l'ordinateur est essentiellement individuel.