Intermodal Improvement

Nudging Users to Use Keyboard Shortcuts

Raursø, Niels Erik; Kappel Persson, Mikkel ; Garðarsson, Kristinn Bragi ; Mazán, Daniel ; Andreasen, Simon ; Avotina, Elizabete ; Ventegodt, Alex ; Triantafyllou, Evangelia

Published in:
Design, User Experience, and Usability. Interaction Design - 9th International Conference, DUXU 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Proceedings

DOI (link to publication from Publisher):
10.1007/978-3-030-49713-2_30

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Raursø, N. E., Kappel Persson, M., Garðarsson, K. B., Mazán, D., Andreasen, S., Avotina, E., Ventegodt, A., & Triantafyllou, E. (2020). Intermodal Improvement: Nudging Users to Use Keyboard Shortcuts. In A. Marcus, & E. Rosenzweig (Eds.), *Design, User Experience, and Usability. Interaction Design - 9th International Conference, DUXU 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Proceedings* (Vol. 12200, pp. 439-451). Springer. https://doi.org/10.1007/978-3-030-49713-2_30

Intermodal Improvement: Nudging Users to Use Keyboard Shortcuts

Niels Erik Raursø[0000−0002−2140−9089], Mikkel Kappel Persson[0000−0002−6851−2008], Kristinn Bragi Garðarsson[0000−0001−6915−2932], Daniel Mazáň[0000−0001−5180−113X], Simon Andreasen[0000−0002−3196−1109], Elizabete Avotiņa[0000−0003−1715−0163], Alex Ventegodt[0000−0002−1124−1665], and Evangelia Triantafyllou[0000−0001−8764−2118]

Department of Architecture, Design and Media Technology, Aalborg University, Copenhagen, Denmark
`{nraurs18, mperss18, kgarda18, dmazan18, sban18, eavoti18, avente18 }@student.aau.dk, evt@create.aau.dk`

Abstract. Keyboard shortcuts have been proven to be the most efficient method of issuing commands in computer software. Using the mouse in graphical user interfaces provides an intuitive but slow method for executing functions in a software. Many fail to make the transition to faster modalities, such as keyboard shortcuts. This is not just the case for novices, but also users with years of experience. This study examines the research on this behavior, as well as how the concepts of nudging and ambient suggestion can be used to actively encourage and support the usage of keyboard shortcuts. Based on this research, a design is proposed and implemented in a simple word processor application. A user experience evaluation was done, by having participants perform writing and formatting tasks inside the application. Using the Microsoft Reaction Card Method followed by a semi-structured interview, the users elaborated on their experience. In the interviews topics and questions of motivation, distraction, and annoyance were raised. The results showed that most participants found the system convenient and helpful in learning shortcuts without being too obtrusive. There are promising first indications of it having potential in promoting the usage of keyboard shortcuts, however further research is required in order to make any generalizations.

Keywords: User-Interface Design · Keyboard Shortcuts · Nudging · Ambient Suggestion · User Experience

1 Introduction

KeyBoard Shortcuts (KBSs) have been proven to be the most efficient method of issuing commands in computer software, nevertheless they are remarkably underutilized [17, 23]. This is not solely the case for novices, but also for users with years of experience within a given software. While the efficiency gains might appear small in isolation, with regular use of a software the time saved will accumulate considerably over time.

The behavior of adhering to methods already learned is understandable, as learning a new method requires further effort, and could take away time and focus from the task at hand. As a consequence, users can underestimate the efficiency gains of adopting faster methods [18]. This is a fundamental consequence of Graphical User Interfaces (GUIs). Presenting options in a visually salient manner is useful for novices, but may avert users from adopting expert methods, as users are biased towards incremental interactive actions [11]. If users choose to adopt another modality, e.g. KBSs instead of clicking toolbar icons, a performance dip is likely to occur, further dissuading users [22].

Taking inspiration from existing approaches, this paper attempts to apply mechanisms of nudging [4] and ambient suggestions [8] in order to shift user's inertia. To this end, we have adjusted and tested the GUI of a Word Processing Application (WPA), which aims at nudging users to use KBSs.

2 Background

Software solutions that assist users in learning and using KBSs do exist, but they often require the user to be intrinsically motivated and proactive in learning and utilizing KBSs. Some aid the user by providing overview of an application's KBSs [1] [2], but this requires that the user makes an effort to adopt this higher-level strategy, and remembers it in the heat of completing a main goal. A complete overview of all KBSs could present itself as information overload and be slower for the user if the KBSs are not retained.

Other applications apply transient notifications whenever a KBS could have been usedz [3] [4]. They contain basic features such as disabling notifications for certain functions and tracking missed opportunities for using the KBS. The fact that these notifications appear *after* choice and action, mean their preventive power is small. They require that the user takes note and remembers the shortcut for future use.

More forceful approaches have experimented with using obtrusive deterrences such as limiting functionality, adding time buffers, or requiring actions from the user [16, 12]. Although they might increase KBS usage, such approaches might not be appropriate in practice due to their obtrusive nature.

Inspired by the approaches above, in this study we investigate a possible middle ground solution that both facilitates KBS usage and proactively dissuades the user from using the mouse.

2.1 User Behavior

While people are usually aware of the benefits and the efficiency gains of using KBSs, few make an effort in trying to switch to this modality [11, 23]. The

[1] https://www.cheatsheetapp.com/CheatSheet/ last accessed 28th of January 2020
[2] https://www.ergonis.com/products/keycue/ last accessed 8th of October 2019
[3] http://www.veodin.com/keyrocket/ last accessed 13th of March 2019
[4] https://github.com/halirutan/IntelliJ-Key-Promoter-X last accessed 28th of January 2020

reasoning behind the lack of effort can be described with the term *satisficing*. The term was coined by Herbert A. Simon in 1947 and is a portmanteau of 'suffice' and 'satisfy'. It covers the tendency to accomplish the task at hand to a *sufficient* level, instead of accomplishing it at the most efficient and optimal level.

A reason for this may be as Czerwinski et al.'s [9] study shows, *"that users can mistrust their abilities, leading to underestimates of potential benefit with the new modality"* [8]. Wai-Tat Fu et al. state as well that *"In interactive tasks, people are biased towards the use of general procedures that start with interactive actions. These actions require much less cognitive effort, as each action results in an immediate change to the external display that, in turn, cues the next action"* [11]. It is commonly agreed that recognition is easier than recalling [1]. Therefore one reason why people tend to access commands by using the menus or clicking the icons, is that it is easier cognitively than remembering the exact shortcut for the exact command they wish to use.

Charman et al. [7] states that breaking habits is difficult and takes additional cognitive effort. This coupled with the fact that the aspirational level of the user often is to just invoke a command but not maximize the efficiency of the operation [23], creates an environment with little reason to improve. This motivational paradox as referred to by John M. Carroll et al. [6], can be used to explain why even experienced users cling to less efficient methods.

One way to increase users' motivation is through the use of gamification. McGonigal [21] defines the following four traits of game: A *goal* specifies what the player works for, providing a sense of purpose. *Rules* set up constraints and a system on how to achieve the goal. *Feedback* provides the player with progress. The last trait is *voluntary participation*, which concerns acceptance of the other traits.

2.2 Performance within a UI

As a new user of a software interface, one must rely on any previous experiences and the visual cues within the interface. Interface designers have the challenge of developing interfaces that support both novice and high-performance expert use. Due to users' tendency to satisfice and the reluctance to explore faster strategies, high-performance functionality often has a low usage rate. Designers must therefore be aware of this, and develop interfaces that support the transition from novice to expert.

Cockburn et al. [8] deconstructs the interaction with interfaces into the ***functions***, the commands and capabilities within software, and the various ***methods*** with which these can be accessed. An example of this is the bold function, available in word processing software by various methods. Each *method* has a *performance characteristic* that include the user's performance within that method, as well as a floor and ceiling of possible performance. Performance has before been estimated using the Keystroke-Level Model [5], which shows show how keystrokes a far faster than aiming with the mouse.

By this deconstruction Cockburn et al. introduce the domain of intermodal performance improvement. It concerns transitioning users to faster modalities such as KBSs. Scarr et al. [22] suggests with the switch, a dip in performance is likely to occur as it requires further cognitive resources and time to become acquainted with the new modality or method. It is therefore of importance that perceived cost of making the switch is minimized, and that the switch is rapidly rewarded. The first step in this transition is making the user aware of the new modality. Once the user is aware of the new modality, whether or not they choose to use it, is dependent on their perception of the potential gain in efficiency.

2.3 Ambient Suggestion

Another domain that Cockburn et al. [8] put forth concerns making the user aware of the unused functionality. Here focus is on using ambient suggestions and recommendations rather than explicit instruction. The presented themes by Cockburn et al. concern how to generate useful recommendations and how to present them best.

Cockburn et al. state that *"Presentation should be dynamically updated to the user's context, should be continuously available to the user, and should be presented in an ambient manner that allows quick access without interrupting task execution"* [8].

Cockburn et al. note the importance of feedback's relevance and the likelihood of disrupting the user for any interface. Citing research by Bødker [3], Cockburn et al. describes the two types of task disruption as *"**Breakdowns** result in severe disruption, forcing the user's attention to a new activity. **Focus-shifts** cause only a brief attention switch and cause less disruption"* [8]. Interface designers needs to balance between being obtrusive and going unnoticed.

Concerning how information is presented and perceived by the user, Cockburn et el. present the following factors: "*the probability that system feedback accurately reflects the user's intention, the ease with which causal relationships between action and effect can be learned, their stability and predictability, the temporal connection between action and response, the user's degree of focus on the work environment, and the potential costs of interruption*"[8]

Matejka et al. [20] argue that in order for a recommendation to be good, it should be both unfamiliar to the user, and useful within the context.

An example of ambient suggestions is Matejka et al.'s Patina [19] which overlays a heat map onto the interface of an application. The heat map highlights interface elements using a color mapping with opacity showing the usage of interface elements, and hue distinguishing if the usage is by the user, other users, or both.

A field in which information likewise must be pushed gently, in a way that does not take focus away from the task at hand, is video games. Dyck et al. [10] describe how 'calm messaging' with the use of transient text, animations crafted to match visibility with importance, serves as a fluid way to deliver information to the user. Attention-aware elements that, for example, modify the transparency based on the users attention. This way elements are still accessible and is a part of the users awareness, but not necessarily occluding their view.

2.4 Nudging

Nudging is the concept of influencing behavior towards a predictable outcome by the way options are presented. In a digital context this has been defined as *"the use of user-interface design elements to guide people's behavior in digital choice environments"* [24].

Nudges have been categorized along two axes, transparency and mode of thinking [4, 13]. The mode of thinking refers to whether the nudge engages mainly our automatic or reflective mind, or respectively system 1 & 2 as put forth by Kahneman and Egan[15]. Automatic thinking is the fast, effortless thinking that guides most of our decision-making. Reflective thinking takes over when slow, rational and effortful thinking is required. Transparency refers to how clear the intentions and working of a nudge is to the user.

In a paper by Caraban, Karapanos, Gonçalves, and Campos [4], 23 nudge mechanisms are found by review of the use of nudging in HCI research. The mechanisms are categorized into categories such as: *facilitate*, *reinforce*, *fear*, and *confront*.

Facilitate nudges reduce mental or physical effort, thus making a particular choice easier to choose. They can be designed in a way that aligns with the user's own interests and goals. One way this nudge can be performed is by *suggesting alternatives* that might otherwise not have been considered. This type of nudges is at the transparent and reflective end, as a proposed option is up for the user to reflect upon.

Reinforce nudges attempts to reinforce behavior by situating them/it in the user's mind. This can be done with *just-in-time prompts*, that attempts to highlight behavior at a well timed moment. *Ambient feedback* too attempts to reinforce behavior, but with minimizing disruption of the user's task. A less transparent way such nudges can work, is using *subliminal priming* per the *mere exposure effect*, as exposure could develop a preference based on familiarity.

Fear nudges work by invoking negative feeling such as fear, loss, and uncertainty, in order to dissuade users from certain behaviors.

Confront nudges try to stop an undesired action by causing doubt. *Reminding of the consequences* is also a transparent mechanism that attempts to cause the user to reflect on the consequences of their choice.

Friction nudges are less intrusive, as they do not necessarily demand attention or action. They therefore only offer slight reflection.

Hassenzahl and Laschke [14] has elaborated on this concept with the term *Pleasurable Troublemaker*. Unlike purely automatic nudges, pleasurable troublemakers should ideally create just enough friction at the moment of choice to cause reflection and sustained behavioral change. It should allow sidestepping it, they argue that this adds an ironic element to the object, and that it emphasizes the personal choice, thus becoming more likable. A slightly annoying object should have expressive character and be understanding and naive in order for it to create a bond with a person.

3 Methods

To evaluate if nudging and ambient suggestion can be used to support and actively encourage the use of KBSs, a solution incorporating these elements was designed. Following qualitative semi-structured interviews on proposed design ideas, a final design was chosen. The UI is implemented in Java and built on-top of an existing WPA demo[5], which contains a toolbar with icons for basic text formatting.

3.1 Design Choices

The proposed prototype features *suggestions*, UI-elements that consist of the icon and KBS of a function in the WPA as seen in figure 1.

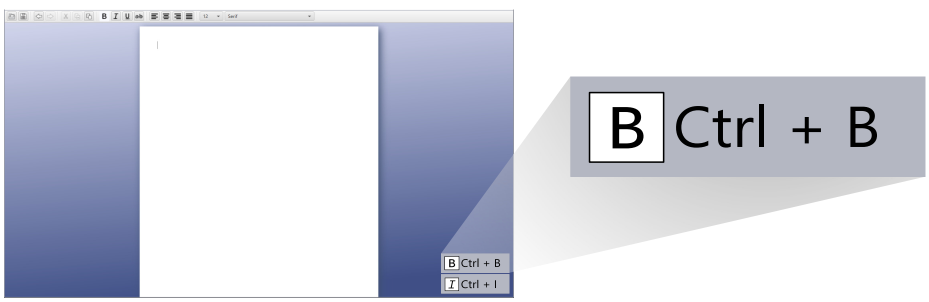

Fig. 1: Illustration of the modified WPA, and a close-up of a *suggestion*.

Whenever a toolbar icon is clicked, a *suggestion* appears in the bottom-right corner. As more icons are clicked, the added suggestions form a list. A suggestion does not disappear until its corresponding KBS has been used.

Each suggestion can shake, change transparency and color based on the interaction with the system. The goal of these visual states is to attract no or varying attention depending on the context. The three main visual states can been seen in figure 2.

[5] https://github.com/FXMisc/RichTextFX

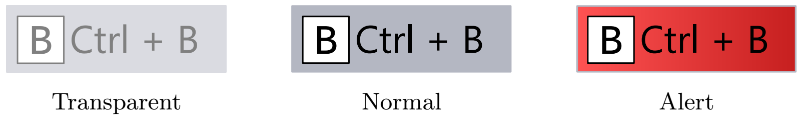

Fig. 2: Illustration of the three different states of a suggestion.

By default, focus should be on the task within the application. The suggestion will therefore be transparent most of the time. Hovering with the cursor over the suggestions' area makes them all fully opaque.

When a suggestion appears, they fade-in and alert the user for a brief moment. Whenever the user is likely to apply a function, the system fades into attention by becoming non-transparent. Moments in the WPA that are indicative of such are whenever text is selected by either mouse or keys.

A behavior the system attempts to proactively dissuade is when the user is going for the toolbar area using the mouse. In this case, suggestions call further gradual attention to themselves. When the cursor gets closer to the toolbar, all suggestions become gradually less transparent. The corresponding suggestion becomes increasingly red as the cursor is moved closer to a specific toolbar icon, and the suggestion shakes as the cursor enters the icons area. This is in an attempt to invoke the feeling of doing something wrong. When a KBS for a suggestion is performed, the suggestion exits with a bouncing animation before exiting on the right. These animations, as seen in fig. 3, intends to add an expressive element to the suggestions.

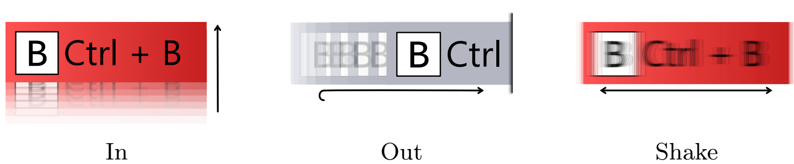

Fig. 3: Illustration of the suggestions' three different animations.

With this study, we wanted to investigate whether these suggestions could nudge the user into using KBSs by their presence and selective calls to attention.

Their persistent presence makes them different from transient notifications that, by their nature, do not necessarily demand attention or action. The suggestions appear as a consequence of the user's action, and embodies a single alternate choice, when taken, also makes them disappear. Their persistent but actionable nature also set up rules. If the goal of using KBSs is accepted and the user engages, progress and feedback is seen by the suggestions' existence, and possibly performance improvement. It is voluntary to participate, with the only obstacles being calls to attention.

Friction is added with attempts to dissuade the user with animation. This friction does not prevent user action, unless it disrupts by demanding too much attention.

3.2 Evaluation

To investigate to what extend nudging and ambient suggestion can be used to support and actively encourage the use of KBSs, a user experience evaluation was conducted. The evaluation involved two writing tasks, which the participants were asked to perform.

The participants were provided with a printed example for each task. The participants were asked to write new content, but to follow the formatting from the examples. The first tasks required them to describe their favorite things. The second task was a fictional CV with sentences to fill in. Each task required them to use several formatting functions multiple times. The first task was for the participants to become familiar with the WPA, and did not include this papers implementation. For the second task, the implementation was included. The participants were not informed of this change. The interviewer took notes of unexpected occurrences and the strategy the participants took. Following the tasks, the Microsoft Reaction Cards Method (MRCM) [2] was used. The participants were asked to pick five cards with an adjective written on them and elaborate on their selection. The cards were picked from a pile of 30 cards, 12 of which were negative, 13 positive, and 5 neutral. The answers served as a starting point for the semi-structured interviews. In the interviews the participants were asked if they noticed each element of the system, and elements they did not notice were shown to them. During the interview, questions relating to motivation, distraction, and annoyance were posed.

The application was tested on ten university students from different lines of study, seven of which were male and three female. The participants were equal parts Windows and macOS users. The interviews were conducted by two interviewers.

4 Results

All cards chosen in the MRCM, are presented in figure 4. Most participants chose positive words over negative when choosing cards.

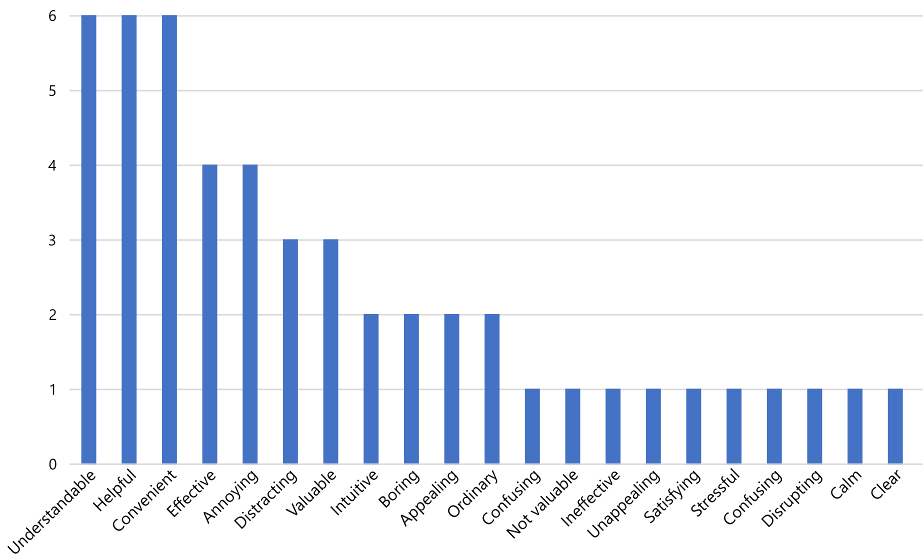

Fig. 4: The MRCM cards and frequency with which they were chosen by the participants.

A common theme throughout the MRCM evaluation was that the users found the system helpful and convenient for learning new shortcuts. Multiple participants mentioned that they could see it being effective for people that use an extensive amount of time working in WPAs, especially while formatting a document.

In terms of motivation, several participants said the system would convince them to use KBSs more and would be a convenient way to learn KBSs, as it was easy to understand and comprehend instantly. One participant noted that it would be especially useful when learning to use a new program, like applications for mathematics or photo manipulation programs. One participant acknowledged that such a system could be helpful for other people, but did not find it useful for himself. The participant could not see the benefit of saving two seconds once in awhile, and did not think he would remember all the KBSs. One participant noticed and acknowledged the intent of the system, but rejected learning and using these specific shortcuts. The participant knew KBSs relevant for him in WPA, but did not perform formatting of text often enough to use or bother learning all the specific KBSs of the WPA.

Regarding distraction, the vast majority found neither the design of the system to be intrusive nor distracting enough to disrupt their workflow. One participant explicitly mentioned liking that the suggestions faded out whenever not in use and that they would fade in when a toolbar icon was about to be clicked. Several mentioned they found the suggestions helpful and ready at hand without being too obstructive. A little less than half noticed the system right away, a few noticed it, but chose to ignore them afterwards. One participant attempted to select multiple words one at a time, which was not a feature of the WPA. The participant then thought it was not possible to use KBSs, and then chose to ignore the suggestions. Another participant thought the suggestions were notifications from other applications and therefore ignored them, having a habit of ignoring notifications in general. Two participants found either the color changes or shaking animation to be too distracting. When directly asked, a few others said that the animation and color changes could potentially be too distracting. A couple of participants thought the placement of the suggestions were too far from the toolbar, as they were not noticeable when clicking a toolbar icon.

As for annoyance, half of the participants did not find the system annoying. Some noted that they did not find it annoying after a while, or after getting used to it. A couple of participants found the effects like shaking and the color changing to be specifically annoying. Some mentioned it was annoying that a suggestion for rarely used function would stay until the corresponding KBS was used.

One topic raised by several participants concerns customization of the interface, such as placement of KBS suggestions, or the ability to disable the system or parts of it. Some suggested to limit the KBS suggestions to a max of five visible at a time, others that they should be sorted after relevance. It was observed that several participants tried to remove the suggestions by clicking on them.

5 Discussion

Some participants considered both the WPA and suggestions as a single product when choosing MCRM cards, instead of focusing only on the suggestions. Therefore, it is not possible to decide with absolute certainty which parts of the system caused them to pick the cards, as the cards reflect their overall experience. The interviews used them as a starting point for discussing, in order to invite participants to elaborate on their experience. As they were performed by two different interviewers, some variation is expected.

The application was largely successful in making participants more motivated in learning shortcuts. When asked about choosing a word, several participants talked about how the application helped and reminded them to utilize KBSs. It was mentioned that to learn and retain the KBSs, the disappearance of the suggestion was not useful.

The suggestions were less conspicuous than expected, as some participants did not notice the system. They noticed the entrance of the suggestions, but paid no particular attention to them, as they were dismissed as just another notification. Had the suggestions been introduced to the participants this probably wouldn't be the case, but this finding could still indicate that the design is too similar to other pushed information that is easily dismissed.

The fact that some participants were overly focused on the task of writing could be a consequence of the unnatural testing environment. When discussing KBSs usage during the task some participants mentioned that they used KBSs less than normal due to the fact that they were not familiar with the keyboard used during the testing.

Some participants were unwilling to use KBSs which highlights the importance of relevance. They rejected learning the particular KBSs for the WPA, as they would not be of future use of the participant. The suggestions shown should make sense for the user to learn, and be perceived to be of importance. If the suggestions' visual urgency does not match the quality of the recommendation it would be highly disruptive and just a nuisance. Choosing to highlight particular repetitive actions could therefore be considered, as this would more rapidly pay off in performance and perceived relevance. The level of urgency conveyed visually should be tuned to just highlight the option enough for users, but not causing too much disruption making them reject it.

The attempt to invoke a fear of doing something wrong using colors and animation was by some participants recognized to have an affect on their choice by reminding them. The most common comment was that it might be too disruptive and annoying. It was however not consistently noticed or perceived, possibly because the animation occurred outside of their attention on the toolbar icon. Whether these calls for attention could be performed closer to their center of attention, like the toolbar, would be up for further research as to the effect and how disruptive it would be.

As the tasks given to test participants were short and with a considerable amount of formatting required, the suggestions possibly appeared more often than they would under ordinary circumstances. This could have caused the suggestions and their animations to be considered more distracting. Further evaluation in a natural setting would therefore be of importance.

In some cases the system failed to convince the test participants, stating they would not be able to recall the shortcuts and was therefore reluctant to learn them. This is likely underestimating their own ability to remember KBSs if used regularly.

While the solution focuses on its use in WPAs, there could be potential for a system such as this to assist the user to improve intermodal performance in a wide range of applications. However, this raises further challenges concerning prediction of context and relevance, and designing the suggestions to not appear obtrusive. As the dormant gray rectangles were generally considered not to be a distraction, more elaborate designs that both convey urgency at an appropriate level, and facilitate the use of KBSs could be investigated further.

6 Conclusion

Looking at optimizing performance while using a user interface, research has shown there is space of improvement. As described with term *satisficing*, even experienced users within a given software are reluctant to optimize their efficiency. KBSs are considered the fastest method when it comes to productivity, but many people are not using them.

This study aimed at investigating if nudging and ambient suggestions could be used to support and actively encourage the use of KBSs. The proposed prototype featured *suggestions*, UI-elements that appear whenever a toolbar icon has been clicked, and are removed when the corresponding KBS has been used. The suggestions are displayed in an ambient manner by using animations, transparency, and color for selective calls for attention. Furthermore, they attempt to actively dissuade the user from clicking toolbar icons without preventing them from doing so. Participants were asked to perform two writing tasks with and without the prototype activated. The user experience was evaluated by using the MRCM to measure participants' emotions and desirability towards the interface. Following the MRCM, a semi-structured interview was conducted to evaluate if and how the UI, among other things, motivated or distracted the user.

Promising results were found from the user experience evaluation. The majority of the participants found the system helpful, convenient and motivating. Furthermore, most participants did not find the system to disrupt their workflow, or be too obtrusive. There are first indications that nudging and ambient suggestions could be used to facilitate KBSs and proactively remind users to use them. Further research into different elements of the design, as well as testing in a more natural environment is required in order to make any generalizations.

Most software include various methods for functions allowing better performance, so there is a good reason for developers to consider ambient suggestion and nudging to provide a more user friendly experience when the user transitions to expert methods.

References

1. Anderson, J., Bower, G.: Recognition and retrieval processes in free recall. Psychological Review **79**, 97–123 (03 1972). https://doi.org/10.1037/h0033773
2. Benedek, J., Miner, T.: Measuring desirability: New methods for evaluating desirability in a usability lab setting. Proceedings of Usability Professionals Association **2003**(8-12), 57 (2002)
3. Bødker, S.: Applying activity theory to video analysis: How to make sense of video data in hci. In: Context and consciousness: Activity theory and human computer interaction, pp. 147–174. MIT press (1996)
4. Caraban, A., Karapanos, E., Gonçalves, D., Campos, P.: 23 ways to nudge: A review of technology-mediated nudging in human-computer interaction. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300733, https://doi.org/10.1145/3290605.3300733
5. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale, NJ, USA (1983)
6. Carroll, J., Rosson, M.B.: Paradox of the active user. pp. 80–111 (01 1987)
7. Charman, S.C., Howes, A.: The adaptive user: an investigation into the cognitive and task constraints on the generation of new methods. Journal of experimental psychology: Applied **9**(4), 236 (2003)

8. Cockburn, A., Gutwin, C., Scarr, J., Malacria, S.: Supporting novice to expert transitions in user interfaces. ACM Computing Surveys (CSUR) **47**(2), 31 (2015)
9. Czerwinski, M., Horvitz, E., Cutrell, E.: Subjective duration assessment: An implicit probe for software usability. In: Proceedings of IHM-HCI 2001 conference. vol. 2, pp. 167–170 (2001)
10. Dyck, J., Pinelle, D., Brown, B.A., Gutwin, C.: Learning from games: Hci design innovations in entertainment software. In: Graphics interface. vol. 2003, pp. 237–246. Citeseer (2003)
11. Fu, W.T., Gray, W.D.: Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. Cognitive science **28**(6), 901–935 (2004)
12. Grossman, T., Dragicevic, P., Balakrishnan, R.: Strategies for accelerating on-line learning of hotkeys. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 1591–1600. ACM (2007)
13. Hansen, P.G., Jespersen, A.M.: Nudge and the manipulation of choice: A framework for the responsible use of the nudge approach to behaviour change in public policy. European Journal of Risk Regulation **4**(1), 3–28 (2013)
14. Hassenzahl, M., Laschke, M.: PLEASURABLE TROUBLEMAKERS, pp. 167–196. Mit Press (2014), http://www.jstor.org/stable/j.ctt1287hcd.13
15. Kahneman, D., Egan, P.: Thinking, fast and slow. Farrar, Straus and Giroux New York (2011)
16. Krisler, B., Alterman, R.: Training towards mastery: overcoming the active user paradox. In: Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges. pp. 239–248. ACM (2008)
17. Lane, D.M., Napier, H.A., Peres, S.C., Sandor, A.: Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. International Journal of Human-Computer Interaction **18**(2), 133–144 (2005)
18. Malacria, S., Scarr, J., Cockburn, A., Gutwin, C., Grossman, T.: Skillometers: reflective widgets that motivate and help users to improve performance. In: Proceedings of the 26th annual ACM symposium on User interface software and technology. pp. 321–330. ACM (2013)
19. Matejka, J., Grossman, T., Fitzmaurice, G.: Patina: Dynamic heatmaps for visualizing application usage. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3227–3236. ACM (2013)
20. Matejka, J., Li, W., Grossman, T., Fitzmaurice, G.: Communitycommands: command recommendations for software applications. In: Proceedings of the 22nd annual ACM symposium on User interface software and technology. pp. 193–202. ACM (2009)
21. McGonigal, J.: Reality is broken: Why games make us better and how they can change the world. Penguin (2011)
22. Scarr, J., Cockburn, A., Gutwin, C., Quinn, P.: Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 2741–2750. CHI '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1978942.1979348, https://doi.org/10.1145/1978942.1979348
23. Tak, S., Westendorp, P., Van Rooij, I.: Satisficing and the use of keyboard shortcuts: being good enough is enough? Interacting with computers **25**(5), 404–416 (2013)
24. Weinmann, M., Schneider, C., vom Brocke, J.: Digital nudging. Business & Information Systems Engineering **58**(6), 433–436 (2016)