

Vers l'identification des fonctions d'utilisabilité à fort impact sur la conception

Elodie Bouzekri

elodie.bouzekri@estia.fr

Univ. Bordeaux, ESTIA INSTITUT DE TECHNOLOGIE, Bidart, France

MOTS-CLÉS

Fonctions d'utilisabilité, conception, techniques d'interaction

ACM Reference Format:

Elodie Bouzekri. 2023. Vers l'identification des fonctions d'utilisabilité à fort impact sur la conception. In *IHM '23 : 34^e conférence Francophone sur l'Interaction Humain-Machine, April 03–07, 2023, Troyes, France*. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Recenser les fonctions d'utilisabilité peut être utile afin de permettre un meilleur accompagnement des concepteurs et développeurs de systèmes interactifs. Certaines fonctions d'utilisabilité découlent directement de leur heuristique associée (ex. *undo global* pour fournir des sorties explicites), d'autres sont plus difficilement identifiables par la multiplicité des stratégies d'implémentation possible (ex. les mécanismes de retour d'information : immédiat, pro-actifs, explicatifs, etc.). Si les méthodes se basant sur une approche centrée utilisateur ont pour but de soutenir l'utilisabilité, le recensement de ces fonctions particulières pourrait les enrichir. Parmi ces méthodes, certaines identifient la conformité de systèmes interactifs avec les heuristiques d'utilisabilité par l'analyse de logs basés sur les modèles [3]. Une autre approche possible pourrait être d'étendre les approches recommandant des commandes aux développeurs de systèmes interactifs [2] aux fonctions d'utilisabilité.

2 HEURISTIQUES ET IDENTIFICATION DES FONCTIONS D'UTILISABILITÉ

Afin d'identifier ces fonctions d'utilisabilité, une piste possible est de recenser les solutions proposées par les concepteurs de systèmes interactifs afin de soutenir les heuristiques d'utilisabilité. Cependant, comme ces fonctions ont pour objectif de soutenir l'utilisabilité du système et non pas directement la finalité de celui-ci, elles peuvent être incomprises ou sujettes à ambiguïté lors de la spécification des exigences du système. Ayant identifié ce problème et afin de faciliter la compréhension entre parties prenantes et la prise en compte des heuristiques d'utilisabilité dès la phase de spécification des exigences, Juristo et al. [4] ont identifié et hiérarchisé certaines de ces heuristiques. Les heuristiques sélectionnées sont celles ayant un fort impact sur la conception et le développement du système. Les

auteurs ont proposé un pattern pour la traduction en exigences de la prise en compte de ces heuristiques. Celles-ci concernent : le retour d'information (statut du système, retour sur l'interaction, barre de progrès), le *undo/annulation* (*undo global*, *undo objet-spécifique*, annulation de l'exécution, retour à un état particulier), la prévention et correction des erreurs d'entrées de l'utilisateur, l'exécution pas-à-pas, le profil utilisateur (préférences et favoris), les différents niveaux d'aide, l'agrégation de commandes et les raccourcis. Le retour sur l'interaction, l'annulation de l'exécution d'une action ou de l'application et les préférences de l'utilisateur ont été évaluées comme ayant un fort impact sur la conception et le développement du système : elles requièrent la mise en place d'une architecture spécifique et peuvent impacter autant la présentation, le dialogue et que le noyau fonctionnel du système. Carjaval et al. [1] ont proposé des recommandations centrées sur l'implémentation de ces heuristiques afin de les traduire en fonctions. Par exemple, pour la prévention et la correction des erreurs d'entrées de l'utilisateur (*warning*), l'application de leur méthode résulte en un diagramme de classe contenant des classes dédiées et proposant les fonctions suivantes : *deleteTask()*, *displayWarning()* et *undo()*. Ces fonctions peuvent être considérées comme des fonctions d'utilisabilité.

3 CONCLUSION

Les travaux présentés se concentrent sur les heuristiques ayant un fort impact sur les efforts de conception et de développement. Leur implémentation soutenant l'utilisabilité, ces travaux les recensant peuvent constituer un point de départ afin d'identifier les fonctions d'utilisabilité requérant le plus d'effort pour d'être implémentées. Également, de nombreuses heuristiques ont été développées pour des contextes particuliers (ex. jeux vidéo, mobile, site web, etc.) permettant ainsi d'identifier des fonctions d'utilisabilité 'standards' de celles spécifiques à ces contextes particuliers. Cependant, se limiter à cette approche pourrait limiter le recensement aux fonctions les plus courantes et les plus étudiées.

RÉFÉRENCES

- [1] Laura Carvajal, Ana M. Moreno, Maria-Isabel Sanchez-Segura, and Ahmed Seffah. 2013. Usability through Software Design. *IEEE Transactions on Software Engineering* 39, 11 (Nov. 2013), 1582–1596. <https://doi.org/10.1109/TSE.2013.29>
- [2] Marko Gasparic, Andrea Janes, Francesco Ricci, and Marco Zanellati. 2017. GUI Design for IDE Command Recommendations. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI '17)*. Association for Computing Machinery, New York, NY, USA, 595–599. <https://doi.org/10.1145/3025171.3025200>
- [3] Philippe Palanque, Eric Barboni, Célia Martinie, David Navarre, and Marco Winckler. 2011. A model-based approach for supporting engineering usability evaluation of interaction techniques. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. Association for Computing Machinery, New York, NY, USA, 21–30. <https://doi.org/10.1145/1996461.1996490>
- [4] Francy D. Rodriguez, Silvia T. Acuña, and Natalia Juristo. 2015. Design and programming patterns for implementing usability functionalities in web applications. *Journal of Systems and Software* 105 (July 2015), 107–124. <https://doi.org/10.1016/j.jss.2015.04.023>

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or professional use, not for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IHM '23, April 03–07, 2023, Troyes, France

© 2023 Association for Computing Machinery.

ACM ISBN XXXX-X-XXXX-XXXX-X/XX/XX...\$15.00

<https://doi.org/10.1145/XXXXXXX.XXXXXXX>

2023-01-31 11:00. Page 1 of 1–1.