Evanescent Adaptation on Small Screens

Sara Bouzit[12] **Denis Chêne**[1] **Gaëlle Calvary**[2]

[1]Orange Labs
28 chemin du Vieux chêne
38240 Meylan, France
firstname.lastname@orange.com

[2]Univ. Grenoble Alpes, LIG, F-38000
Grenoble, France
CNRS, LIG, F-38000 Grenoble, France
firstname.lastname@imag.fr

ABSTRACT

This paper addresses the problem of mastering the complexity of interacting with a large set of applications on smartphones. In one hand, number of applications increases. In the other hand, screen size reduces. To tackle this paradoxical evolution, we investigate adaptive user interfaces. We assume that it is possible to predict the applications of interest for a user in a given situation. Based on this hypothesis, our challenge is to accelerate user interaction when prediction is correct, without penalizing it when prediction is wrong. The paper proposes the concept of Evanescent Adaptation. The principle is a two-layer based representation: the predicted items (first layer) are displayed above the full list of items (second layer). The first layer is said to be evanescent in the sense that it automatically disappears progressively. The paper claims for putting this disappearing process under the control of the end-user. Thereby the user can close the first-layer as soon as s/he perceives prediction as irrelevant.

Author Keywords

Adaptive interfaces, ephemeral adaptation, evanescent adaptation, user performance, time, error.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Small screen size together with numerous applications make it difficult for people to navigate on smartphones. Therefore adaptive User Interfaces (UI) are essential for pushing forward relevant applications, and even more than on large screens [9, 12].

UI contextual adaptation has become an urgent necessity [11]. Research is active with two main classes of approaches: spatial and graphical adaptation [1, 3, 10, 13, 14, 15, 16, 17, 19, 20, 22, 23, 24]. In both cases, solutions work well when prediction is correct, but fail when prediction is wrong. We overcome this limitation by incorporating advances in Design for All [7] and ergonomics. We propose the concept of Evanescent Adaptation which grounding makes it suitable to address any context of use. We experiment it with regular users using smartphones, which is the context of use our concept is claimed to be efficient.

The paper is structured into three core sections devoted to the state-of-the-art, the concept of Evanescent Adaptation, and its evaluation. Findings open perspectives for future research.

OzCHI '15 , December 07 - 10 2015, Melbourne, VIC, Australia
Copyright © 2015 ACM xxx-x-xxxx-xxxx-x/xx/xx... $xx.00
http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx.

BACKGROUND LITERATURE

Spatial adaptation [6, 21, 25] consists to reorganize UI elements in order to push predicted items on top of the screen. Examples are frequency-based approach [21] and split menu [25]. These approaches are criticized for their lack of stability. UI changes jeopardize the creation of a user mental model about the UI, which is disturbing, and even more in case of cognitive or visual disabilities.

Conversely graphical adaptation ensures spatial stability. It consists in applying graphical effects for outlining predicted items. Thereby user attention is attracted to predicted elements, thus reducing navigation time and visual search time. One example is highlighting approach [27] that changes the background color of predicted items. However highlighting is not sufficient on smartphones as all items might not be displayable on a single screen. If the number of items to be displayed requires several pages (screens), the user has to browse all the pages and to search for highlighting.

Design for All [7] brings interesting techniques. One example is the training wheels [4], related to the Multi-layer UI concept [26]. The training wheels target novice learners and/or cognitive disabilities. The principle is to present only the basic functionalities at first time. Then the UI is enriched along the user learning progress. Several studies [5, 18] were conducted to improve the concept, especially by giving control to the user for personalizing the layers.

In the state of the art, there is one approach that implicitly, but partially, combines UI adaptation and Multi-layer UI: In Context Disappearing (ICD) [2]. ICD is a focus+context approach inspired from Ephemeral Adaptation [8], but applicable on smartphones. ICD is based on two layers: above, the predicted items; below, the full list of items. The objective is to attract user attention to the predicted items in order to accelerate user interaction, especially when the list of items is long. The first layer (named prompting window) disappears gradually after a delay of 5000ms. ICD keeps the context (complete list of items) accessible at any interaction stage not to penalize interaction when prediction is wrong. ICD was shown as efficient but the two lists overlapping generated errors.

This paper improves ICD by integrating user control for closing the prompting window and thus accessing the full list as soon as the predicted items are perceived as irrelevant. It is applied to home screens of smartphones whilst ICD was only applied to lists of items (as Ephemeral Adaptation).

FOUR PROTOTYPES

First step consisted in applying ICD (initially limited to menus (linear list)) to grids of icons on smartphones (e.g., home screens). This gave rise to prototype D, D standing for Disappearing (Fig. 1). At opening the home page, user finds two layers of icons. Grid below is the full grid of icons. Grid above is a prompting window.

Figure 1. Prototype D (Disappearing): the prompting grid appears above the main grid, and gradually disappears.

The prompting window is a small grid with three predicted icons, prediction being based on frequency. It is a grid to ensure spatial stability. It displays the icons at the same position as in the full grid. An icon from the bottom of the home screen or from the second underneath page is placed at the bottom of the prompting window. The prompting window disappears within 5000ms resulting in an automatic three-step process: 1) prompting window+full list; 2- disappearing prompting window; 3) full list. At first step, user starts searching for target on the prompting window. When prediction is correct, user selects target directly in the prompting window. Otherwise (incorrect prediction), user can either wait for complete disappearance of the prompting window or search for the target directly on the full grid (context) without having to wait until complete disappearance. Of course, for the items hidden by the prompting window, user has to wait for complete disappearance before being capable of selecting target. This may slow down user interaction. Also the two grids overlapping may generate errors, especially when prediction is wrong and user tries to select the target in the main grid. In summary, in prototype D, the context (complete grid of icons) is accessible, but partially only and subject to errors.

Second prototype consisted in overcoming these limitations by incorporating a user control into the prompting window. This gave rise to prototype DU, DU standing for Disappearing+User control (Fig. 2). The user control is an Exit button for closing the prompting window. It is a user controlled three-step process: 1) prompting window+full list; 2- disappearing closable prompting window; 3) full list. If prediction is correct, user selects target directly in the prompting window. Otherwise (incorrect prediction), user can switch to full grid of icons by clicking on Exit button which accelerates access to the main grid. Of course, like in prototype D, user can also click items in the context or wait for complete disappearance of the prompting window.

Figure 2. Prototype DU (Disappearing+User control): prototype D is enhanced with a user control.

Third prototype consisted in questioning the disappearing feature compared to a pure user control. This gave rise to prototype U, U standing for User control (Fig. 3). In this prototype, the prompting window does not disappear gradually resulting in a user controlled two-step process: 1-prompting window+full list; 2-full list. Control of the prompting window is fully given to user. The principle is shown in Fig. 3.

Figure 3. Prototype U (User control): the prompting window does not disappear gradually.

Those three prototypes were completed by a fourth one dedicated to control condition: the full list of icons is displayed without any effect, neither prompting window nor graphical effect.

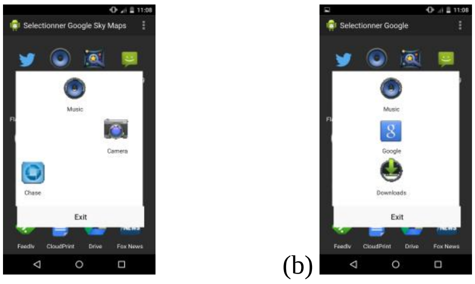
COMPARATIVE EVALUATION

The goal is two-fold: first to test whether D (initially applied to menu) remains effective when applied to grid of icons; secondly to compare D to DU, U and Control condition in order to verify if user control impacts adaptation usability.

Four tests were implemented and performed on smartphones with grids of 40 icons.

First test is control condition without any prediction.

Second test is D condition, including two variations. Icons in the prompting window are displayed either in the same order as in complete grid (order condition) or according to a probabilistic criterion (Fig. 4).



(a) (b)

Figure 4. (a) Order condition: icons have approximately the same distribution in the prompting window than in complete grid. (b) Probabilistic condition.

Third test is DU condition, with the same two variations same order or probabilistic distribution.

Fourth test is U condition with the same two variations: same order or probabilistic distribution.

Hypotheses

We made the following assumptions:

H1. Speed

- For a high prediction level

D, DU and U will be faster than control. When prediction is correct (which means that target is on the prompting window), access to target will be faster than in control condition where target can be on first or second screen.

U will be faster than D and DU. In U, the prompting window does not disappear. Thereby user has time to see and select the target while in D and DU the prompting window disappears gradually, which means that user must select target before complete disappearance of the prompting window.

In predictive conditions (D, DU and U), prediction displayed in probabilistic manner will be faster than by order. In probabilistic display, icon that is most likely useful for user will always be at top of the prompting window. This makes access to this target faster than when prediction is displayed by order which means that target can be anywhere on the prompting window.

- For a low prediction level

DU and U will not be worse than Control. Indeed, the objective of these three predictive conditions is to not slow down user interaction in case of incorrect prediction.

DU and U will be faster than D. When prediction is incorrect, user has to switch as soon as possible to the complete grid of icons. In DU and U conditions, user can control disappearance of the prompting window and thus accelerate transition to the complete grid without having to wait for complete disappearance of the prompting window. This differs from D where user cannot control prompting window disappearance and thus must wait until complete disappearance of the latter to be able to get hands on complete grid of icons.

DU will be faster than U. In this case, when prediction is incorrect, prompting window — that brings no benefit in this case — should be removed as soon as possible so that user can find the target in complete grid. In DU, the prompting window disappears gradually even if user does nothing. In addition, s/he can click on the button in order to accelerate disappearance. In U, the prompting window is fixed which means that if user does not click on Exit button, it will never disappear. So in U disappearance of the prompting window depends on user action. Automatic disappearance in DU should make it faster than U.

In all conditions (Control, D, DU, U), when target is on first screen, interaction will be faster than when it is on second screen. Access to target on first screen doesn't require any action from user, as opposed to target on second screen that requires the user to scroll in order to move from screen 1 to screen 2. So when target is on screen 1, interaction will be faster than when target is on screen 2.

H2. User preference

- For a high prediction level

At least D, DU or U will be preferred to Control. When prediction is correct in predictive conditions, user accesses to target faster and always finds what s/he wants to see. This makes predictive conditions preferred to Control condition in which user must seek target in first or second screen.

U will be preferred to D and DU. This assumption is based on the fact that the prompting window display time is longer in U than in D and DU. In U, user controls the prompting window disappearance without being stressed. When prediction is correct, user has more time and is more comfortable in U.

- For a low prediction level

Control will neither be preferred to D, DU or U. The main objective of predictive conditions is to not slow down user interaction when prediction is incorrect. Thereby Control condition should not be preferred to predictive conditions (D, DU, U).

Methodology

There are four independent factors (Fig. 5). First factor is grid type and is a within-subject factor. Control grid is static; D, DU and U grids are adaptive. Second independent factor is prediction display. For D, DU and U conditions, predicted grid contains three icons displayed at the same place than in the full grid or according to a probabilistic criterion. Third independent factor is target location. 40 icons were divided into two vertical screens in each condition. User target can be in screen 1 or 2 (accessible by scrolling) and target distribution was controlled. Fourth independent factor is prediction accuracy level. High and low accuracy levels are the same as defined in Findlater's study [8], also reused in ICD [2]. High level prediction (correct prediction) means that target is one of the predicted icons. Low level prediction (incorrect prediction) means that target is none of the predicted icons. In both cases target is always on the complete grid of icons (screen 1 or 2).

Task

The experimental task consists to ask users to perform a sequence of targets selection. For each selection, a message displayed at top of the screen specifies the icon to be selected. Then grid of icons appears. Target name remains displayed for preventing user from forgetting it. User selects target from the prompting window and / or full grid. When user succeeds in selecting the target, a new message appears specifying the name of the new target to select. If selection is incorrect, an error message is displayed; user must find the target before moving to a new selection. At the end of the test, a thank you message is displayed informing the user that test is complete.

Selections were performed with finger. We did not use stylus.

In each grid, order of icons as well as selection sequence were controlled by a random draw. Six distributions were made to define order of conditions. Users were assigned randomly to distributions. Target position in screen 1 or 2 and prediction accuracy level were also controlled.

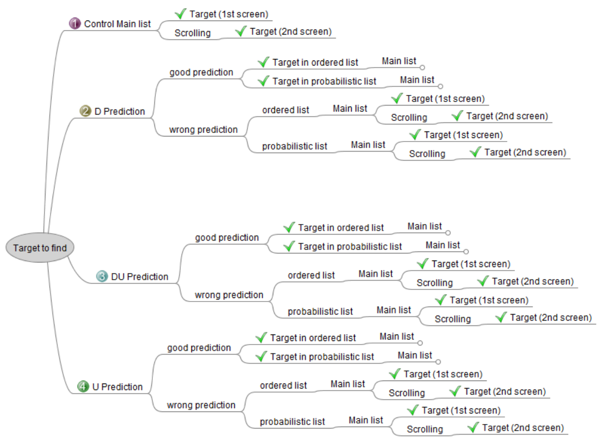


Figure 5. Tests overview.

Quantitative and Qualitative Measures

Three dependent variables were measured. The first dependent variable is speed selection. Speed selection is measured in seconds as the time taken from opening the grid of icons until correct selection of target. The second dependent variable is task fulfillment. For that one, error rates were recorded. The third dependent variable is scrolling time. Finally, subjective data related to difficulty, satisfaction and aesthetics were collected along a Likert scale of 5 points. We also asked users to make a preferential ranking for different conditions.

Apparatus

Materials used are Android smartphones. Experimentation is coded in Java for Android. Selection time, scrolling time and error rate are recorded in a database.

Participants

Thirty-three volunteers (men and women) participated to the experiment. They were recruited in our research laboratory, aged between 23 and 59, all being regular users of smartphones.

Procedure

Before starting, the principle of each condition was explained to participants, but without mentioning the two prediction accuracy levels (high and low). Each participant completed a short training test. It consisted in successfully selecting 10 targets. Grids of icons used in this training test were different from test conditions. The test is compound of 140 targets: 120 on adaptive designs (D, DU, U), plus 20 in Control condition. Each subject performs 40 selections for each adaptive design (D, DU, U = 40*3). Those 40 selections are divided in two parts: 20 include the correct prediction (high level), half of them are displayed by similar order as in main grid whilst others are displayed according to probabilistic criterion (the most probable icon at the top of prediction window). 20 cover wrong predictions, half of them when targets are distributed on first screen and others on second screen. There are also 20 selections for Control condition. In all cases, targets are distributed equally among first or second screen (10 selections with target in first screen and 10 selections with target in second screen).

There are 24 conditions: S33*I3*E2*P2*O2 with S: Subjects, I3: adaptive designs (D, DU, U), E2: (screen1 or 2), P2: low vs high prediction level, O2: order vs probabilistic display. In addition, a Control condition was managed with two factors (E1: target on screen 1 and E2: target on screen 2).

Results

Speed

Brown-Forsythe's test and Levene's test were applied to test homogeneity of variance. This later could not be distinguished, so for data analysis non parametric Friedman's ANOVA by Ranks was applied.

As expected when prediction level is high, predictive conditions (D M = 1.75, SD = 1.31, DU M = 1.70, SD =1.26, U M = 1.40, SD = 0.0.64) are significantly faster than control condition (M = 6.06, SD = 3.40). Target selection time in good predictive conditions is much shorter than in Control condition $F(3) = 102.35$, $p < 0.0001$. This confirms the result found in [2] that when prediction is correct, D applied to menu is faster than Control. Thereby D can be generalized to grid of icons.

For DU predictive condition, there is no significant difference between order-based (M = 1.56, SD = 0.97) and probabilistic-based presentation (M = 1.84, SD = 1.50), $F(1) = 0.04$, $p > 0.5$. It is similar for U based on order (M = 1.40, SD = 0.68) and probabilistic (M = 1.41, SD = 0.60), $F(1) = 0.15$, $p > 0.5$. For D a slight difference occurs ($F(1) = 6.53$, $p = 0.01$) between D ordered condition (M = 2.04, SD = 1.30) and D probabilistic-based presentation (M = 1.46, SD = 1.27). For good predictions, D probabilistic-based presentation is faster than order-based.

When prediction is incorrect, performance depends significantly on conditions $F(3) = 35.71$, $p < 0.0001$. Users are significantly faster in DU and Control (DU M = 6.97, SD = 2.64; Control M = 6.06, SD = 3.40; D M = 8.30, SD = 3.79; U M = 8.0, SD = 3.37).

When prediction is incorrect and target is in first screen (D M = 6.27, SD = 2.34, DU M = 6.1, SD = 2.38, U M = 7.20, SD = 3.02, Control M = 3.72, SD = 1.75), users are significantly faster than when target is in second screen (D M = 10.33, SD = 3.90, DU M = 7.85, SD = 2.62, U M = 8.81, SD = 3.54, Control M = 8.39, SD = 3.02), ($F(7)$ = 103.84, $p < 0.001$).

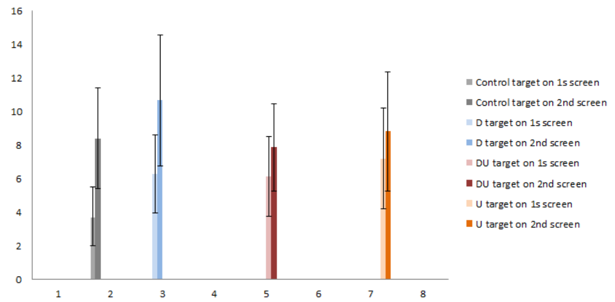


Figure 6. Selection time for Control condition and predictive conditions (D, DU, U) when prediction is wrong.

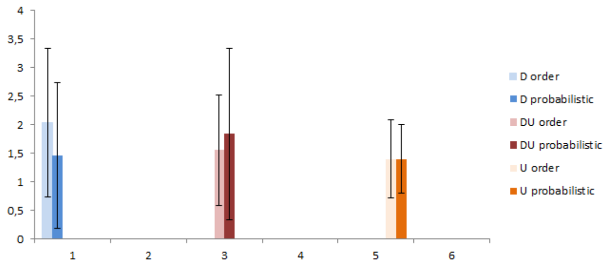


Figure 7. Selection time for predictive conditions (D, DU, U) when prediction is correct.

Errors

There is a clear difference between Control, D, DU, and U (Control M = 0.60, SD = 0.83; D M = 4.18, SD = 3.32; DU M = 1.03, SD = 1.10; U M = 0.63, SD = 0.93; $F(3)$ = 42.21, $p < 0.001$). Indeed the fixed prompting window (two steps without disappearing effect) in U doesn't generate an overlap between the prompting window and the complete grid. On the opposite, DU and D which both include three-step with disappearing effect show higher error rates. Nevertheless DU seems much more efficient than D, $F(1)$ = 15.12, $p < 0.001$ and not so far from U. Gradually disappearing effect may lead to errors but user control helps to avoid them.

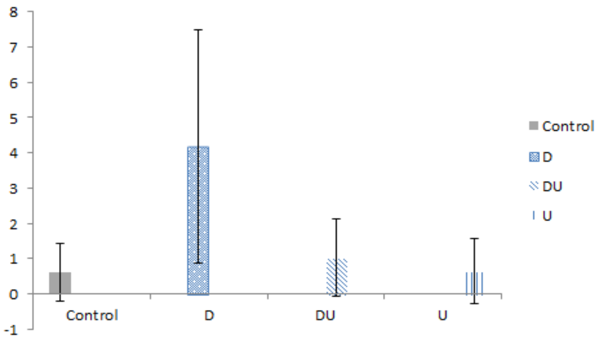


Figure 8. Error rate for all conditions.

User preferences

Results suggest that users prefer to have control over the interface (Control M = 2.36, SD = 1.08, D M = 1.73, SD = 0.91, DU M = 2.85, SD = 1, U M = 3.06, SD = 1.03), $F(3)$ = 20.82, $p < 0.001$. U and DU are appreciated by users whereas D is the less preferred. Results show a clear difference between D and DU ($F(1)$ = 10.94, $p < 0.001$), also between D and U ($F(1)$ = 10.94, $p < 0.001$). There is no difference between DU and U ($F(1)$ = 0.27, $p > 0.5$).

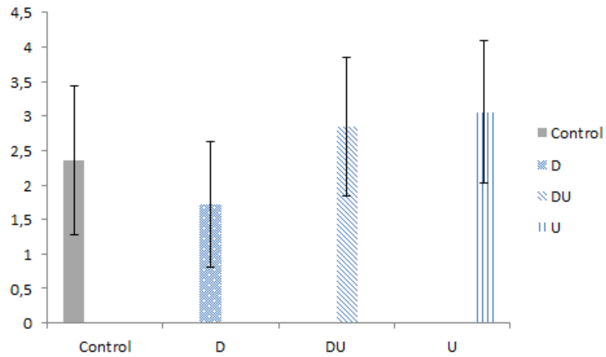


Figure 9. User preferences for all conditions.

Discussion

H1. Speed

- For a high level prediction

D, DU and U will be faster than Control condition. Supported

When prediction corresponds to user needs, it increases efficiency as it reduces visual search time of targeted item. Conversely when prediction doesn't match his/her needs, and also in Control condition, user must find the target on two different screens. It takes more time and effort. This is not that surprising but experience was needed before further analysis.

U will be faster than D and DU. Supported

This hypothesis is confirmed. Indeed, when prediction is correct, a short presentation time of predicted item is helpful to user. A two-step interaction (without disappearing effect) is more efficient in case of correct prediction. There is no significant difference between D and DU. In this case user does not need to use Exit button in order to put away the prompting window as relevant information is inside. Thus user control isn't useful in this context.

- For a low prediction level

In all conditions (Control, D, DU, U), when target is on first screen interaction will be faster than when it is on second screen. Supported

This supported hypothesis is a prerequisite to further analysis as is it obvious to be faster on a near target than on a far one. In all conditions when user searches for target in first screen, s/he will access directly to this target. Whereas if target is in second screen, user takes more time because s/he has to scroll from screen 1 to screen 2 in order to reach the target. This justifies the fact that user is faster when target is in screen 1.

DU and U will be faster than D. Partially supported

DU will be faster than D. Supported

On the contrary, when prediction is wrong, user needs to quit the prompting window as quick as possible. Users are faster in DU condition than in D condition. When prediction is irrelevant, it can slow down interaction. Indeed user has no control over the prompting window and must wait for its complete disappearance. In DU, user controls the prompting window and can make it disappear at any time. Also the prompting window gradually disappears which allows user to see hidden parts and to prepare his motor action towards the targeted item. This may explain the partial support of this hypothesis.

U will be faster than D. Not supported

No significant time difference between U and D was revealed. The prompting window in U condition is fixed and applies in a shortest adaptation process (two steps) that can be accelerated by user (control available). In a low prediction level, it should be an advantage. However this one seems to be cancelled due to the hiding of a large part of the complete grid, which is not the case in D condition. Indeed, an advantage of this condition is the rapid access to the complete list hidden by the prompting window. The disappearance effect facilitates preview in case of wrong prediction. To summarize, in a low prediction level, U condition benefit (user control to accelerate interaction) is counterbalanced with D condition benefit (disappearance helps to preview other items), and finally no clear difference between the two can be stated.

DU will be faster than U. Supported

When prediction is incorrect user interest is to switch as soon as possible to the complete grid of icons. Both DU and U enable user to put the prompting window out. But in DU condition it will be done automatically after 5000ms delay. It gives then access to the complete list of items. Moreover, DU is a three-step process, with a progressive disappearance. Thus it enables, as in the previous hypothesis, to preview underlying list of items. Those two combined factors might explain that DU is faster than U.

H2. User preference

- For a high level prediction

At least D, DU or U will be preferred to Control. Supported

In front of a large set of items (40 icons here) user needs prediction and is glad to rapidly find the target.

U will be preferred than D and DU. Supported

When prediction is correct, in U condition user has no disappearing effect of the prompting window, target selection is easier than in D and DU where user must make a selection before the prompting window disappears. This justifies the fact that U is preferred to D and DU.

- For a low level prediction

Control will not be preferred to D or will not be preferred to DU or will not be preferred to U. Supported

When prediction is incorrect, DU and U conditions are clearly preferred. We justify this result by the fact that in DU and U conditions user can control the prompting window and can fire it without waiting for its complete disappearance. One can deduce from this expected result that user always wants to have control over what happens. Adding a button reinforces the proposed approaches because it makes these approaches with double profits especially DU. On one hand, if prediction is correct, these approaches make user faster and can reduce navigation time as well as visual search time. On the other hand, when prediction is incorrect, user does not have to absorb the impact induced by a wrong prediction. In this case user controls the interface and there is no waiting time which never slows down user interaction in case of incorrect prediction.

As a result, user always prefers to have control over the interface, even if it adapts appropriately to him/her.

CONCLUSION AND PERSPECTIVES

This paper addresses the problem of how to interact with an increasing number of applications with devices which display surface is paradoxically decreasing. The approach is adaptive UIs. It applies the “In Context Disappearing” approach to smartphones, thereby generalizing the technique from one-dimensional navigation (linear menus) to two-dimensional navigation in a grid of icons (e.g., home screens of smartphones). Two variants were investigated, namely DU and U.

DU is a three-step process (prediction, transition, then complete grid), enhanced with a user control. Experience shows that the approach is efficient in both cases of good or bad prediction thanks to the user control.

U focuses on user control in a two-step process (prediction then complete grid).

This work calls for further research on user control with possibly different degrees of control and presentation that go beyond the Exit button. Also rethinking disappearing effect might be valuable as it still generates some errors. Improvement may also be done on how to display the transition between still clickable items and just disappeared ones. Finally the rationale of the prediction and the locations items come from also deserve to be studied. Currently they come from the current list which seems to be well understood and appreciated by users. However prediction could come from deeper hierarchy levels that question the impact on user model. This is related to the intelligibility [28] property which is key in Human Computer Interaction.

REFERENCES

1. Bederson, B.B. Fisheye menus. ACM UIST, (2000), San Diego, USA.p, 217-225.

2. Bouzit, S., Chêne, D., and Calvary, G. From Appearing to Disappearing Ephemeral Adaptation for Small Screens. In *Proc OZCHI'14* Proceedings of the 16th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design. Pages 41-48.

3. Bridle, R., and McCreath, E. Inducing shortcuts on a mobile phone interface. In. *Proc. IUI, (2006), 327-329*.

4. Caroll, J. M. and Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM, 27(8):800-806*.

5. Clark, B., Matthews, J., Deciding Layers: Adaptive Composition of Layers in a Multi-Layer User Interface. *Proceedings of 11th International Conference on Human-Computer Interaction, Volume 7, July 2005*.

6. Cockburn, A., Gutwin, C. and Greenbrerg, S. A predictive modal of menu performance. In. *Proc CHI'07, (2007), 627-636*.

7. Constantine, S. Adaptive Techniques for Universal Access. User Modeling and User-Adapted Interaction 11: 159-179, 2001.

8. Findlater, L Moffatt, K., McGrenere, J., and Dawson, J. (2009). Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. *ACM CHI, 1655-1664*.

9. Findlater, L., and McGrenere, J. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pages 1247-1256. ACM, 2008.

10. Findlater, L., and Mc Grenere, J. A comparison of static, adaptive, and adaptable menus. In proceedings of the SIGCHI conference on Human factors in computing systems, pages 89-96. ACM, 2004.

11. Findlater, L., and Gajos., K.Z. Design space and evaluation challenges of adaptive graphical user interfaces. AI Magazine, 30(4) : 68, 2009.

12. Findlater, L. and McGrenere, J. Evaluating reduced functionality interfaces according to feature findability and awareness. In. *Proc. IFIP Interact 2007, (2007), 592-605*.

13. Francone, J., Bailly, G., Lecolinet, E., Mandran, N., and Nigay, L. Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode. ACM AVI, 2010.

14. Gajos, K.Z., Czerwinski, M., Tan, D.S., and Weld, D.S. Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI'06, (2006),201-208.*

15. Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S. Predictability and accuracy in adaptive interfaces. In*. Proc. CHI '08*, (2008), *1271-1274.*

16. Huhtala, J., Mantyjarvi, J., Venta, L. and Isomursu, M. Animated Transitions for Adaptive Small Size Mobile Menus. *In Proc of the 12th IFIP TC 13 Int. Conf on Human-Computer Interaction Interact'2009*.

17. Jones, M., Marsden, G. Mohd-Nasir, N., and Boone, K. Improving Web Interaction on Small Displays. In 8th World Wide Web conference, Toronto 1999.

18. Kang, H., Plaisant, C., Shneiderman, B., New approaches to help users get started with visual interfaces: multi-layered interfaces and integrated initial guidance. *Proceedings of the 2003 annual national conference on Digital government research*.

19. Lee, D.S. and Yoon, W.C.: Quantitative results assessing design issues of selection-supportive menus. International Journal of Industrial Ergonomics 33 (1), 2004, pp. 41-52.

20. Matejka, J., Grossman, T., and Fitzmaurice, G. Patina: Dynamic Heatmaps for Visualizing Application Usage. ACM CHI, 2013.

21. Mitchell, J. and Shneiderman, B. Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin 20, 4(1989), 33-37.*

22. Patel, D., Marsden, G., Jones, M., and Jones, S. An Evaluation of Techniques for Image Searching and Browsing on Mobile Devices. Proceedings SAICSIT 2009, pp 60-69.

23. Raptis, D., Tselios, N., Kjeldskov, J., and Skov, M. Does size matter? Investigating the impact of mobile phone screen size on users' perceived usability, effectiveness and efficiency. MobileHCI, 2013.

24. Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In *Conference Proceedings of INTERACT'09, 2009.*

25. Sears, A., and Shneiderman, B. Split menus: Effectively using selection frequency to organize menus. *ACM TOCHI 1, 1(1994), 27-51.*

26. Shneiderman, B. (2003). Prompting universal usability with multi-layer interface design. In *Proc of. CUU 2003*, 1-8. ACM Press.

27. Tsandilas, T. and Schraefel, M.C. An empirical assessment of adaptation techniques. In *CHI '05 Extended Abstracts*, (2005), 2009-2012.

28. Vermeulen, J. Improving Intelligibility and Control in Ubicomp. *In Proceedings of the 12th ACM Interaction Conference on Ubiquitous Computing*-Adjunct Papers, Ubicomp'10 Adjunct, pages 485-488, New York, NY, USA, 2010. ACM. ISBN 18.978-1-4503-0283-8.