UNE TAXONOMIE POUR LES METHODES D'EVALUATION DES INTERFACES HOMME-MACHINE

Sandrine Balbo, Joëlle Coutaz

Laboratoire de Génie Informatique, IMAG
BP 53 X, 38041 Grenoble Cedex
tel. : (+33) 76.51.48.54
E-mail : balbo@imag.fr, coutaz@imag.fr

Résumé

L'évaluation des interfaces homme-machine est un problème auquel les informaticiens commencent à s'intéresser depuis peu. Cette tâche était jusqu'à présent plutôt réservée aux ergonomes. Nous présentons dans cet article une taxonomie des systèmes et techniques d'évaluation projeté selon trois axes : l'axe de l'automatisation, l'axe du savoir et l'axe de l'utilisateur. L'axe de l'automatisation repère les états d'automatisation, au sein des ordinateurs, des systèmes étudiés : pas d'automatisation, capture automatique, analyse automatique et critique automatique. L'axe du savoir indique les connaissances que possède l'ordinateur pour l'évaluation : savoir empirique, savoir sur l'interface, savoir sur l'utilisateur et savoir sur la tâche. Enfin, l'axe de l'utilisateur précise si le mécanisme d'évaluation observe l'utilisateur : évaluation avec ou sans observation de l'utilisateur.

Mots clés : Evaluation, Automatisation, Interface Homme-Machine.

1. Introduction

Les techniques du génie logiciel proposent des cycles de développement pour les logiciels où se situent les étapes d'analyse du problème, d'implémentation et de maintenance. Plus généralement, le cycle de vie classique pour le développement d'un logiciel est le suivant :

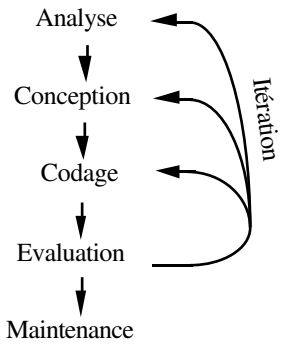


Figure 1 : cycle de développement usuel

L'ingénierie des interfaces homme-machine (IHM) propose bon nombre d'outils pour les phases d'analyse, de conception et de codage. Nous trouvons, entre autres, pour la phase d'analyse : des modèles de la tâche (GOMS [Card83], CLG [Moran81], JSD* [Lim92] ou MAD [Scapin89]), et pour les phases de conception et de codage : des générateurs d'interfaces (SIROCCO [Normand92] ou Interface Builder [Webster89]), des squelettes d'application (Serpent [Serpent89] ou MacApp [Schmucker86]) ainsi que des boîtes à outils (Motif [Motif90] ou la Toolbox du Macintosh). Pour la phase d'évaluation, il existe cependant peu de techniques et d'outils éprouvés. Dans cet article, nous cherchons à organiser l'ensemble des méthodes disponibles au moyen d'un espace de classification. L'objectif de cette taxonomie est de localiser de manière non ambigüe les techniques existantes et de situer notre propre travail [Balbo93] sur l'évaluation automatique des IHMs.

2. Notre taxonomie

Dans la littérature, les classifications pour les méthodes d'évaluation abordent très généralement la distinction entre évaluations expérimentales et évaluations à fondements théoriques (cf. [Nielsen90], [Senach90] ou [Whitefield91]). Cependant la dimension d'automatisation, au sein de l'ordinateur, des processus d'évaluation est très peu prise en compte. C'est sous cette angle nouveau que nous nous proposons d'aborder les méthodes pour l'évaluation.

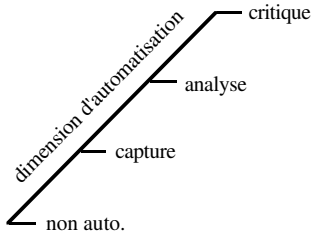
Cette présentation se scinde en deux parties : un exposé général des trois dimensions de notre taxonomie, suivi d'une présentation de quelques systèmes illustratifs que nous situons dans cette taxonomie.

2.1. Trois dimensions pour notre taxonomie

Notre taxonomie se présente selon les trois dimensions suivantes : les dimensions de l'automatisation, du savoir utilisé pour l'évaluation ainsi que de l'utilisateur.
• La dimension de l'automatisation se décompose en quatre états :

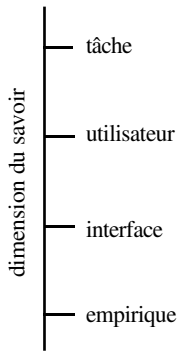
- Les méthodes ***non automatiques*** menées de bout en bout par l'évaluateur.
- Les méthodes pour lesquelles la ***capture*** des informations est automatique. Par informations nous entendons aussi bien informations visuelles, frappe au clavier, manipulation de la souris, parole, etc.

- Les méthodes qui fournissent une ***analyse*** automatique de l'interface à évaluer, c'est-à-dire qui précisent et localisent les problèmes relevés.
- Les méthodes qui fournissent automatiquement un ***critique***, c'est-à-dire qui proposent des évolutions afin d'améliorer la qualité ergonomique de l'interface.



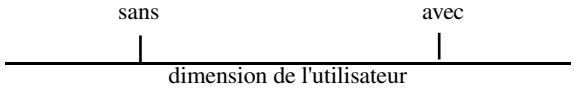
- La dimension du savoir utilisé pour l'évaluation se décompose aussi en quatre états :

  - Le savoir ***empirique***. "Empirique : qui s'appuie principalement sur l'expérience et non pas sur les données scientifiques ou rationnelles" (définition du Petit Robert). A ce niveau l'évaluateur (l'expert en évaluation) recherche directement dans l'interface, ou en observant l'utilisateur, les propriétés dont il sait, par expérience, qu'elles mettront en défaut des principes ergonomiques reconnus.
  - Le savoir sur l'***interface***, sur l'***utilisateur*** ainsi que sur la ***tâche***. Ces trois sources de savoir relèvent d'une source autre que le savoir empirique. Ce savoir peut provenir de théories ou de modèles construits au préalable comme nous l'observerons au paragraphe 2.2.



- La dimension de l'utilisateur se décompose en deux états :

  - Les évaluations ***sans*** observation de l'utilisateur.
  - Les évaluations ***avec*** observation de l'utilisateur.

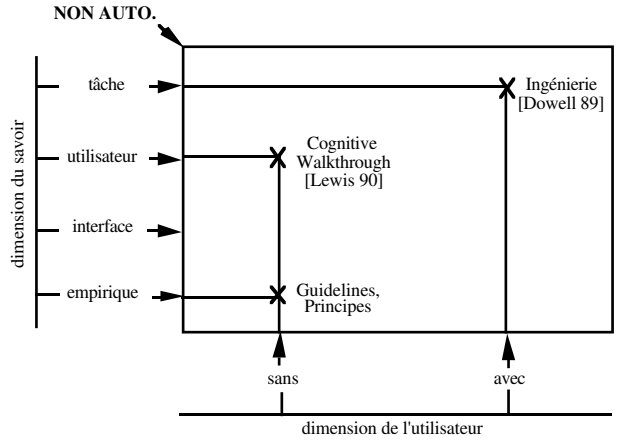


2.2. Etat de l'art

Nous présentons maintenant quelques unes des méthodes d'évaluation étudiées en les plaçant dans notre taxonomie. Cette présentation sera décomposée en fonction des plans de l'automatisation.

2.2.1. Le plan des évaluations non automatique

[Smith86] propose des recommandations construites à force d'observation et d'expérimentation. Ces recommandations semblent, à première vue, simples et faciles à suivre pour un programmeur, mais il ne faut pas sous-estimer le fait que ce rapport propose 944 conseils pour un document de 480 pages!



[Scapin86], [Barthet88] ou [Nielsen90] proposent des principes plus synthétiques que les conseils proposés par [Smith86], comme par exemple les neuf principes de base proposés par [Nielsen90].

Dialogue simple et naturel
Parler le langage de l'utilisateur
Miminiser la charge mémoire de l'utilisateur
Cohérence
Fournir des retours d'information
Fournir des sorties clairement visibles
Fournir des raccoucis (clavier)
Messages d'erreur informatifs
Prévention des erreurs

Figure 2 : neuf principes de [Nielsen90]

Ces principes donnent au concepteur une démarche à suivre, permettent de structurer son espace de travail par des règles. D'autre part, l'utilisation de tels principes n'est possible qu'à condition, par exemple, de savoir en quoi un message d'erreur sera plus informatif qu'un autre, ou être capable de construire le modèle mental de l'utilisateur. Ces principes demandent une expérience dans le domaine de l'ergonomie et des sciences cognitives bien plus importante que pour l'application des conseils proposés par [Smith86].

Ces méthodes pour l'évaluation requièrent une expérience préalable importante dans la mesure où elles fournissent des conseils ou des principes, mais pas de moyens pratiques pour mettre en œuvre ces conseils. Nous présentons maintenant quelques systèmes qui font un premier pas dans ce sens en fournissant une ébauche de support à l'évaluateur.

[Whiteside88] propose comme support pour l'évaluation une mesure objective de la performance de l'utilisateur. La mesure effectuée est la suivante : **S = (1/T) PC** où

S = Taux d'achèvement de la tâche,
T = Temps passé sur la tâche par l'utilisateur,
P = Pourcentage de la tâche effectué par l'utilisateur,
C = Temps nécessaire à un expert pour exécuter la tâche.

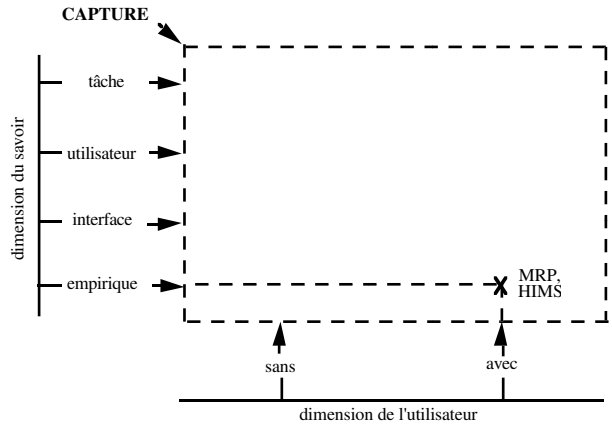
Ces mesures sont enrichies par des enregistrements audio-visuels, des interviews et des questionnaires dont l'évaluateur soustraira une évaluation qualitative qui viendra appuyer l'évaluation quantitative précédente. Cette évaluation reste pour une grande part le produit d'une évaluation artisanale au sens de [Long89].

En conclusion, [Polier91] aussi bien que [Nielsen90] énoncent qu'au moins trois ergonomes sont nécessaires pour une évaluation efficace. Bien que ce soient des techniques simples à mettre en œuvre, elles sont toutefois coûteuses car elles requièrent la présence et la disponibilité de ces évaluateurs.

Toutes ces remarques nous confortent dans l'idée de fournir des outils informatiques d'aide à l'évaluation.

2.2.2. Le plan de la capture automatique

S'appuyant sur une capture automatique des actions de l'utilisateur, [Siochi91] propose la détection de répétitions de schémas d'actions : MRP (Multiple Repeating Pattern). L'étude, par un expert, des répétitions d'actions de l'utilisateur va permettre de cerner plus rapidement certains problèmes associés à l'interface. Cette technique rejoint les observations via des techniques audiovisuelles [Hammontree92] en permettant à l'ergonome de revoir la séquence des actions de l'utilisateur autant de fois que nécessaire.



Quelques essais d'automatisation de la capture ont été tentés avec plus ou moins de succès. Nous pouvons citer Playback [Neal83] limité aux interfaces à langage de commande, ou HIMS [Theaker89] (Human Interface Monitoring System) tourné vers des interfaces graphiques et qui enregistre directement les signaux vidéos envoyés à la console.

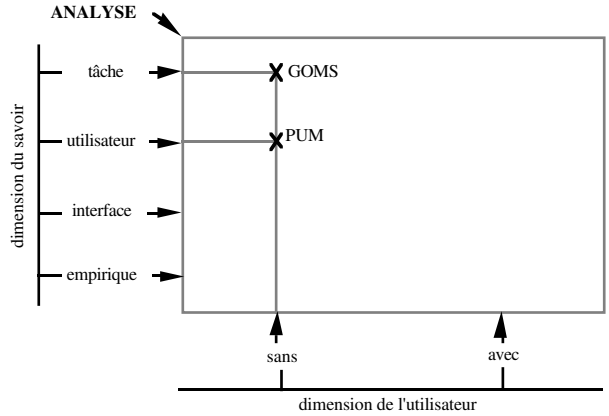
Ce que proposent MRP ou HIMS est insuffisant. Les données brutes sont difficiles à analyser car trop abondantes (200 MO/secondes pour HIMS). C'est pourquoi nous introduisons les notions d'analyse puis de critique automatique.

2.2.3. Le plan de l'analyse automatique

[Card83] préconise l'emploi de méthodes de psychologie pour l'évaluation des interfaces homme-machine. Le modèle qu'il développe, GOMS, est un modèle de description de l'exécution d'une tâche par un individu. Ce modèle décompose une tâche complexe en un arbre de buts et de sous-buts. Cette décomposition peut descendre jusqu'au niveau le plus concret qui représente les actions physiques, sur les dispositifs d'entrées, de l'opérateur. Keystroke est l'application de GOMS qui descend à ce niveau. La prédiction de la durée d'exécution d'une tâche complexe est alors réalisée par simple addition de la durée de réalisation des tâches élémentaires qui la composent. Ces tâches élémentaires sont, par exemple, la frappe de touches du clavier, l'acte de désignation ou le temps de réponse du système.

Cette approche hiérarchique descendante permet de choisir les méthodes et les opérateurs qui vont réaliser les tâches. La faiblesse relevée à ce niveau provient du fait que les

exécutions ainsi modélisées sont le reflet d'actions expertes qui supposent une exécution sans erreur.

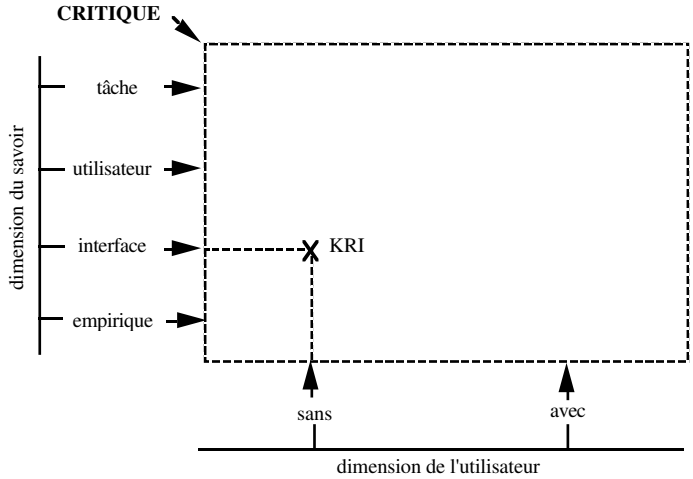


Tout en se détachant du modèle GOMS et d'un quelconque modèle de la tâche, [Young90] nous présente PUM (Programmable User Model). PUM est un logiciel qui simule certaines des capacités de l'architecture cognitive de l'utilisateur. Cet outil permet de prédire si une conception sera aisée à utiliser. Pour ce faire, le concepteur décrit les connaissances que l'utilisateur doit avoir au sujet de la tâche et de l'interface. Cette description est ensuite traduite en règles pour le PUM. Puis ces règles sont exécutées et fournissent une simulation de l'utilisateur exécutant la tâche avec l'interface spécifiée. Une planification infructueuse doit attirer l'attention du concepteur sur un défaut potentiel d'utilisabilité ou une difficulté d'apprentissage.

Ces techniques vont permettre de relever certains problèmes, ou de comparer une interface avec une autre, mais n'apportent aucune solution aux problèmes rencontrés. Ceci implique l'introduction de la dimension du critique automatique.

2.2.4. Le plan du critique automatique

KRI (Knowledge-based Review of user Interfaces), est le seul système, à notre connaissance, proposant automatiquement des améliorations de l'interface évaluée [Löwgren90].



Il propose un outil d'évaluation qui utilise une collection de règles (conseils ergonomiques du style de ceux proposés par [Smith86]) pour l'évaluation d'une description formelle de la conception d'une interface utilisateur. Cette évaluation produit des commentaires ainsi que des suggestions d'évolution.

Ce système s'adresse à des interfaces de remplissage de formulaires conduits par des menus dont l'accès s'effectue via des touches de fonction. De plus ce système ne prend en compte que les aspects syntaxiques et de mise en page de l'interface utilisateur. A première vue, ce système n'apporte que des éléments d'évaluation que nous appellerons figés. Exemple : “Tel bouton est trop près de tel autre” ou “L'ordre proposé ne suit pas l'ordre lexicographique”, etc.

La technique proposée par le système KRI tend à évaluer la conception de l'interface mais elle en oublie l'utilisateur et les remarques constructives qui découlent de l’observation de ce dernier.

3. Conclusion

Au sein de cette taxonomie, nous distinguons deux grands ensembles de systèmes d'évaluation : les évaluations expérimentales et les évaluations reposant sur l'étude d'un modèle formel. Cette distinction est reprise par la figure 3 qui donne une vue d'ensemble des quelques systèmes présentés.

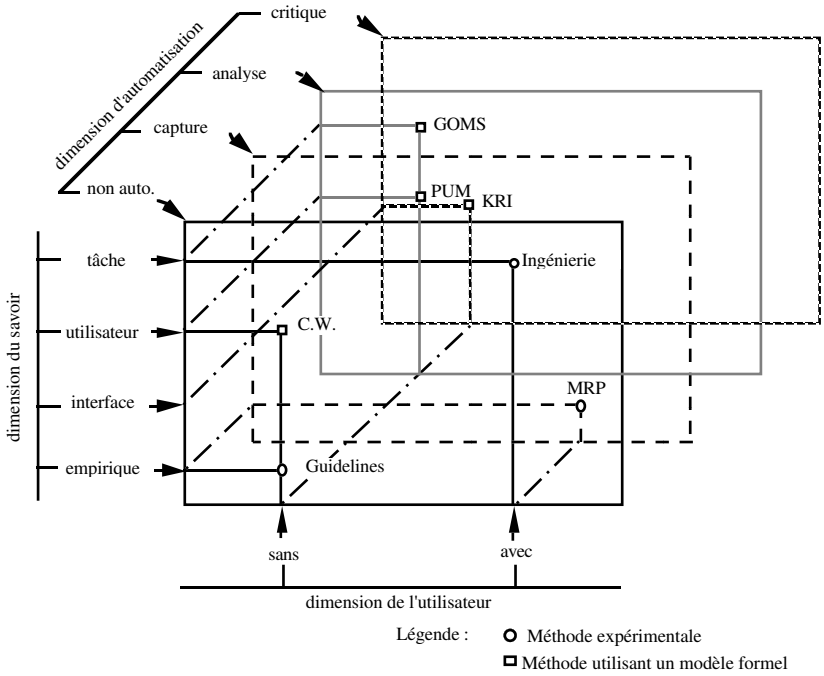


Figure 3 : vue d'ensemble de notre taxonomie

- L’évaluation expérimentale s’effectue à partir de données observées sur le cas réel. Elle requiert une réalisation partielle du système. A l'heure actuelle, l'une des caractéristiques principales de ces évaluations est l'importance du rôle joué par l’évaluateur (généralement un expert ergonome) qui va apporter son jugement sur l’interface à évaluer.
- L’évaluation à fondements théoriques, quant à elle, met en application des modèles. S’appuyant sur une représentation formelle de l'utilisateur ou de l'interface, parfois même de la tâche, elle ne nécessite pas une réalisation totale du système. Sa fonction est essentiellement prédictive.

Tous les systèmes étudiés se représentent dans notre taxonomie par un point à l'intersection d'un état de chacune de ces trois dimensions. Mais dans le cas général un système ou une technique d'évaluation se représente par un nuage de points. En effet, un système d'évaluation pourrait proposer une capture ainsi qu'une analyse automatique en se basant sur un savoir sur la tâche comme pour l'outil automatique d'aide à l'évaluation que nous proposons [Balbo92].

4. Références

[Balbo92]
S. Balbo & J. Coutaz, “Un pas vers l’évaluation automatique des interfaces homme-machine”, Congrès ERGO-IA’92, Biarritz, 7-9 Oct. 1992

[Balbo93]
S. Balbo, J. Coutaz & D. Salber, "Towards automatic evaluation of multimodal user interfaces", 1993 International Wokshop on Intelligent User Interfaces, Orlando (USA), January 1993

[Barthet88]
M.F. Barthet, "Logiciels interactifs et ergonomie, modèles et méthodes de conception", Editions Dunod Informatique, 1988

[Card83]
S.K. Card, T.P. Moran & A. Newell, “The psychology of Human Computer Interaction”, Lawrence Erlbaum Associates, 1983

[Dowell89]
J. Dowell & J. Long, " Toward a conception for an engineering discipline of human factors", Ergonomics, vol. 32, No 11, pp. 1513-1535, November 1989

[Hammontree92]
M.L. Hammontree, J.J. Hendrickson & B.W. Hensley, "Integrated data capture and analysis tools for research and testing on graphical user interfaces", in Proceedings of the CHI'92 Conference, ACM Press, Monterey, 3-7 May 1992, pp. 431-432

[Lewis90]
C. Lewis, P. Polson, C. Wharton & J. Rieman, “Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces”, Proceedings of the CHI’91 Conference on Computer Human Interaction. April 1990, Seattle. ACM New York, pp. 235-242

[Lim92]
K.Y. Lim, J.B. Long & N. Silcock, "Integrating human factors with system development methods: an overview of a structured human factors method", in Proceedings of the IDASCO'92 Conference, Oxford(UK), April 6-9 1992, IEE, D.K. Hitchins (Eds)

[Long89]
J. Long & J. Dowell, "Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering", in Proceedings of the Fifth Conference of the BCS HCI SIG, A. Sutcliffe and L. Macaulay (Eds), Cambridge University Press, 1989

[Löwgren90]
J. Löwgren & T. Nordqvist, “A Knowledge-Based Tool for User Interface Evaluation and its Integration in a UIMS”, Human-Computer Interaction-INTERACT'90, pp. 395-400

[Moran81]
T. Moran, “The Command Langage Grammar : a representation for the user interface of interactive computer systems”, International Journal of Man-Machine Studies, 15, 1981, pp. 3-50

[Motif90]
OSF/Motif, Programmer’s Guide, Open Software Foundation, Prentice Hall, 1990

[Niel 83]
Neal & Simon, "Playback: a method for evaluating the usability of software and its documentation", in Proceedings of the CHI'83 Conference, 1983

[Nielsen90]
J. Nielsen & R. Molich, "Heuristic evaluation of user interfaces", Proceedings of the CHI’90 Conference on Computer Human Interaction. 1990, Seatle. ACM New York, pp. 349-256

[Normand92]
V. Normand, “Le modèle Sirocco : de la spécification conceptuelle des interfaces utilisateur à leur réalisation”, thèse de l’Université Joseph Fourier, 1992

[Pollier91]
A. Pollier, “Evaluation d’une interface par des ergonomes : diagnostics et stratégies”, Rapport de recherche INRIA no 1391, Février 1991

[Scapin86]
D. L. Scapin, “Guide ergonomique de conception des interfaces homme-machine”, Rapport Technique INRIA no 77, Octobre 1986

[Scapin89]
C. Pierret-Goldreich, I. Delouis & D.L. Scapin, "Un Outil d'Acquisition et de Représentation des Taches Orienté-Objet", rapport de recherche INRIA no 1063, Programme 8 : Communication Homme-Machine, août 1989

[Schmucker86]
K. Schmucker, "MacApp: an application framework", Byte 11(8), 1986, pp. 189-193

[Senach90]
B. Senach, "Evaluation de l'ergonmie des interfaces homme-machine: du prototype aux systèmes experts", Congrès ERGO-IA'90, Biarritz, 19 au 21 Septembre 1990

[Serpent89]
SEI Serpent Application Develloper's Guide, *User's Guide CMU/SEI-89-UG-6, ESD-TR-89-12* (Januuary 1989)

[Siochi91]
A.C. Siochi & D. Hix, “A study of computer-supported user interface evaluation using maximal repeating pattern analysis”, Proceedings of the CHI’91 Conference on Computer Human Interaction. 1991 May, New Orleans. ACM New York, pp. 301-305

[Smith86]
S.L. Smith & J.N. Mosier, “Guidelines for designing user interface software”, Report MTR-10090 ESD-TR-86-278, The MITRE Corporation, Bedford, MA, August 1986

[Theaker89]
C.J. Theaker, R. Phillips, T.M.E. Frost & W.R. Love,"HIMS: a tool for HCI evaluations", in Proceedings of the Fifth Conference of the BCS HCI SIG, A. Sutcliffe and L. Macaulay (Eds), Cambridge University Press, 1989

[Webster89]
B.F. Webster, “The NeXT Book”, Addison-Wesley, Reading, Mass., 1989

[Whitefield91]
A. Whitefield, F. Wilson & J. Dowell, "A framework for human factors evaluation", Behaviour and information technology, vol. 10, no. 1, pp. 65-79, 1991

[Whiteside88]
J. Whiteside, J. Bennett & K. Holtzblatt, "Usability engineering : our experience and evolution", in Handbook of Human-Computer Intercation, M. Helander (Ed.), Elsevier Science Publishers, 1988

[Young90]
R.M. Young & J. Whittington, "A knowledge Analysis of interactivity", Proceedings of INTERACT'90, edited by D. Diaper, G. Cockton & B. Shackel, Elsevier Scientific Publishers B.V. pp. 207-212, 27-31st August 1990, Cambridge (United Kingdom)