# Interface Adaptivity by Widget Promotion/Demotion


**Sara Bouzit, Gaëlle Calvary**
Université Grenoble Alpes
CNRS, Grenoble INP, LIG
Grenoble, France
{sara.bouzit,gaelle.calvary}@imag.fr

**Denis Chêne**
Orange Labs
Meylan, France
denis.chene@orange.com

**Jean Vanderdonckt**
Université catholique de Louvain
Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be


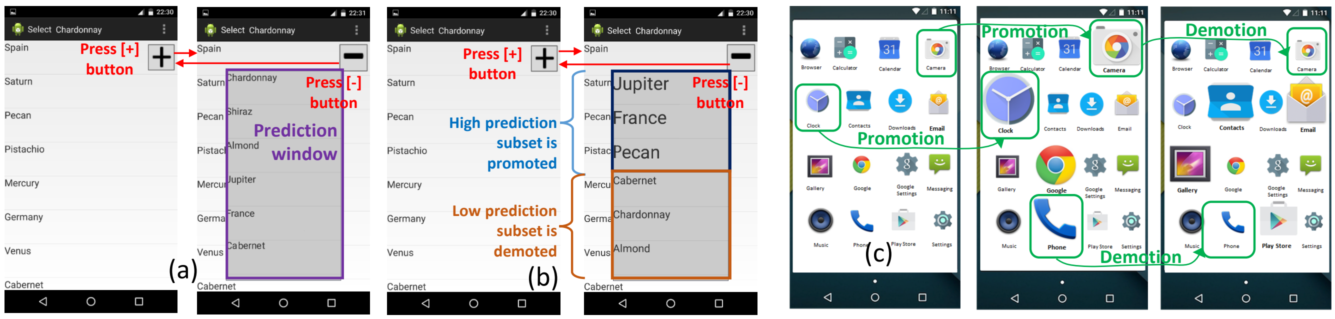


**Figure 1: Examples of bimotion interfaces: (a) a menu with one prediction window, (b) a menu with two prediction sub-sets: high and low prediction, and (c) a toolbox of icons with promotion and demotion based on icon size.**


## ABSTRACT

Promotion and demotion are a typical adaptive navigation technique making a page or a link easier to select by emphasizing it or de-emphasizing it depending on its popularity. This technique, which was successfully applied to adaptive web sites, is now generalized to mainstream graphical user interfaces by introducing bimotion user interfaces, which constantly and dynamically perform adaptivity by promoting the most predicted widgets and demoting the least predicted ones either in context or in a separated prediction window. Promoted widgets that are less frequently used become demoted, demoted widgets that are more frequently used become promoted.



## CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**; **Ubiquitous and mobile devices**; **Smartphones**; *User studies*; *Gestural input*; *Empirical studies in interaction design*;

## KEYWORDS

Adaptivity, graphical adaptive menu, promotion/demotion.



Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).


*EICS '19, June 18–21, 2019, Valencia, Spain*


© 2019 Copyright held by the owner/author(s).


ACM ISBN 978-1-4503-6745-5/19/06.
https://doi.org/10.1145/3319499.3328237

**ACM Reference Format:**
Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2019. Interface Adaptivity by Widget Promotion/Demotion. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '19), June 18–21, 2019, Valencia, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3319499.3328237


## 1 INTRODUCTION

Presentation, navigation, and contents are the three main categories of web site elements subject to adaptivity [3]. In adaptive navigation, promotion [12] aims at making a web page or a link easier to find and faster to select by moving it on top of its menu, while demotion moves it at the bottom to reflect its decreasing popularity. By generalizing this idea, *promotion* of any user interface element aims at emphasizing it according to a prediction rendering technique to reflect its increasing prediction, while *demotion* aims at overshadowing this element to reflect its decreasing prediction.

Prediction could be computed according to several prediction schemes [2, 5, 10, 14]: Most Popular Usage, Most Frequently Used, Most Recently Used, etc. Any widget receiving a higher prediction from a prediction scheme could be rendered by several grouping/distinction techniques based on the location [3, 10] (e.g., placing a link closer to the home page, moving a link closer to the top, page reorganization, clustering) and/or the format (e.g., highlighting, bolding, coloring, increasing font size, changing font type, marking). Conversely, any widget receiving a lower prediction could be rendered based on the location (e.g., removing a link from the home page, moving a link lower in a list) and/or the format (e.g., greying, deactivating, blurring, or reducing font size). As for any adaptivity technique, promotion and demotion have advantages and shortcomings [6, 7, 11]]: spatial instability (the initial layout is altered after adaptivity), user disruption (the user is always astonished to see some change), extraneous cognitive effort (an effort is needed to acquire the newly adapted contents). For these reasons, neither machine learning analysts nor user interface developers are quite ready for any arbitrary adaptivity. The *prediction window* is hereby referred to as any area containing widgets added for promotion or removed for demotion. Promotion and demotion, whilst being popular in web engineering [10, 12], have never been applied to mainstream GUIs.

*Graphical Adaptive Interfaces* (GAMs) have been studied extensively [7], many of them performing adaptivity on a menu or a list [9], provoking some spatial instability [2]. Adaptivity could be recursively applied to the prediction window, which has never been investigated. Since the prediction window is always constrained by a limited size, every promotion inevitably implies a corresponding demotion and any element selected inside or outside the prediction window also contributes to updating the contents of the prediction window. For instance, an element belonging to the prediction window that is selected out of its scope probably means that it was not salient enough, an element belonging to the prediction window that is selected in its scope should be strengthened and an element that is never selected from the prediction window should be weakened. Demotion, in contrast, moves an element computed as less predicted further away. A element should not be subject to demotion from the natural location where the end user expects to find it. Demotion of elements should be based on their low prediction: unpopular ones should be moved further away, aiming not to overload the users with navigation to contents that are rarely accessed or relevant only for a few users.

To address the aforementioned challenges and to fill the gap created by the absence of promotion and demotion in GAMs, this paper provides the following contributions:

- The concept of *bimotion interface*, which dynamically performs adaptivity by promoting the most predicted widgets and demoting the least predicted ones either included in the interface itself or in a separated prediction window.
- An instantiation for a menu or a toolbox (Fig. 1).
- A bimotion menu, a new widget implemented in Java for Android SDK and a user study investigating its effect on adaptivity for smartphones as their small screen size and resolution impact usability [14] and performance [16].

## 2 RELATED WORK

Bimotion interfaces overlap two areas of research: graphical adaptive interfaces and adaptive navigation.

### Background on Adaptive Interfaces

Graphical Adaptive Interfaces exhibit a certain amount of potential benefits at a certain cost [14]: the gain expected from promoting predicted items to speed up activation is unfortunately sometimes counter-balanced by the end user's need to constantly adapt to the altered layout, especially for novice users who hesitate between alternate areas. This introduces a first comparison criteria: spatial stability is preserved when the layout of the initial interface is left unchanged after adaptivity [6]. The cost/benefit ratio becomes positive when adaptivity improves the performance, such as reducing the menu selection time in a hierarchical menu [8] or when limited screen resolution induces long scrolling, thus affecting visual search performance [16]. Other observed shortcomings are [11]: adaptivity does not work well with small interfaces, when the end user alternates between items, when spatial instability is provoked by altering the initial layout. In particular, several forms of GAMs have been introduced [1] and demonstrated their potential benefits under given circumstances. Most of them [1, 6] concentrate on experimenting various presentations of the adaptive menu and investigate some forms of promotion: only menu items subject to a better prediction are subject to emphasizing. Nothing is said about those promoted items whose prediction is low and those promoted items whose interest decrease over time [9].

### Background on Adaptive Navigation

Reviewing adaptive techniques for web sites and hypermedia is beyond the scope of this paper, since it can be found in [3, 10]. These techniques basically fall into three categories:

1. *Adaptive presentation techniques*, which adapt the presentation of elements depending on any contextual parameters belonging to any model, e.g. a user model, a task model, a platform model, an environment model.
2. *Adaptive navigation techniques*, which do the same of elements regulating the navigation or the dialogue of an application (e.g., page browsing, link traversing, page reading, local and global behaviors).
3. *Adaptive contents techniques*, which are relevant to the domain of contents recommendation.

Adaptive navigation is achieved either through selecting links from a larger list (and hiding, removing the non-recommended links) or by automatically generating targets for predefined links, therefore inducing spatial instability as soon as the initial page is altered. Most adaptive navigation research focuses on adaptive navigation that does not restrict the user but rather provides suggestions as to which links or paths are more appropriate. Promotion and demotion are interesting techniques for supporting adaptive navigation [10], such as by clustering, where similar pages are formed into groups by promotion/demotion, where pages are assumed to

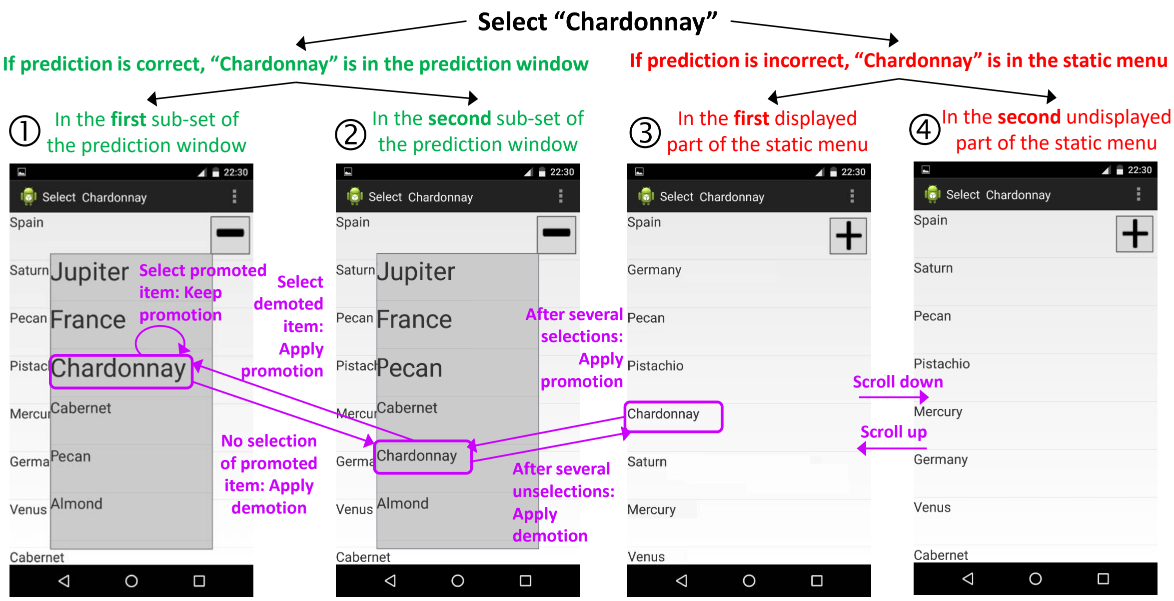


**Figure 2: State-transition diagram of the bimotion menu tested in the user study.**

be arranged hierarchically, and pages are moved closer to and farther away from the home page according to use. Another method consists of building a Markov model to characterize navigation links and apply promotion/demotion to adapt this navigation depending on navigation history. In conclusion, since graphical adaptive interfaces primarily rely on layout-changing techniques that do not preserve spatial stability, other techniques might be investigated to preserve some stability. On the other hand, promotion, demotion of links has been extensively applied to support adaptive navigation in web applications [4]. While all the aforementioned adaptive interfaces provide some degree of adaptivity, none of them is as adaptive as we are proposing: enabling widget adaptivity based on a recursive prediction window intended for predicted widget activation by promotion/demotion. None has examined which promotion/demotion might be used to render adaptivity with respect to existing graphical adaptive menus in particular. Pro/demotion has never been applied to mainstream GUIs.

## 3 DESIGN OF THE BIMOTION MENU

Based on our experience and on related work, requirements were elicited for the bimotion menu: **(R1)** provide an adaptive list that improves interaction when item prediction is accurate without depreciating it when prediction is inaccurate, **(R2)** support item adaptivity by promotion/demotion, **(R3)** maintain spatial stability and consistency of the initial list, and **(R4)** support item promotion/demotion recursively in the prediction window. In order to satisfy **R1**-**R3**, a bimotion list consists in two parts: the *initial menu* (i.e., the static part containing all the menu items left untouched throughout the whole interaction to satisfy R3) and the *prediction window* (i.e., the adaptive part containing predicted items). Bimotion menus were developed as a new widget in Java for Eclipse based on Android Software Development Kit with:

**Prediction rendering**: among all techniques used for rendering the prediction, emphasizing is applied for item promotion and de-emphasizing for item demotion since they are similar and congruent, which make them easier to compare. In our algorithm, for example, a probability of .8, resp. .6, will be rendered with a emphasizing factor of 1.4, resp. 1.2. Other renderings could be also considered, such as twisting and pulsing [13], which work for promotion, but not for demotion: if an item is pulsing to emphasize it, it cannot be "un-pulsed" to de-emphasize it. The prediction window should preferably be displayed in context of the menu [15].

**Prediction organization**: Bimotion Menus sort items in decreasing order of prediction and by group of similar prediction levels, but also organize them in the prediction window according to existing grouping relationships.

**Prediction recursion**: detecting items that are selected or not from the prediction window leads to pro/demotion.

**Consistent item selection**: Bimotion Menus do not provide any other interaction with any domain specific functionality not already available in some way via the graphical interaction, thus assuming that the initial menu covers all the functions to preserve consistency.

**Implicit item selection**: acceptance or rejection of item can be issued graphically with a single implicit action: pressing `[Escape]` or `[-]` or tapping outside the prediction window for rejection and pointing for acceptance (Fig. 1b). No further confirmation is required after item selection.

Bimotion Menus appear in two forms.

*Uni-dimensional linear vertical menu* (Fig. 2), where the prediction window is divided into two subsets: a first containing items with high prediction and a second with low prediction. Promotion is applied to the first subset: items are emphasized with a larger font size to attract user attention. Depending on the accuracy, these items are of interest to the user and demotion could be applied by de-emphasizing with a smaller font size. If an item is selected inside the prediction window, for an accurate prediction, promotion was efficient and demotion is not required. If an item was selected outside the prediction window, for an inaccurate prediction, promotion was inefficient and demotion is applied.

*Two-dimensional horizontal/vertical menu* (Fig. 1c), where the behavior is similar to the 1D menu, except that promotion/demotion of icons are ensured by fade in/out with size increase/decrease of icon sizes. Ponsard et al. [13] compared icon emphasizing techniques on a smartphone: direct highlighting effects include icon shrinking (playing with size variable), rotating (playing with orientation visual variable), pulsing (multiplying the size by 2), and twisting (multiplying orientation by 2 by back and forth rotation) ; indirect effects include grey scale (playing with color), transparency (playing with opacity), and blur. Twist and pulse improve search time performance by 8-10% and pulse is preferred.

## 4 USER STUDY

To investigate the impact of bimotion on adaptive menus, a user study is conducted comparing three conditions:

(1) *Static menu*: the Control condition (**C**), consisting in a static vertical menu with a full static menu.
(2) *Adaptive menu*: the Adaptive condition (**A**), consisting in a graphical adaptive menu with a 3-6 items prediction window (Fig. 1a). This menu is introduced to test whether the bimotion Menu has some impact in itself, and not just because of its adaptivity. Otherwise, there is a risk of attributing the potential benefit of bimotion Menu to its adaptive behavior and not to its working.
(3) *Bimotion menu*: the bimotion condition (**BM**), consisting in an adaptive menu with a 6-item prediction window with pro/demotion (Fig. 1b). BM therefore considers prediction recursion as opposed to none in A condition. Six items are considered to distinguish the promoted subset from the demoted (Fig. 2).

### Method

To shape our method, the Findlater's test [6] on ephemeral adaptation, was considered as a reference method since it has been extensively used in various studies for consistent comparison: any menu item with high prediction (probability < 80%) is located at the top of the prediction window, any item with a low prediction (60% < probability < 80%) is located at the bottom of the prediction window, and any item with a prediction below a tailorable threshold (threshold < probability < 60%) is not displayed in the prediction window. In Findlater's test, the prediction is rendered by progressive fade-in of non-predicted item: there is no separate prediction window. The full menu contains 4 groups of 4 related items (i.e., England, France, Germany, Spain - Venus, Mercury, Jupiter, Saturn - Cabernet, Chardonnay, Merlot, Shiraz - Almond, Pecan, Pistachio, Walnut) and the prediction was defined as follows: Venus=80%, Spain, Shiraz=70%, Pecan, Cabernet, Pistachio=60%, all other items sharing the same normal probability. Although this menu was originally tested on desktop, we reused this configuration because a Zipf distribution (Zipfian $R^2 = .99$) across only 8 randomly chosen items out of the 24 items was used to determine them, the semantics of items do not preclude any prior knowledge, and they are all understandable by a person. Similarly to [6], there are three independent variables:

(1) The within-subject Menu Type (I3): it is defined as the initial static menu without any adaptation which serves as the control condition (C), the adaptive menu with its prediction window (A), or Bimotion (BM).
(2) The Prediction level (P2): it is said to be *high*, respectively *low*, when the target item to select belongs, respectively does not belong, to the prediction window. Let $P+$ denote a high prediction and $P-$ a low prediction in the rest of this paper.
(3) The Target location (T2): the 16 items are distributed across two screens, one immediately visible and the other browsable by vertical scrolling, each containing 8 items. The target item can therefore be located either on the first screen or on the second screen. Each time the first or the second screen will be involved, the condition will be suffixed by '1st' or '2nd'. For instance, the first screen of the Bimotion will be referred to as 'BM-1st' whereas the second screen of the adaptive menu will be referred to as 'A-2nd'.

### Hypotheses

The hypotheses formulated for this user study are the following: ***Speed with high prediction level***

$H_{11}$ = *A and BM will be faster than C.*

$H_{21}$ = *BM will be faster than A when A prediction window contains six predicted items.*

$H_{31}$ = *BM is not worse than A when A prediction window contains three predicted items.*

***Speed with low prediction level***

$H_{41}$ = *BM and A are not worse than Control condition.*

***Error with high prediction level***

$H_{51}$ = *Errors are less frequent in A than in BM.*

***Error with low prediction level***

$H_{61}$ = *There is no difference between A and BM in terms of error.*

### Task

Participants were instructed to perform a sequence of item selections. For each test condition, i.e., static, adaptive, bimotion, the user has to select "Start" button for starting the test (which also starts the chronometer), a message appears indicating the target item to be selected. Then the menu appears. In all conditions, the item name remains displayed at top of the screen as a reminder. The user has to select the requested target in order to move to a new selection. If the user makes a wrong selection, an error message appears inviting user to find the requested target. When the user succeeds in selecting the right target, a new message appears specifying the name of the new target. At the end of the test, a thank you message is displayed informing the user that the test is complete. In each menu, the item ordering was randomized after ten selections in order to avoid any learning effect on the menu layout. The sequence of item selection, the ordering of conditions, and the assignment of participants to these distributions, were all controlled by a random draw. Target position on first screen or on second screen and prediction accuracy level were also controlled in the same way. Once the initial menu is presented, the participant selects an item inside or outside the prediction window.

*Quantitative and Qualitative Measures.* Three dependent variables were measured: 1) speed (menu item selection time) that was measured by the time elapsed from opening the menu until final selection of target (in seconds); 2) task completion, based on recorded error rates; 3) scrolling time.

*Apparatus.* Android-based Google Nexus smartphones were used, with 2 Gb LPDDR3 RAM, 16 Gb of storage and a 1920 × 1080 pixel screen resolution (423 ppi). The prediction window displays maximum 6 items in an area representing 44%.

*Participants.* Fifteen subjects (6 females, 9 males, aged between 23 and 55 years) participated in this experiment. All participants were regular smartphone users and they were recruited in other organizations through a mailing list.

*Procedure.* Each participant performed the task in a controlled environment. Prior to the task, each participant was welcomed, signed the consent form, and filled in a short questionnaire on their profile and background. After the questionnaire was completed, the researcher demonstrated the usage of Adaptive and Bimotion Menus to participants and explained the principle of each condition without mentioning the two prediction accuracy levels. The participants were then given a short training period (5 min.) during which ten selections different from the test were completed. An entire test for each participant is thus composed of 130 targets:

- **20** selections for **C**: 10 items were located on the first screen and 10 items were located on the second screen.
- **70** selections for **A**: 20 selections when prediction is correct in a 6-item prediction window, half of them located in the 3 first predicted items and half located in the last 3 predict-ed items; 20 selections when prediction is wrong in a 6-item prediction window, half of them located in the 3 first predicted items and half located in the last 3 predicted items; 10 selections when prediction is correct in a 3-item prediction window; and 20 selections when prediction is wrong in a 3-item prediction window, half of them located on the first screen and half of them located on the second screen.
- **40** selections for **BM**: 20 selections for high prediction, half of them with target on first subset, and others with target on the second (complementary) subset; and 20 selections when prediction is wrong, half of them with target on first subset and others on the second subset.

In summary, the design was as follows:

| | | |
|---|---|---|
| | 15 | participants |
| × | 130 | target items (20 **C** + 70 **A** + 40 **BM**) |
| = | 1950 | item selections in total |

*Analysis.* After each participant, the questionnaire, selection times and error rates were added into a database, along with the results of the experiment. The data was entered in an anonymous format so the participants could not be identified. The data was analyzed by a dedicated MS Excel sheet combining graphs, descriptive, and inferential statistics.

## 5 RESULTS AND DISCUSSION

As we are in case of a skewed distribution, a non-parametric Friedman's test of differences among repeated measures with Bonferroni Type I correction was conducted. Wilcoxon signed-rank tests were used for post-hoc analysis. In case of correct prediction, a good adaptive condition should accelerate interaction. As expected, when prediction is correct and prediction window contains three predicted items in A ($M = 1.4$, $SD = .346$), selection time is significantly shorter ($\chi^2(3) = 24.52$, $p^{***}$) than when prediction is correct and prediction window contains six predicted items in A, BM, and C.

- **A** correct prediction with 3 predicted items ($M = 1.4$, $SD = .346$), **A** correct prediction and target in the $1^{st}$ subset ($M = 3.73$, $SD = 1.53$, $W_{14} = 3.20$, $p^{**}$), **A** target in the $2^{nd}$ subset of prediction window ($M = 4.50$, $SD = 2$, $W_{14} = 3.29$, $p^{***}$).
- **A** correct prediction with 3 predicted items ($M = 1.4$, $SD = .346$), BM correct prediction and target in the $1^{st}$ subset of prediction window ($M = 3.37$, $SD = 1.41$, $W_{14} = 3.17$, $p^{***}$), BM target in the $2^{nd}$ subset ($M = 4.01$, $SD = 1.92$, $W_{14} = 3.12$, $p^{**}$).
- **A** correct prediction with 3 items ($M = 1.4$, $SD = .346$), Control ($M = 3.04$, $SD = .63$, $W_{14} = 3.41$, $p^{**}$).

No significant difference was found between C and BM ($W_{14} = 1.36$, $p > .05$, *n.s.*). We were expecting some reduced selection time after promotion/demotion, but nothing appears clearly: Control condition ($M$ = 3.04, $SD$ = .63), BM correct prediction and target in the $1^{st}$ subset of prediction window ($M$ = 3.37, $SD$ = 1.41, $W_{14}$ = .85, $p > .05$, *n.s.*), BM target in the $2^{nd}$ subset of prediction window ($M$ = 4.01, $SD$ = 1.92, $W_{14}$ = 1.82, $p^{**}$). The same observation appears for A when prediction is correct and prediction window contains 6 predicted items (Control condition: $M$ = 3.04, $SD$ = 0.63, A correct prediction and target in the 1st subset of prediction window: $M$ = 3.73, $SD$ = 1.53, $W_{14}$ = 1.50, $p > .05$, *n.s.*), A target in the $2^{nd}$ subset of prediction window ($M$ = 4.50, $SD$ = 2, $W_{14}$ = 2.39, $p^{*}$). We also computed Rosenthal's $r$ coefficient for those significant cases since we are in the case of a Wilcoxon signed-rank test: in all these cases, Rosenthal's $r$ is above .5, which suggests that the effect size is large. We explored results further through some interaction steps analysis. In A condition with 6 items, the predicted item was halftimes among the 3 first items belonging to the 1st subset, and halftimes among the 3 last items belonging to the $2^{nd}$ subset. The same sub-sets were inside the BM condition ($1^{st}$ subset is promoted and $2^{nd}$ is demoted). There is a highly significant difference between all conditions ($\chi^2(6) = 33.23$, $p^{***}$). As expected, BM-1st subset facilitates rapid action ($M$ = 3.37, $SD$ = 1.42) and BM-2nd subset is significantly longer ($W_{14}$ = 2.07, $p^{*}$). Similarly, **A** condition with 6 predicted items and the target item located in the 1st subset is significantly shorter than A with 6 items with a target item located in the $2^{nd}$ subset ($W_{14}$ = 2.39, $p^{*}$). BM-1st subset is significantly longer than A with 3 predicted item ($W_{14}$ = 3.2, $p^{***}$), even if they should be equivalent as they both display the target item through a list of 3 items. Globally, 3 predicted items displayed together enable rapid action whereas 6 predicted items slow down interaction. Nevertheless, BM-1st subset is significantly shorter than A-2nd subset with 6 predicted items ($W_{14}$ = 2.05, $p^{*}$). Promotion effectively helps to display less elements, but a negative effect occurs, probably due to animation. When prediction is wrong, BM ($M$ = 3.01, $SD$ = 1.81), A (3 predicted items) ($M$ = 4.25, $SD$ = 0.88) and A (6 predicted items) ($M$ = 3.40, $SD$ = 2.83) are not worse than Control: (Control: $M$ = 3.04, $SD$ = .63, $\chi^2(3) = 7.79$,$p > .05$, *n.s.*). Finally, a good adaptive condition should generate fewer errors: errors are significantly different ($\chi^2$ = 7.35, $p^{*}$) between Control ($M$ = .33, $SD$ = .61), A ($M$ = 1.93, $SD$ = 2.12) and BM ($M$ = .86, $SD$ = 1.18). A pairwise analysis shows that A generates significantly more errors than control ($W_{14}$ = 2.57, $p^{**}$)) and than BM ($W_{14}$ = 1.9, $p^{*}$). These results contradict the initial expectation that adaptivity should become stable after a certain amount of promotions and demotions. Conversely, the control condition has a lower percentage of errors in all cases. In conclusion, only $H_{21}$ and $H_{41}$ are supported.

## 6 CONCLUSION

This paper introduced adaptivity by bimotion, a combination of promotion and demotion. Evaluation of a bimotion menu shows that promotion displaying 6 predicted items in two steps does not quite improve performance compared to control condition as well as to a adaptive menu displaying only 3 predicted items. But compared to an adaptive menu with 6 predicted items, both speed and error rate were better. Promotion could increase the number of predicted items without penalizing interaction, as opposed to a maximum of 3 items [6, 7]. Further research should explore promotion and demotion in a more parametrizable environment to identify the threshold beyond which bimotion becomes efficient.

## REFERENCES


[1] Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2016. A Design Space for Engineering Graphical Adaptive Menus. In *Proc. of EICS '16*. ACM, New York, NY, USA, 239–244.

[2] Robert Bridle and Eric McCreath. 2005. Predictive Menu Selection on a Mobile Phone. In *Proc. of ECML/PKDD'2005*. 327–329.

[3] Peter Brusilovsky. 2007. *Adaptive Navigation Support*. Springer, Berlin.

[4] Sven Casteleyn, Olga De Troyer, and Saar Brockmans. 2003. Design Time Support for Adaptive Behavior in Web Sites. In *Proc. of SAC '03*. ACM, New York, NY, USA, 1222–1228.

[5] Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A Predictive Model of Menu Performance. In *Proc. of CHI'07*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/1240624.1240723

[6] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. 2009. Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. In *Proc. of CHI '09*. 1655–1664.

[7] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proc. of AVI '06*. 201–208.

[8] Sheng-Cheng Huang, I-Fan Chou, and Randolph G. Bias. 2006. Empirical Evaluation of a Popular Cellular Phone's Menu System: Theory Meets Practice. *J. Usability Studies* 1, 2 (Feb. 2006), 91–108.

[9] Kyungdoh Kim, Robert W. Proctor, and Gavriel Salvendy. 2009. Menu Design in Cell Phones: Use of 3D Menus. In *Proc. of the 13th Int. Conf. on Human-Computer Interaction*. Springer, Berlin, 48–57.

[10] Evgeny Knutov, Paul De Bra, and Mykola Pechenizkiy. 2009. AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hyper. and Multi.* 15, 1 (2009), 5–38.

[11] Talia Lavie and Joachim Meyer. 2010. Benefits and costs of adaptive user interfaces. *Int. J. of Human-Comp. Studies* 68, 8 (2010), 508 – 524.

[12] Mike Perkowitz and Oren Etzioni. 2000. Towards adaptive Web sites: Conceptual framework and case study. *AI* 118, 1 (2000), 245 – 275.

[13] Antoine Ponsard, Kamyar Ardekani, Kailun Zhang, Frederic Ren, Matei Negulescu, and Joanna McGrenere. [n. d.]. Twist and Pulse: Ephemeral Adaptation to Improve Icon Selection on Smartphones. In *Proc. of GI*.

[14] Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proc. of UbiComp '12*. 173–182.

[15] Michelle Wiebe, Denise Y. Geiskkovitch, and Andrea Bunt. 2016. Exploring User Attitudes Towards Different Approaches to Command Recommendation in Feature-Rich Software. In *Proc. of IUI '16*. 43–47.

[16] Yaohua Yu and Zhengjie Liu. 2010. Improving the Performance and Usability for Visual Menu Interface on Mobile Computers. In *Proc. of AVI '10*. 369–372.