



HAL
open science

La conception ergonomique d'interfaces : problemes de methode

Dominique Scapin, P. Reynard, A. Pollier

► **To cite this version:**

Dominique Scapin, P. Reynard, A. Pollier. La conception ergonomique d'interfaces : problemes de methode. RR-0957, INRIA. 1988. inria-00075602

HAL Id: inria-00075602

<https://inria.hal.science/inria-00075602v1>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCOUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tel: (1) 39 63 55 11

Coll. Inf
Rapports de Recherche

N° 957

Programme 8

LA CONCEPTION ERGONOMIQUE D'INTERFACES : PROBLEMES DE METHODE

Dominique L. SCAPIN
Pascale REYNARD
Agnès POLLIER

Décembre 1988



**La conception ergonomique d'interfaces :
problèmes de méthode¹**

**Human factors in the design of user interfaces :
methodological problems**

Dominique. L. Scapin, Pascale Reynard et Agnès Pollier

...scap@archie-ergo.inria.fr

Programme 8

Décembre 1988

¹ Ce rapport a été effectué dans le cadre de la convention INRIA/CNET n° 88 78016 00 790 92 45
LAA/TSS/DAP.



Résumé

Ce rapport tente d'identifier un certain nombre de problèmes méthodologiques posés par la conception ergonomique d'interfaces utilisateurs. Du plus général au plus spécifique sont évoqués des principes généraux de conception, des pré-requis de la conception, certaines théorisations cognitives, des modèles abstraits à niveaux, et enfin quelques étapes générales de conception. Ce rapport conclut sur quelques besoins de recherche majeurs dans le domaine de la conception ergonomique d'interfaces utilisateurs, tout d'abord en ce qui concerne l'organisation des recommandations ergonomiques, puis en ce qui concerne la modélisation, en particulier la modélisation du processus de conception.

Mots clefs : ergonomie du logiciel - interfaces utilisateurs - conception et évaluation d'interfaces - modèles de l'interaction homme-ordinateur - recommandations - processus de conception

Abstract

This paper attempts to identify various methodological problems with the design of user interfaces, from a human factors point of view. From the most general to the most specific, this review focuses on general design principles, on pre-requisites to the design, on some cognitive theorizations, on abstract layered models and finally on a few general design steps. This paper concludes on some major existing research needs in the field of user interface design, first in the area of human factors guidelines organization, and second in the area of modelling, particularly concerning the design process.

Key-words : software psychology - user interfaces - design and evaluation of interfaces - human-computer interaction models - recommendations - design process

1. Introduction

L'ergonomie de l'informatique a beaucoup évolué lors de cette dernière décennie. C'est devenu une discipline importante et reconnue, pour des raisons économiques (baisse des coûts, élargissement du parc des micro-ordinateurs), des raisons sociologiques (les utilisateurs ne sont plus seulement des professionnels de l'informatique) et enfin des raisons technologiques (de plus en plus d'informatique interactive, distinction entre couche application et couche interface, outils de prototypage). En particulier, l'ergonomie des logiciels a enregistré une forte progression dans la recherche, l'industrie et bien d'autres secteurs d'activité. Les interfaces utilisateurs représentent maintenant une part très importante des travaux en ergonomie, ce qui explique qu'on soit passé en quelques années à un corpus de résultats extrêmement volumineux.

Scapin (1986) relevait un certain nombre de déficiences caractérisant la conception des logiciels (e.g., manque de connaissances préalables des tâches et des utilisateurs, manque de méthodologie de conception, orientation fonctionnelle plutôt qu'opérationnelle, etc.). Ce constat était attribué à l'attitude "technologique" (la technologie résoud tout) et au recours trop fréquent à l'introspection ou au dit "bon sens" chez les concepteurs. On peut estimer que ces critiques sont moins actuelles en raison d'un plus grand intérêt, chez les informaticiens, et aussi une meilleure connaissance, voire une participation aux travaux en ergonomie. Les concepteurs deviennent conscients des problèmes et s'informent (e.g., Coutaz, 1988).

Malheureusement, le transfert des connaissances d'un domaine à un autre (de l'ergonomie vers l'informatique) pose un certain nombre de difficultés. On peut donc s'interroger sur la manière dont sont fournis les résultats en ergonomie, en particulier quant à leur applicabilité dans le cadre des processus d'évaluation et de conception d'interfaces, et réfléchir sur les moyens d'améliorer ce transfert.

Dans un rapport récent (Scapin, 1988), selon une orientation qui mettait l'accent sur les besoins en matière d'analyse conceptuelle préalable des tâches, ont été considérés les méthodes informatiques, les études sur les concepteurs, les modèles de l'interaction homme-ordinateur et les recommandations. Ici, on s'intéresse plutôt aux étapes générales d'une démarche de conception, ce qui constitue un complément de l'étude précédente.

Pour faire le point sur les problèmes et les perspectives de recherche actuels en ergonomie des interfaces, quelques aspects de la littérature ont été examinés du

point de vue des approches méthodologiques proposées quant à la conception d'interfaces. Seront donc évoquées des notions allant du plus général au plus spécifique : des principes, des pré-requis de la conception, des théorisations cognitives, des modèles à niveaux, et quelques grandes étapes de conception, pour conclure sur les besoins actuels dans le domaine en matière d'organisation des recommandations et de modélisation, en particulier pour le processus de conception.

2. Principes généraux de conception ergonomique d'interfaces

Les enseignements "pratiques" de Norman (1986) reflètent un peu l'état de l'art du domaine. En effet, celui-ci suggère :

- de créer une science de la conception d'interface centrée utilisateur, ce qui dénote qu'elle n'en est pas encore une;
- de prendre les choses au sérieux, ce qui est une évidence quand on sait les ressources affectées à l'étude des premiers téléphones (Dreyfuss, 1955) ou à l'interface du Xerox Star (Smith et al, 1982), précurseur de l'interface si louangée du MacIntosh;
- de séparer la conception du système et la conception de l'interface : c'est ce qui se passe actuellement du point de vue des nouveaux logiciels d'application, ce qui permet de modifier une partie du système sans affecter l'interface et inversement;
- et de commencer par les besoins de l'utilisateur, lesquels restent à définir.

Ce dont la littérature ergonomique abonde, ce sont de grands principes pour une approche ergonomique des interfaces. Quelques uns d'entre eux seront mentionnés ici. Ainsi Norman (1983) propose :

- d'essayer de montrer au concepteur que les utilisateurs des systèmes ont des besoins spécifiques;
- de fournir des méthodes et des guides;
- de fournir des outils logiciels pour la conception d'interfaces, de séparer la conception de l'interface de la tâche de programmation,

ou bien suggère :

- que la spécification de l'interface soit parmi les premières tâches de la conception du système;
- que le feed-back soit non ambigu pour que l'utilisateur sache ce que le système a interprété des entrées précédemment effectuées;

- qu'une confirmation des commandes ayant des effets potentiellement catastrophiques soit prévue;
- que l'utilisateur ait la possibilité de toujours contrôler le dialogue;
- que les noms de variables, de commandes, d'objets, de fichiers soient suggestifs de ce qu'ils représentent;
- que l'utilisateur puisse définir un langage de commande abrégé.

On voit donc qu'il y a à la fois suggestion d'étapes et de quelques recommandations importantes.

D'autres principes ont été décrits, par exemple : connaître l'utilisateur, minimiser les mémorisations, optimiser les opérations, prévoir les erreurs (Hansen, 1971); prévoir toutes les actions possibles de l'utilisateur, minimiser les besoins d'apprentissage, fournir un grand nombre de diagnostics explicites et une forte assistance en temps réel, prévoir des raccourcis pour les plus expérimentés, autoriser les utilisateurs à s'exprimer de différentes manières (Wasserman, 1973); connaître la population des utilisateurs, répondre de façon consistante et claire, représenter les données comme l'utilisateur se les représente, adapter les termes au vocabulaire de l'utilisateur, fournir le plus possible des moyens à l'utilisateur de corriger ses propres erreurs, encourager l'utilisateur ! (Pew and Rollins, 1975); faciliter la découverte, le feedback immédiat; utiliser le modèle de l'utilisateur, consistance et uniformité, éviter l'a-causalité (sorties non comprises), fournir des aides tutorielles, fournir des choix possibles, faire en sorte que le système soit sous le contrôle de l'utilisateur (Gaines and Facey, 1975); simplicité, réponses rapides et significatives, flexibilité, stabilité (pas de voie sans issue), protéger des erreurs graves (écraser un fichier), auto-documentation (les commandes parlent d'elles mêmes, les help sont faciles d'accès), modifiable par l'utilisateur (Cheriton, 1976); et il en existe bien d'autres...

Tous ces points sont très utiles, et permettent de signaler aux concepteurs les aspects les plus saillants de l'interface, d'un point de vue ergonomique. Cependant, ceci est insuffisant comme outils d'ingénierie.

Il est absolument nécessaire de donner à ces principes des définitions opératoires, tendre vers des concepts bien définis et applicables, en particulier des critères explicites de conception.

3. Pré-requis de la conception

Deux domaines sont en général cités comme primordiaux pour toute conception : les utilisateurs et la tâche. En ce qui concerne la tâche, il semble que l'on ne soit pas encore en mesure de fournir des méthodes faisant le lien entre les caractéristiques des tâches et la conception d'interfaces. Dans Scapin (1988, op. cit.), un certain nombre de lacunes dans la prise en compte des aspects tâches ont été identifiés, et une ébauche de modèle a été proposée. Une perspective intéressante concerne par ailleurs la définition d'une taxonomie des activités dans le cadre de l'interaction homme-ordinateur (Lenorowitz et al., 1984).

En ce qui concerne les utilisateurs, deux constats sont souvent faits :

- une seule interface ne peut être adaptée à la fois à tous les utilisateurs potentiels. Pour qu'elle soit adaptée à une population particulière, les caractéristiques de ces utilisateurs doivent être bien connues. En effet, des différences de population en termes de capacités, expérience, formation, etc., peuvent influencer la performance des utilisateurs sur des tâches informatisées. Cependant, les types d'utilisateurs sont en général seulement distingués par grandes catégories, par exemple les managers, les spécialistes, les administratifs, ou bien par grandes classes de connaissances, e.g., connaissances sur les systèmes informatiques en général, expériences avec un système particulier, connaissances du domaine de la tâche :
- les caractéristiques des utilisateurs vont évoluer à mesure que ces derniers acquièrent de l'expérience avec le système, ce qui implique idéalement, qu'un système devrait être flexible, en fonction de l'expérience des utilisateurs.

Il serait bon que ces typologies soient affinées, en particulier pour que l'on puisse appliquer des règles de conception du genre : Si l'utilisateur est... expérimenté (avec attributs = temps de formation, type de formation, etc.), alors... proposer tel ou tel type de dialogue (avec attributs : possibilité d'abréviations, d'anticipation, etc.).

Une recommandation générale concerne la participation des utilisateurs. On sait que les utilisateurs potentiels seront d'autant plus enclins à accepter un système qu'ils ont été associés à l'établissement du cahier des charges et qu'ils constatent qu'un certain nombre de leurs suggestions ont été implémentées. Ainsi, les utilisateurs

but. L'émetteur (ou concepteur) envoie un message dont il pense qu'il conduira au résultat espéré une fois qu'il sera traduit par un interprète (l'utilisateur).

L'intérêt d'une telle orientation est de se représenter la conception de l'interface comme d'autres aspects de la conception. Plutôt que d'ajouter de l'ergonomie au système a posteriori, il s'agirait de faire en sorte que les objectifs de communication du système (et sa documentation) fassent partie intégrante du travail du concepteur.

Les auteurs ont montré que si, dès le départ du projet de conception, l'information sur les buts est explicitement structurée, il y aura une plus grande convergence dans l'activité de résolution des problèmes de conception. Il s'agit en somme de la mise en place d'un plan préalable, permettant de "baliser" la progression des différents cycles de conception jusqu'au produit final.

Bien que les auteurs s'intéressent plutôt aux aspects dénomination et langage (quantifieurs, connecteurs logiques, etc.), la métaphore générale de la communication semble être adéquate pour penser la conception. Cependant, comme pour la théorie de l'action, il s'agit encore d'un cadre de description, et pas d'une méthode, mais on retiendra en fait l'intérêt de l'approche orientée communication concepteur-utilisateur. C'est en effet dans ce domaine que l'ergonomie peut apporter un plus, en fournissant des outils appropriés de communication entre le concepteur et l'utilisateur.

5. Modèles à niveaux

Les modèles à niveaux (layered models) offrent des perspectives intéressantes, en particulier en structurant les couches de l'interface et en permettant d'envisager la conception à différents niveaux.

Par exemple, Foley and Van Dam (1973) font la distinction entre :

- le niveau conceptuel, qui représente la couche la plus abstraite du modèle, la façon dont la tâche est représentée dans le système;
- le niveau sémantique qui détermine l'existence et les effets des fonctions sur leurs objets correspondants;
- le niveau syntactique qui détermine la validité des séquences d'entrée/sortie selon les règles syntactiques définies (e.g., règles de séquençement des commandes, de paramétrage, etc.);

- le niveau lexical qui détermine l'apparence physique des entrées-sorties (ceci concerne aussi bien les écrans, le clavier, les dispositifs de désignation que la dénomination des commandes, les modes de codage, etc.).

L'objectif des auteurs étant de disposer d'un modèle de conception "top-down" de l'interface, il reste à inclure dans ce modèle la prise en compte d'itérations et de conception "bottom-up" et mixte.

CLG (Moran, 1981) utilise aussi des couches hiérarchiques mais plutôt avec l'objectif de modéliser la façon dont l'utilisateur se représente le système.

A ce type de modèle, Buxton (1983) a ajouté un composant pragmatique destiné à mieux représenter les aspects strictement physiques de l'interface. L'approche de Nielsen (1986) paraît particulièrement intéressante dans la mesure où, par l'utilisation de la notion de protocole virtuel (notion introduite par Gaines and Shaw, 1984, et qui considère des machines virtuelles communiquant à différents niveaux), elle illustre la correspondance entre les deux entités : le système et l'utilisateur. Par ailleurs, cet auteur prend en considération un niveau "Buts" et un niveau "Tâche".

Ces divers modèles à niveaux restent très généraux, en particulier quant au contenu des couches de type conceptuel. Leur utilité est essentiellement d'ordre classificatoire, cependant, ils représentent un cadre d'examen fertile pour la conception, qu'il convient de valider de manière pratique. Ce sera un des objectifs d'une étude en cours. Cette dernière consiste à répertorier des recommandations ergonomiques et à cette occasion de définir pour chacune d'entre elles le niveau auquel elles peuvent intervenir dans le processus de conception de l'interface.

5.1 Un modèle possible

Rapidement, le modèle à niveaux qui est actuellement envisagé est le suivant :

- utilisation de la notion de protocole virtuel (cf. Nielsen, op. cit.);
- définition du modèle comme représentant l'interaction selon différents niveaux d'abstraction dans la communication;

Trois niveaux sont envisagés : "Basique", "Fonctionnel", et "Visibilité".

Le niveau "Basique" concerne les caractéristiques de plus haut niveau de l'interface, en particulier les caractéristiques générales de la tâche, les buts des utilisateurs, les

eux-mêmes peuvent aider à la définition des caractéristiques de l'interface. Par le moyen d'interviews et de mises en situation d'essai, on peut s'assurer avec les utilisateurs les plus expérimentés que toutes leurs exigences sont prises en compte. Il est souhaitable d'obtenir le concours des utilisateurs, mais il faut cependant savoir que ces derniers ont des limitations. Certains auteurs (e.g., Gould and Lewis, 1985) proposent une façon d'inclure les utilisateurs dans le processus de conception :

- un échantillon d'utilisateurs attendus doit travailler en collaboration avec l'équipe de concepteurs à la formulation des premières étapes;
- très tôt dans la conception, les utilisateurs attendus doivent simuler et exécuter le travail réel pour que leurs performances et réactions soient mesurées;
- si après le test d'utilisation, des problèmes sont constatés, ils doivent alors être résolus.

Nombre des exigences des utilisateurs sont implicites et sont souvent au départ des difficultés de conception. Plutôt que de demander à l'utilisateur d'essayer de dire précisément et explicitement toutes ses exigences au début de la conception, une autre solution serait d'améliorer l'interaction utilisateur-concepteur, tout au long de la conception, et pour cela il faudrait des méthodes.

4. Théorisations cognitives

De nombreux travaux théoriques ont été menés sur le thème des interfaces homme-ordinateur. On a évoqué les modèles d'interaction homme-ordinateur dans Scapin (1988, op. cit.). Certaines théorisations sont quant à elles intéressantes, bien qu'elles n'aient pas donné lieu à des modèles formels. Dans cette catégorie, on évoquera des conceptualisations théoriques à but descriptif sur l'interaction homme-ordinateur, notamment la théorie de l'action de Norman (1986, op. cit.) et la conception basée sur la communication (Thomas and Carroll, 1981). Un système doit être "convivial", mais qu'est ce que cela signifie, quels en sont les paramètres, et quels en sont les facteurs ?

- Pour ce qui est des paramètres de la convivialité (i.e., l'ensemble des descripteurs possibles), un gros travail est encore nécessaire pour aboutir à une description stable et cohérente.
- Pour ce qui est des facteurs de la convivialité, certaines approches théoriques ont tenté de définir le problème, dans un cadre général (Norman, 1986, op. cit.), comme étant celui de la conception de systèmes logiques, conceptua-

lisables permettant aux utilisateurs de développer des modèles mentaux cohérents avec le modèle du concepteur. Ces modèles mentaux dérivés de l'image du système vont aider à l'apprentissage et l'interprétation des états des systèmes. Le problème général de l'interaction est posé en termes de compatibilité entre le modèle de l'utilisateur (le modèle conceptuel construit par l'utilisateur) et l'image du système (l'image physique du système à partir de laquelle les utilisateurs développent leur modèle conceptuel, ainsi que la documentation, les instructions, les manuels, les aides...), laquelle est en fait le résultat du modèle du concepteur (lequel a un certain modèle de l'utilisateur, etc.).

Norman distingue sept étapes dans cette interaction : intention, spécification de l'action, exécution, perception, interprétation et évaluation. Ces niveaux d'activité pourraient être propices à une meilleure répartition des recommandations ergonomiques. Il peut être intéressant de décrire les différentes étapes de l'activité, utiles au guidage de la conception et à l'analyse des systèmes, car un point important souvent oublié par les concepteurs, c'est qu'une interface conçue pour une étape n'est pas obligatoirement adaptée à une autre. Ce point est en interaction avec un autre fait : il n'y a pas de réponses correctes aux questions de conception, seulement des compromis. Ceci pose d'ailleurs avec grande acuité le problème de la documentation de conception, notamment celui de faire apparaître les critères de décision et les compromis.

Le problème fondamental, selon Norman, du désaccord entre les variables psychologiques et physiques doit être résolu lors de la conception du système. Pour cela, il sera nécessaire de réaliser une étude de la complexité de la tâche en différenciant les variables psychologiques des variables physiques c'est à dire de réaliser un modèle conceptuel pertinent dès les premières étapes du projet.

Les théorisations de Norman représentent une aide à la conceptualisation du problème de la conception d'interfaces, dans la mesure où elles offrent un cadre de pensée. Cependant, il ne s'agit pas encore d'une méthode applicable. Il reste beaucoup à faire pour tenter d'opérationnaliser ce genre d'idées. En particulier, il faudrait être en mesure d'affecter des règles de conception tenant compte des divers niveaux proposés par l'auteur.

Selon Thomas and Carroll (1981, op. cit.), l'interaction homme-ordinateur peut être pensée comme une véritable conversation où chaque participant a ses propres

relations existantes avec les événements extérieurs, lesquels ne sont pas traités dans les modèles actuels. Le niveau "Basique" se décrit en deux composants : un composant "Conceptuel" et un composant "Pragmatique".

Le composant "Conceptuel" représente les objectifs des utilisateurs, leurs méthodes, les informations qu'ils utilisent, etc. Ce composant, similaire du niveau conceptuel de Foley and Van Dam (op. cit.) ou de Buxton (op. cit.), nous est par ailleurs utile pour représenter (par exemple selon "MAD", Scapin, 1988, op. cit.) la façon dont les utilisateurs se représentent leur tâche, telle qu'on peut la décrire à partir d'analyses du travail. Ce niveau concerne donc la décomposition des tâches, leur fréquence, leur relations, etc. C'est aussi à ce niveau qu'il semble utile de pouvoir décrire les caractéristiques des utilisateurs (profils, expérience passée, etc.).

Le composant "Pragmatique", différent de la notion utilisée par Buxton (op.cit.), concerne la représentation des événements extérieurs (les informations arrivant à l'opérateur et celles qu'il transmet, les interactions avec d'autres systèmes et d'autres utilisateurs, etc.). On peut voir ce composant comme un ensemble de contraintes sur la conception, auxquelles peuvent s'ajouter diverses autres exigences, comme le choix de matériels ou de logiciels compatibles avec les systèmes existants, l'existence de standards, ou bien encore les critères de performance attendus, l'étendue des formations disponibles, etc. On peut considérer ce niveau basique comme celui auquel se définissent, puis se précisent les différentes exigences de conception à respecter, lesquelles constitueront l'ensemble des objectifs de conception qui seront modifiés au cours des itérations au travers des niveaux suivants. La différence essentielle de ce niveau (pour la partie conceptuelle) par rapport à CLG est par exemple le fait qu'il s'agit de se donner des objectifs de conception et non pas de décrire la tâche future.

Le niveau "Fonctionnel" représente les essais de traduction des objets et opérations définis au niveau conceptuel en des objets et fonctions système. Il est décrit selon deux composants : un composant "Sémantique" et un composant "Logique".

Le composant "Sémantique" définit les objets du système, leurs catégories, types, relations (e.g., hiérarchie) et attributs, ainsi que les fonctions qui opèrent sur ces objets. C'est à ce niveau que doit par exemple être considérée l'utilisation de métaphores.

Le composant "Logique" représente, quant à lui, la traduction en termes de système (ou de procédures non informatisées) des entrées/sorties vis-à-vis de l'environnement de la tâche, i.e., la gestion des événements extérieurs. De la même façon que le composant sémantique est une traduction du composant conceptuel, le composant logique est une traduction du composant pragmatique.

Le composant sémantique est commun à tous les modèles existants, à quelques variations près.

Le niveau "Visibilité", comme son nom l'indique représente tous les aspects de l'interface visibles pour l'utilisateur. Ce niveau se divise en deux composants, tout comme Foley and Van Dam (op. cit.), un composant syntactique et un composant lexical.

Le composant "Syntactique" concerne la "grammaire" du dialogue, i.e., des aspects tels que les modes de dialogue, la structure séquentielle des commandes et de leurs paramètres, le feed-back, les formats, méthodes, gestion des erreurs, etc.

Le composant "Lexical" concerne les aspects entrées/sorties individuels de l'utilisateur, comme les caractéristiques des icônes, les clicks de la souris, les touches fonction, le codage couleur, etc.

La documentation et les manuels, lesquels illustrent les diverses options choisies à tous ces niveaux, seront simplement considérés comme partie du composant lexical (un des aspects de l'image du système selon Norman, op. cit.).

En tout état de cause, quel que soit le modèle choisi, il est important de le valider, un des moyens nous paraissant être d'observer si, comment, et quand des considérations ergonomiques (des recommandations) peuvent être associées de manière non ambiguë aux différents niveaux du modèle.

En étant conscient qu'un des domaines pour lesquels des outils de modélisation paraît essentiel est celui de la formalisation des tâches, car c'est à partir de cette dernière que l'on peut tenter de poser les bonnes questions de conception d'interfaces, il est utile de progresser dans la définition d'un cadre théorique adéquat et pratique (i.e., donnant lieu à des méthodes) pour la conception.

Il reste par ailleurs un effort important à fournir pour coordonner ces aspects niveaux d'abstraction avec divers aspects évoqués plus loin (critères de conception,

stratégies et méthodes de conception, etc.) à l'aide d'une architecture, d'un modèle général approprié. Etant donné le caractère opportuniste de la conception, un cadre utile pourrait être celui des architectures de type "blackboard" (e.g., Nii, 1986).

6. Etapes de conception ergonomique

On peut suggérer deux étapes différentes dans le développement de systèmes : la conception et l'implémentation (production du système). Lorsque les fonctions du système sont bien structurées et clairement comprises, ces deux activités s'effectuent successivement, la conception est terminée bien avant que l'implémentation commence. Par contre si la complexité augmente (ce qui est difficilement spécifiable par avance), comme c'est souvent le cas actuellement, il serait souhaitable que les concepteurs et les utilisateurs attendus explorent ensemble les différentes possibilités du système dans un effort de convergence.

De nombreux articles suggèrent un certain nombre d'étapes pour une approche ergonomique des interfaces. A partir d'un certain nombre de ces suggestions (Bailey, 1982; Hendricks et al., 1982; Shneiderman, 1980, 1987; Scapin, 1986, op. cit.), quelques grandes étapes peuvent être identifiées:

- identifier les besoins système;
- effectuer le recueil d'informations (organisation de l'équipe de conception, obtention de la participation de la hiérarchie, passation de questionnaires aux utilisateurs de tous niveaux, interviews, vérification de la littérature, consultation des concepteurs et utilisateurs de systèmes similaires, estimation des coûts et des bénéfices attendus, préparation d'un agenda, avec les "délivrables", i.e., les productions attendues);
- déterminer les concepts généraux du système;
- déterminer les objectifs et spécifications de performance (exigences, contraintes) et les besoins et caractéristiques des utilisateurs;
- conception des aspects sémantiques (définition des buts et établissement des exigences, définition des organigrammes de tâches, organisation des opérations en unités de tâches, définition des structures de données, définition des aspects de sécurité et de confidentialité, obtention d'agrément);
- définir le système (exigences fonctionnelles, de déterminer les fonctions et les entrées/sorties);
- effectuer la conception de base (allocation des fonctions, procédures de travail, analyse de la tâche, etc.);

- conception des structures syntactiques (comparaison des alternatives d'affichage, définition de la syntaxe des fonctions, préparation des formats de réponse système, développement des modules de diagnostic d'erreur, spécification des temps de réponse, définition des procédures de help, évaluation des spécifications, test expérimentaux) pour:
 - spécifier les moyens d'entrée/sortie (claviers, modes de désignation, périphériques, etc.);
 - . déterminer le dialogue,
 - . concevoir les entrées,
 - . concevoir les sorties, pour lesquelles des étapes de conception sont détaillées;
 - faciliter la conception (conception des aides et manuels, de la formation);
 - tester;
 - développement du logiciel (modularité, maintenance, etc.);
 - plan d'implémentation (participation des utilisateurs, écriture des manuels, définition de la formation, etc.);
 - définition d'un bon environnement de communication (fournir des consultants, une boîte à suggestions, prévoir une lettre d'information, des réunions de groupe, etc.);
 - préparation de l'évolution future.

Des interactions fréquentes avec les utilisateurs potentiels sont proposées, pour les diverses étapes de conception, par exemple : proposer une solution et la communiquer avec les utilisateurs, évaluer le feed-back des utilisateurs, affiner la conception, documenter, communiquer et persuader, valider, et continuer les interactions, etc.

7. Discussion

Tout ceci est très classique et un peu général mais correspond assez bien aux découpages habituels. L'aspect le plus difficile est de savoir précisément où et quand introduire de l'ergonomie. Il est donc nécessaire de replacer les besoins en ergonomie vis à vis des différentes étapes et d'établir les liens entre les divers niveaux d'analyse.

Toutes ces étapes sont certes utiles, en particulier l'analyse des utilisateurs, l'évaluation et le test, les allers-retours, etc., mais pour ce qui est de la manière dont on peut effectuer la conception de l'interface, peu de recommandations détaillées sont fournies.

Une approche intéressante pour la conception de logiciels serait de pouvoir accéder à des modules, selon une structure hiérarchique, chaque module décrivant un aspect particulier du système. Cette structure permettrait au concepteur d'accéder aux informations à différentes profondeurs. Si on imagine donc une conception par niveau où chacun est représenté par un module indépendant, on pourra attribuer pour chacun un nombre de règles pratiques spécifiques, facilement utilisables pour les concepteurs. Selon cette perspective, il faut définir un modèle cohérent pour chaque étape de conception qui tienne compte des méthodes et des règles disponibles dans la littérature.

Trois soucis majeurs méritent d'être mentionnés : les spécifications formelles, le test et les itérations.

Le développement de spécifications formelles peut être considéré comme première étape de la conception d'interfaces. Par exemple, Malhotra et al. (1980) suggèrent :

- de meilleures méthodes pour élaborer des buts (e.g., illustrer des alternatives de conception)
- des aides à la génération de buts (e.g., aider au travail en groupe, conduire les concepteurs à explorer tous les aspects du problème, aider à l'accès de solutions de conception connues, telles que standards, algorithmes existants, etc.).

Il s'agit donc là de fournir au concepteur des moyens de mieux appréhender la conception des interfaces dès le début de la conception, en particulier dans l'identification d'un modèle conceptuel approprié. Pour mettre à disposition du concepteur des outils d'aide qui puissent le guider dans sa progression, un travail de recherche important reste à faire.

Par ailleurs, avoir l'objectif de réaliser dès le premier essai une interface satisfaisante est un but louable, mais pas toujours atteint. Les utilisateurs évoluent et toutes les questions de conception n'ont pas encore reçu de réponse(s). Le test expérimental d'interfaces est donc souvent nécessaire. Pour cela, des règles pratiques, applicables et implémentables permettant d'identifier, puis de corriger certains aspects des interfaces, sont nécessaires. Il s'agit là notamment de fournir de bons critères pour l'évaluation ergonomique.

Enfin les différents moyens de conception doivent être itératifs, il doit y avoir un cycle de conception, de test (évaluation) puis de re-conception. Ces diverses étapes doivent être répétées aussi souvent que ce sera nécessaire. Gould and Lewis (op.

cit.) notent que même si le fait de réaliser une conception itérative paraît essentielle aux concepteurs, beaucoup semblent estimer qu'une seule itération ou révision est suffisante. Bien que ce type de conception paraisse coûteuse, c'est le seul moyen, vu l'état actuel de nos connaissances sur la conception, pour assurer la réalisation de systèmes pertinents d'un point de vue ergonomique.

Ces phases d'itération et de redéfinition des buts de conception peuvent être plus ou moins complexes suivant les problèmes d'utilisation rencontrés, aussi faudrait-il des principes que les concepteurs puissent appliquer dès les premières étapes, c'est à dire des outils de simulation pour mettre en place un projet de conception d'interface, bien avant que l'on débute la mise en place du système en tant que tel. Il devrait y avoir un continuel va et vient entre les tests et les améliorations des interfaces.

En résumé, des cadres de pensée, des démarches générales, des points d'attention sont proposés dans la littérature, cependant, un effort important est nécessaire en matière de mise en oeuvre détaillée de concepts et de recommandations ergonomiques. Ce sont quelques uns de ces besoins qui sont maintenant examinés.

8. Besoins

Suite à l'examen de la littérature, on constate un certain nombre de besoins d'études. Ces besoins concernent essentiellement deux aspects : l'organisation des recommandations et les modèles.

8.1 Recommandations

Dans la littérature, on constate une extension continuelle des nouvelles formes de guidage pour la conception d'interfaces-utilisateur. Smith and Mosier (1986) citent quatre possibilités différentes :

- les standards de conception : leur nombre augmente. Des groupes ISO ou AFNOR sont créés sur les aspects ergonomie des interfaces. Cependant, on peut se demander si les standards peuvent assurer de façon consistante une bonne conception. Ils pourront en effet ne pas être pertinents pour une application particulière. De plus, les standards devront être sans cesse réadaptés en fonction des progrès de la technique, les standards actuels sont relatifs aux connaissances actuelle;

- les guides de conception : il s'agit de séries de recommandations pour la conception de logiciels d'interfaces utilisateur, avec des exemples, des explications et des commentaires adaptés, selon les sujets concernés. Les guides permettent une adaptation flexible à l'état des connaissances, alors que les standards une fois établis sont difficilement modifiables;
- les règles de conception : il s'agit de séries de spécifications déterminées pour des systèmes particuliers. De nombreuses grandes entreprises conçoivent leurs propres standards ou guides (Apple / IBM / AT&T, etc.);
- les algorithmes de conception : il s'agit de règles d'implémentation de programmes de micro-ordinateurs contrôlant la production automatique de logiciels d'interfaces-utilisateur. Ces algorithmes en sont à leur début; dans le futur, ils seront très certainement utilisés plus couramment. Cet aspect semble l'un des plus prometteurs. En tout cas, c'est un problème de recherche intéressant dans la mesure où il pose avec acuité le problème de la cohérence, de l'applicabilité, et du caractère implémentable des recommandations.

Selon le constat de Smith and Mosier (1984), ou les précautions de Scapin (op. cit.), la meilleure aide disponible actuellement pour les concepteurs, c'est de se référer à un guide général et d'en dériver des règles spécifiques à l'application désirée, car certains systèmes sont d'un certain point de vue uniques, ce qui empêche d'appliquer strictement les connaissances obtenues sur d'autres systèmes; certaines questions de conception sont plus complexes que d'autres et/ou spécifiques d'une situation particulière, ce qui nécessite le recours à des spécialistes, et/ou l'utilisation de méthodes expérimentales; nombre de décisions de conception sont le fruit de compromis entre plusieurs critères, comme entre temps d'apprentissage, fonctionnalité, temps de réponse, temps de transaction, clarté, brièveté, erreurs, etc. Quelquefois, améliorer la performance sur une dimension n'améliore pas forcément la performance selon une autre dimension (e.g., performance d'utilisation et performance d'apprentissage). Les guides ne pourront non plus remplacer une bonne analyse de la tâche permettant de déceler ses particularités et ses exigences.

Des règles concrètes manquent généralement de cohésion et sont disséminées dans la littérature sans avoir été intégrées dans une structure conceptuelle. Smith and Mosier (op. cit.) notent que de façon générale, ce sont des experts en ergonomie qui rédigent les guides et de ce fait, ils sont difficilement accessibles aux informaticiens-concepteurs de logiciels. Il faudrait donc rendre les guides plus utilisables pour les concepteurs. Combattre leur trop grande généralité en les rendant plus

spécifiques ne constitue pas la véritable solution, car plus ils seront spécifiques, moins ils seront applicables pour la variété des systèmes existants.

Alors que les auteurs utilisent les classifications habituelles de découpage des interfaces, il faut aller plus loin selon les directions suivantes, en particulier dans la perspective d'algorithmes d'aides à la conception:

- montrer les interactions et faire apparaître les compromis.
- réorganiser les recommandations, par exemple selon des critères de conception
- aboutir à des règles stables, selon un format générique.

8.1.1 Montrer les interactions et les compromis

Norman (1983, op. cit.) a cherché à analyser certains traits de systèmes, avant que ces derniers aient été développés. Il suggère qu'il n'existe pas de réponses simples aux questions de conception, seulement des compromis. Chaque application a des principes de conception forts et faibles, qui doivent être interprétés dans leur contexte propre. Un objectif intéressant serait donc de rendre les compromis explicites. Un défrichage des problèmes peut se faire à partir de la littérature, mais nécessite aussi des études détaillées de la conception ergonomique d'interfaces.

8.1.2 Réorganisation des recommandations

La réorganisation des recommandations selon des critères de conception semble intéressante. Un travail en cours consiste à examiner des recommandations ergonomiques, pour identifier et définir explicitement un certain nombre de critères de conception permettant d'une part de classer les recommandations, et d'autre part de se donner les moyens de définir des recommandations de plus haut niveau, utilisables lors du processus de conception. Une première tentative a déterminé quelques 16 critères candidats : Compatibilité, Consistance, Brièveté, Flexibilité, Feed-back immédiat, Prise en compte de la charge mentale, Actions minimales, Actions explicites, Signifiante des codes et dénominations, Contrôle utilisateur, Guidage utilisateur, Fonctions intelligentes, Expérience des utilisateurs, Groupement d'items, Distinction entre items, Prise en compte des tâches. Il conviendra bien entendu de donner des définitions opératoires à ces critères, d'en évaluer le bien-fondé, et éventuellement de leur donner une métrique d'évaluation (e.g., une mesure de consistance, de compatibilité, etc.).

8.1.3 Meilleure définition des règles

Voici quelques exemples de recommandations : "c'est une bonne chose si les utilisateurs peuvent converser avec leur ordinateur dans un langage aussi naturel que possible et non contraignant"; "le langage généré par le système doit être aussi près du langage humain que possible"; "si un algorithme de traitement pour une tâche peut-être écrit, il doit l'être et la tâche doit être accomplie par l'ordinateur"; "la tâche de l'utilisateur dans les systèmes homme-machine doit être conçue aussi simple que possible", etc.

L'examen de telles recommandations révèle que bien souvent, d'une part une recommandation peut renfermer en fait plusieurs règles; d'autre part, les prémisses et conclusions ne sont pas toujours évidentes à appliquer, car trop vagues (e.g., des utilisateurs expérimentés, des messages clairs, des codages signifiants,...). Il convient donc d'examiner les recommandations, d'évaluer si elles sont pertinentes et de donner des définitions précises de leurs prémisses et conclusions.

Par ailleurs, il sera utile de définir un format générique de règles qui tienne compte de la tâche effectuée, des caractéristiques des utilisateurs si cela est pertinent. Par exemple :

Si T (tâche) = Entrée de données

avec Ox (objet, appartenant à classe n) = clavier alphanumérique

et Oz (objet, appartenant à classe m) = non confidentiel (e.g., pas un mot de passe)

Alors,

(en utilisant C5 (critère) = Feed-back immédiat)

- Ox doit être affiché. (ici le type d'utilisateur ne s'applique pas)

Si T = Entrée de données, avec modification

avec Op = Ancienne donnée

et On = Nouvelle donnée

Alors,

(en utilisant C5 (critère) = Feed-back immédiat)

- Op et On doivent être affichés

(exception = traitement de texte)

et - une fonction Fy = confirmation doit être fournie.

Ceci est évidemment en relation avec la définition d'un modèle cohérent des objets de l'interface (e.g., caractéristiques structurelles, d'affichage et de comportement des divers objets de l'interface) et la validation de cette base d'objets.

8.2 Modélisation

Outre la modélisation des tâches qu'on a mentionné auparavant, les besoins en modélisation concernent d'une part la modélisation des objets de l'interface, et d'autre part la modélisation du processus de conception.

Les aspects modélisation des objets de l'interface ne seront pas développés ici. Etant donné que les recommandations concernent, en fonction de divers critères, le choix de certaines caractéristiques de l'interface, il est essentiel de disposer d'un descriptif cohérent des divers éléments de l'interface. Un travail est nécessaire, à partir des définitions et classifications offertes par les divers outils de prototypage existants, mais aussi à partir du décryptage des recommandations elle-mêmes. Il pourrait concerner des éléments tels que : objets de l'interface (e.g., menus, fenêtres, valeurs par défaut, soulignement pour guidage, etc.); grandes fonctions transversales (confirmation, sauvegardes, copies, etc.); sous-tâches significatives (e.g., entrée de données, modification d'entrées, définition de positions, etc.; cf. Lenorowitz, op. cit.).

Un autre aspect de la modélisation concerne le processus de conception d'interfaces. Bien qu'il soit difficile de donner une définition précise et complète de la conception, certains auteurs s'y sont essayés. Ainsi, Hammond et al. (1987) : "La conception est par nature un processus dynamique : durant le développement, de nouveaux buts prendront de l'importance et d'autres s'amenuiseront; les solutions apparemment solides à un état intermédiaire ne seront plus présentes dans le produit final." La conception débute avec la description de la tâche conceptuelle. Le système doit atteindre certains objectifs, permettant l'évaluation de solution proposée par le concepteur. Selon Mostow (1985), la conception est une importante activité humaine. Comme phénomène psychologique, c'est une sorte intéressante de résolution de problème complexe, qui est encore mal comprise. De fait, nombre d'auteurs considèrent les concepteurs comme de véritables "résolveurs" de problèmes, et considèrent la conception comme une tâche mal définie au départ, qui a un but final (le produit) qui ne pourra être satisfait qu'en fonction de certaines contraintes et exigences (notamment le cahier des charges). La tâche de conception fait aussi (de façon similaire) référence à la notion de "problèmes mal définis"

(Figeac-Letang et al., 1987; Malhotra et al., op. cit.), car le concepteur ne possède pas toutes les informations nécessaires afin de résoudre le problème (réaliser le but). Il doit fournir les informations manquantes grâce à ses connaissances antérieures qui seront source d'inférences et façonneront les prises de décisions. A partir de là, la tâche sera spécifiée et la forme du produit final définie.

Ces faits vont bien évidemment avoir des conséquences sur le processus de conception en tant que tel; toute prise de décision à un moment donné va influencer l'environnement futur du projet, c'est à dire la solution du problème.

Selon Mostow (op. cit.), le processus de conception opère à plusieurs niveaux d'abstraction, c'est à dire que le produit final peut être décrit à différents niveaux de détail et ne constituerait qu'un des espaces problèmes compris dans le processus de conception. Chaque critère et contrainte préalablement définis délimitent l'espace problème au travers duquel les concepteurs vont devoir cheminer pour arriver à la solution finale (le produit conçu). Les spécifications diverses (temps de formation, l'utilisateur, rapidité d'exécution, pourcentage de réussite et d'erreur...) et les critères fonctionnels ou techniques sont explicités dès le début de la conception et représentent le produit en termes de "propriétés désirées", alors que d'autres critères restent implicites dans la tête des concepteurs, par exemple les connaissances acquises au cours de leur expérience. Ce n'est qu'en fonction des prises de décision, au fur à mesure du processus, qu'elles seront formalisées et que le champ des possibles (e. g., solutions envisageables) se restreindra. Parmi cet ensemble, les contraintes jouent un rôle essentiel, mais dans la plupart des cas elles ne donnent pas de définition du produit à concevoir.

Ce sont ces compromis, ces niveaux d'abstraction qu'il faut identifier et valider pour la conception d'interfaces.

8.3 Etude de l'expertise en conception ergonomique d'interfaces

Pour une meilleure insertion de l'ergonomie dans le processus de conception, deux activités paraissent essentielles :

- l'une concerne l'examen de la conception de systèmes telle qu'elle est effectuée à l'aide de méthodes informatiques,
- l'autre concerne la façon dont l'ergonomie est mise en oeuvre, en particulier par les ergonomes eux-mêmes, à quel moment, selon quelles étapes, selon quels critères, etc.

Dans le premier cas, il conviendrait de suivre des projets à long terme qui utilisent des méthodes de conception informatique de manière régulière (e.g., MERISE).

Dans le second cas, il conviendrait d'étudier la conception d'interfaces en situation réelle et dans une deuxième étape, en laboratoire, pour affiner les hypothèses. C'est cette dernière démarche qui sera adoptée dans un travail en cours. Dans cette étude, il s'agit d'identifier et de modéliser les stratégies d'évaluation et de conception mises en oeuvre par les experts ergonomes lorsqu'ils évaluent des interfaces ou qu'ils participent à des projets de conception. La méthodologie utilisée a trois caractéristiques: (a) études de projets ou d'interfaces réels, (b) interviews et observations, (c) analyse des documents de conception. On s'intéresse donc aux productions verbales et écrites des concepteurs, aux connaissances utilisées, aux sources d'informations consultées, à la planification, aux stratégies, aux prises de décisions, aux compromis, en particulier afin d'illustrer les points d'ancrage de décisions d'ordre ergonomique.

9. Conclusion

Dans la littérature, l'interaction homme-ordinateur n'est pas traitée explicitement comme une variable de conception. Les modèles de conception proposés actuellement sont trop généraux, trop informels et manquent pour la plupart d'organisation pour être utilisables directement par les concepteurs. Ils sont souvent difficilement adaptables à des applications spécifiques. Dans la plupart des cas, ils énumèrent les différents aspects que doit posséder une interface pour être pertinente du point de vue de l'utilisateur, en donnant des sortes d'évaluation de la qualité ergonomique de l'interface une fois qu'elle a été mise en place. Les modèles proposés sont en particulier peu satisfaisants du point de vue de la prise en compte de l'utilisateur et des caractéristiques conceptuelles de sa tâche. Les concepteurs "ouverts" à l'introduction des facteurs humains ont des difficultés pour accéder à et choisir les bonnes recommandations et à tenir compte des interactions, sans compter que les guides généraux tendent à être trop vagues.

D'impérieux besoins d'organisation des connaissances ergonomiques se font donc sentir pour que la conception d'interfaces, si elle demeure un art, devienne aussi une discipline d'ingénierie. Il reste donc encore à développer et à valider expérimentalement des outils formels de conception d'interfaces, c'est à dire mettre en place un ensemble d'instructions ou d'heuristiques, dont l'objectif sera de guider le comportement du concepteur.

Sans exclure la possibilité, assez peu réaliste, de laisser aux utilisateurs le soin de construire ou de modifier eux-mêmes les interfaces du système, en fonction de leurs besoins, il s'agit bien d'aider le concepteur. Pour ce faire, il semble essentiel de connaître précisément les démarches des concepteurs, ce qui permettra à moyen et long termes, de mettre en place des méthodes de conception. De tels logiciels d'aide pourraient présenter des exemples d'application pour assister l'élaboration des buts et les prises de décision des concepteurs, contenir des architectures types pour les applications ayant les mêmes objectifs et les divers critères et fonctionnalités nécessaires à l'assistance d'une conception adaptée à l'utilisateur (mise en place du dialogue, des entrées, des sorties, messages d'erreur...).

En tout état de cause, il s'agit de permettre une meilleure collaboration entre divers intervenants (les concepteurs, les ergonomes et les utilisateurs) qui disposent de connaissances différentes :

- . connaissances sur la conception, la programmation et la technologie;
- . connaissances sur les futurs utilisateurs, les principes de la communication homme-machine;
- . connaissances sur la tâche devant être accomplie.

Des aides adaptées à la conception, des méthodes devront permettre au concepteur l'accès aux trois types de connaissances mentionnées ci-dessus, alors qu'il ne possède en général que les premières.

10. Bibliographie

- Bailey, R. W. (1982) Human performance engineering : a guide for system designers. Prentice Hall, Englewood Cliffs, N. J.
- Buxton, W. (1983) Lexical and pragmatic considerations of input structures. ACM SIGGRAPH : Computer Graphics 17 (1), 31-37.
- Cheriton, D. R. (1976) Man-machine interface design for time-sharing systems. Proceedings of the ACM National Conference.
- Coutaz, J. (1988) De l'ergonome à l'informaticien : pour une méthode de conception et de réalisation des systèmes interactifs. Colloque ERGO-IA, Biarritz, 4-6 Octobre, France.
- Dreyfuss, H. (1955) Designing for people, Simon and Schuster, New York.
- Figeac-Letang, C., Falzon, P., & Bisseret, A., (1987) Analyse de l'activité de conception du système de feux d'un carrefour. Rapport Technique INRIA (février 1987).

- Foley, J. D. and Van Dam, A. (1984) Fundamentals of interactive computer graphics, Addison Wesley.
- Gaines, B. R., and Facey, P. V. (1975) Some experience in interactive system development and application. Proceedings of the IEEE, 63,(6), 894-911.
- Gaines, B. R., and Shaw, M. (1984) Principles of dialog engineering. NATO Workshop on Research needs in computer-user interaction. Loughborough, September 1984.
- Gould, J.D. and Lewis, C. (1985) Designing for usability : key principles and what designers think. Communications of ACM, 28, 300-311.
- Hammond, N.V., Gardiner, M.M., Christie, B. & Marshall, C.J., (1987) Applying cognitive psychology to user-interface design. Gardiner, M.M., & Christie, B, (Eds.) 372 pages.
- Hansen, W. J. (1971) User engineering principles for interactive systems. Proceedings of the Fall Joint Computer Conference, 39, AFIPS Press, Montvale, N.J.
- Hendricks, D., Kilduff, P., Brooks, P., Marshak, R., and Doyle, B. (1982) Human engineering guidelines for management information systems. US Army Human Engineering Laboratory.
- Lenorowitz, D. R., Philips, M. D., Ardrey, R. S., and Kloster, G. V. (1984) A taxonomic approach to characterizing human-computer interfaces. in Human-Computer Interaction, G. Salvendy (Ed.), Elsevier.
- Malhotra, A., Thomas, J. C., Carroll, J. M., and Miller, L. A. (1980) Cognitive processes in design. International Journal of Man-Machine Studies, 12, 119-140.
- Moran, T.P. (1981) The Command Language Grammar : a representation for the user interface of interactive computer systems. International Journal of Man Machine Studies, 15, 3-50.
- Mostow, J., (1985) Towards better models of the design process. The AI Magazine, 44-57.
- Nielsen, J. (1986) A virtual protocol model for computer-human interaction. International Journal of Man-Machine Studies, 24, 301-312.
- Nii, H. P. (1986) Blackboard systems. Knowledge Systems Laboratory, Report No. KSL 86-18, Stanford University.
- Norman, D. A. (1983) Some observations on mental models. In D. Gentner and L. A. Stevens (eds.) Mental models. Hillsdale, N.J., Lawrence Erlbaum.
- Norman, D. A., (1983) Design principles for human-computer interfaces. In A. Janda (Ed.), Proceedings of CHI'83 : Human Factors in Computing Systems, Boston, December 1983, New-York : ACM.

- Norman, D.A. (1986) Cognitive Engineering, in : User centered system design, Norman, D.A. & Draper, S.W., (Ed.), Hillsdale, N.J. : Erlbaum. 31-61.
- Pew, R. W., and Rollins, A. M. (1975) Dialog specification procedure. BBN Report No. 3129, Cambridge, MA.
- Scapin, D. L. (1986) Guide ergonomique de conception des interfaces homme-machine. Rapport de Recherche INRIA 77.
- Scapin, D.L. (1988) Vers des Outils Formels de Description des Tâches de Bureau Orientés Conception d'Interfaces. Rapport de Recherche INRIA 893.
- Shneiderman, B. (1980) Software psychology. Winthrop, Cambridge, MA.
- Shneiderman, B. (1987) Designing the user interface : strategies for effective human-computer interaction. Addison-Wesley.
- Smith, D. C., Irby, C., Kimball, R., Verplank, W., and Harslem, E. (1982) Designing the star user interface. Byte, 74, 242-282.
- Smith, S.L., and Mosier, J. N. (1984) The user interface of computer-based information systems : a survey of current software design practice. Behaviour and Information Technology, Vol. 3 (3), 195-203.
- Smith, S.L., and Mosier, J. N. (1986) Standards versus guidelines for designing user interface software. Behaviour and Information Technology, Vol. 5, (1), 47-61.
- Thomas, J.C., and Carroll, J.M., (1981). Human factors in communication. IBM. Systems Journal, 20 (2), 236-263.
- Visser, W. (1987). Abandon d'un plan hiérarchique dans une activité de conception. In Proceedings Cognitiva'87, 18-22 Mai, Paris, France, 366-371.
- Wasserman, T. (1973) The design of idiot-proof interactive systems. Proceedings of the National Computer Conference, 42, AFIPS Press, Montvale, N.J.

