

Critères d'évaluation et propriétés des systèmes de menu

Gilles Bailly^{1,2}

¹(GET/ENST) – CNRS LTCI UMR 51141
46 rue Barrault, 75013, Paris,
France

{gilles.bailly, elc}@enst.fr

Eric Lecolinet¹

²Lab. CLIPS-IMAG, Université de Grenoble 1
385, rue de la Bibliothèque, 38041 Grenoble Cedex 9,
France

{gilles.bailly, laurence.nigay}@imag.fr

ABSTRACT

Un grand nombre de nouveaux menus ont été conçus ces dernières années. Nous proposons de les organiser dans un canevas cohérent. D'une part, nous reprenons des critères d'utilisabilité comme l'efficacité ou l'apprentissage. Nous distinguons en particulier le mode novice du mode expert. D'autre part, nous exposons des critères d'utilité comme l'espace occupé que nous appliquons aux systèmes de menus. L'ensemble de ces critères permet de comparer les différents menus existants. Nous fournissons ensuite un ensemble de propriétés afin de concevoir de nouveaux menus adaptés à leur contexte d'utilisation.

Categories and Subject Descriptors: H.5.2. [Information Interfaces and Presentation]: User Interfaces. I.3.6 [Methodology and Techniques]: Interaction techniques.

General Terms: Design, Human factors

Keywords: Système de menus, Marking menus.

1. INTRODUCTION

Les systèmes de menus sont devenus incontournables dans les applications WIMP afin de supporter le nombre grandissant commandes. Les plus utilisés: les menus déroulants (menus cascades) et les menus contextuels proposent aux utilisateurs un ensemble de commandes organisés hiérarchiquement.

Une part importante du temps passé dans un logiciel consiste à explorer, rechercher ou activer des commandes dans ces menus. Ainsi, une amélioration, même limitée de la navigation dans les menus peut améliorer la productivité de manière significative.

Face à cet enjeu, de nouveaux systèmes de menus ont été conçus: par exemple les Toolglass [2] ou Marking menus [3] pour les menus contextuels et les Fish-eye menus [1] ou la hotBox [4] pour la barre de menus. Ces techniques se sont principalement focalisées sur deux aspects: (a) augmenter le nombre de commandes dans les menus (b) améliorer le mode expert des menus c'est à dire l'efficacité du menu pour des utilisateurs experts.

Devant l'importance des techniques proposées, nous proposons de les organiser dans un canevas cohérent. D'une part, nous reprenons des critères d'utilisabilité [6] comme l'efficacité ou l'apprentissage. Nous distinguerons en particulier le «mode novice» qui correspond à l'utilisation du menu pour une tâche d'exploration au «mode expert» qui consiste à rappeler rapidement des commandes pour les utilisateurs experts.

D'autre part, nous exposons des critères d'utilité, comme l'espace occupé que nous appliquons aux systèmes de menus. L'ensemble de ces critères permet de comparer les différents menus existants. La seconde partie de l'article fournit un ensemble de propriétés

pour concevoir de nouveaux menus adaptés à leur contexte d'utilisation. Enfin nous concluons et donnerons quelques pistes pour élaborer de nouveaux systèmes de menus.

2. CRITERES D'UTILISABILITE POUR LES MENUS

Cette section présente 5 critères d'utilisabilité pour évaluer des systèmes de menus. Des travaux antérieurs [9] ont présenté certains de ces critères d'utilisabilité, mais ils se sont focalisés uniquement sur 3 de ces critères en mode expert: efficacité, erreurs et apprentissage. Nous proposons ici de décrire 5 critères, qui concernent à la fois le mode expert et pour le mode novice.

2.1 Efficacité

Nous distinguons l'efficacité du mode novice pour la tâche d'exploration, de l'efficacité du mode expert pour la tâche de rappel (resélection) de commandes [3].

L'efficacité en mode expert indique la rapidité à laquelle un utilisateur est capable de sélectionner ou de rappeler une commande fréquente. Ce critère est à rapprocher de celui du nombre d'erreurs car en augmentant la vitesse, on augmente aussi le risque de faire des erreurs. Au bout d'un certain temps, les utilisateurs experts sont censés trouver un bon compromis entre vitesse et nombre d'erreurs.

L'efficacité en mode novice indique si le menu peut être utilisé facilement et efficacement pour les utilisateurs qui découvrent les commandes qu'il contient. L'efficacité en mode novice est difficile à estimer car elle dépend à la fois des intentions et des connaissances de l'utilisateur.

Le cas d'un utilisateur qui parcourt le menu dans le but de découvrir ce que lui propose le système est bien différent du cas d'un utilisateur qui cherche une commande qu'il sait qu'elle existe mais qui ne connaît pas sa position exacte dans la hiérarchie. Dans ce dernier cas, la connaissance de l'utilisateur sur l'organisation générale, la sémantique des menus (par exemple «copier» se trouve systématiquement dans «édition»), ou la connaissance du domaine lui permet de filtrer un ensemble de sous-menus et d'accélérer sa recherche.

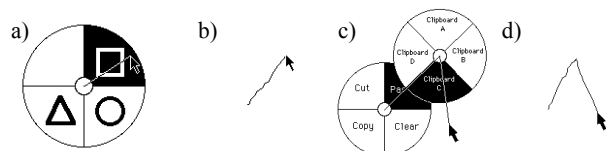


Figure 1: A gauche, Marking menu en a) mode novice : menu affiché, b) mode expert : une trace comme simple retour visuel). A droite, Marking menu hiérarchique en c) mode novice, d) mode expert.

2.2 Fiabilité

Alors que le nombre d'erreurs en mode novice est souvent très faible et comparable entre les techniques, il peut varier beaucoup plus significativement en mode expert. Une technique qui favorise le mode expert c'est-à-dire l'accès rapide aux commandes risque d'augmenter aussi le nombre d'erreurs.

2.3 Apprentissage

Nous distinguons deux types d'apprentissage.

Le premier type d'apprentissage correspond au temps nécessaire pour passer du mode novice au mode intermédiaire [3]. Il indique la rapidité et la facilité à laquelle le menu permet de retrouver le chemin d'un ensemble limité de commandes que l'on a déjà visité en mode novice. C'est-à-dire la vitesse à laquelle l'utilisateur va devenir familier avec le menu. Il ne connaît pas à priori le chemin exact des commandes désirées, mais sera capable de le reconstruire pas à pas sans jamais se tromper de sous-menu.

Le second type d'apprentissage a été le plus étudié ces dernières années avec l'introduction des Marking-menus. Il correspond au temps nécessaire à un utilisateur pour devenir expert. C'est-à-dire pour passer du mode intermédiaire au mode expert.

La plupart des travaux récents ont regroupé le mode novice et le mode intermédiaire et parle simplement de l'apprentissage du mode expert. Cependant nous pensons qu'il est important de distinguer ces deux modes comme dans [3] car ils correspondent à deux types de tâches différentes : (A) la tâche exploratoire où l'utilisateur navigue dans l'arbre des commandes de haut en bas et de bas en haut (retour arrière), (B) la tâche de « rappel » où l'utilisateur ne fait que descendre dans le menus (il ne se trompe plus de branche) pour les commandes qu'il a pris l'habitude d'utiliser.

2.4 Satisfaction

La satisfaction indique le plaisir de l'utilisateur à utiliser un système de menu. Or un menu de commandes est rarement l'objet principal de la tâche courante mais plutôt un artifice articulatoire. L'utilisateur doit donc pouvoir sélectionner une commande sans perdre son attention sur le contexte. Pour cela, le menu doit distraire au minimum l'utilisateur. Ainsi un menu qui satisfera les utilisateurs n'est pas forcément un menu esthétique mais au contraire un menu qui saura se faire oublier.

2.5 Facilité de compréhension

Ce critère indique la capacité d'une technique à être pris en main rapidement. Par exemple, les Marking menus hiérarchiques sont de nouveaux menus hiérarchiques circulaires. Pour activer un item d'un sous menu, il faut d'abord passer par le centre du sous-menu pour l'activer comme le montre la Figure 2-a. Le fait de devoir d'abord passer par le centre du sous-menu n'est pas forcément intuitif et peut poser des difficultés pour démarrer avec la technique.

Nous venons de présenter 5 critères d'utilisabilité appliqués aux systèmes de menus. Tous ces critères nécessitent des tests avec les utilisateurs afin d'être évalués de manière qualitative ou quantitative. Les critères d'utilité qui suivent ne nécessitent pas l'implication de l'utilisateur final. Par exemple, il est possible de mesurer l'espace occupé par le menu à l'écran sans utilisateur et de s'apercevoir que son adaptation sur PDA peut se révéler difficile.

3. CRITERES D'UTILITE POUR LES MENUS

Dans cette partie, nous proposons 4 critères d'utilité pour les menus. Ces critères sont rattachés au système.

3.1 Occupation de l'espace

Le premier aspect est l'espace occupé de manière permanente à l'écran. C'est-à-dire l'espace qu'utilise le menu lorsque l'utilisateur ne s'en sert pas. Par exemple la barre de menu affiche de manière systématique le premier niveau de la hiérarchie en haut de l'écran, soit environ 8 éléments. Les menus contextuels n'utilisent aucun espace à l'écran car ils sont affichés seulement lorsque l'on interagit avec eux. Le menu Hotbox [4] s'affiche seulement si l'utilisateur tient la barre d'espace appuyé afin de masquer le moins de temps possible la zone de travail.

Par contre, les palettes ou les Toolglass [2] présentent un ensemble de commandes qui occupent de manière permanente l'écran. Cela peut se révéler problématique pour des applications de visualisation par exemple.

Un autre aspect important est l'espace occupé une fois le menu déployé. Par exemple, les Marking menus hiérarchiques [3] de profondeur 3 utilisent quatre fois plus d'espace qu'un menu hiérarchique traditionnel. Cet aspect peut devenir critique si l'interface graphique est un PDA. L'espace utilisé par un menu déployé peut varier entre le mode expert et le mode novice. Il est souvent moins important en mode expert. Par exemple les Marking-menus hiérarchiques prennent beaucoup moins d'espace en mode expert. Leur mode expert consiste à faire un geste. Or ce geste ne dépend pas de l'échelle. Il est alors souvent beaucoup plus petit.

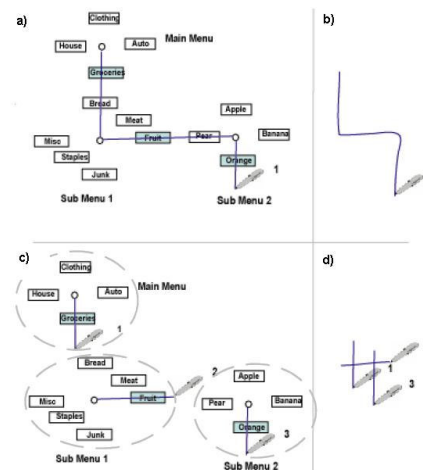


Figure 2: En haut, Marking menu hiérarchique en a) mode novice, b) mode expert. Composition spatiale de marques. En bas, Mutli-Stroke en c) mode novice et d) mode expert. Composition temporelle de marques simples.

3.2 Contextualité

Les menus contextuels (ou pop-up menus) sont des menus qui adaptent leur contenu en fonction du contexte. Ils sont souvent invoqués à l'aide du clic droit de la souris au-dessus de l'objet d'intérêt. Seules les commandes en relation avec l'objet sont affichées, ceci permettant de filtrer les commandes.

Ces menus ont la particularité de pouvoir être activés partout sur l'interface graphique et donc même près des bords de l'écran. Dans le cas des menus hiérarchique traditionnels, Le déploiement des sous-menus s'adapte à la disponibilité de l'interface (la position des sous-menus n'est pas fixée à l'avance). Par contre dans le cas des Marking-menus hiérarchique, ils requièrent tellement de place qu'ils ne peuvent être contextuels que pour les objets situés au centre de l'écran. Il est donc difficile de les utiliser comme des menus contextuels.

3.3 Nombre maximal de commandes

Certaines applications fournissent plus de 1000 commandes à l'utilisateur. Tous les menus ne supportent pas un tel nombre de commandes. Ce critère indique donc la capacité des menus à supporter un très grand nombre de commandes.

Il peut également indiquer pour les menus hiérarchiques, le nombre de commandes supportées en largeur c'est-à-dire le nombre de commandes disponibles à un niveau.

Lorsqu'il y a trop de commandes à un niveau, la solution est d'augmenter la profondeur du menu qui a souvent un coût en terme de temps et de nombre d'erreurs.

De plus, la plupart des menus radiaux ne supportent pas plus de 8 items en largeur ce qui peut se révéler problématique (le menu « File » contient généralement 12 items). Polygon menu [9] est un menu circulaire proposant une solution originale qui permet d'afficher jusqu'à 16 items sur un niveau en considérant à la fois la position et la direction des marques pour sélectionner un élément comme le montre la Figure 3. Le FishEye menu [1] permet d'afficher plusieurs dizaines d'éléments à un niveau pour les menus traditionnels mais n'a pas de mode expert.

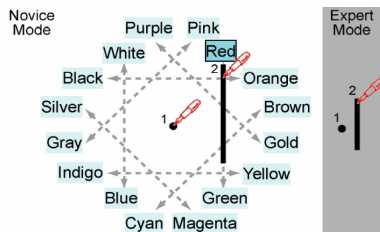


Figure 3: Polygon menu permet d'afficher jusqu'à 16 items au même niveau. Le clic détermine l'origine du menu. A la fois la position et l'orientation de la trace relativement à l'origine du menu déterminent l'item sélectionné.

3.4 Taille des libellés

Un aspect souvent sous-estimé est la capacité des menus à supporter des labels de longue taille. Les applications se spécialisent de plus en plus et le nom des commandes s'allonge.

Dans Word par exemple, on peut trouver « Mise en forme automatique... » ou « comparaison et fusion de documents... » qui comportent plus de 25 caractères. Les Marking menus hiérarchiques sont très sensibles à la taille des libellés. Leurs largeurs à l'écran correspondent au produit de la taille d'un item par 2 fois sa profondeur. Les palettes et les ToolGlass ne supportent généralement que des icônes ce qui n'est pas toujours idéal pour les utilisateurs novices.

Ces quatre critères d'utilité permettent d'établir si le menu est en adéquation avec les contraintes du système. Les critères

d'utilisabilité et d'utilité permettent de choisir quel menu correspond le mieux à des besoins dans un contexte donné. Cependant, ils ne disent pas comme faire pour satisfaire ces critères. Nous proposons donc maintenant un ensemble de propriétés qui peuvent améliorer les menus. La plupart favorisent souvent un critère au détriment d'un autre.

4. PROPRIETES

4.1 Minimisation du changement d'attention

L'utilisateur doit pouvoir sélectionner une commande sans perdre son attention sur la tâche courante. Pour cela, le menu doit distraire au minimum l'utilisateur.

Il est donc préférable de pouvoir activer une commande à l'endroit de la zone d'intérêt. La barre de menu ou la palette ne respectent pas cette propriété. En effet, d'une part la cible du regard est modifiée temporairement et d'autre part le chemin parcouru par la souris est souvent plus long, plus fatigant et donc plus distrayant.

Au contraire un menu contextuel apparaît sur l'objet courant, ainsi le regard ne change pas et le déplacement de la souris est faible. Malheureusement, le menu peut alors cacher l'objet d'intérêt ce qui peut aussi se révéler une nouvelle source de distraction. Une solution pour le mode novice et le mode expert a été envisagée pour les menus circulaires. Pour le mode novice, [3] a proposé de ne pas afficher le camembert (Figure 1-a) mais d'adopter une stratégie « seulement les labels » qui consiste à n'afficher que les labels comme le montre la Figure 2 a). Les zones d'occlusion sont alors réduites. Dans le mode expert, le menu n'apparaît plus et seulement une trace est affichée à l'écran comme le montre la Figure 2 b). Supprimer le retour visuel permet donc de mieux conserver le contexte mais augmente le temps de sélection et le taux d'erreurs [3].

4.2 Indépendance à l'échelle

Cette propriété permet à l'utilisateur de ne pas se focaliser sur la taille de ses gestes. Dans les menus traditionnels, les commandes sont à une position bien précises ce qui oblige l'utilisateur d'une part à être précis et d'autre part de regarder ce qu'il fait.

En développant des menus radiaux, les commandes ne sont plus associées à une position, mais à une direction. Cela permet de sélectionner un élément sans regarder. En effet, la plupart d'entre nous sommes capable de dessiner les 8 directions d'une boussole, alors qu'il est beaucoup plus difficile de dessiner une marque de 40 pixels sans regarder. En considérant seulement l'orientation, plutôt que la position, la taille de la marque n'a plus d'importance et permet de faire des gestes de n'importe quelle taille.

L'indépendance à l'échelle améliore l'utilisabilité. Les Marking menus ont montré qu'ils étaient significativement plus rapide et produisaient significativement moins d'erreurs [3].

4.3 Transition novice expert

Cette propriété permet d'accélérer d'une part le passage du mode novice au mode intermédiaire et le passage du mode intermédiaire au mode expert. Cette propriété améliore l'apprentissage. Elle consiste à ne pas changer de technique d'interaction entre les différents modes, mais au contraire de proposer une transition fluide.

Dans les menus traditionnels, le mode novice est identique au mode intermédiaire et consiste à naviguer dans le système menus.

Par contre, il y a une rupture avec le mode expert qui consiste à utiliser des accélérateurs claviers comme «Ctrl+C» pour Copier.

Dans les Marking menus, le menu s'affiche seulement au bout de 0.3s d'inactivité. Ainsi, si l'utilisateur est novice, il attend la fin de ce délai pour voir le menu s'afficher puis active la commande désirée. Par contre, si l'utilisateur est expert, il peut faire une marque sans attendre l'affichage du menu. Si au cours de l'interaction, il a des doutes (mode intermédiaire), le menu se réaffichera au bout de 0.3s d'inactivité. Dans tous les cas, quelques soit le mode, l'utilisateur a réalisé le même geste. Il n'y a pas de rupture dans l'interaction.

De la même manière, le KeyMenu [5] (menu basé sur les touches d'un clavier) propose une transition fluide entre le mode novice et expert en conservant la même succession de touches pour activer les commandes.

4.4 Prévisualisation

C'est la capacité qu'a un menu hiérarchique à montrer le sous-menu d'un item avant de l'activer. Ceci permet, avant de descendre dans la hiérarchie, de s'apercevoir si oui ou non l'item d'intérêt s'y trouve. La prévisualisation est d'autant plus efficace que l'objet recherché s'identifie facilement par rapport aux autres. Il est souvent plus rapide de pré-visualiser les sous-menus que de «descendre-remonter». Le multi-stroke [10] ne permet pas la prévisualisation afin de restreindre l'espace occupé pour le déploiement du menu. Cela a pour conséquence de ralentir la navigation en mode novice et de frustrer l'utilisateur.

4.5 Composition spatiale vs temporelle

Les Marking-menus et les multistrokes [10] ont adopté deux stratégies différentes. Les Marking menus utilisent la composition spatiale, c'est-à-dire que l'utilisateur fait une marque composée spatialement comme le montre la Figure 2-a. Le geste est continu.

A l'opposé, le multi-stroke repose sur une composition temporelle comme le montre la Figure 2-c. L'utilisateur fait plusieurs marques simples dessinées les unes sur les autres successivement. L'interaction n'est plus continue dans ce cas.

Il a été montré que le multi-stroke est un peu plus rapide que le Marking menus hiérarchique mais surtout qu'il diminue significativement le nombre d'erreurs [10]. Une explication théorique est que le multi-stroke nécessite seulement de dessiner des marques simples (mais plusieurs fois) alors que le Marking menu hiérarchique nécessite une marque plus compliquée.

4.6 Fusion sélection + contrôle

Des techniques comme les Controlmenus [7], Flow menus ou ToolGlass rappelées dans [2] permettent de sélectionner une opération et de la contrôler continuellement en un simple geste. C'est ce que Guimbretière nomme «merging» ou «la fusion de la sélection et de la manipulation directe d'une commande» [2]. Les techniques ayant cette propriété se sont révélés significativement plus rapide que les autres [2].

4.7 Modalité

La modalité est une propriété importante pour les actions répétées. C'est la capacité à conserver le mode (ou l'outil), c'est-à-dire ne pas être obligé de re-sélectionner la même commande entre chaque action. Les palettes conservent généralement le mode (une fois un outil sélectionné, il est conservé jusqu'à ce qu'il soit remplacé). Les Floating Pie menus [8] ont par exemple enrichi les pie menus de cette propriété. Il est ainsi possible de «bloquer» un

pie menu à n'importe quel niveau de la hiérarchie. Ils évitent donc pour une opération répétée de descendre à chaque fois dans le même sous-menu. Noter qu'une idée similaire est utilisée pour les menus cascades de certaines boîtes à outils (Motif) : ça s'appelle des tear Off menus.

5. CONCLUSION

L'importance des techniques développées ces dernières années nécessitait de capitaliser les connaissances dans ce domaine. Nous avons donc appliqué les critères d'utilisabilité et d'utilité aux menus en prenant soin de séparer le mode expert du mode novice (alors que la plupart des travaux s'intéressent principalement au mode expert). Ces critères permettent de comparer les différents systèmes de menus. Nous avons ensuite présenté un ensemble de propriétés susceptible de permettre aux concepteurs de favoriser tel ou tel critères. Ces critères et propriétés ont permis également de présenter la plupart des systèmes de menus de commandes.

Aucune technique, pour l'instant ne remplit tous les critères d'utilisabilité et d'utilité. La prochaine étape dans la suite de nos travaux consiste à concevoir un menu qui rassemblera la plupart des avantages cités tout en considérant les spécificités du contexte d'utilisation.

6. REFERENCES

- [1] Bederson, B. B. (2000). Fisheye menus. *ACM UIST'00 Symposium on User interface Software and Technology*, pp. 217-225.
- [2] Guimbretière, F., Martin, A. & Winograd, T. (2005). Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.*, pp. 460-476.
- [3] Kurtenbach, G. P. (1993). *The Design and Evaluation of Marking Menus*. Doctoral Thesis. UMI Order Number: UMI Order No. GAXNN-82896., University of Toronto.
- [4] Kurtenbach, G., Fitzmaurice, G. W., Owen, R. N., and Baudel, T. (1999). The Hotbox: efficient access to a large number of menu-items. *ACM CHI'99 Conference on Human Factors in Computing Systems*, pp. 231-237.
- [5] Lyons, K., Patel, N. J., and Starner, T. (2003). KeyMenu: A Keyboard Based Hierarchical Menu. *IEEE ISWC'07 Symposium on Wearable Computers*.
- [6] Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. *ACM CHI'93 Conference on Human Factors in Computing*, pp. 206-213.
- [7] Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. (2000). Control menus: execution and control in a single interactor. *ACM CHI '00 Extended Abstracts on Human Factors in Computing Systems*, pp. 263-264.
- [8] Rubio, J.M. and Janecek, P (2002). Floating pie menus: enhancing the functionality of contextual tools. *ACM UIST'02*, pp. 39-40.
- [9] Zhao, S., Agrawala, M., and Hinckley, K. (2006). Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. *ACM CHI'06*, pp. 1077-1086.
- [10] Zhao, S. and Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. *ACM UIST '04*, pp. 33-42.