MenuErgo: Conception assistée de menus par évaluation automatique de règles ergonomiques

Iyad Khaddam, Sara Bouzit, Gaëlle Calvary, Denis Chêne

▶ **To cite this version:**

Iyad Khaddam, Sara Bouzit, Gaëlle Calvary, Denis Chêne. MenuErgo: Conception assistée de menus par évaluation automatique de règles ergonomiques. Actes de la 28ième conférence francophone sur l'Interaction Homme-Machine, Oct 2016, Fribourg, Suisse. pp.36-47, 10.1145/3004107.3004130 . hal-01383738

HAL Id: hal-01383738
https://hal.science/hal-01383738

Submitted on 19 Oct 2016

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.

MenuErgo: Conception assistée de menus par évaluation automatique de règles ergonomiques

Iyad Khaddam[1], Sara Bouzit[2,3], Gaëlle Calvary[2], Denis Chêne[3]
[1]Université catholique de Louvain, Louvain School of Management, Louvain Interaction Lab.
B-1348 Louvain-la-Neuve (Belgium) - iyad.khaddam@uclouvain.be
[2]Université Grenoble Alpes, Laboratoire d'Informatique de Grenoble, F-38000 Grenoble (France)
[1]Orange Labs, 28 chemin du Vieux Chêne, F-38240 Meylan (France)
{sara.bouzit, gaelle.calvary}@imag.fr – {sarah.bouzit, denis.chene}@orange.com

RÉSUME
Cet article présente MenuErgo, un environnement logiciel de conception d'une barre de menu avec ses menus déroulants et sous-menus d'une interface graphique par évaluation automatique de règles ergonomiques propres aux menus selon quatre stratégies d'évaluation: la stratégie active à l'initiative du système, la stratégie passive à l'initiative du concepteur, la stratégie mixte partagée par les deux et la stratégie par unités conceptuelles basée sur le domaine sémantique. L'architecture logicielle de MenuErgo est composée d'un déclencheur d'évaluation, d'un moteur d'évaluation, d'un présentateur d'évaluation et d'une base de 58 règles ergonomiques dont l'évaluation est assurée automatiquement. Une étude qualitative exploratoire rapporte les réactions de neuf concepteurs ayant utilisé MenuErgo sur une étude de cas de conception de menu pour une application multimédia.

Mots Clés
Barre de menu ; évaluation automatique ; menus ; menus déroulants ; règles ergonomiques ; stratégie d'évaluation.

ABSTRACT
This paper presents MenuErgo, a software environment for designing a menu bar along with its pull-down menus and sub-menus of a graphical user interface by automatic evaluation of menu usability guidelines according to four evaluation strategies: an active strategy initiated by the system, a passive strategy initiated by the designer, a mixed strategy shared by both of them, and a strategy by conceptual units based on the semantic domain. The MenuErgo software architecture consists of an evaluation trigerrer, an evaluation engine, a presenter, and a knowledge base of 58 usability guidelines whose evaluation is automated. A qualitative exploratory study reports on the reactions of nine designers after their usage of MenuErgo on a case study for designing menus for a multimedia application.

© ACM, 2016. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 28ème conférence francophone sur l'Interaction Homme-Machine, 2016.
http://dx.doi.org/10.1145/3004107.3004130

Author Keywords
Automatic evaluation ; evaluation strategy ; menu bar ; pull-down menus ; sub-menus ; usability guidelines.

ACM Classification Keywords
• **Human-centered computing ~ Graphical user interfaces** • *Human-centered computing ~ Heuristic evaluations* • Human-centered computing ~ User interface management systems • **Software and its engineering ~ Software design engineering** • Information systems ~ Information systems applications.

INTRODUCTION

Une longue tradition de recherche a été établie dans le domaine de l'évaluation automatique d'une interface sur base de diverses règles de conception, règles ou recommandations ergonomiques, d'utilisabilité ou d'accessibilité [1,4,9,12,19, 22,36]. Typiquement, l'interface graphique d'un site Web, d'une application web ou interactive est passée au crible de règles d'utilisabilité et/ou d'accessibilité en vue d'identifier et de corriger des problèmes potentiels d'utilisabilité [22]. Cette démarche impose deux conditions : la *disponibilité de l'interface* (pour un site web, il faut que le code HTML soit disponible pour le soumettre à un outil d'évaluation automatique [15] ; pour une application graphique, il faut accéder à tout ou partie de son code [19]) et l'*évaluation hors contexte de conception* (l'évaluation s'effectue après la conception, ce qui oblige le concepteur à revenir en arrière en cas de problème d'utilisabilité détecté, idem pour le développeur).

D'autre part, une autre tradition de recherche concerne les outils de critiques de conception [6,14,18,20,26]. Typiquement, le concepteur manipule une certaine représentation de sa future interface, souvent sous la forme d'un modèle, et l'outil analyse [6], commente [18], critique [20] la conception en cours sur base de ce qui peut être détecté automatiquement. Cette démarche induit également deux limites majeures : la *discontinuité* entre conception et développement (la critique est effectuée en phase de conception sans possibilité pour le développeur de reprendre les résultats [15]) et le *niveau de fidélité* de la représentation (la critique porte sur une représentation de l'interface dont le niveau de fidélité n'est pas nécessairement celui de l'interface finale et porte seulement sur une partie des propriétés ergonomiques [8]).

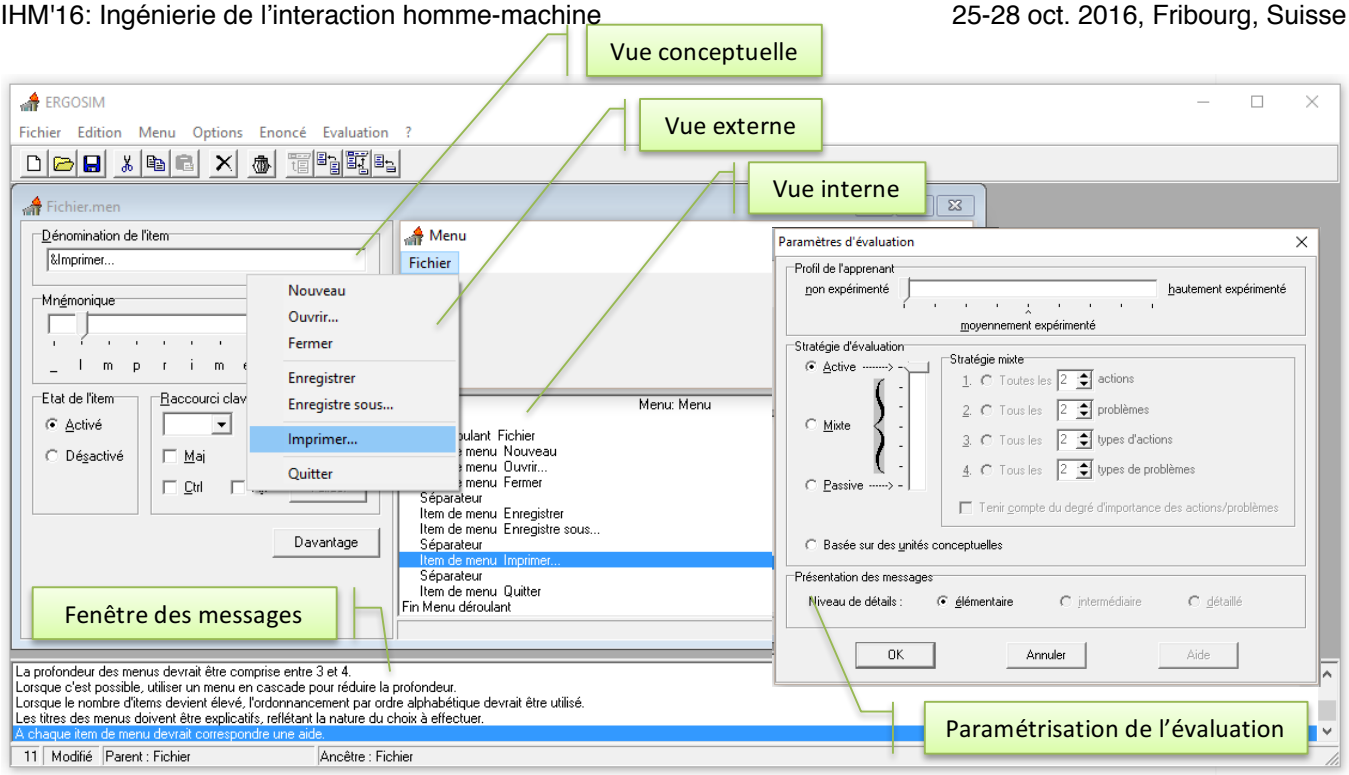

Figure 1. Environnement de travail de MenuErgo.

Cet article présente MenuErgo (Fig. 1), un environnement logiciel de conception d'une barre de menu avec ses menus déroulants et sous-menus qui tente de bénéficier du meilleur des deux approches : une critique en cours de conception (et non pas après la conception) sur une interface finale (et non pas seulement une interface représentée) sur base d'une évaluation automatique qui est paramétrable en fonction de la préférence du concepteur (et non pas fixe dans tous les cas de figure) et qui permet d'assurer la continuité avec le reste du cycle de vie de développement de l'interface (ici par génération automatique d'un code correspondant), comme le font les outils les plus avancés [17,37]. Dans MenuErgo, le concepteur conçoit sa barre de menus exactement comme il le ferait dans un environnement de développement intégré, sur base des mêmes paramètres. A tout moment, suivant certaines conditions, une évaluation automatique de règles ergonomiques peut être lancée afin de critiquer la conception en cours, en espérant que le concepteur puisse y répondre et mieux manipuler les règles au moment opportun.

Nous avons choisi de centrer un tel environnement logiciel sur une phase particulière du cycle de vie : la conception d'une barre de menu avec ses menus déroulants et ses sous-menus. Plusieurs raisons motivent ce choix : le menu reste parmi les techniques d'interaction les plus couramment utilisées [3,27], il existe de nombreuses manifestations différentes de menu (par exemple, Bailly *et al.* ont répertorié 64 techniques de menu différentes [2]), un nombre conséquent de règles relatives à la conception de menu reste valable [27,31], ces recommandations sont éparpillées dans la littérature [35], la conception de menu est une activité familière du concepteur [6] et un objet plus facilement contrôlable dans une expérimentation que d'autres [21].

Le reste de cet article est structuré comme suit : la section 2 revisite certains travaux liés d'une part à l'évaluation automatique de règles ergonomiques et d'autre part les travaux liés à l'aide à la conception en général et pour les menus en particulier ; la section 3 introduit, motive et décrit MenuErgo, l'environnement logiciel concerné ; la section 4 rapporte les résultats d'une étude qualitative exploratoire destinée à identifier les réactions de concepteurs ayant utilisé MenuErgo; la section 5 résume les contributions de cet article et esquisse des pistes de réflexion pour poursuivre ce travail.

TRAVAUX LIES

Etant donné la nature du travail consacré à MenuErgo, les travaux liés relèvent principalement de deux domaines de recherche qui sont résumés dans les deux sous-sections suivantes : l'évaluation automatique d'une interface à l'épreuve de règles et l'aide à la conception. Le cas échéant, une focalisation sur les menus est opérée.

Evaluation automatique de règles

L'évaluation automatique d'une interface, en général graphique, à l'épreuve de règles ou recommandations ergonomiques ou d'accessibilité constitue un domaine privilégié de recherche pour lequel de nombreux travaux ont été effectués. Ivory & Hearst [22], de même que Charfi *et al.* [10] en ont effectué une revue de l'état de l'art. On trouve notamment l'évaluation de sites web à l'épreuve de règles ergonomiques et/ou de règles d'accessibilité. Nous laissons ici de côté mes outils évaluant les règles d'accessibilité car ils forment une famille d'applications à part entière et un autre domaine de recherche. Etant donné le nombre important d'outils et leur variété, seuls quelques exemples considérés comment représentatifs sont discutés dans cette section, sans souci d'exhaustivité.

METROWEB [14] permet au concepteur d'accéder à une ou plusieurs bases de règles ergonomiques relatives à différents types d'interfaces qui sont structurées suivant les *critères ergonomiques* de Bastien & Scapin [7]. Il n'offre que de la documentation structurée en matière d'ergonomie du logiciel, mais qui peut être affinée sur base de requête que l'on peut formuler sur ces bases, du genre « lister toutes les règles ergonomiques relatives à la conception d'un menu déroulant valides en général ».

ERGOVAL [19] évalue automatiquement un ensemble de règles ergonomiques pour une interface graphique d'une application interactive suivant la méthode d'évaluation heuristique, ces recommandations étant codées dans l'outil même. Toute occurrence d'un problème ergonomique potentiel est enregistrée dès qu'une déviation par rapport à la règle encodée est détectée.

KWARESMI [4] évalue automatiquement un ensemble de règles d'ergonomie et d'accessibilité d'un site web par analyse statique de son code HTML. Chaque règle n'est plus programmée, mais spécifiée dans le langage à balises Guideline Definition Language (GDL). Il en résulte l'autonomie de la base de règles par rapport au moteur d'évaluation : puisqu'il n'est plus nécessaire de programmer une règle, mais seulement la spécifier, le moteur d'évaluation ne doit pas être actualisé si une nouvelle règle doit être incorporée. L'inconvénient de cette approche concerne la difficulté [35] potentielle à transformer une règle initialement énoncée dans une source ergonomique comme un guide. De plus, l'apprentissage d'un langage de spécification supplémentaire peut être considéré comme contraignant.

Plusieurs outils adoptent la même démarche, comme ErgoManager [1], ErgoCoIn [25], MAUVE [30], avec un degré de flexibilité plus important en ce qui concerne la manipulation des règles au sein des bases et en personnalisant l'évaluation sur un ensemble restreint de règles plutôt que sur l'entièreté de la base de règles.

RITA [10] élargit le cadre de l'évaluation automatique d'interfaces graphiques en prenant en compte les règles ergonomiques, mais aussi d'autres variables quantitatives comme le temps d'exécution d'une tâche, le taux d'erreur, les traces d'interaction et les met en relation pour évaluer de manière plus précise davantage de règles.

Il est possible de développer des widgets graphiques [11] dont le code est étendu pour mieux capturer certaines données d'interaction et donc mieux les analyser. EISEVAL [11,12] évalue automatiquement des règles ergonomiques encodées dans une application développée selon le paradigme multi-agent, dont un agent est particulièrement chargé du processus complet de l'évaluation, allant de la collecte des données jusqu'à leur interprétation finale, en passant par la participation de l'utilisateur final [13].

Une question récurrente posée par ces outils concerne la limite jusque laquelle l'évaluation automatique des règles est réalisable : entre une source de règles ergonomiques et une base de règles ergonomiques finalement automatisées, combien de règles restent? Farenc *et al*. [18] rapportent que la limite oscille vers les 40% de règles considérées pour une interface graphique : 40% des règles ergonomiques du guide ergonomique de La Poste ont pu être évaluées automatiquement avec des degrés divers. Cooper *et al.* [15] montrent que dépasser le seuil de 50% est possible, surtout lorsqu'il s'agit d'un site Web dont le code HTML est téléchargeable, au contraire d'une interface codée d'une application interactive.

Aide à la conception d'interface

Depuis longtemps, l'aide à la conception d'interface a pris la forme d'outil de critique de la conception [20,26], notamment parce qu'il a été empiriquement prouvé que la production de critique par un outil influençait favorablement le processus cognitif de développement dans la tête du concepteur [6], qu'il travaille isolément [6] ou de manière collaborative [26]. L'intervention d'un tel outil se veut *formative*, surtout pour les concepteurs novices, *sommative*, surtout pour les concepteurs experts [6]. Pour eux, le rappel circonstancié de règles pertinentes est estimé plus important que leur apprentissage, qui est l'apanage des novices. Un outil d'aide à la conception par critique fonctionne typiquement sur une certaine représentation de l'interface qui peut aller d'une représentation informelle jusqu'à un modèle formel permettant d'évaluer une technique d'interaction complète [28]. Par définition, cette représentation réduit l'objet de conception à un ensemble plus réduit de caractéristiques qui sont sujettes à la critique. Produire une critique constitue une activité complexe sur une phase complexe qu'est la conception [20]. Dès lors, produire une critique opportune considérée comme suffisamment de haut niveau par un concepteur constitue encore aujourd'hui un défi à relever.

En ce qui concerne METROWEB, il a été montré que les concepteurs manipulent davantage de règles ergonomiques lorsqu'elles sont suggérées par un outil d'aide à la conception que sans cet outil et que l'interface résultant de cette conception satisfaisait davantage de règles que sans lui [14].

MenuDesigner [33] génère automatiquement une barre de menus avec ses menus déroulants et sous-menus à partir d'une représentation de l'interface : un graphe d'enchaînement des activités, une forme simplifiée de modèle de tâches dans lequel les tâches sont liées par des opérateurs ET, OU, XOU. La structure du menu est générée à partir du modèle : à part modifier ce modèle et re-générer le résultat, il n'y a aucune adaptation ni intervention de la part du concepteur dans le processus. De plus, certains menus déroulants sont constitués en fonction de l'ordre d'apparition dans le graphe, ce qui est une forme réduite de groupement sémantique.

MenuSelector [31] est un outil de prototypage rapide de menu pour le web prenant en compte divers paramètres physiques, comme la localisation (p. ex. dans une colonne à gauche), le mécanisme de sélection, l'ordonnancement des items de menu, la répartition en groupes, etc. Cette approche est purement syntaxique car aucune prise en considération de la tâche et de son contexte d'usage n'est assurée.

MenuOptimizer [3] fournit une aide à la conception de menu aux concepteurs et développeurs par optimisation de la structure du menu en maximisant sa cohérence par rapport à sa performance sur base d'un algorithme de type 'fourmi'. Cet outil révèle bien la popularité des items de menus par leur coloration, ce qui préserve la stabilité spatiale du menu, au contraire des menus adaptatifs qui peuvent affecter la configuration spatiale, comme dans les menus adaptatifs [21].

Matsui & Yamada [24] ont introduit un algorithme génétique pour générer automatiquement la structure du menu en vue de maximiser le temps de sélection d'un item de menu considéré comme fréquent. La configuration d'un tel menu peut changer, donc mettant en échec la stabilité spatiale [21].

En conclusion, on observe d'une part des outils d'aide à la conception, essentiellement basé sur des techniques d'optimisation, avec peu ou pas d'intervention du concepteur et d'autre part des outils d'évaluation automatique basée sur des règles, mais qui interviennent après que la conception et/ou le développement aient eu lieu.

MenuErgo entend marier les avantages de ces deux mondes en offrant au concepteur une aide à la conception de menu sous la forme d'une évaluation automatique de la conception de menu en situation de conception pour ne pas reporter les bienfaits de cette évaluation à une phase ultérieure de développement tout en permettant de générer le code directement pour poursuivre le développement.

CONCEPTION ET DEVELOPPEMENT DE MENUERGO

Options de conception de MenuErgo

Les grandes options de conception principales qui ont été décidées pour MenuErgo sont décrites ci-après.

1. Visualisation multi-vues du menu. On considère généralement que trois vues différentes peuvent être offertes simultanément sur un objet de l'interface en fonction du profil de l'utilisateur et de ses besoins (Fig. 1) [16,20]: une *vue externe* qui la représentation réelle du menu telle qu'elle apparaît réellement dans l'application interactive, qui est essentiellement destinée à la valider avec l'utilisateur final ; une *vue interne* qui donne la représentation interne au système du menu sous la forme d'une hiérarchie de ressources, qui est essentiellement la vue du développeur ; une *vue conceptuelle* qui donne la représentation d'un item de menu en termes d'options de conception utiles au concepteur. Toute modification opérée sur la vue conceptuelle (p. ex. par assignation d'une valeur à un paramètre) ou interne (p. ex. déplacement d'un item d'un sous-menu à l'autre) est immédiatement propagée à la vue externe, qui n'est pas manipulable. La vue conceptuelle couvre les propriétés suivantes (Fig. 2):

- la *dénomination* de l'item, qui peut comprendre, en plus du libellé de l'item, le caractère « & » afin de spécifier visuellement (c'est-à-dire en le soulignant) le caractère utilisé comme mnémonique d'accès de l'item.
- la *mnémonique* d'accès de l'item, c'est-à-dire le caractère que l'on peut utiliser en combinaison avec la touche « Alt » pour sélectionner l'item de menu à l'aide du clavier au lieu d'utiliser la souris. La gamme des mnémoniques autorisées est automatiquement proposée par MenuErgo dès que l'utilisateur a acquis la dénomination. L'utilisateur peut alors manipuler directement le curseur de défilement sur ces mnémoniques et voir simultanément le résultat, par exemple toutes les lettres de l'item « Enregistrer » à la Figure 2. Ceci peut provoquer évidemment des collisions entre mnémoniques ou des conflits entre règles conflictuelles [23].
- l'*état* d'activation de l'item, qui indique si l'item est activé ou non. Lorsque l'item est désactivé, il apparaît grisé dans la barre de menus.
- le *raccourci clavier* ou accélérateur de l'item, qui définit la combinaison de touches normales (A, B, C, ..., X, Y, Z, 1, 2, ...9, 0, F1, F2, ..., F11, F12, Del, Ins, ...) et spéciales (« Ctrl », « Alt » et « Maj ») utilisée pour accéder de manière unique à l'item de menu dans la barre de menus.
- le *type d'attachement* de l'item, qui spécifie si l'item est attaché à l'affichage d'un sous-menu (ou d'un menu déroulant), à l'ouverture d'une boîte de dialogue ou d'une fenêtre secondaire, au branchement/débranchement du paramètre associé ou encore au déclenchement d'une action de l'application.
- le *message d'aide contextuelle* associé à l'item, qui s'affiche, lorsque l'item est sélectionné, dans la barre d'état spécifique au menu en cours de conception située entre la vue globale textuelle et la fenêtre des messages.

La barre d'outils reprend les fonctionnalités les plus fréquemment utilisées : les commandes de gestion de fichier (par exemple, « Nouveau », « Ouvrir » et « Enregistrer »), les fonctionnalités d'édition (« Couper », « Copier », « Coller » et « Supprimer ») sur un item de menu, un sous-menu, un menu déroulant ou un séparateur, les opérations sur les menus (par exemple, insérer un menu déroulant, insérer un sous-menu, insérer un item de menu et insérer un séparateur) ainsi que la commande de nettoyage de la fenêtre des messages.

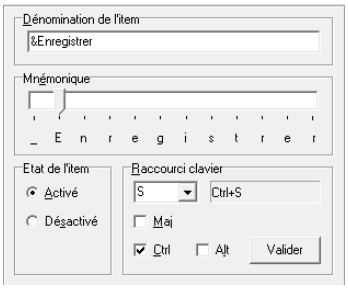

Figure 2. Vue conceptuelle d'un item de menu.

2. Evaluation multi-stratégies. Les utilisateurs cibles de MenuErgo sont principalement les concepteurs, les développeurs, voire tout intervenant dans le cycle de vie de développement. Ces personnes ayant des profils et des expériences diverses en matière d'ergonomie du logiciel, l'évaluation doit être flexible en fonction de ces profils [6,18] : une *fenêtre de paramétrisation* (partie droite de la Fig. 1) offre plusieurs stratégies de déclenchement de l'évaluation :

1. une *stratégie d'évaluation active* où l'utilisateur décide lui-même du moment où le logiciel évalue le menu en cours de conception. Cette stratégie est qualifiée d'active dans la mesure où c'est l'utilisateur final qui initie le contrôle de l'évaluation. Cette stratégie s'avère pertinente pour les utilisateurs qui souhaitent ne pas être interrompus dans leur processus cognitif de réflexion [6].
2. une *stratégie d'évaluation passive* où le logiciel évalue, sans la moindre intervention de l'utilisateur, chacune des actions opérées par ce dernier sur la conception en cours. Cette stratégie est qualifiée de passive dans la mesure où le système initie l'évaluation. Cette stratégie s'avère pertinente pour les utilisateurs qui souhaitent intervenir dès qu'un problème ergonomique se pose [14].
3. une *stratégie d'évaluation mixte*, c'est-à-dire située à mi-chemin entre les stratégies active et passive, où l'utilisateur peut paramétrer l'évaluation sur base : du nombre d'actions de l'utilisateur (p. ex., l'évaluation est déclenchée après chaque suite de 5 opérations de l'utilisateur), du nombre de types d'actions (p. ex. l'évaluation est déclenchée après qu'un item de menu soit complètement spécifié et pas avant pour ne pas perturber l'activité), du nombre de problèmes ergonomiques détectés (p. ex. dès que 3 problèmes ergonomiques ont été détectés, ce qui est utile notamment lorsqu'une erreur déclenche des problèmes en cascade) et du type de problème ergonomique (p. ex. l'évaluation est déclenchée dès qu'une erreur ergonomique importante est décelée). Pour l'instant, le déclenchement s'effectue dès qu'un seuil de référence est atteint pour une de ces conditions, mais on pourrait envisager des conditions logiques plus complexes combinant ces paramètres (p. ex., l'évaluation est déclenchée soit lorsqu'une règle ergonomique à haut impact est violée soit lorsqu'un petit nombre de règles à faible impact est enfreint).
4. une *stratégie d'évaluation basée sur des unités conceptuelles*, proche de la stratégie active, où l'utilisateur décide du moment où le logiciel évalue les unités conceptuelles de la conception en cours, une *unité conceptuelle* étant définie ici une entité non élémentaire [5] de la barre de menus, telle que, par exemple, le menu déroulant « Fichier » et l'ensemble des items de menus (et sous-menus) qu'il contient. Cette stratégie est qualifiée de « basée sur des unités conceptuelles » dans la mesure où l'évaluation porte globalement sur un concept.

Un curseur gradué entre actif et passif permet à l'utilisateur de situer le niveau d'intervention désiré.

3. Feedback paramétrable. L'utilisateur peut régler le *niveau de détail* avec lequel le résultat de l'évaluation automatique est présenté (Fig. 1) [14,35]:

- *élémentaire*, c'est-à-dire laissant apparaître un message comprenant l'intitulé de la règle ergonomique enfreinte (ou des règles) suite à la dernière évaluation déclenchée (cfr. la fenêtre des messages à la Fig. 1);
- *intermédiaire*, où le message contient, en plus de l'intitulé de la règle détectée dans la dernière évaluation déclenchée, des informations complémentaires relatives à la règle ergonomique elle-même, comme le lieu, le niveau d'importance ;
- *détaillé*, où le message contient l'ensemble des informations relatives à la règle enfreinte : l'intitulé, le critère ergonomique de Bastien & Scapin [7], le niveau linguistique [35], une à plusieurs illustrations (positives et/ou négatives) et les références bibliographiques de la règle.

Le contenu de la fenêtre des messages peut être nettoyé à tout instant à l'aide d'un bouton de commande disposé sur la barre d'outils ou via un autre bouton de commande dans la boîte de dialogue des options d'affichage.

4. Gestion des profils des utilisateurs. Un profil de l'utilisateur peut être enregistré concernant les paramètres suivants :

- le *profil de l'utilisateur* (en haut à droite de la Fig. 1) qualifie le niveau d'expérience de l'utilisateur tant d'un point de vue général de l'ergonomie du logiciel qu'au niveau de la plate-forme, de la conception de menus, de la conception d'interfaces homme-machine ou de menu : non expérimenté, moyennement expérimenté, hautement expérimenté. Sur base de cette valeur, MenuErgo propose de manière automatique des valeurs pour les autres paramètres d'évaluation que l'utilisateur peut toujours modifier. Ainsi, s'il choisit le profil « *non expérimenté* », le logiciel choisit la stratégie d'évaluation passive et le niveau de détail de présentation des messages élémentaire. Si l'utilisateur se considère comme « *moyennement expérimenté* », le logiciel lui suggère la stratégie d'évaluation mixte, où l'évaluation n'est déclenchée que toutes les 5 actions. Enfin, si l'utilisateur est « *hautement expérimenté* », le logiciel lui propose la stratégie d'évaluation active.
- la *stratégie d'évaluation*, p. ex. le système intervient-il après 4 actions ou 3 types de problèmes ?
- le *niveau de détail* de présentation des messages résultant de l'évaluation ergonomique de la conception en cours (qui peut être élémentaire, intermédiaire ou détaillé).

L'utilisateur peut aussi à tout moment régler les options d'affichage (Fig. 3) et d'évaluation (Fig. 4), qui peuvent être associées au profil.

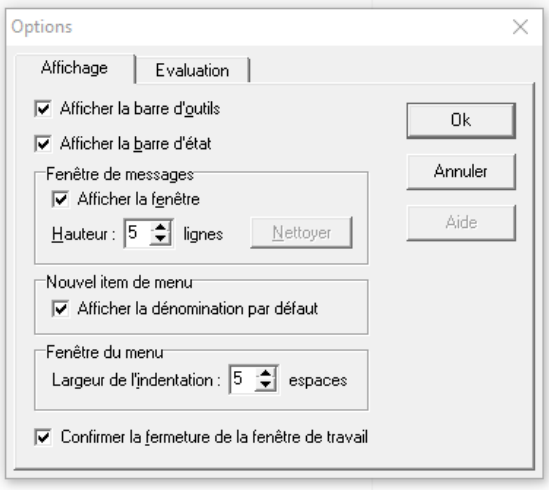

Figure 3. Options d'affichage de l'évaluation automatique.

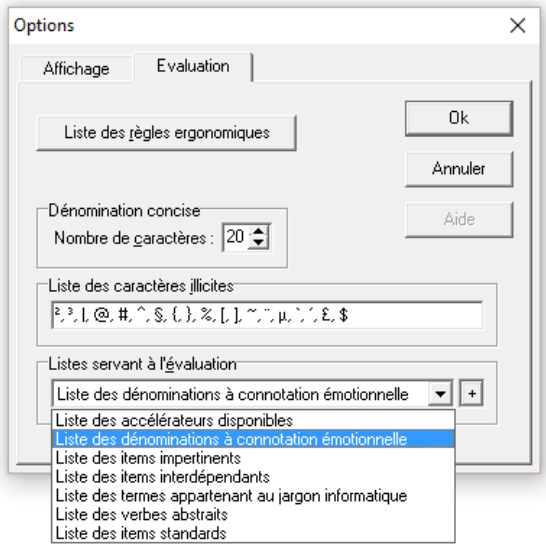

Figure 4. Options d'évaluation de l'évaluation automatique.

Les options relatives à l'évaluation automatique permettent de paramétrer différents aspects qui ont été jugés pertinents dans le cadre de la conception de menu, essentiellement basé sur l'observation et sur l'expérience, notamment des concepteurs en situation de conception [6,14,18,20]. Premièrement, l'ensemble des règles ergonomiques sujettes à évaluation peut lui-même être paramétré et enregistré de façon à n'évaluer qu'un sous-ensemble des règles implémentées, ce qui permet de cibler l'évaluation à certains aspects (Fig. 5). Ceci s'est avéré probant à plusieurs reprises [13,30].

Deuxièmement, différentes listes utiles à l'évaluation (cfr. partie basse de la Fig. 4) peuvent aussi être spécifiées, typiquement par un spécialiste en ergonomie du logiciel. Ceci couvre notamment les items de menus standardisés, tels qu'on les trouve dans les guides de style, comme IBM CUA, MS Windows ou MacOS avec tous leurs paramètres nécessaires (Fig. 6), mais aussi les termes considérés comme ergonomiquement appropriés (p. ex. les verbes d'action à faire suivre par un objet dans le paradigme « Action + objet ») ou inappropriés (p. ex. la traduction malheureuse de l'item « Abort » en anglais par « avorter » et le terme de jargon « Booter » à éviter - Fig. 7).

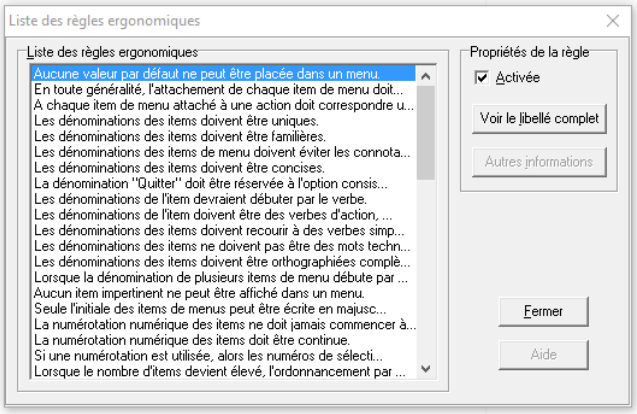

Figure 5. Paramétrage des règles ergonomiques à évaluer.

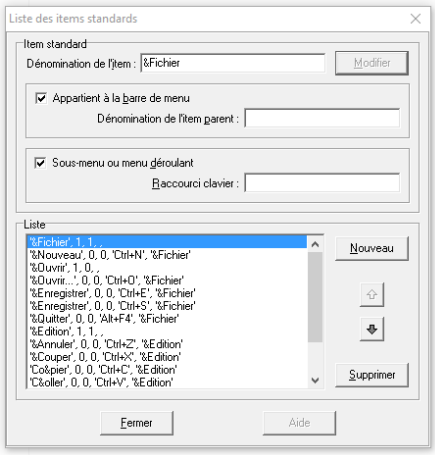

Figure 6. Définition des items de menus standardisés.

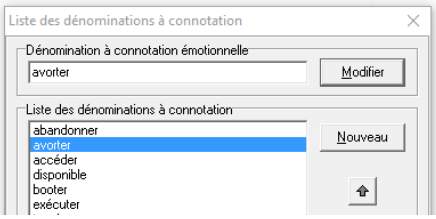

Figure 7. Définition des items appropriés ou non.

Implémentation de MenuErgo

Les grandes options de l'implémentation qui ont été décidées pour MenuErgo sont décrites ci-après, d'abord dans son architecture logicielle, puis discutées dans la manière qui a été choisie d'implémenter l'évaluation automatique des règles, un sujet de recherche complexe et non encore élucidé.

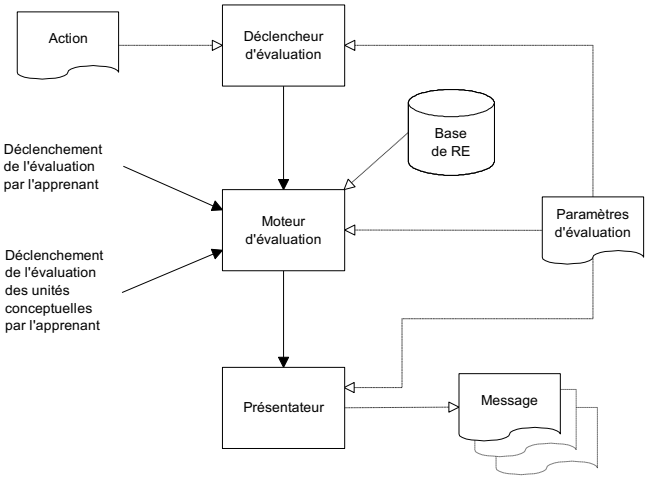

Figure 8. Paramétrage des règles ergonomiques à évaluer.

Architecture logicielle de MenuErgo. MenuErgo a été développé en Borland Pascal for Windows 7.0 (3.5), un environnement de développement intégré choisi pour son langage procédural et sa facilité d'intégration d'une base de données orientée objet pour le stockage des règles ergonomiques et de leur « parsing » pour l'évaluation automatique. L'architecture logicielle (Fig. 8) est composée des modules suivants :

- le *déclencheur d'évaluation* : il reçoit en entrée chaque action opérée par l'utilisateur sur la représentation interne ou conceptuelle de la barre de menus et de ses menus déroulants en cours de conception, puis les paramètres d'évaluation définis dans le profil (p. ex. la stratégie d'évaluation). Le déclencheur d'évaluation est chargé de déclencher une évaluation sur base de ces paramètres en envoyant un signal au moteur d'évaluation, qui analyse les actions opérées depuis la dernière évaluation déclenchée. Ces actions peuvent être : la modification de la dénomination de l'item de menu courant, l'insertion d'un menu déroulant dans la barre de menus, l'insertion d'un item de menu (dans la barre de menus, dans un menu déroulant ou dans un sous-menu), la désignation du caractère mnémonique d'accès d'un item de menu, la modification ou la suppression du caractère mnémonique d'accès d'un item de menu, la création d'un raccourci clavier, la modification d'un raccourci clavier existant, la désactivation d'un menu déroulant, l'insertion d'un item de menu, d'un sous-menu.
- le *moteur d'évaluation* : il reçoit en entrée la base de règles ergonomiques et produit un résultat d'évaluation suite à son activation par le déclencheur d'évaluation. Celui-ci ayant comptabilisé les différentes actions effectuées par l'apprenant depuis le dernier déclenchement du moteur d'évaluation, il n'intervient qu'une seule fois pour un même signal de déclenchement. Après quoi, il réinitialise, si nécessaire, le nombre d'actions, de types d'action, de problèmes ou de types de problèmes. Le résultat d'évaluation produit par le moteur d'évaluation est envoyé au présentateur qui présentera le résultat de l'évaluation.
- le *présentateur de l'évaluation* : il reçoit du moteur d'évaluation un résultat d'évaluation et produit en sortie, suite à l'analyse des paramètres d'évaluation un message qui est affiché dans la fenêtre de message (Fig. 1), permettant ainsi à l'utilisateur de corriger. Le (ou les) message(s) est (sont) affiché(s) lorsque le présentateur juge que le résultat d'évaluation doit être communiqué en fonction des paramètres issus du profil de l'utilisateur.
- la *base de règles ergonomiques* : elle contient le code des règles ergonomiques prêt à être exécuter automatiquement à la demande du moteur. Cette base a été implémentée sous la forme d'une base de données orientée objet dont chaque objet contient le code abstrait de chaque règle dont l'implémentation est discutée ci-après.

Sélection des règles ergonomiques. Premièrement, une collecte des règles ergonomiques relatives à la conception des menu a été opérée à partir de deux guides ergonomiques contenant un ensemble significatif de règles : 46 règles ergonomiques élémentaires issues du guide ergonomique de Bastien & Scapin [29] (chaque règle a été sélectionnée sur base du critère « menu ») et 402 règles issues du guide ergonomique de Vanderdonckt [32] (ces règles sont simplement la somme de toutes les règles issues des sections relatives à la sélection de menu et ont été traitées automatiquement avec l'outil SIERRA [35]). Après avoir retiré les doublons, un ensemble initial de 362 règles ergonomiques de conception de menu a été constitué. Ces règles ont donc été soumises au processus de transformation d'une règle initiale énoncée textuellement en une règle fonctionnelle automatisable en suivant les jalons de développement [35].

Programmation des règles ergonomiques. Pour évaluer une règle ergonomique automatiquement sur la représentation interne d'un menu, plusieurs paradigmes de programmation sont candidats à l'implémentation. La programmation *logique*, *fonctionnelle*, ou *par contraintes* sont des paradigmes de programmation séduisants quant à la manière logique et théorique d'exprimer une règle ergonomique (p. ex. sous la forme de clauses Prolog ou en formules de la logique des prédicats du premier ordre, sous la forme de contraintes à respecter en CSP ou sous la forme de listes fonctionnelles contraintes en Lisp). Un moteur d'évaluation autonome est alors chargé d'évaluer ces expressions. Dans KWARESMI [4], les règles ergonomiques sont spécifiées à l'aide d'une formule de la logique des prédicats du premier ordre évaluée par un moteur d'évaluation dont le fonctionnement est indépendant de la définition des règles.

Ce paradigme nous a posé finalement plusieurs problèmes : la complexité de réalisation du moteur, la difficulté d'expressivité du langage, la difficulté de gérer les méta-priorités (ce qui est particulièrement important lors de la résolution de conflits [23]), la difficulté d'exprimer des contraintes fines, la performance lors du passage à l'échelle avec un plus grand nombre de règles. C'est pourquoi nous avons préféré le paradigme de la programmation *procédurale* dans lequel chaque règle a fait l'objet d'un développement idoine en programmation orientée objet, stocké comme un objet dans la base de données.

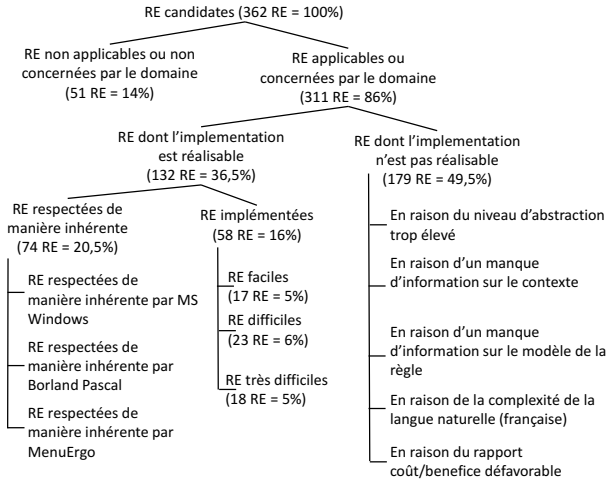

Figure 9. Explications sur les raisons d'implémentation.

Praticabilité des règles ergonomiques. Pratiquement, 58 règles ergonomiques ont pu être implémentées sur 362 règles de la sélection initiale, soit un taux de 16%. Ce taux largement inférieur à 40% [19] ou 50% [15] pose la question qu pourquoi cette faible proportion de règles implémentées par rapport à l'ensemble initial. Pour répondre à cette question, les règles ont été réparties dans quatre catégories (Fig. 9):

1. les règles non applicables ou non concernées par le domaine (51 règles = 14%) : cette catégorie regroupe les règles applicables à la conception de menu en général, mais qui ne sont pas applicables à la barre de menu ou aux menus déroulants en question ici. Par exemple, la règle « Les menus plein écran doivent être ordonnancés par ordre décroissant de fréquence » ne concerne pas le type de menu traité par MenuErgo.
2. les règles dont l'implémentation n'est pas réalisable (179 règles = 49,5%) : cette catégorie regroupe les règles dont l'évaluation automatique est impossible à réaliser pour différentes raisons telles que l'expression à un haut niveau d'abstraction non interprétable par un automate (par exemple, la règle « la conception de menu peut faire appel à une métaphore du mini-monde »), le recours à des informations contextuelles non disponibles (par exemple, les règles « la complexité d'un menu doit refléter le niveau d'expérience de l'utilisateur », « le nombre d'items de menu doit tenir compte de la résolution d'écran de la plateforme »), la difficulté d'interprétation du langage naturel (par exemple, la règle « les messages d'erreur de menu doivent éviter de faire appel à l'humour »).
3. les règles respectées de manière inhérente à la plateforme (74 règles = 20,5%) : cette catégorie regroupe les règles qui sont directement respectées par le système d'exploitation (ici, MS Windows) et/ou par l'environnement de développement (ici, Borland Pascal). Par exemple, la règle « Chaque item de menu doit être activé ou désactivé » est par nature respectée par MenuErgo puisqu'une option de conception la couvre. De même, la règle « chaque menu déroulant doit être attaché à un seul item de la barre de menu » est intrinsèquement assurée.
4. les règles dont l'implémentation a été réalisée (58 règles = 16%) : cette catégorie regroupe les règles dont l'implémentation a pu être réalisée avec trois niveaux de difficulté différents :
   a. les règles faciles, qui sont entièrement réalisables par un nombre de lignes de code raisonnable. Par exemple, la règle « la barre de menu ne devrait pas comporter plus de 9 items » a nécessité 20 LOC.
   b. les règles moyennes, qui sont implémentées partiellement par un nombre de lignes de code important. Par exemple, « les mnémoniques des items de menus doivent être phonétiquement distincts » a eu besoin de 43 LOC pour être réalisée.
   c. les règles complexes, qui sont implémentée par une restriction de celle-ci par un nombre élevé de lignes de code. Par exemple, la règle « les items de menu ne doivent pas faire appel à un verbe abstrait » se borne à vérifier qu'aucun terme appartenant à la liste des verbes abstraits n'est présent sous une forme ou sous une autre dans les dénominations. Il faut 87 LOC pour assurer cette règle.

Les catégories 1 à 3 sont indépendantes de l'outil MenuErgo car elles concernent des raisons situées en dehors du paradigme de programmation choisi et de l'environnement de développement utilisé pour MenuErgo.

ETUDE QUALITATIVE EXPLORATOIRE

Comme il n'existait pas d'outil offrant un point de référence utile pour une comparaison, il est apparu pertinent de conduire une étude plutôt qualitative que quantitative destinée à récolter les réactions des concepteurs avec MenuErgo.

Procédure

Il a été demandé aux concepteurs de concevoir un menu complet pour une application interactive de gestion de CD-ROM et de DVD, un domaine dont la sémantique a été estimée abordable et ne nécessitant pas de connaissances particulières au domaine. Afin que le point de départ soit le même pour tous les participants, ceux-ci ont reçu les spécifications suivantes:

- un document hypertexte comprenant un énoncé textuel de l'étude de cas, un modèle des données sous la forme d'un schéma entité-association (Fig. 10)
- un modèle de tâches, exprimant la vue de l'utilisateur
- un modèle de la statique des traitements, spécifiant les fonctions sémantiques du noyau fonctionnel
- un modèle de la dynamique des traitements, spécifiant les conditions d'enchaînement des fonctions.

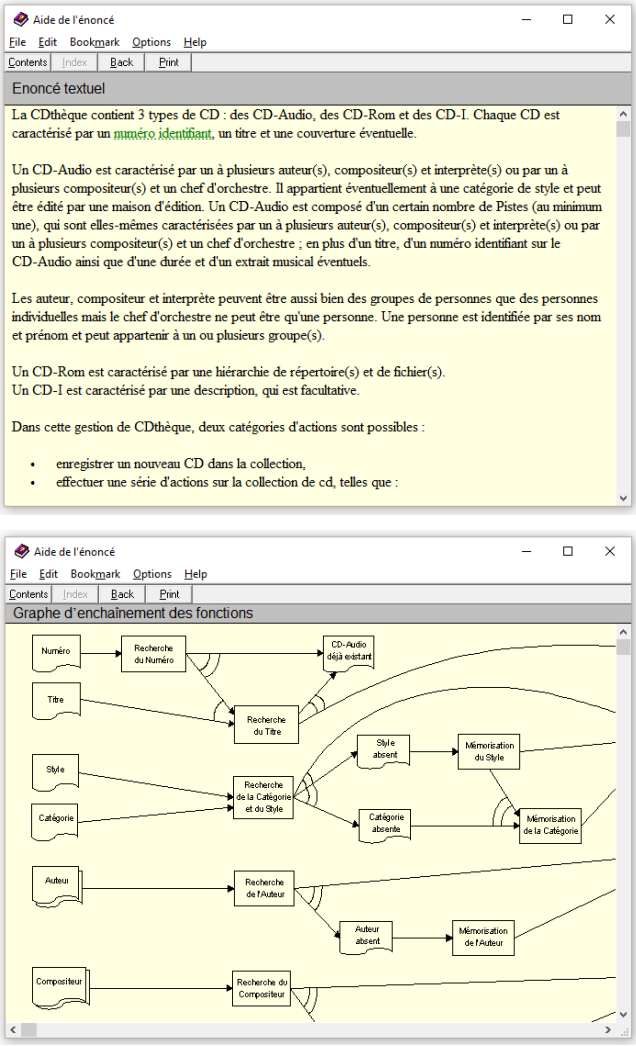

Figure 10. Enoncé textuel et modèle de tâche de l'étude de cas.

Seul l'énoncé textuel est indispensable à la réalisation de l'expérimentation, les autres documents étant présents car ils sont souvent utilisés par ces concepteurs. Une demi-heure leur a été accordée pour concevoir la barre de menus, les menus déroulants, les sous-menus de l'application multimédia. Aucune formation ou période de familiarisation n'a été octroyée afin de reproduire des conditions exigeantes.

Participants

Neuf concepteurs (9 hommes, de 25 à 43 ans) d'applications interactives ont été sélectionnés à partir d'une liste de distribution de concepteurs travaillant dans différentes organisations, comme des agences web, des services informatiques, des départements informatique et organisation. Il n'a pas été tenu compte de leur niveau d'expérience afin de ne pas biaiser l'échantillonnage. Chaque concepteur s'est déclaré suffisamment compétent quant aux modèles invoqués.

Questionnaire

A l'issue de la procédure, un questionnaire a été administré à chaque participant en trois parties sous la forme de questions fermées à choix simple dont la réponse est exprimée sur une échelle de Likert à 5 points toujours croissante (1=le plus faible, 2=faible, 3=moyen, 4=élevé, 5=le plus élevé):

1. leur *profil* de chaque concepteur (Q1-Q6): quel est votre niveau d'expérience dans l'utilisation de la plate-forme MS Windows en général (Q1), dans cette version de Windows en particulier (Q2), dans la compréhension d'un schéma Entité-Association (Q3), dans la conception de barres de menus et de menus déroulants (Q4), dans la conception d'autres types de menus (Q5), dans la conception d'interfaces (Q6)?
2. leur *apprentissage perçu* à l'issue de l'expérience (Q7-Q11): qu'avez-vous appris globalement (Q7), au niveau de la barre de menus (Q8), au niveau des menus déroulants (Q9), au niveau des menus Windows (Q10), au niveau des règles ergonomiques (Q11)?
3. leur *évaluation subjective* de MenuErgo (Q12-Q16): êtes-vous satisfait de l'interface de MenuErgo en général (Q12), de sa facilité d'utilisation (Q13), de la compréhension de l'expérience (Q14), de la qualité des messages produits (Q15), de la qualité de l'aide fournie (Q16)?

Au terme du remplissage de ce questionnaire qui n'a pas été contraint en temps, chaque concepteur a été interviewé de manière semi-dirigée personnellement en lui demandant au maximum trois points positifs et trois points négatifs assortis de suggestions d'amélioration. Ces suggestions ont été consignées et triées en trois catégories en fonction de leur niveau de priorité: indispensables, souhaitables et optionnelles.

Résultats et discussion

La figure 11 représente graphiquement la distribution des réponses aux seize questions par les neuf concepteurs selon le codage de couleur suivant : rouge=le plus faible, orange= faible, jaune = moyen, vert clair = élevé, vert foncé = le plus

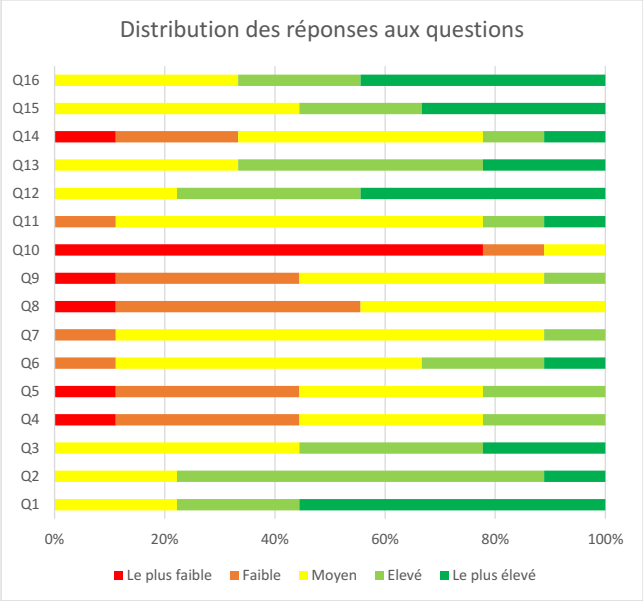

Figure 11. Distribution des réponses aux questions.

élevé. Pour les questions Q7 à Q11, les « mauvaises » couleurs sont à interpréter par un apprentissage limité, pas nécessairement par un « mauvais » apprentissage.

Concernant le *profil* des concepteurs, on peut estimer qu'ils bénéficient d'une excellente connaissance du système d'exploitation Windows en général (Q1 : μ=4,33, M=5, σ=0,82), un peu moins de la version de Windows utilisée dans l'expérience (Q2 : μ=3,89, M=4, σ=0,57). Leurs connaissances en modélisation des données s'avèrent suffisantes (Q3 : μ= 3,78, M=4, σ=0,79). Par contre, ils estiment que leurs connaissances en matière de conception de barres de menu et menus déroulants (Q4 : μ=2,67, M=3, σ=0,94) et d'autres types de menus (Q5 : μ=2,67, M=3, σ=0,94) sont inférieures à leurs connaissances en matière de conception d'interface en général (Q6 : μ=3,33, M=3, σ=0,82).

Concernant l'*apprentissage perçu* par les concepteurs, ils expriment avoir appris globalement quelque chose (Q7 : μ=3,00, M=3, σ=0,47), mais moins en matière de barres de menus (Q8 : μ=2,33, M=2, σ=0,67) et de menus déroulants (Q9 : μ=2,56, M=3, σ=0,83) et encore moins pour les menus standardisés par MS Windows (Q10 : μ=1,33, M=1, σ=0,67). Ceci est explicable par le fait que les menus standardisés par MS Windows, p. ex. « Fichier », « Edition », « Aide » sont eux-mêmes issus de la norme IBM CUA que l'on retrouve sur tous les systèmes d'exploitation modernes. Curieusement, les concepteurs ont le sentiment d'avoir plus appris en matière de règles ergonomiques (Q11 : μ=3,22, M=3, σ=0,79) que pour tous les autres aspects. Ceci semble indiquer qu'ils connaissent sans doute bien les bonnes pratiques, mais qu'ils ne connaissaient pas nécessairement les règles ergonomiques associées, ce qui corrobore le résultat avancé préalablement par Chevalier *et al.* [14].

Concernant l'évaluation de la satisfaction subjective ressentie par les neuf concepteurs, ils rendent compte d'une grande satisfaction à utiliser l'interface de MenuErgo (Q12 : μ=4,22, M=4, σ=0,79) et de son aide (Q16 : μ=4,11, M=4, σ=0,87) et des messages produits par le présentateur de l'évaluation (Q15 : μ=3,89, M=4, σ=0,87). La facilité d'utilisation est jugée globalement satisfaisante (Q13 : μ=3,89, M=4, σ=0,79). En revanche, l'exercice a été très mal estimé (Q14 : μ=2,89, M=3, σ=1,10). Les résultats de l'interview expliquent partiellement cette mauvaise performance : le temps d'utilisation a été trop court (30 minutes seulement), les concepteurs ont eu tendance à lire trop la documentation jugée volumineuse et n'ont pas bénéficié d'assez de temps pour l'assimiler. Tous les écarts types sont inférieurs à l'unité à part pour la question Q14 (σ=1,10) : les neuf concepteurs semblent concourir dans leurs avis dans l'ensemble, à part la question trop court de l'expérience, à l'une ou l'autre exception près. En prenant en compte l'ensemble des observations positives-négatives et des suggestions, on observe ceci :

1. En ce qui concerne la visualisation multi-vues, les participants ont fortement apprécié de pouvoir visualiser la barre de menus et les menus déroulants qu'ils étaient en train de construire. Ils avaient ainsi un retour d'information immédiat quant à la réalisation de leur exercice. Cette caractéristique, propre à un apprentissage par l'action, semble donc être bien exploitée par MenuErgo. Le paramétrage de l'évaluation, jugé indispensable, a été reconnu par les participants en fonction de leur niveau d'expérience : plus le niveau d'expérience était élevé, plus le concepteur avait tendance à recourir à une stratégie active (sous son contrôle), contrairement aux concepteurs moins expérimentés préférant la stratégie passive.
2. En ce qui concerne les paramètres d'évaluation, certains évaluateurs les ont apprécié dans leur ensemble tandis que d'autres ont surtout été séduits par l'une ou l'autre caractéristique bien précise de ces paramètres : un évaluateur a apprécié le choix de la stratégie d'évaluation, un second a préféré la détermination des paramètres d'évaluation sur base du profil de l'utilisateur.
3. En ce qui concerne la vue conceptuelle, il nous semble qu'elle doit effectivement être conservée puisqu'elle permet de visualiser d'une part le menu en cours de conception selon une autre vue que la vue externe, et d'autre part, de visualiser, de manière permanente, un menu déroulant ou un sous-menu entier tout en poursuivant sa réalisation sur la vue conceptuelle ou interne.
4. D'autres caractéristiques du logiciel, moins importantes dans un environnement d'apprentissage mais néanmoins nécessaires, ont aussi été citées par les évaluateurs : les raccourcis clavier associés aux opérations d'insertion des différents types d'item de menu et les fonctions d'édition copier-coller.

Les suggestions jugées indispensables sont : les messages résultant d'une évaluation ergonomique de l'exercice en cours devraient être contextualisés, la présentation des messages d'évaluation pourrait être améliorée, les niveaux de détail de présentation intermédiaire et détaillé pourraient être contextualisés, on pourrait étendre l'enregistrement d'un fichier dans un format autre que UsiXML ou un fichier de ressource Windows, afin de l'utiliser dans une autre plateforme ou de récupérer un menu existant, un historique des messages d'évaluation permettrait de mieux suivre, l'évaluation devrait se faire aussi par désignation d'une unité conceptuelle particulière (par exemple, évaluer le menu « Untel »). Toute unité conceptuelle devrait pouvoir aussi être « gelée » ou « dégelée » lorsqu'elle doit sortir de l'évaluation.

Les suggestions souhaitables sont : proposer une vue externe sur différentes plateformes avec différentes résolutions, transformer un item de menu en menu déroulant ou sous-menu et réciproquement, ajouter des menus contextuels notamment dans le cadre d'interface à documents multiples. Les suggestions optionnelles sont : autoriser un tirer-lâcher pour manipuler directement les items, offrir un menu d'édition contextuel, réafficher à la demande les messages passés relatifs à un item, produire un signal sonore pour une erreur en fonction de son facteur d'impact,…

CONCLUSION

Cet article a présenté MenuErgo, un environnement logiciel de conception de barre de menu, de menus déroulants, et de sous-menu d'une interface graphique par évaluation automatique de 58 règles ergonomiques sur 362 relatives au menu selon quatre stratégies d'évaluation différentes (active, passive, mixte, par unités conceptuelles) qui sont paramétrables en fonction du niveau d'expérience de l'utilisateur. Une première étude qualitative exploratoire menée avec neuf sujets a mis en évidence les réactions des concepteurs suite à l'utilisation de MenuErgo sur une étude de cas.

Une nouvelle expérimentation pourrait se centrer sur l'utilisation des quatre stratégies d'évaluation et de leur paramétrisation en fonction du profil des concepteurs afin de valider les paramétrages. Elle pourrait également couvrir l'intégration avec les données d'usage en fonction du contexte. Comme MenuOptimizer [3] qui optimise l'ordonnancement des items de menu suivant un algorithme de type « fourmi », les items de menu pourraient être associés à des données qui pourraient elles-mêmes être prises en compte dans l'évaluation, comme dans RITA [10]. L'étude pourrait alors se baser sur un enregistrement horodaté des actions des concepteurs en fonction de leur évolution dans le temps au cours de l'expérience. Ceci permettrait de dégager différentes approches de conception de menu (p. ex. en largeur d'abord, en profondeur d'abord, sur des aspects syntaxiques ou sémantique d'abord) et d'en déduire une interface d'aide à la conception plus intelligente. Un modèle multi-paramétrique de la qualité de chaque menu, prenant en compte chaque critère de manière pondérée, pourrait ainsi être mis au point.

La limite majeure de l'évaluation automatique des règles reste tout de même leur caractère uniquement syntaxique puisqu'elles ne tiennent pas compte du contexte d'usage. On pourrait y remédier en incorporant un modèle du contexte

(utilisateur, plateforme, environnement), mais ceci engendre une autre complexité. Les règles restantes exigent un effort de développement bien au-delà du bénéfice potentiel de leur incorporation. Cependant, une fois cette implémentation réalisée, le bénéfice devrait perdurer.

REFERENCES

1. Walter de Abreu Cybis, Dominique L. Scapin, and Marcelo Morandini. 2005. ErgoManager: A UIMS for monitoring and revising user interfaces for Web sites. In *Proc. of 1st Int. Conf. on Web Information Systems and Technologies* (WEBIST'2005), 281–286. http://dx.doi.org/10.5220/0001230602810286
2. Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2009. MenUA: A Design Space of Menu Techniques. Consulté le 2 mai 2016 à http://www.gillesbailly.fr/menua/
3. Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: interactive optimization of menu systems. In *Proc. of the 26th ACM Symposium on User interface software and technology* (UIST'2013). ACM Press, New York, 331–342. http://dx.doi.org/10.1145/2501988.2502024
4. Abdo Beirekdar, Jean Vanderdonckt, and Monique Noirhomme-Fraiture. 2002. A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. In *Proc. of 4th Int. Conf. on Computer-Aided Design of User Interfaces* (CADUI'2002). Kluwer Academic Publisher, Dordrecht, 337–348. http://dx.doi.org/10.1007/978-94-010-0421-3_29
5. François Bodart, Anne-Marie Hennebert, Jean-Marie Leheureux, and Jean Vanderdonckt. 1994. Towards a Dynamic Strategy for Computer-Aided Visual Placement. In *Proc. of 2nd ACM Int. Working Conf. on Advanced Visual Interfaces* (AVI'1994). ACM Press, New York, 1994, 78-87. http://dx.doi.org/10.1145/ 192309. 192328
6. Nathalie Bonnardel, Tamara Sumner, and Benedikte Harstad Kallak, 1997. The Cognitive Ergonomics of Knowledge-Based Design Support Systems. In *Proc. of ACM Conf. on Human Aspects in Computing Systems* (CHI'97). ACM Press, New York, 22–27. http://dx.doi.org/10.1145/258549.258613
7. Christian Bastien and Dominique L. Scapin. 1995. Evaluating a User Interface with Ergonomic Criteria, *International Journal of Human-Computer Interaction* 7, 2, 1995, 105–121. http://dx.doi.org/10.1080/10447319509526114
8. Syrine Charfi, Dominique L. Scapin, and Emmanuel Dubois. 2008. Identification et prise en compte de propriétés ergonomiques pour la modélisation et la conception de SIM. In *Proc. of 20th Int. Conf. on Interaction Homme-Machine* (IHM'2008). ACM Press, New York, 55–62. http://dx.doi.org/10.1145/1512714.1512726
9. Selem Charfi, Abdelwaheb Trabelsi, Houcine Ezzedine, and Christophe Kolski. 2012. Graphical Controls Based Environment for User Interface Evaluation. In *Proc. of the 4th Int. Conf. on Human-Centered Software Engineering* (HCSE'2012). Springer, Berlin, 360–367. http://dx.doi.org/10.1007/978-3-642-34347-6_28
10. Selem Charfi, Houcine Ezzedine, and Christophe Kolski. 2015. RITA: a useR Interface evaluaTion frAmework. *Journal of Universal Computer Science* 21,4, 526–560. http://dx.doi.org/10.3217/jucs-021-04-0526
11. Selem Charfi, Abdelwaheb Trabelsi, Houcine Ezzedine, and Christophe Kolski. 2014. Widgets Dedicated to User Interface Evaluation. *International Journal of Human-Computer Interaction* 30, 5 (2014), 408–421. http://dx.doi.org/10.1080/10447318.2013.873280
12. Selem Charfi, Houcine Ezzedine, and Christophe Kolski. 2014. Evaluation Based Graphical Controls: A Contribution to Mobile User Interface Early Evaluation. In *Proc. of 16th Int. Conf. on Human-Computer Interaction* (HCI International 2014). Lecture Notes in Computer Science, Vol. 8512. Springer, Berlin, 114–123. http://dx.doi.org/10.1007/978-3-319-07227-2_12
13. Selem Charfi and Houcine Ezzedine. 2014. Evaluation Tools Through User Participation Techniques: Features, Limitations, and New Perspectives. In *Proc. of Int. Conf. on Advanced Logistics and Transport* (ICALT'2014). IEEE Press, Piscataway, 13–18. http://dx.doi.org/10.1109/ICAdLT.2014.6864074
14. Aline Chevalier, Nicolas Fouquereau, and Jean Vanderdonckt. 2009. The influence of a knowledge-based system on designers' cognitive activities: a study involving professional web designers. *Behaviour & Information Technology* 28,1 (2009), 45–62. http://dx.doi.org/10.1080/01449290701288833
15. Michael Cooper, Quentin Limbourg, Céline Mariage, and Jean Vanderdonckt. 1999. Integrating Universal Design into a Global Approach for Managing Very Large Web Sites. In *Proc. of the 5th ERCIM Workshop on User Interfaces for All* (UI4All'99). GMD Report 74, GMD - Forschungszentrum Informationstechnik GmbH, Sankt Augustin, 131–150.
16. Charles-Eric Dessart, Vivian Genaro Motti, and Jean Vanderdonckt. 2012. Animated Transitions between User Interface Views. In *Proc. of ACM Int. Working Conf. on Advanced Visual Interfaces* (AVI'2012). ACM Press, New York, 341–348. http://dx.doi.org/10.1145/2254556.2254623
17. Chi Dung Tran, Houcine Ezzedine, and Christophe Kolski. 2013. EISEval, a generic reconfigurable environment for evaluating agent-based interactive systems. *International Journal of Human-Computer Studies* 71, 6 (June 2013), 725–761. http://dx.doi.org/10.1016/j.ijhcs.2013.02.001
18. Michael Ericsson, Magnus Baurén, Jonas Löwgren, and Yvonne Wærn. 1998. A Study of Commenting Agents as Design Support. In *Proc. of Conference Summary on Human Factors in Computing Systems* (CHI'1998), ACM Press, New York, 225-226. http://dx.doi.org/10.1145/286498.286714
19. Christelle Farenc, Véronique Liberati, and Marie-France

Barthet. Automatic Ergonomic Evaluation: What are the limits ?. In *Proc. of 2nd International Workshop on Computer-Aided Design of User Interfaces* (CADUI'1996). Presses Universitaires de Namur, Namur, 159–170.

20. Gerhard Fischer, Andreas C. Lemke, Thomas W. Mastaglio, and Anders I. Mørch. 1991. The Role of Critiquing in Cooperative Problem Solving. *ACM Trans. on Information Systems* 9, 2 (April 1991), 123–151. http://dx.doi.org/10.1145/123078.128727
21. Kryzstof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proc. of the Working Conference on Advanced Visual Interfaces* (AVI'2006), ACM Press, 201–208. http://dx.doi.org/10.1145/1133265.1133306
22. Melody Y. Ivory and Marti A. Hearst. 2001. The state of the art in automating usability evaluation of user interface. *ACM Computing Surveys*, 470–516. http://dx.doi.org/10.1145/503112.503114
23. Llúcia Masip, Célia Martinie, Marco Winckler, Philippe A. Palanque, Toni Granollers, and Marta Oliva. 2012. A Design Process for Exhibiting Design Choices and Trade-Offs in (Potentially) Conflicting User Interface Guidelines. In *Proc. of 4th Int. Conf. on Human-Centered Software Engineering* (HCSE'2012). Springer, 53–71. http://dx.doi.org/10.1007/978-3-642-34347-6_4
24. Souichi Matsui and Seiji Yamada. 2008. Genetic algorithm can optimize hierarchical menus. In *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems* (CHI'2008). ACM Press, NY, 1385–1388. http://dx.doi.org/10.1145/1357054.1357271
25. Marcelo Morandini, Walter de Abreu Cybis, and Dominique L. Scapin. 2009. A Prototype to Validate ErgoCoIn: A Web Site Ergonomic Inspection Technique. In *Proc. of 13th Int. Conf. on Human-Computer Interaction* (HCI International 2009). Lecture Notes in Computer Science, Vol. 5610. Springer, Berlin, 339–348. http://dx.doi.org/10.1007/978-3-642-02574-7_38
26. Kumiyo Nakakoji, Yunmen Ye, and Yasuhiro Yamamoto. 2010. Supporting Expertise Communication in Developer-Centered Collaborative Software Development Environments. In *Collaborative Software Engineering*. Ivan Mistrík, John Grundy, André Hoek, Jim Whitehead (Eds.). Springer, Berlin, 219–236. http://dx.doi.org/10.1007/978-3-642-10294-3_11
27. Kent L. Norman. 1991. The Psychology of Menu Selection, Designing Cognitive Control at the Human/Computer Interface, Ablex Publishing Corporation, Norwood, New Jersey.
28. Philippe A. Palanque, Eric Barboni, Célia Martinie, David Navarre, and Marco Winckler. A model-based approach for supporting engineering usability evaluation of interaction techniques. In *Proc. of ACM Int. Conf. on Engineering Interactive Computing Systems* (EICS' 2011). ACM Press, New York, 21–30. http://dx.doi.org/10.1145/1996461.1996490
29. Dominique L. Scapin. *Guide ergonomique de conception des interfaces homme-ordinateur*. Rapport INRIA n°77, Institut National de Recherche en Informatique et en Automatique, Le Chesnay, octobre 1986.
30. Antonio Giovanni Schiavone and Fabio Paternò. 2015. An extensible environment for guideline-based accessibility evaluation of dynamic Web applications. *Universal Access in the Information Society* 14, 1 (2015), 111–132. http://dx.doi.org/10.1007/s10209-014-0399-3
31. Jérémy Spoidenne and Jean Vanderdonckt. 2003. MenuSelector: Automated Generation of Dynamic Menus with Guidelines Support. In *Proc. of 10th Int. Conf. on Human-Computer Interaction* (HCI International'2003). Vol. 1, J. Jacko, C. Stephanidis (eds.). Lawrence Erlbaum Associates, Mahwah, 233–237.
32. Jean Vanderdonckt. *Guide ergonomique des interfaces homme-machine*. Presses Universitaires de Namur, Namur, 1994.
33. Jean Vanderdonckt. 1999. Computer-Aided Design of Menu Bar and Pull-Down Menus for Business Oriented Applications. In *Proc. of 6th Int. Workshop on Design, Specification, Verification of Interactive Systems* (DSV-IS'99). D.J. Duke, A.R. Puerta (Eds.). Springer, Vienna, 73–88. http://dx.doi.org/ 10.1007/978-3-7091-6815-8_7
34. Jean Vanderdonckt and Abdo Beirekdar. Automated Web Evaluation by Guideline Review. *Journal of Web Engineering* 4, 2 (June 2005), 102–117. http://dl.acm.org/citation.cfm?id=2011168
35. Jean Vanderdonckt. 1999. Development Milestones towards a Tool for Working with Guidelines. *Interacting with Computers* 12, 2 (December 1999), 81-118. http://dx.doi.org/10.1016/S0953-5438(99)00019-3
36. Joseph Xiong and Marco Winckler. 2008. An investigation of tool support for accessibility assessment throughout the development process of web sites. *Journal of Web Engineering* 7, 4 (December 2008), 281–298. http://dl.acm.org/citation.cfm?id=2011280
37. Joseph Xiong, Mouhamed Diouf, Christelle Farenc, and Marco Winckler. 2006. Automating Guidelines Inspection: From Web site Specification to Deployment. In *Proc. of 5th Int. Workshop on Computer-Aided Design of User Interfaces* (CADUI'2006). Springer, Berlin, 273–286. http://dx.doi.org/ 10.1007/978-1-4020-5820-2_23