Exploring a Design Space of Graphical Adaptive Menus: Normal vs. Small Screens*

JEAN VANDERDONCKT†, Université catholique de Louvain, Belgium
SARA BOUZIT, Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, France
GAËLLE CALVARY, Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, France
DENIS CHÊNE, Orange Labs, France

Graphical Adaptive Menus are Graphical User Interfaces menus whose items predicted of immediate usage can be automatically rendered in a prediction window. Rendering this prediction window is a key question for adaptivity to enable the end user to efficiently differentiate predicted items from normal ones and to select appropriate items consequently. Adaptivity for graphical menus has been more investigated for normal screens, such as desktops, than for small screens, such as smartphones, where real estate imposes severe rendering constraints. To address this question, this paper defines and explores a design space where graphical adaptive menus are structured based on Bertin's eight visual variables (i.e., position, size, shape, value, color, orientation, texture, and motion) and their combination by comparing their rendering for small screens with respect to normal screens. Based on this design space, previously introduced graphical adaptive menus are revisited in terms of four stability properties (i.e., spatial, physical, format, and temporal), new menu designs are introduced and discussed for both normal and small screens. The resulting set of graphical adaptive menu has been subject to a preference analysis from which a particular design emerged: the cloud menu, where predicted items are arranged in an adaptive tag cloud. We investigate empirically the effect of the cloud menu on the item selection time and the error rate, with respect to a static menu and an adaptive linear menu. The paper then suggests a set of usability guidelines useful for designers and practitioners to design graphical adaptive menus in general and cloud menus in particular.

CCS Concepts: • **Human-centered computing → Graphical user interfaces**; **Ubiquitous and mobile devices**; **Smartphones**; *User studies*; *Gestural input*; *Empirical studies in interaction design*; Pointing devices; Touch screens; • **Applied computing → Personal computers and PC applications**;

Additional Key Words and Phrases: Adaptation, Adaptive user interfaces, Adaptivity, Graphical Adaptive Menus, Intelligent User Interfaces, Menu selection, Prediction scheme, Prediction window, Split interface.

ACM Reference Format:
Jean Vanderdonckt, Sara Bouzit, Gaëlle Calvary, and Denis Chêne. 2019. Exploring a Design Space of Graphical Adaptive Menus: Normal vs. Small Screens. *ACM Trans. Interact. Intell. Syst.* 9, 4 (November 2019), 39 pages. https://doi.org/10.1145/3237190

*This paper is an extended version of the paper [78].
†This is the corresponding author.

Authors' addresses: Jean Vanderdonckt, Université catholique de Louvain, Louvain Research Institute in Management and Organizations, Place des Doyens, 1, Louvain-la-Neuve, B-1348, Belgium, jean.vanderdonckt@uclouvain.be; Sara Bouzit, Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, Grenoble, Institute of Engineering, Univ. Grenoble Alpes, Grenoble, F-38000, France, sara.bouzit@gmail.com; Gaëlle Calvary, Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, Institute of Engineering, Univ. Grenoble Alpes, Grenoble, F-38000, France, gaelle.calvary@imag.fr; Denis Chêne, Orange Labs, Chemin du Vieux Chêne, 28, Meylan, F-38240, France, denis.chene@orange.com.

2160-6455/2019/11-ART $15.00
https://doi.org/10.1145/3237190

1 INTRODUCTION

Graphical User Interfaces (GUIs) of interactive applications can be subject to three primary forms of adaptation [32]: *adaptability* when the end user controls its adaptation process, *adaptivity* when the application controls the adaptation, or *mixed-initiative* when the adaptation process is collaboratively managed by both the end user and the application [44]. Between the two extremes exists a large spectrum for various forms of mixed-initiative adaptations [58] depending on the degree of intervention or control of the end user vs. the application. It then follows an adaptation life cycle with two regulation loops between the user and the application [19]: a perception-decision-action (PDA) loop for both the application and the end user and a learning-prediction-adaptation (LPA) for supporting the adaptation, this last being particularly expressive for adaptivity.

Adaptability gives the full potential and control to the end user, which is often appreciated for its flexibility, but depreciated for being time consuming, which is perceived as even more constraining when repeated. The end user tends to enter into adaptability only if the win exceeds the cost. This is the main reason why adaptivity has been introduced: to delegate the execution of adaptation to the application as a function that was previously ensured by the end user. Adaptivity exhibits a series of potential benefits, such as the ability to improve the three usual usability aspects: effectiveness [81], efficiency [39], and subjective satisfaction [32, 69]. By automatically changing the presentation and/or the behavior of the GUI depending on each individual end user, it is expected that these benefits will become ultimately profitable for the end user. In practice, however, many obstacles exist before adaptivity provides its full benefits. Moreover, adaptivity also brings its own limitations such as its perpetual change over time which prevents the end user from learning it and increases her cognitive load [52], the loss of control [41, 52], the rejection of adaptation [5], the high variation of its impact on usability [52, 66, 76], especially depending on end user's characteristics [38] and task [39], the need for accuracy [40], the limited performance [28], the need for ensuring predictability [23, 40, 73], the need for explaining and understanding why a particular adaptation technique has been applied [73, 76], the wish for providing the GUI with feedback on the adaptivity quality [4], and the need for appropriate measures for evaluating the impact of adaptivity [39].

Due to these variations and the large variety of GUI elements subject to adaptivity, we orient the discussion of this paper to graphical menus, which still represent today one of the most frequently used techniques for interacting with any web site or interactive application [6]. A significant body of research and development has been devoted to optimizing their usage [7], in particular by adapting them to any aspect of the context of use [26]: to the end user [23, 38], to the interactive task [39, 50], to the device/platform [16, 24, 32, 45, 69], and to the environment [17, 67] and its location [81]. Moreover, menu selection is an interaction style subject to many design issues [53] that are at the heart of menu engineering, such as automated generation [77] and evaluation of graphical menus [15].

Graphical Adaptive Menus typically present the end user with predicted items in order to speed up their selection without searching for them in deep and wide menus, particularly for feature-rich software where the amount of items becomes important [82]. With the continuous expansion of mobile applications running on an ever-increasing variety of mobile devices, new adaptivity techniques are required [4, 23, 34, 39] that consider constraints imposed by these devices such as a moderate computational power, a limited set of interaction techniques and a reduced screen resolution [33]. This last constraint significantly affects the navigation through several screens or pages, lists like phone setting or address books, and menus, especially when lengthy [69]. Since selection of a target requires a navigation time and a visual search time that are depending on the number of items that can be displayed in the GUI, mobile devices are particularly affected.

We hereby define the *prediction window* as the subset of menu items resulting from a prediction scheme. Although the name ”prediction window” suggests a graphical rendering, it does not preclude any particular modality usage. This paper will cover the graphical modality for addressing the following research question: what kind of graphical technique would be suitable to render a prediction window on small screens vs. normal screens?

This paper will not cover other modalities which could be alternatively used for this rendering, such as vocal, touch, gesture, or haptics. Such modalities, either taken individually or collectively, have been investigated for rendering the static menu, but not the prediction window. For instance, an auditory adaptive menu [83], a menu adaptation pattern [12] emphasizes items in a menu by adding graphical markings or selecting more noticeable colors in the graphical modality or by increasing their volume, adding acoustical markings when they are read in speech modality. The area of Multimodal Adaptive Menu is still in its infancy, with preliminary works such as the Adaptive Multimodal Framework for adapting multimodal applications [13] which comprises a detailed discussion of all adaptation steps or Polymodal Menus [18].

Even the graphical modality has not been fully investigated yet. This paper will focus on Graphical Adaptive Menus in two dimensions, mainly on their visual space. Other spaces, like the motor space and the cognitive spaces are discussed elsewhere [6], as well as three dimensional menus on small screens [49]. This paper also assumes that the *prediction scheme*, i.e., the method used to predict relevant menu items, is selected among the various existing ones: [23, 28, 54, 72, 81]:

- *Most Frequently Used* (MFU): the most frequently used menu items over a certain period of time ranging from a certain date until today.
- *Most Recently Used* (MRU): the most recently used items over a small recent period of time, like the last day, week, or month.
- *Least Recently Used* (LRU): the least frequently used menu items over a certain period of time [37].
- *Degree-of-Interest* (DOI): all the items for which a computed DOI returns a value above a certain threshold. For instance, the DOI can be based on navigation history, repetition, and context-aware interaction.
- *Topic-of-Interest* (TOI): all the items which have been defined as a marker of interest from the end user. For instance, international news, technology, productivity software.
- *Adaptive filtering*: all the items whose a cost function returns a value above a certain threshold. This cost function typically consists of a linear filter with a transfer function controlled by variable parameters to be adjusted depending on the context of use.
- *Context-aware adaptation*: the items are predicted according to end users’ actions depending on the context of use [62]. Different parameters can be examined, such as space, time, task, history, role, user preferences, social aspects, through various techniques borrowed from artificial intelligence and machine learning [13, 23].

The remainder of this paper is structured as follows: Section 2 will examine work related to graphical adaptive interfaces and menus with a particular filter for menus which can serve the prediction window rendering. Section 3 will define the design space used throughout the paper to structure the discussion of graphical adaptive menus. Section 4 will report on the results of a preference analysis concerning previously identified graphical adaptive menus. Section 5 will present two user studies related to the cloud menu, which emerged from the preference analysis. Section 6 will devise some usability guidelines for graphical adaptive menus and for the cloud menus. Section 7 will revisit the research question in the light of the contribution and will present some avenues to this research by structuring them by level.

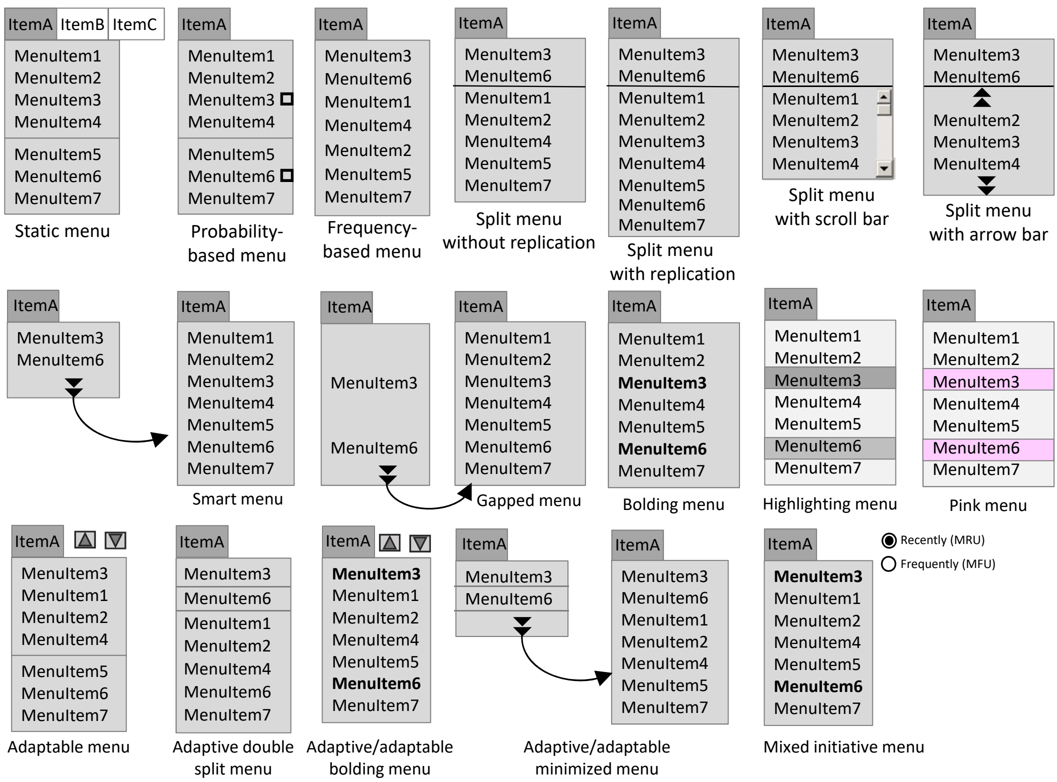

Fig. 1. Catalogue of menus: part 1/3 discussed in the Related Work.

2 RELATED WORK

Graphical adaptive interfaces exhibit a certain amount of potential benefits [41] at a certain cost [52, 76]. The cost/benefit ratio becomes positive when the adaptivity reduces the menu selection time, such as in a hierarchical menu [66] or when limited screen resolution induces long scrolling [53]. By predicting the three best items based on user's contextual information, e.g., activities, location, time, emotion, and weather, an accuracy of up to 69% can be obtained [54].

Other observed shortcomings are: adaptivity does not work well with short menus, when the end user alternates between items whose amount is larger than those contained in the prediction window, when *spatial instability* is provoked by altering the initial menu [39]. Various graphical adaptive menus exist that partially satisfy this property. This section reviews work related to menu selection in chronological order, with a particular attention to graphical adaptive menus. Fig. **??** presents an overview of most of the menus discussed in this section.

We hereby define a *static menu* as the initial non-adaptive version of the graphical menu, consisting of a hierarchy of menu items arranged in a linear way with their related properties, such as position, structure, and ordering which remain constant over time. Hence, a static menu always preserves spatial stability since all items are consistently positioned in the menu over time.

Probability-based menus [74] sort items in decreasing probability of selection: most popular items are positioned near the beginning of the menu. During the initial stages of practice, probability-based menus produce faster mean selection times than the static menu, the random menu (where items are randomly arranged), and the alphabetic menu (where items are sorted in increasing alphabetical order). After some practice, the static menu becomes better than the probability-based menu. End users remember easily the item position in the menu, indicating that spatial stability is an important factor in increasing the efficiency of menu selection.

Frequency-based menus, also called **Dynamic Menus** [63], sort items in decreasing order of frequency, depending on the end user's actions, thus promoting MFU. They are probably the first manifestation of adaptive menu in history during the 1980's with the probability-based menus [6].

Adaptive Prompting [50] predicts a selection of applications and related files based on an application model (which specifies relationships between elements) and a user model (which manually specifies the user's experience level, etc.) to reduce navigation effort. How applications and files are presented is also subject to adaptivity.

Split menus [72] assemble a prediction window (here, a topmost area promoting a small list of predicted items, typically 2-3) and a static menu. Menu items are initially predicted according MFU scheme and later on expanded to other schemes such as MRU, DOI, or any combination of them. Any menu item appears in either the static part or in the predicted window, which may confuse users who constantly oscillate between the two areas to find their item. The split menu was implemented for three conditions [32]: static (top four items remain static), adaptable (top four items can be moved up and down by end users), and adaptive (top four items are predicted according to user's recently and frequently used items). Static menus were found more efficient than both adaptable and adaptive menus, but adaptable menus were favored in terms of satisfaction.

Split menus with replication [39] restore spatial stability: the prediction window remains as stated before but the second part remains unaltered, thus enabling end users to always refer to the static menu as they know it. When this second part contains a more important amount of items, it could become reduced and scrollable either via a scroll bar (split menu with scroll bar) or with an arrow bar (split menu with arrow bar) [10]. Stable user traits, such as users with low-level extraversion and high-level needs for cognition, favor the adoption of split user interfaces [38].

Smart menus [46], featured in Microsoft Office 2003, initially show only the most commonly used items and all the available items by clicking on the arrow at the bottom of the smart menu to expand it. Smart menus track how often an end user invokes each item, in order to predict frequently used and recently used menu items. Smart menus provide beginners with a starting guided path toward learning a new UI. Menus become adaptive when they reflect the users' work habits. In short, a MFU or a MRU scheme automatically hides unused items. This method could be generalized to any prediction scheme: the prediction window is first displayed with unpredicted items hidden, the complete menu is displayed by clicking on the arrow. The lack of observability and understandability of the prediction as well as the dislike of the extra click and delay imposed lead most end users to deactivate this option [46].

Gapped menus [35] attempt to restore spatial stability to smart menus. Gapped menus present the static menu with only predicted items, leaving a blank space as a gap for unpredicted items. Clicking on the arrow at the bottom of the gapped menu displays the entire static menu. An experiment compared a static menu, a smart menu, and a gapped menu: item selection times and error rates were smaller for the static menu than for the smart menu. The gapped menu, whilst being as long as the static menu, was faster than the smart menu, but slower than the static menu.

Bolding menus [66] are a first form of emphasized menus where predicted items are boldfaced. A split menu, a bolding menu, an adaptable menu, and a traditional menu were compared for a normal screen [66]: the adaptable menu outperformed the other menus in terms of overall performance and subjective satisfaction. The split menu was estimated sub-optimal, especially when the predicted frequency changes. The bolding menu was not significantly better than the traditional menu in terms of item selection time, but was preferred by end users since estimated much less sensitive to the variations of prediction that its counterparts.

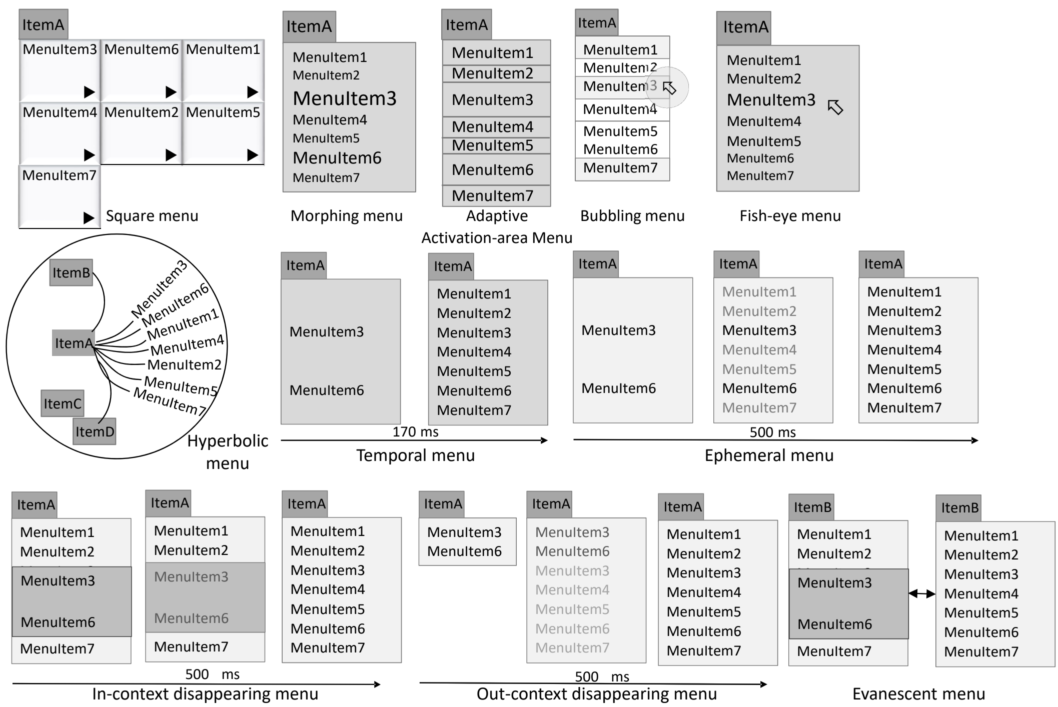

Fig. 2. Catalogue of menus: part 2/3 discussed in the Related Work.

Highlighting menus [66] emphasize predicted items by contrasting them with respect to normal items appearing in the static menu. For instance, Gajos et al. [39] highlight predicted items by colouring their background in pink. A study analyzed the effect of menu size on user satisfaction of five menus with different highlighting [2]: an adaptable menu where participants could move up and down predicted item; an adaptive double split menu divided into a section applying a MFU scheme, a section apply a MRU scheme, and the unpredicted items; an adaptive/adaptable bolding menu where participants could move predicted items up and down after 50 selections; an adaptive/adaptable minimized menu, which is a smart menu divided into a MFU section, a MRU section, and the rest of items to be displayed on demand with a moving facility; and a mixed-initiative menu [44], which is a bolding menu letting the participant to choose between MFU and MRU. In small menus, the minimized condition was the most preferred menus, followed by the adaptable, and bolding menus. The adaptive split and mixed-initiative menus were the least preferred menus. In large menus, the mixed-initiative was the most preferred, followed by the minimized menu; the adaptable menu was the least preferred followed by the adaptive split menu.

Square Menus [1] are re-layouting items of the pull-down menus into square regions in order to improve their selection performance. They performed a large analysis of several research studies to propose the Search, Decision, and Pointing (SDP) model. Square Menus improve specifically pointing performance, especially for experts. It was shown as a promising solution compared to traditional linear menus and to pie menus. It reduces Fitts' Law pointing time for experts and novice users performed better with traditional menus and even worse than with pie menus.

Morphing menus [28] change the font size of each menu item depending on its prediction: the font size is increased, respectively decreased, if the prediction is high, respectively low. While morphing menus preserve spatial stability, they facilitate selecting accurately-predicted items, but complexify selecting items with low or inaccurate prediction. Morphing menus made manipulation easier with enlarged activation areas in cascading menus. The objective was to eliminate the explicit delay for activation found in several implementations. The study shows that enlarged activation area and zero delays improve item selection by up to 29% in comparison to traditional methods.

Adaptive Activation-Area Menu (AAMU) [75] is an adaptive morphing menu containing an enlarged activation area for predicted items which dynamically resizes itself providing a broader steering path for menu navigation. AAMUs, whether they are used in isolation or combined with Force-field menus, outperformed the static menu.

Bubbling menus [76] represent a design for cascading drop-down menus aimed at accelerating the selection of the frequently used items by directly jumping to them one by one. To this end, two techniques are combined: the bubble cursor, whose size dynamically changes as the cursor moves and selects the target within the closest distance, especially for frequently-based items; and directional mouse-gesture techniques, which accelerate reaching predicted items.

Fish-eye menus [10] display items with a font size that increases or decreases depending on the distance with respect to cursor position: the closer, the larger, the further, the smaller. Per se, fish-eye menus are not adaptive since their layout does not change depending on predicted items. But similarly to AAMUs [75] and Bubbling menus [76], they are able to increase the selection area of any item, which might be useful for predicted items.

Hyperbolic menus [51] is a Focus + Context technique for displaying and manipulating large hierarchies. It displays several hierarchy levels at once according to a hyperbolic tree which minimizes screen usage. All menu items are displayed on this hyperbolic tree. Since parts of the hyperbolic view are expanding and collapsing depending on the position of the cursor, menu items are never displayed at the same place, thus inducing spatial instability, which is difficult to use for novice users and without fine visio-motor coordination.

Temporal menus [53] introduce a temporal dimension by displaying items on two stages: at opening, the menu displays only predicted items, after a delay of 170ms, non-predicted items appear. This menu maintains spatial stability, thus helping the end user to maintain a mental model of the menu. Transposing temporal menu to smartphone is not straightforward because all items cannot be displayed on a single screen. Any predicted item located on the subsequent screens requires a cognitive effort to explore the whole set of items.

Ephemeral menus [34] is an adaptive temporal menu where the gradual onset was used in order to display non-predicted items. At opening the menu, user finds predicted items and after a delay of 500ms remaining items appear gradually. This approach suffers from the same problem of temporal menu and items cannot be displayed on single screen in the case of a small screen device. Lee and Yoon [53] examined a dynamic menu, where predicted items appear immediately when a menu is opened (abrupt onset) and those outside of the subset appear after a delay.

In-Context Disappearing (ICD) approach has been applied to menus for small screens [20]: at opening the static menu, the end user finds a superposition of the full list of items with the prediction window prompting predicted items. This latter contains three predicted items and disappears gradually. The presentation of predicted items in the prediction window remains homogeneous, thus preserving spatial stability. ICD menus are a promising approach for accelerating interaction but it generates errors caused by the overlapping of two lists.

Out-of-Context Disappearing (OCD) approach is the inverse [20]: at opening, the prediction window is immediately displayed with the predicted items, like in a split menu; after 500 msec [34], the complete menu is gradually displayed from the back, thus replacing the prediction window.

Evanescent menus [21] are adaptive menu where the prediction window is first presented superimposed to the initial menu and then progressively made transparent to reveal the menu, thus enabling the user to select a predicted item if it belongs to the prediction window and the initial menu after.

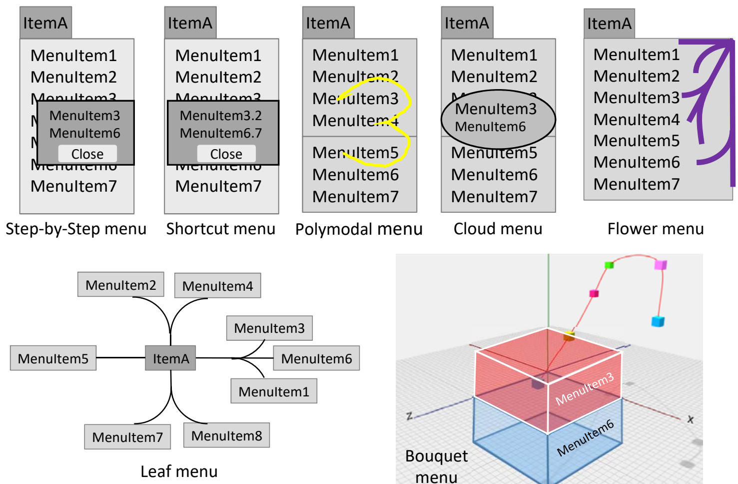

Fig. 3. Catalogue of menus: part 3/3 discussed in the Related Work.

Step-by-Step Menus [17] displays at each level of the hierarchy the prediction window and offers to select the menu item leading to the next level of the target path. **Shortcut Menus** [16] display the target item in a prediction window at the root level of the hierarchy. A comparison between two graphical adaptive menus, i.e., shortcut menus vs. step-by-step menu, was conducted to improve hierarchical navigation in smartphones [16]: step-by-step menus preserve the consistency with the static menu through level-by-level navigation to reach the predicted item. Shortcut menus directly moves the end user to the predicted item in its very right location, thus reducing if not eliminating the effort for navigating in the menu hierarchy.

Polymodal Menus [18] are graphical adaptive menus in which any menu item can be selected graphically (by pointing), vocally (by voice recognition), tactilely (by touching), gesturally (by issuing a gesture representing the menu), or any combination of them. Predicted menus are rendered by graphical or vocal prediction window. This user study suggests that polymodal menus represent a promising interaction technique for adaptive menus on a smartphone, particularly when eye-free condition should be imposed.

Cloud Menus [78] are an adaptive split menu for small screens where the predicted menu items are arranged in a circular tag cloud with a location consistent with their corresponding position in the static menu and a font size depending on their prediction level. An empirical study suggests that cloud menus reduce item selection time and error rate when prediction is correct without penalizing it when prediction is incorrect, compared to two baselines: a non-adaptive static menu and an adaptive linear menu.

From a survey of visual menus [6], several other types of menus could be identified as candidates for becoming an adaptive menu, either for normal screens or for small screens. We hereby review the most significant of them.

Flower menus [?] extend marking menus with opportunity to draw straight lines or curved ones into the eight cardinal directions of a compass, which can optionally be terminated by bended, cusped, and pig-tail endings. A comparative study of flower, linear and polygon menus shows that polygon and flower menus offer better performance for learning in expert mode as compared to linear menus. But as for previous techniques, flower menus do not accommodate well with small screens unless there is space enough to capture gestures on the device surface.

Wavelet Menus [36] promote with a circular and linear layout, while **Leaf Menus** [70] are optimized for item accessibility.

Bouquet Menus [25] consist of a marking menu offering flicks and marks from an origin towards the directions of the eights octants of a cube, thus generalizing the Flower Menu [?] into three dimensions. The area of 3D menus [49] is however out of scope of this paper.

Stacked half-pie menus [43] display menu items as circles in half pie on a tabletop surface. This interaction technique tends to make this design unlimited in terms of menu depth and breadth while still maintaining the initial form of the menu. This menu is limited for small screen devices like smartphones where there is not enough space on the screen. In addition, the navigation in the pie menu may be a constraint for novice users.

PocketMenu [67] exploited the idea of changing the modality for menu selection: menu items are laid out along the border of the touch smartphone within the hand comfort zone, tactile features guide the hierarchical navigation, a vibro-tactile feedback with speech allows identifying the items non-visually. This interaction technique is particularly useful for end users with visual disabilities.

Although the aforementioned menus present significant advantages that have been empirically assessed, these menus are not simultaneously aimed at small screen devices and adaptivity. Menu optimization also represents an active area of interest for graphical adaptive menus.

MenuDesigner [77] is aimed at automatically generating a menu bar, associated cascading menus, and menu items based on an activity chaining graph representing possible hierarchical navigation based on a task model [14]. This approach remains static (the menu structure is generated once for all), without any adaptation and could lead to inconsistent menus when items are arranged.

MenuOptimizer [7] is aimed at helping designers and developers to optimize the menu structure by maximizing consistency vs performance based on ant colony algorithm. While MenuOptimizer reveals the popularity of menu items by a color line under each menu item, thus leaving the menu structure untouched, it does not provide end users with an adaptive menu.

MenuErgo [47] provides a software environment for designing a menu bar along with its pull-down menus and sub-menus of a graphical user interface by automatic evaluation of menu usability guidelines according to four evaluation strategies: an active strategy initiated by the system, a passive strategy initiated by the designer, a mixed strategy shared by both of them, and a strategy by conceptual units based on the semantic domain.

3 DESIGN SPACE FOR GRAPHICAL ADAPTIVE MENUS

To define our design space, we rely on Bertin's semiology defining eight *visual variables* to effectively and efficiently convey a change [11] (Fig. 4): *position* (e.g., change in the x, y, z location), *size* (e.g., change in length, area, or repetition), *shape* (e.g., change by shape, regular or not), *value* (e.g., a change of color saturation, a change from light to dark), *orientation* (e.g., a change in alignment, angle), *color* (e.g., change in hue at a given value), *texture* (e.g., a change in pattern, in gradient), and *motion* (e.g., a change by animated transition or a visual effect). The eight visual variables have become a reference in visual design, information visualization, communication, and Human-Computer Interaction. Bertin also defined five *marks*, which are basic signs representing some piece of information other than itself:

(1) *Points*: are elementary signs that can be modulated by size, shape or color for visualization.
(2) *Lines*: are signs composed of a segment of points modulated by length, line type, thickness.
(3) *Areas*: are signs composed of a series of lines having a length, a width to form a 2D size.
(4) *Surfaces*: are signs composed of a series of areas to form a flat object in a 3D space, i.e., without any thickness.
(5) *Volumes*: are signs having a length, a width, and a depth, thus having a 3D form.

Each visual variable serves one or many purposes: *selective* (is a change perceivable enough to allow us to select it from a set?), *associative* (is a change perceivable enough to allow us to perceive

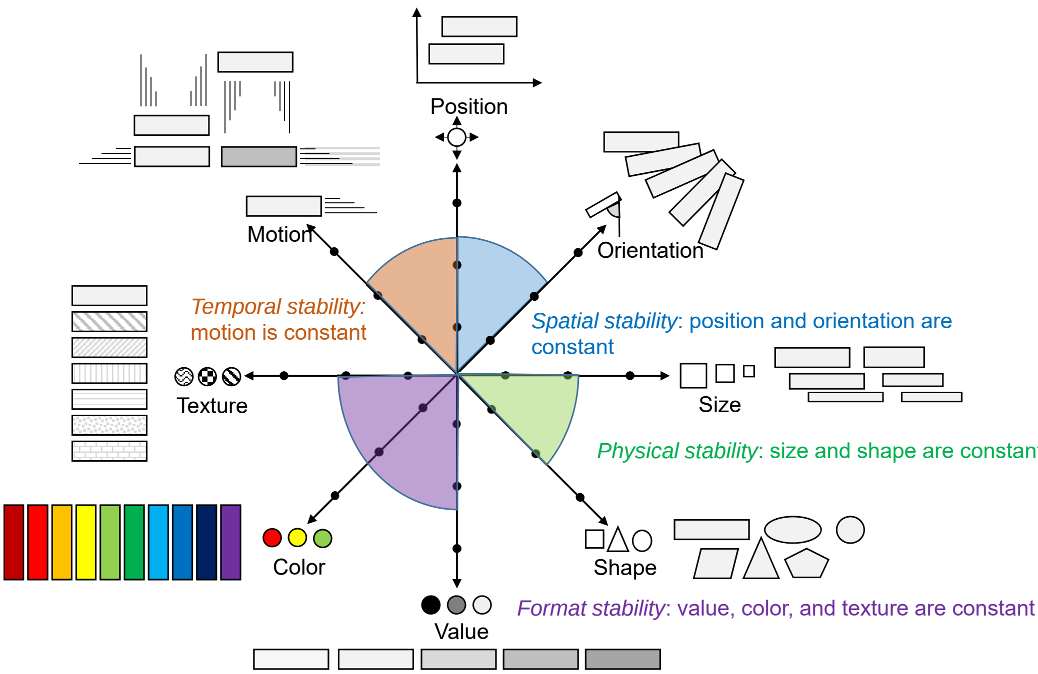

Fig. 4. The Design Space for Graphical Adaptive Menus.

Visual variable	Selective	Order
Position	●	●
Size	●	●
Shape	◑	○
Value	●	●
Color	●	○
Orientation	◕	○
Texture	●	○
Motion	●	○

Table 1. Level of support of visual variables expressed according to Harvey's Balls.

it as a whole?), *quantitative* (is a change perceivable enough to convey a numerical interpretation of it?), *order* (are changes according to this variable perceived as ordered?), and *length* (how many differentiations could be perceived by changing this variable?). Bertin's visual variables have a varying ability to be selective and/or ordered, which are the two main expected properties for adaptivity (Table 1). Four stability properties [17] could be defined depending on which visual variable has not been altered by a graphical adaptive menu:

(1) *Spatial stability* is defined as the ability of a graphical adaptive menu to preserve its spatial layout after adaptivity, thus keeping position and orientation constant.
(2) *Physical stability* is defined as the ability of a graphical adaptive menu to preserve its physical configuration after adaptivity, thus keeping size and shape constant.
(3) *Format stability* is defined as the ability of a graphical adaptive menu to preserve the format of its layout after adaptivity, thus keeping value, color, and texture constant.
(4) *Temporal stability* is defined as the ability of a graphical adaptive menu to preserve its position over time while being adapted, thus keeping motion constant.

Menu type	Property							
	Spatial		Physical		Format			Temporal
	Position	Orient.	Size	Shape	Value	Color	Text.	Motion
Static menu	Constant							
Probability-based menu	var.							
Frequency-based menu	var.							
Split menu	var.							
Split menu with replic.	var.		var.					
Split menu with scroll	var.							var.
Split menu with arrow	var.							var.
Smart menu	var.		var.	var.				
Gapped menu					var.			
Bolding menu					var.			
Highlighting menu					var.			
Pink menu					var.	var.		
Adaptable menu				var.				
Ad. double split menu	var.							
Ad./ad. bolding menu	var.			var.	var.			
Ad./ad. minimized menu	var.		var.	var.				
Mixed initiative menu	var.				var.			
Square menu	var.		var.	var.				
Morphing menu	var.		var.					
Adapt. activation-area m.	var.		var.	var.				
Bubbling menu						var.		
Fish-eye menu	var.		var.	var.				
Hyperbolic menu	var.	var.	var.	var.				var.
Temporal menu					var.			var.
Ephemeral menu					var.			var.
In-Context disapp. m.				var.				var.
Out-of-context dis. m.				var.				var.
Evanescent menu				var.				var.
Step-by-step menu			var.	var.				
Shortcut menu			var.	var.				
Polymodal menu								var.
Cloud menu				var.		var.		var.
Flower menu						var.		var.
Leaf menu	var.		var.	var.		var.		var.
Bouquet menu	var.	var.				var.		var.

Table 2. Existing graphical adaptive menus compared by stability property.

Table 2 compares existing graphical adaptive menus with respect to the four aforementioned stability criteria by detailing which visual variable is affected. The first line of Table 2 characterizes a static menu, which obviously keeps constant all visual variables, thus satisfying the four properties. For each other graphical adaptive menu, only the visual variables that are not preserved are reported in each entry. For instance, probability-based menus, as well as frequency-based menus, do not

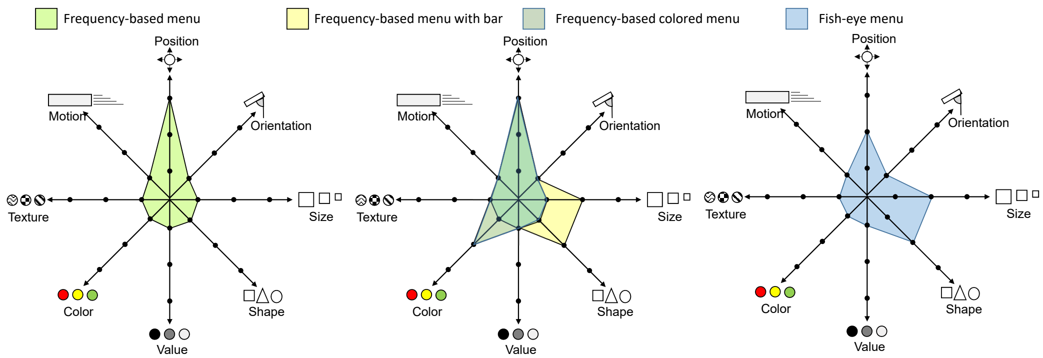

Fig. 5. Menus on the Design Space: (a) Probability-based menu, (b) Various frequencies, (c) Fish-eye menu.

keep constant the position of menu items, therefore spatial stability is not established. Conversely, physical, format, and temporal stabilities are satisfied. Since the prediction window should appear in some different way than the static menu, each visual variable could in principle initiate a family of graphical adaptive menus. The design space is represented as a radar diagram (Fig. 4), where each axis denotes one visual variable at a time, with a scale ranging from the central point (no coverage) to low, medium, and high coverage. By connecting the dots on each scale, a graphical adaptive menu is graphically represented by a footprint on the radar diagram. Exploring the design space induced by the eight visual variables should serve three virtues (Fig. 4) [9]:

(1) *Descriptive virtue*: each menu selection technique could be characterized on the design space with the same terms. For instance, Fig. 5a represents the probability-based menu in terms of the design space. According to Table 2, only the position is variable, which represented as a higher value on its corresponding axis.
(2) *Comparative virtue*: two or more menu selection techniques could be consistently compared against the same criteria, representing the dimensions of the design space. For instance, Fig. 5b superimposes three ways to represent the frequency of menu items: by position, by color, by a histogram representing the frequency.
(3) *Generative virtue*: the analysis of existing techniques based on the design space should enable us to report on already well-covered areas and to identify gaps to be filled in by suggesting new, potentially unexplored, techniques. For instance, Fig. 5c represents how fish-eye menus could be exploited for focusing on predicted items as opposed to unpredicted items. This usage of fish-eye menus has never been considered.

Based on this design space, we now investigate eight families of graphical adaptive menus by systematically examining each visual variable. For each family, either new designs or extrapolated designs are introduced and discussed. When a graphical adaptive menu is extrapolated, its basic reference will be mentioned. For instance, [68] invented the twisting icons, it is transposed here to graphical adaptive menus as twisting menus. When a graphical adaptive menu touches several visual variables at once, it will fall in the family corresponding to the most relevant or salient visual variable. How these designs preserve stability properties is summarized in Table 3.

3.1 Position-Changing Menus

Graphical adaptive menus where the position of predicted items changes depending on the prediction scheme are the primary form of adaptivity investigated ever, with probability-based menus [74], frequency-based menus [63], split menus with replication [39] or without [72] as their most representative examples. Despite their reported benefits, such as the fastest overall performance [28, 72], these menus are challenging when machine learning algorithms pick the wrong items in

Menu type	Property							
	Spatial		Physical		Format			Temp.
	Position	Orient.	Size	Shape	Value	Color	Texture	Motion
Static menu	Constant							
Pushpin menu	var.		var.	var.				
Time-based menu	var.		var.					
Localized menu	var.	var.						
Flippable menu	var.	var.						var.
Twisting menu	var.	var.						var.
Rotating menu	var.	var.						var.
Pulsing menu			var.	var.				var.
Prediction by bar			var.	var.				
Prediction by scale			var.	var.				
Rating menu			var.	var.				
Greyscaling menu					var.			
Italicizing menu					var.			
Underlining menu					var.			
Boxing menu			var.		var.			
Font-changing menu			var.		var.			
Transparency menu					var.			
Blurring menu					var.			
Blinking menu					var.		var.	
Rainbow menu					var.	var.		
Glowing menu				var.	var.	var.		
Prediction by color						var.		
Heatmap menu					var.	var.		
Prediction by line				var.		var.		
Prediction by rainbow				var.	var.	var.		
Fish-eye colored m.	var.		var.			var.		
Textured menu							var.	
Patined menu							var.	
Weared menu					var.		var.	

Table 3. New or extrapolated graphical adaptive menus compared by stability property.

the prediction window, thus making it more difficult to complete low frequency tasks. When menu items are moving around, they undermine the end user's memorability of the system, especially in multi-tasking and multi-device environments. All these menus are largely criticized for endangering their spatial stability, which may confuse end users in the ultimate item positions that are perpetually changing, thus preventing them from building a permanent mental model based on their layout. Split menus with replication partially escape from this drawback: the end user may first check whether the desired item belongs to the prediction window and, if not, browse the normal menu. The static part is position-invariant whereas the prediction window is not. Other suggested forms of position-changing menus are (Fig. 6a): the **pushpin menu** (a split menu where predicted items can become permanently placed by locking them with a pushpin) and the **time-based menu** (a multi-split menu where predicted items are sorted in chronological time). This induces a new category of split menus where different portions could contain different split areas according to

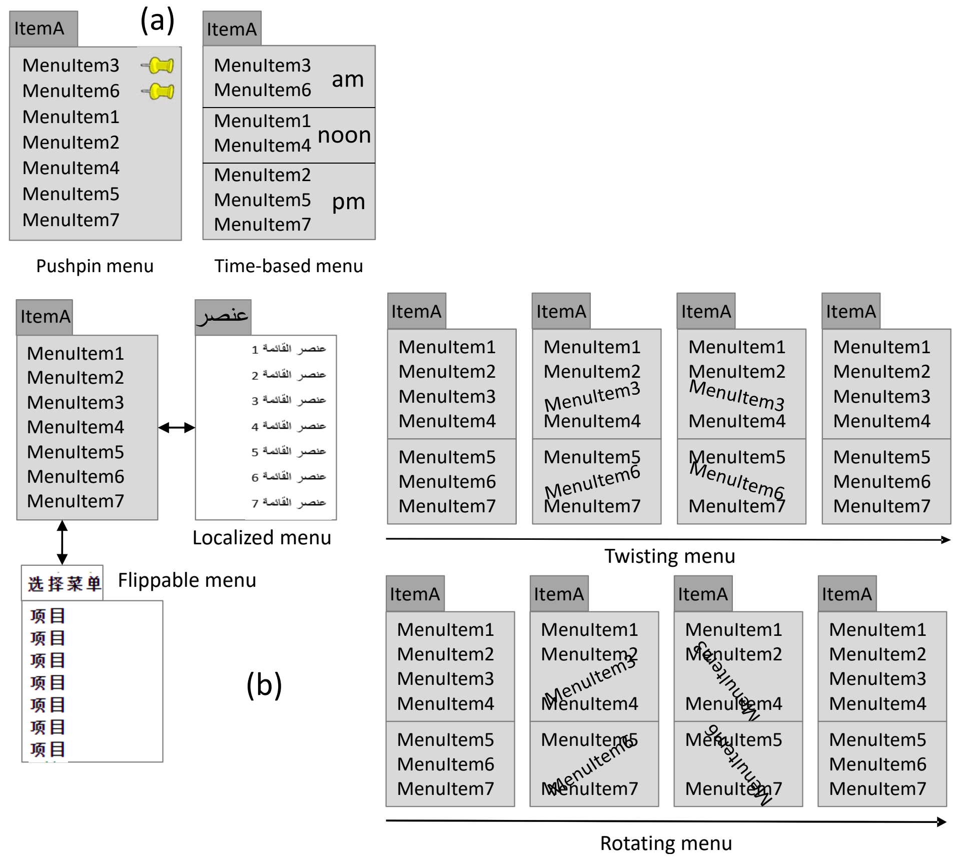

Fig. 6. Catalogue of menus: part 1/4 of new designs with position-changing (a) or orientation-changing (b).

different prediction scheme: single split, double split (as in Fig. 1), or multi-split menu. Later in this exploration, other families of variable-changing menu will inevitably affect the position and thus the spatial stability: menus with layout and/or a selection area modified by the prediction. Adaptive activation-area menus [75] are representative as they change the selection area depending on the prediction of each item. Similarly, morphing menus [28] or square menus [1], enlarge this zone by resizing it as a rectangle or a square.

3.2 Orientation-Changing Menus

Any technique changing the orientation of predicted items falls in this category. A typical example concerns hyperbolic menus [51], which are not a graphical adaptive menu per se. But the visualization technique can expand sub-trees unveiling predicted items and collapse sub-trees containing unpredicted items. Since items are distributed along a hyperbola, their orientation changes as the tree is expanded or collapsed. This visual variable has never been subject to any investigation as far as we know, although it has some potential to be further examined. For instance, label orientation, i.e., horizontal, vertical, angular, could be considered for emphasizing a predicted item, especially in cloud menus [78]. When a **localized menu** is adapted according to the end user's culture and language, items can be flipped automatically [48] between a Left-to-Right (LTR) format as used in Western cultures and a Right-To-Left (RTL) format as used in Arabian languages, or from Top-To-Bottom (TTB) to Bottom-To-Top (BTT), thus giving rise to a **flippable menu**. Similarly to twisting icons [68], **twisting menus** briefly change the orientation of predicted items by twisting

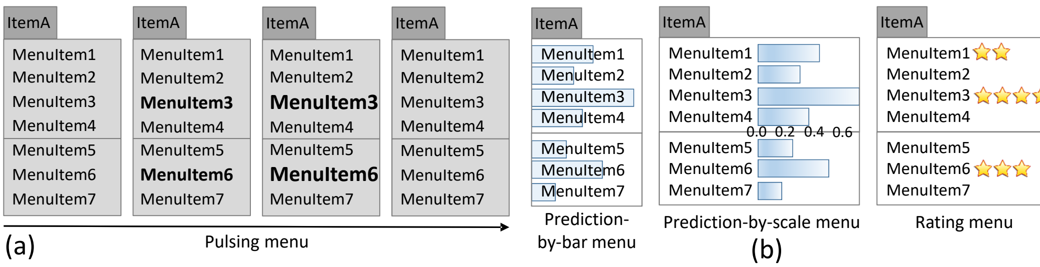

Fig. 7. Catalogue of menus: part 2/4 of new designs with size-changing (a) or shape-changing (b).

them with respect to the horizontal, while in **rotating menus**, predicted items are subject to a 360° rotation. The speed and the frequency to be used for these animations are yet to be determined, although we know that the speed should last more or less 500 msec and the frequency should not repeat the same animation too many times in order not to induce any boredom.

3.3 Size-Changing Menus

Four types of actions can modify the menu size: modifying the selection area (such as in morphing menus [28]), adding another menu part (such as in split menus with replication [39], adding another user interface element for shortcutting the menu hierarchy (such as in shortcut menus [16], and translating/localizing their labels (e.g., translating an English item produces a longer item in French and even longer in German). Size can be further decomposed into four sub-variables depending on how many dimensions are considered: line (for a one-dimensional change), surface (for a two-dimensional change), multi-surface (for a $2D1/2$ change when a surface is projected onto another one), and volume (for a three-dimensional change). The **pulsing menu** (Fig. 7a), inspired by the pulsing icons [68], pulse forward predicted items, thus changing their size until they return to their initial state. Pulsing is perceived less intrusively than strong animations found in a rotating menu.

3.4 Shape-Changing Menus

While the rectangle remains the predominant shape for menus, other shapes have been considered, but not necessarily for adaptivity. For instance, square menus [1] delineate a rectangular area for each menu item appearing in a squared menu to improve item selection. **Radial menus** [57] present menu items according to a (semi-)circular layout, showing that for different breadths and depths, they can be superior to their equivalent cascading menu counterpart. What has not been investigated so far for adaptive menu are shape-changing menus: depending on the amount of predicted items and the prediction scheme, the menu could gracefully evolve from one shape to another that is more suitable for displaying the items as they are: circle, oval, square, rectangle, pentagon, hexagon, heptagon, octagon, parallelogram, trapezium, etc. Some of these shapes have been particularly exploited for gesture-based menus. Shape-changing menus should be possible to highlight predicted items when hovering for instance: the shape of the menu bar would change from a square to a somewhat rounded shape or other shape when you hover over them, thus revealing items only on demand. The menu shape is also affected when the prediction window is moved onto a separate area, which makes it more distinguishable by end users [82], as in split menus with replication [39]. This prediction area is itself subject to shape-changing: line (e.g., for expressing the frequency of a menu item in frequency-by-line menus), histogram, square, rectangle, circle as in cloud menus. Fig. 7b suggests three shapes to convey the likelihood of predicted items: a bar superimposed to the item in the **prediction-by-bar menu**, a bar juxtaposed to the item with a scale such as a histogram in the **prediction-by-scale menu**, or a rating scale in the **rating menu**. The last two menus considerably increase the menu width.

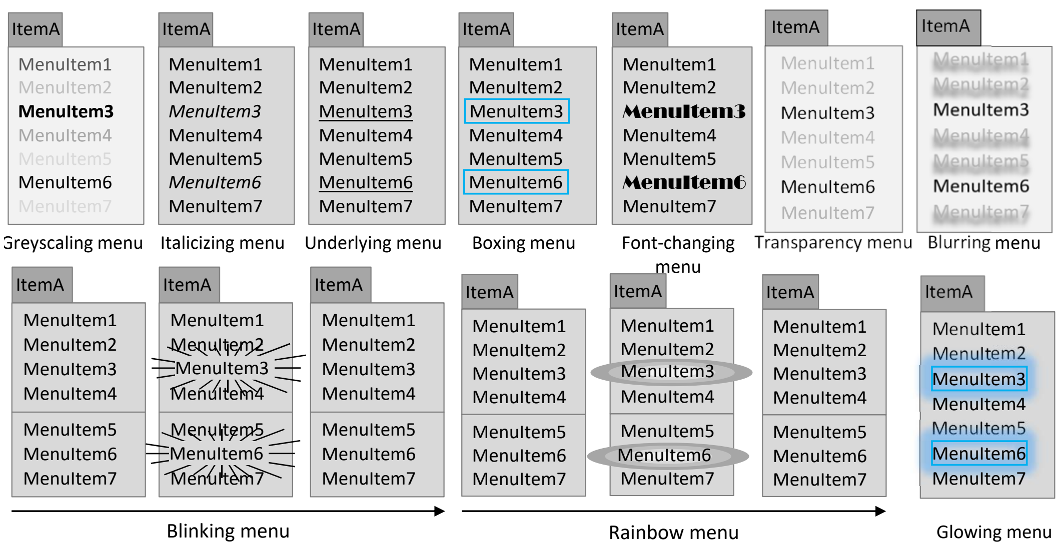

Fig. 8. Catalogue of menus: part 3/4 of new designs with value-changing.

3.5 Value-Changing Menus

Any user interface highlighting technique can be applied to predicted items in value-changing menus: bold, italics, underscores, boxes, capitalization, single or double quotation marks, alternate fonts, emphasis techniques in computer text, or any combination of the preceding. Color is considered in the next family. Highlighting menus [3] highlight predicted items by contrast, which is viable since up to three levels of contrast are usually distinguishable by end users. Bolding menus [3] apply a bold font to predicted items. These two menus were introduced for a comparison with split menus, adaptable menus, and traditional menus on a desktop [32]. This study showed that the adaptable menu outperformed the other menus in terms of overall performance and subjective satisfaction. The split menu was estimated sub-optimal, especially when the predicted frequency changed. The bolding menu was not significantly better than when working with variations in the traditional menus, but was preferred by end users since less sensitive to the variations of prediction that its counterparts. Fig. 8 suggests several forms of value-chaining menus: the **greyscaling menu** where items are greyscaled depending on their prediction (the more greyscaled, the less predicted), the **italicizing menu** where predicted items are formatted in italics, the **underlying menu** where labels of predicted items are underlined, the **boxing menu** where predicted items are surrounded by a visual frame, or the **font-changing menu**, where a different font is used for representing predicted items as opposed to unpredicted ones. This technique heavily depends on the font legibility (e.g., sans-serif fonts are more legible on screen than serif fonts) and their recognizability (some popular fonts can be recognized but not all). Many other visual effects could be envisioned but their effectiveness and efficiency is not demonstrated yet: unpredicted items could be subject to a transparency stencil in a **transparency menu**, blurred in a **blurring menu**, subject to blinking with a small rate in a **blinking menu**, animated with a rainbow in a **rainbow menu**, or glowing to make them more salient in a **glowing menu** as in Phosphor Widgets [8].

3.6 Color-Changing Menus

Color-changing menus [68] were proposed in order to reduce visual search time: frequently used items are highlighted by changing their background or foreground (font) color or both. The study compared color menus to fish-eye menus [10]: on smartphones, color menus require a lot of

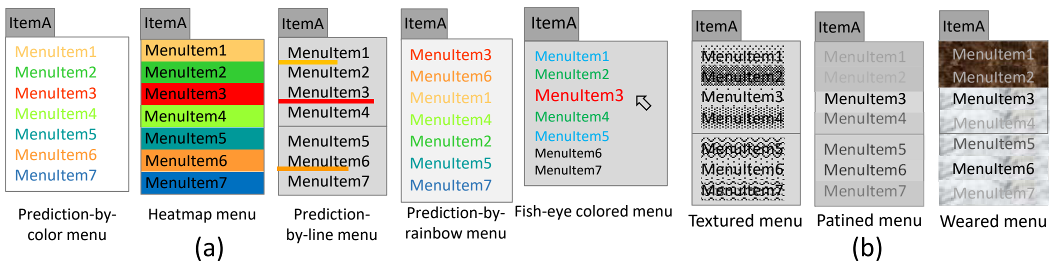

Fig. 9. Catalogue of menus: part 4/4 of new designs with color-changing (a) or texture-changing (b).

concentration from the user, especially when predicted item is located at the bottom of the screen. The user must scroll the window to see the item. For instance, a **prediction-by-color** menu (Fig. 9a) would display all items with a color code associated to the prediction level, while in a **prediction-by-rainbow menu**, menu items would be sorted and displayed according to a rainbow color scheme. We suggest that a **heatmap menu** color its items in a heatmap depending on their frequency of use. Contrarily to split menus where frequent menu items are first presented or to morphing menus where frequent items are enlarged, heatmap menus do not change their layout. The perception of the heat map may induce some usability problems, since rainbow scales are not always correctly interpreted. Heatmap menus could be presented in two ways: the heatmap shows all dots generated by interaction on a menu item (e.g., a mouse click, a finger touch, an eye saccade captured by an eye tracker), or on the entire item region by aggregating the dots and attribute them to the menu item that actually generated them.. Since visual variables are independent of each other, more than one variable could be used to reinforce the adaptivity when critical. For instance, a prediction-by-color menu sorts items in decreasing order of prediction rendered on a rainbow scale: both position and color are altered. A **fish-eye colored menu** plays mainly with size (since the size of items depends on the cursor position), but also with position because the change of size implies a change of absolute position, but not a change in the item ordering.

3.7 Texture-Changing Menus

A **textured menu** results from applying any change of the texture of predicted and/or unpredicted items without changing the rest of its format. Patina [61] dynamically creates a heatmap depicting how frequently a user interface element, such as a menu item or an icon, is used. The Patina is overlaid to the initial user interface, for instance a toolbar, an icon palette. This inspires a **patined menu**, where menu items are overlaid with a transparency texture depicting their usage: the more predicted, the more visible. Instead of progressively hiding unpredicted items, which may prevent end users from appropriately locating them, we can also introduce a **weared menu**, which change their texture based on their usage frequency (MFU) and recency (MRU). Computational wear [59] textures web site links based on three measures:

(1) The *last update date*, which estimates the usefulness of a web site: a newly updated site usually contains latest and timely information. The length of the period from the last updated date to the current is mapped to the magnitude of the rust of its metallic-like icon: the older the last update is, the more rusty the link appears.
(2) The *total number of access times*, which estimates the popularity of the site: a well-accessed site usually contains useful information. This number is mapped to the wear and smudge of the icon: the more often the link is used, the more worn out and smudged it appears.
(3) The *last access date*, which measures the recent popularity and also the usefulness of a site. The length of the period from the last accessed date to the current is mapped to the amount of dust accumulated on the surface of the icon: the older the last access is, the more dusty the link appears.

3.8 Motion-Changing Menus

Although the time dimension expresses a fourth sub-variable of position (i.e., add t after x, y, and z), *time* is typically involved in the motion variable, especially in information visualization. Motion could be even further decomposed into sub-variables, such as: direction, speed, frequency, rhythm, flicker, trails, and style. Temporal menus [53] introduce a temporal dimension by displaying items on two stages. At opening, the menu only displays predicted items, and, after a delay of 170ms, non-predicted items appear. This menu maintains spatial stability, thus helping the end user to maintain a mental model of the menu. Transposing temporal menu to smartphone is not obvious because all items cannot be displayed on a single screen. Any predicted item located on the subsequent screens requires a cognitive effort to explore the whole set of items. All menu types relying on animation of predicted items fall in this category:

- *Ephemeral menus* [34] are adaptive temporal menus where the gradual onset was used in order to display non-predicted items. Upon opening the menu, user finds the predicted items and, after a delay of 500ms, remaining items appear gradually. This approach suffers from the same problem as the temporal menu: items cannot be displayed on single screen in the case of a small screen device.
- *In-Context Disappearing menus* [20] open the traditional menu with a superimposed prediction window prompting three predicted items that disappears gradually.
- *Out-of-Context Disappearing menus* [20], at opening, display the prediction window immediately; after a delay of 500 msec, the complete menu is gradually displayed from the back, replacing the prediction window.
- *Evanescent menus* [21] superimposes the prediction window to the static menu and then progressively makes it transparent to reveal the menu, thus enabling the user to select a predicted item if it belongs to the prediction window and the static menu after. The main difference with ephemeral menus is that a prediction window is subject to animation as opposed to the static menu. The difference with ICD menus is that a continuous morphing between the prediction window and the menu is performed.

In principle, any computer-based 2/3D animation technique can produce several variations of motion-changing menus. The impact of the animation technique, which ranges from subtle transitions (e.g., fade, push, wipe, split, cover, uncover) to more striking ones (e.g., zoom, switch, flip), should be chosen depending on the criticality of predicted items [30].

4 PREFERENCE ANALYSIS

To get a first idea of how end users perceive graphical adaptive menus, a preference analysis was conducted to determine their preference. Preference is often compared to performance [64]: end users sometimes prefer user interfaces that are not necessarily performant and they also become efficient with user interfaces they are not satisfied with. In the previous section, 49 types of graphical adaptive menus were identified. Unlike other research experiments, we did not choose to present one menu at a time and asking each participant to evaluate it separately: such a repetitive task would lead to participant's fatigue. Instead, we chose to conduct a preference analysis with a comparison method. This approach benefits from several advantages: participants do not need to remember the assessment given to previous menus; this avoids inconsistent and conflicting results; the assessment is simplified by avoiding each participant to evaluate different measures for each menu; instead of evaluating qualitative measures on a Likert scale and/or quantitative measures for one menu at a time, the participants are presented with a pair of two candidate menus to be compared. Preference analysis can be supported using any paired comparison model.

4.1 Method

4.1.1 Apparatus. We developed and installed an on-line A/B testing survey application where two graphical adaptive menus were randomly selected from the pool of 49 menus and presented individually to the user so as to select by clicking on the most preferred one from a visual viewpoint. For every positive selection, one point is given to the global score of the menu. If the participant is undecided, there is the possibility to click on the "Tt is a draw" push button, no point is assigned and two new menus are further presented. Since the pool comprises 49 graphical adaptive menus, a complete comparison would generate $49^2 = 2,401$ menu pairs to be compared per participant, which is of course prohibitive. Therefore, for each participant, the system randomly generates 50 menu pairs. In order to prevent responses biases, the menus are also displayed in a random order and each menu pair is verified as being unique per participant to avoid any duplicate.

4.1.2 Stimuli. For each of the 49 graphical adaptive menus in the pool, a high resolution GIF image was created depicting the menu's behavior based on Figs. 1 to 3 and 6 to 9. For each menu involving some visual effect such as an animation, an animated GIF file was produced respecting the guidelines issued, such as 170 msec for the temporal menu [53], 500 msec for the ephemeral menu [34]. For new menus, similar timing was chosen and the animation was repeated in a loop. All these GIF files were uploaded in the A/B testing application along with a short textual description displayable on-demand.

4.1.3 Procedure. The preference analysis requested each respondent to compare 50 randomly generated pairs of menus and select the most preferred. No time constraint was imposed to participants. Each experiment takes approximately fifteen minutes.

4.1.4 Participants. Participants were recruited from a mailing list maintained at Université catholique de Louvain. No compensation was offered to volunteers. The experiment took place remotely via our A/B testing system setup with the fifty menu pairs. Before starting the experiment, the participants were asked to provide some personal information for statistics such as year of birth, gender. Eighty-seven participants conducted the experiment from nine different countries speaking five different languages (i.e., English, French, German, Dutch, and Spanish). Ten participants were removed as outliers for different reasons: the data were incomplete, the experiment was not finished, the data captured from the participant are strange (e.g., "It is a draw" is selected repeatedly, the left vs. right choice is captured repeatedly with a small timeout between, which may reflect some boredom of a user clicking on the same entry all the time). Finally, the amount of data generated by this experiment is 81 participants × 50 menu pairs = 4,350 samples.

4.1.5 Measures. Based on the point allocation, the system computes two measures: the *number of viewings* (the total amount of times that this menu has been included in a comparison) and the *preference percentage* (the ratio between the number of times the menu has been marked as preferred and the number of viewing). Second, we draw a symmetric matrix of results associating for each pair of menu a and b a score between +4 and -4, where a score of +4 would mean that menu a is preferred to menu b by all respondents, whereas a score of -4 would mean that menu b is preferred to menu a all the time. We also used the raw data obtained through the experiment in order to compute a latent score of preference for each menu. For this purpose, we used the Bradley-Terry-Luce (BTL) model to establish a ranking attached with a score for each menu giving an order of magnitude to the same ranking. The raw data feed a preference vector for each menu containing the matches (+1 if preferred, -1 if unpreferred, and 0 otherwise) and then applied the BTL method summing the probabilities for one menu to be preferred to all others. This sum of probabilities can be seen as a latent score for the menu. The Bradley-Terry model [22] is a probability model that can predict the outcome of a comparison. Given a pair of individuals i and j drawn from some population, it estimates the probability that the pairwise comparison $i > j$ turns out true, as [1] $P(i > j) = \frac{p_i}{p_i + p_j}$, where p_i is a positive real-valued score assigned to individual i. The comparison $i > j$ can be read as "i is preferred to j", "i ranks higher than j", or "i beats j", depending on the application. It is particularly used when participants have to assess and compare many stimuli [29].

4.2 Results and Discussion

Table 4, continued in Table 5, reproduces the two measures for all 49 graphical adaptive menus sorted in decreasing order of preference percentage. The symmetric 49 ×49 matrix and the table with the ranking of each menu with its computed score are provided as supplemental material. Tables 4 and 5 reveal several observations:

- The fourteen most preferred graphical adaptive menus all preserve spatial stability, being both position-invariant and orientation-invariant. The frequency-based menu (#15) is the first one appearing in the list with position variance, if we exclude the split with replication menu (#8), which somewhat preserves the same stability. Many other menus with spatial stability are also in the top list. This indicates a strong preference for spatial stability.
- These fourteen first menus mostly play with value-changing capabilities. Out of these fourteen menus, the rating menu (#4) and the pushpin menu (#6) are the only two instances introducing a shape-changing (with the rating bar and the pushpin, respectively) while the fish-eye (#9) and the morphing menu (#11) are the only one with size-changing after them. This reveals a preference for menus which also preserve physical stability, with shape and size after.
- Surprisingly, the patined menu (#14) is the first occurrence of a texture-changing menu, long before any other of the same category (e.g., the weared menu appears at the #38 place).
- In the category of motion-changing menus, the smart menu (#18) is well placed contrarily to previously expressed criticism [46]. But this menu is mostly a two-state menu and is the first one below the 50% barrier. Among all genuine motion-changing menus, the ephemeral menu is the great winner (#22), followed by the glowing menu (#25). Other members of this category come long after: temporal menus (#35), ICD (#37), evanescent menus (#42), and OCD (#43) with low preference percentages.
- Color-changing menus are not well appreciated. The first instance is the prediction-by-line menu (#19), probably because the color is supplemented by the bar length indicating the prediction. In other menus, the color coding scheme is not favored not because people cannot differentiate colors (studies show that the human being is capable of distinguishing up to 9 colors without any trouble), but because they cannot easily associate the color to a value, even in the prediction-by-rainbow menu (#27) or in the heatmap menu (#34). These color coding schemes seem to be more appropriate for visualization (e.g., to show how frequent items are globally) than for selecting items.
- The cloud menu (#32) is the first instance of a graphical adaptive menu with a separate prediction window, as opposed to a close one, as in split with replication (#8), which is related to the observation that people prefer a prediction close to their locus of interest [82].
- Menus with unusual shapes are ranked low, such as the square menu (#41), the leaf menu (#44), and the hyperbolic menu (#45). This suggests that shape-changing menus as a first attempt to convey adaptivity is not very much appreciated.
- The rotating menu (#49) is the least appreciated menu since items are rotating, thus preventing them from being readable.

[1] https://en.wikipedia.org/wiki/Bradley-Terry_model

Ranking	Graphical adaptive menu	Number of viewings	Preference %
1	Greyscaling menu	122	74%
2	Transparency menu	160	71%
3	Highlighting menu	141	70%
4	Rating menu	158	67%
5	Underlying menu	148	66%
6	Pushpin menu	141	60%
7	Boxing menu	107	59%
8	Split menu with replication	120	57%
9	Fish-eye menu	138	57%
10	Bolding menu	119	56%
11	Morphing menu	148	55%
12	Prediction-by-bar menu	138	54%
13	Blurring menu	138	51%
14	Patined menu	114	51%
15	Frequency-based menu	120	50%
16	Font-changing menu	140	50%
17	Probability-based menu	139	50%
18	Smart menu	135	45%
19	Prediction-by-line menu	121	45%
20	Split menu without replication	128	44%
21	Bubbling menu	133	43%
22	Ephemeral menu	124	42%
23	Time-based menu	135	41%
24	Prediction-by-scale	112	41%
25	Split menu with scroll bar	140	40%
26	Glowing menu	146	40%
27	Prediction-by-rainbow menu	139	39%
28	Fish-eye colored menu	139	39%
29	Blinking menu	135	38%
30	Pulsing menu	139	35%
31	Square menu	126	35%
32	Cloud menu	127	35%
33	Italicizing menu	142	34%
34	Heatmap menu	123	33%
35	Temporal menu	157	32%
36	Twisting menu	120	32%
37	In-context disappearing menu	143	31%
38	Weared menu	150	30%
39	Split menu with arrow bar	141	30%
40	Step-by-step menu	132	30%

Table 4. Results of the preference analysis: menus in decreasing order of preference (1/2) percentage.

Ranking	Graphical adaptive menu	Number of viewings	Preference %
41	Prediction-by-color menu	145	28%
42	Evanescent menu	139	27%
43	Out-context appearing menu	132	27%
44	Leaf menu	140	25%
45	Hyperbolic menu	133	23%
46	Polymodal menu	111	23%
47	Prediction-by-rainbow menu	137	20%
48	Flower menu	118	15%
49	Rotating menu	134	10%

Table 5. Results of the preference analysis: menus in decreasing order of preference (2/2) percentage.

Fig. 10. The various pull-down menus used in Findlater et al.'s [34] experiment.

5 USER STUDIES

To systematically explore the design space induced by the eight variables, the menu used by Findlater et al. [34] for their experiment on ephemeral adaptation will be considered as a reference menu (Fig. 10): each pull-down menu contains 4 unrelated groups of 4 related items (i.e., England, France, Germany, Spain – Venus, Mercury, Jupiter, Saturn – Cabernet, Chardonnay, Merlot, Shiraz, – Almond, Pecan, Pistachio, Walnut); the prediction was defined with probabilities such as Venus=80%, Spain, Shiraz=70%, Pecan, Cabernet, Pistachio=60%; all other items having the same normal probability. Although this menu was originally tested on normal screens, we reused their 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 each menu item do not preclude any prior knowledge, and they are all understandable by a normal person.

5.1 Exploratory Study

Since the design space induced by the eight visual variables is wide and deep, an exploratory study was conducted to determine the preference of end users for some variables. To this end, a series of graphical adaptive menus were prototyped in Adobe Flash V27.0 changing one visual variable at a time. This series was presented to gather end users' informal feedback about their preference (Fig. 11): *size* (1D vertical vs. 1D horizontal vs. 2D), *shape* (rectangle, circle, oval), *value* (highlighting of current item, zooming in, zooming in with rotation in case of a vertical label), *color*, *font size*, *orientation* by changing label angle (horizontal, vertical, angular), *texture* by changing font family (regular, bold, italic, or combined), and *motion* by animation (without vs. with animation of non-horizontal items). Note that some prototypes combine more than one visual change.

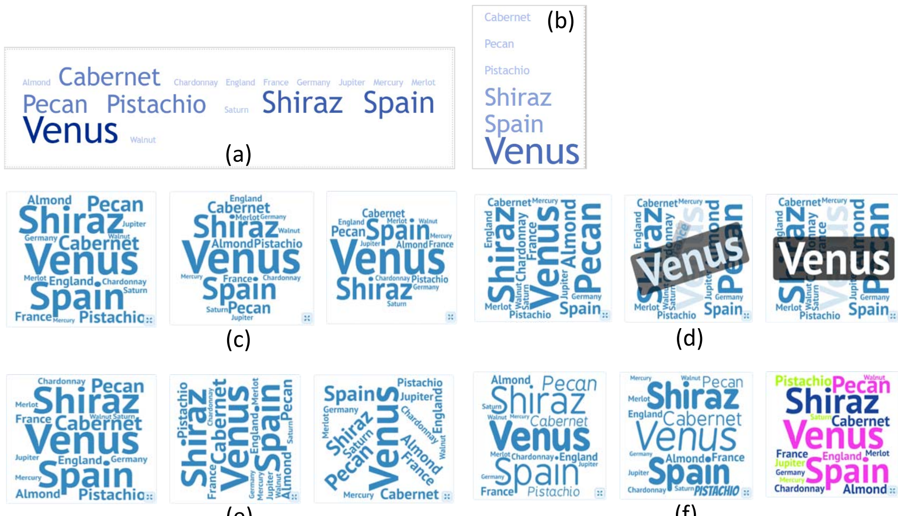

Fig. 11. Menu prototypes based on Findlater's menu with visual variable changing: size (a,b), shape (c), value (d), orientation (e), and texture (f).

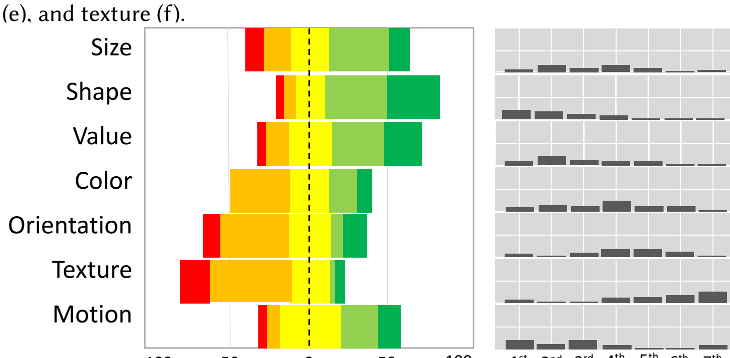

Fig. 12. Participants' rating and ranking for each visual variable.

5.1.1 Method. Each participant performed the task in a controlled environment. Prior to the task each participant was welcomed, had the process explained to them, signed a consent form, and filled in a questionnaire on their background. After the questionnaire was completed, the experimenter demonstrated the initial prototypes on screen. The participants were given 5 min. to familiarize themselves with the prototypes and ask any question. The participant could finish this part early. The participants were then given 15 min. to browse the series of prototypes. During this experiencing time, the experimenter sat next to the participant and observed them. In the end, each participant rated each family of prototypes (one family per visual variable) using a five point Likert scale [56] (1= strongly disagree through to 5=strongly agree) and ranked the families in decreasing order of preference.

5.1.2 Analysis. After each participant, the questionnaire, ratings and ranking data was entered into a spreadsheet in an anonymous format so the participants could not be identified.

5.1.3 Results and Discussion. A total of thirty participants (M = 32.3 years, SD = 6.2 years, 12 female and 18 male) participated in this experiment. All participants were regular computer users and recruited in our organizations through a mailing list. They have different backgrounds such as: accounting, finance, information systems, management, marketing, and human resources. They were all volunteers: they were not given any remuneration (financial or otherwise). Fig. 12

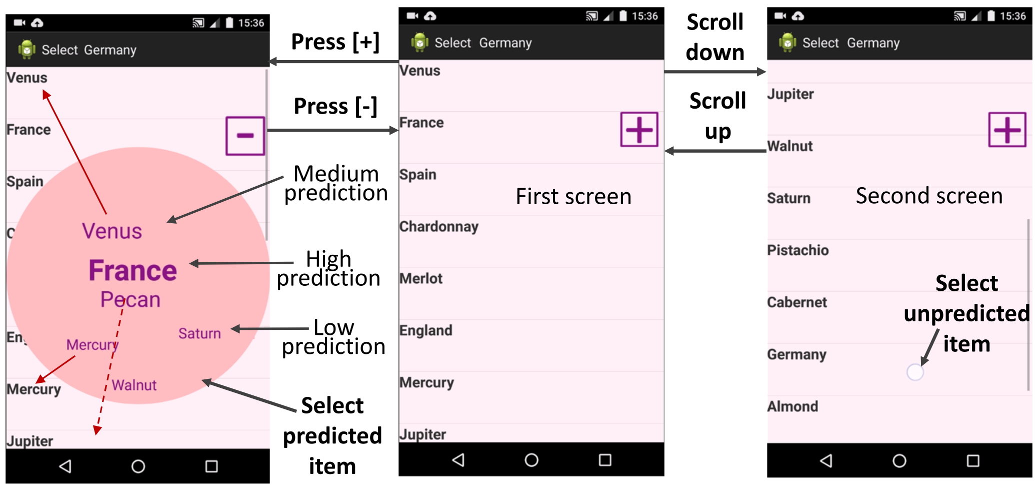

Fig. 13. The Cloud Menu: a circular adaptive cloud contains predicted items that can be made (in)visible by pressing [+], [-].

graphically depicts the correspondence between the distribution of visual variables in terms of preference (represented as a divergent horizontal stacked bar in the left part of Fig. 12 with this coding: red=strongly disagree, orange=disagree, yellow= neutral, light green=agree, dark green= strongly agree, the yellow neutral part is divided into two equal parts) and the distribution of ranking (represented as vertical bars from the fist place to the last place). The preferred visual variables are respectively: shape (82% of the 30 participants), value (71%), size (66%), motion (61%), color (46%), orientation (40%), and texture (22%). Shape was also ranked first followed by motion. Value was ranked second the most frequently, followed by shape. Motion was ranked third the most frequently, followed by shape. With seventh rank most frequently assigned, texture was judged the least preferred variable for conveying adaptivity because of its questionable legibility. In conclusion, shape was selected as the visual variable for further investigation in the next controlled experiment.

5.2 Controlled Experiment

The final design of the cloud menu resulted into a widget in Java for Eclipse based on Android Software Development Kit. The cloud menu consists of a linear list for the static menu superimposed by a prediction window materialized as a circular word cloud with three prediction levels (Fig. 13): (1) any item with high prediction (probability > 80%) is located in the center of the circle highlighted with large font size; (2) any item with medium prediction (60% > probability > 80%) is presented in the periphery with a decreasing size font and a larger distance from the center depending on the probability (the lower the prediction, the more far and the smaller the item becomes) ; (3) any item with low prediction (probability < 60%) is displayed only in the static menu.

The Cloud Menu for Findlater test (Fig. 10) is depicted in Fig. 13: the France item with high prediction is presented in the center with the largest font size; Venus and Pecan with medium prediction are presented afterwards on a position indicating the original position of item in the static menu; Mercury, Walnut and Saturn with low prediction are located in the periphery with an increasing distance from the center and a decreasing size font. Note that the Pecan item is located on an imaginary line indicating an off-screen location. When prediction is correct, the user selects the item directly from the Cloud Menu. When prediction is incorrect, the user makes the

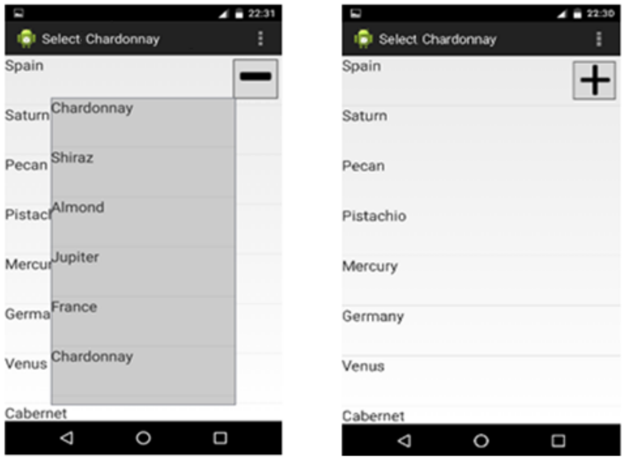

Fig. 14. The Linear Menu: a linear list with six predicted items in the prediction window.

cloud menu disappearing by pressing the [-] button and selects the item from the static menu after scrolling down/up.

A controlled experiment was conducted based on two conditions: a Control condition (or baseline), which presents the Findlater test in a static menu without any adaptation and prediction and a Cloud Menu, which presents the Findlater test as a Cloud Menu with 6 items in the prediction window. To test the influence of the circular layout of the Cloud Menu, a third condition was developed: the Linear Menu, where both prediction window and full list of items are presented as superimposed linear lists (Fig. 14), but without any cloud. To move backward and forward from/to the prediction window, the two [-] and [+] push buttons are also added. Regarding screen real estate, menu items in the prediction window occupy the same height than normal items in the linear menu, but thinner to avoid complete occlusion and to let underlying items be partially visible.

5.2.1 Hypotheses. We made the following assumptions:

Speed with high prediction

H_{11} = *The Cloud Menu and the Linear Menu will be faster than Control condition.* When prediction is correct, the user finds the target item among 6 predicted items in the Cloud Menu and in the Linear Menu more quickly than in Control condition where the target belongs to a list of $4 \times 4 = 16$ items.

H_{21} = *The Cloud Menu will be faster than the Linear Menu.* The circular layout of the 6 predicted items with different font sizes and positions is faster than a linear list without any visual distinction between predicted items.

H_{31} = *A target item located in the center of the Cloud Menu will be selected faster than when located in its periphery.* Indeed, a target item located in the center is considered easy to find because it is emphasized with a large font size (thus inducing a larger activation area as in AAMU [75]), contrarily to a target item located in another part of the cloud menu.

Speed with low prediction

H_{41} = *The Cloud Menu and the Linear Menu will not be worse than Control condition.* Cloud and Linear adaptive menus will not be worse than the static menu as they avoid penalizing interaction when prediction is incorrect. The button should enable the end user to escape from the Cloud Menu and Linear menu in case of an incorrect prediction.

H_{51} = *The Cloud Menus will be faster than the Linear Menu.* Exploration of predicted items will be easier in the Cloud Menu, where items are emphasized with different font sizes and positions, than in the Linear Menu.

H_{61} = *In all conditions (Cloud Menu, Linear menu and Control condition), the target item will be selected faster when located on the first interaction screen than on the second.* Accessing a target item on the first screen will be faster and easier than on the second screen requiring vertical scrolling.
Error rate with high prediction
H_{71} = *The error rate of the Cloud Menu is similar to the Linear Menu.* When the target item is one of the 6 predicted items in Cloud and Linear Menu, the visual search time and selection time are reduced, which also reduces error rate.
H_{81} = *When target item is located in the center of the cloud, errors will be less frequent than when target is in its periphery.* User attention will be attracted to the center where an item is displayed with a larger font size compared to items in other parts of the cloud, thus facilitating selection.
Error rate with low prediction
H_{91} = *No significant difference between all conditions.* When prediction is incorrect in both adaptive conditions (Cloud and Linear), the prediction window will not be used and will disappear, thus generating the same error rate.

5.2.2 Method. Our study was between-subjects with three independent variables:

(1) The Menu Type, a nominal variable with three conditions, representing the the baseline with and without adaptivity and the cloud menu subject to testing: Control Condition (C), Cloud Menu (CM), and Linear Menu (L).
(2) The Target Location, nominal variable with conditions, representing the location of the menu item to be selected: in the cloud (C), in the first screen (F), or in the second part of the screen (S). The first screen contains the observable portion of items while the second screen is browsable by navigation, such as scrolling, panning.
(3) The Level of Prediction, ordinal variable with two conditions, i.e., low and high, which were decided to test the performance when prediction algorithm works well or not.

Each menu is divided into two screens of 8 items, all coming from Findlater's test [34], in its periphery, or outside in case of incorrect prediction. Accurate prediction level is when prediction is correct and target item is inside the Cloud Menu without any restriction. Inaccurate prediction level is when prediction is incorrect and target item is outside the Cloud Menu. The same behavior occurs in the Linear Menu where the target item was also controlled randomly between two prediction accuracy levels. High prediction level is when prediction is correct and target is one of the six predicted items present-ed on the prediction window. Low prediction level is when prediction is wrong and target is outside the prediction window. For both high and low prediction levels, the target item always appears in the static menu.

5.2.3 Task. A between-subjects design was decided to avoid any carryover effects such as practice (learning effect) and fatigue: two independent groups of participants were asked to perform a sequence of item selections. Participants of the first group tested Cloud Menu and Control condition, while the second group tested Linear Menu and Control Condition. Participants were divided into two groups using matched-group design, through which the subjects were matched according to their age and then allocated into group. For each condition, first, a message appeared indicating the target item to select. Then a list of items appeared and the target item was displayed at the screen top as a reminder.

In Cloud Menu, participants selected the target item from the Cloud of predicted items and/or from the static menu. In Linear Menu, participants selected the target item from the prediction window and/or full list of items.

In Control condition, participants selected target items from the main list on the first or on the next screen. When the selected item matched the requested target item, a new message appeared indicating next target item until the test was complete. When the selected item did not match the requested target item, an error message prompted the participant to select the right target before moving to the next target. When the test was complete, a "thank you" message was displayed.

Selections were performed by finger touch on the touchable surface of the smartphone, no stylus or pen were used. Generally, participants were holding the smartphone in their left hand and had to point with right hand (index finger). In each menu, order and position of items were controlled and changed randomly after ten selections in order to avoid any learning effect. Selection sequence (target selection) was also randomly controlled. Target position on first screen or on second screen and prediction accuracy level were also controlled. Each participant had to execute 20 item selections in the Control condition, 20 item selections in the Cloud+High prediction condition and 20 item selections in the Cloud + Low prediction condition, 20 item selections in the Linear +High prediction condition and 20 items selections in the Linear+Low prediction condition. Selections in the Cloud and Linear were mixed to avoid any learning effect induced by a repetitive usage.

5.2.4 Quantitative and qualitative measures. The dependent variables measured were:

(1) The menu item selection time (in seconds), which was measured as the time taken from opening the menu until final selection of requested target.
(2) The error rate (in percentage %), which was measured as the ratio of successfully achieved selections by the total amount of selections.

5.2.5 Apparatus. Android-based Google Nexus smartphones were used, with 2 Gb LPDDR3 RAM, 16 Gb of storage and a 1920 x 1080 pixel screen resolution (423 ppi). The cloud diameter is equal to the screen width. Hence, the cloud menu surface in this case is: $S = \pi \times (l/2)^2 = 916,088$ pixel^2 where l denotes the screen length in pixels. The unused surface is: $F = (l \times h) - S = 2,073,600$ pixel^2 $- 916,088$ pixel^2 $= 1,157,512$ pixel^2, which represents a portion of 56%.

5.2.6 Participants. Two independent groups of nineteen subjects each participated in this experiment. All participants were regular smartphone users and they were recruited in our organization through a mailing list. These thirty-eight subjects are all different from those who participated in the exploratory study to avoid any carry-over effect. They were randomly selected from a list of volunteers belonging to our organization, working in non-computer areas. No compensation was offered.

5.2.7 Procedure. Before starting the test, the principle of each condition (C, L, and CM) was explained to participants but prediction levels were not. Each participant trained with a pretest composed of ten targets. A different item list was used in the pretest than the one used in the test. Both groups selected 60 target items as follows:

(1) **Group 1**: 40 target items for Cloud Menu (20 with high prediction and 20 with low prediction) = 20 items when prediction is correct (10 located in the center of the Cloud Menu and 10 when target is in the periphery) + 20 items when prediction is wrong (items are all outside the Cloud Menu: 10 located on the first screen and 10 on the second screen). There were also 20 items for Control condition: 10 items located on the first screen and 10 on the second.
(2) **Group 2**: 40 targets for Linear Menu = 20 items when prediction is correct and 20 items when prediction is wrong (target is outside the prediction window: 10 located on the first screen and 10 on the second screen). Similarly to group 1, there were also 20 items for Control condition: 10 items located on the first screen and 10 on the second screen.

In summary, the between-participant design was as follows: 19 participants $\times$ 2 groups $\times$ 60 targets = 2,280 menu item selections in total.

Row	Group	Menu	Selection time (sec.)			Error rate (%)		
			M	SD	W/Z	M	SD	W/Z
1	G1	Control	3.40	1.07	0.93	.17	.29	0.22
	G2	Control	3.04	1.07		.33	.62	
2	G1	Cloud (P+)	1.76	0.61	3.08***	.95	1.20	1.47
	G2	Linear (P+)	4.12	3.08		1.93	2.12	
3	G1	Cloud (P+)	1.76	0.61	4.11***	.95	1.20	3.04**
	G1	Control	3.40	1.07		.17	.29	
4	G1	Cloud (P-)	5.60	1.65	3.08***	.74	1.74	1.47
	G2	Linear (P-)	3.40	2.84		2.66	2.79	
5	G1	Cloud (P-)	5.60	1.65	4.17***	.74	1.74	1.22
	G1	Control	3.40	1.07		.17	.29	
6	G2	Linear (P+)	4.12	3.08	2.04*	1.20	2.11	2.54*
	G2	Control	3.04	1.07		.33	.62	
7	G2	Linear (P-)	3.40	2.84	0.2	2.66	2.79	3.54**
	G2	Control	3.04	1.07		.33	.62	

Table 6. Experiment results: $P+$ =high prediction, $P-$ =low prediction, No significance= $p > .05$, $^{*}= p \leq .05$, $^{**}= p \leq .01$, $^{***}= p \leq .001$.

5.2.8 Results and Discussion. Levene's test [55] was applied to verify homogeneity of variance between the two independent samples. Since this later was partially determined, non-parametric Mann-Whitney Comparison test was used for analysis between independent samples (groups), and Wilcoxon Signed Ranks test was applied in the case of within-subjects conditions (Table 6). Data were submitted to a Bonferroni Type I correction before handling.

Selection time. Selection time for all conditions is reported in the third column of Table 6 and graphically depicted in Fig. 15 with a 95% confidence interval for difference between normal means ($\alpha = .05$). First of all, according to row 1 in 6, there is no significant difference ($Z = .93$, $p = .35$) in Control condition between group G1 who tested the cloud menu ($M = 3.40$, $SD = 1.07$) and group G2 ($M = 3.04$, $SD = 1.07$), which allows us to properly compare these two independent groups. According to row 3, Cloud condition with high prediction ($P+ : M = 1.76$, $SD = 0.61$) is significantly faster ($W_{22} = 4.11$, $p = .00004$) than Control condition ($M = 3.40$, $SD = 1.07$). Similarly, Cloud condition with high prediction ($P+ : M = 1.76$, $SD = .61$) is significantly faster ($Z = 3.08$, $p = .002$) than Linear condition in both cases as indicated in row 2: when target is in the center of the Cloud ($M = 1.19$, $SD = .54$) and when it is in the periphery ($M = 2.35$, $SD = 1.04$). Interestingly, when target is located in the center of the Cloud, participants are also significantly faster ($W_{23} = 3.71$, $p = .0002$) than when it is located in the periphery. Usually, corner locations are faster to reach.

Row 6 suggests that Control condition ($M = 3.04$, $SD = 1.07$) is faster ($W_{14} = 2.04$, $p = .05$) than Linear condition when prediction is high ($P+ : M = 4.12$, $SD = 3.08$). In addition in row 4, when prediction is low, users are significantly faster ($Z = 3.08$, $p = .002$) in Linear condition ($P- : M = 3.40$, $SD = 2.84$) than in Cloud condition ($P- : M = 5.60$, $SD = 1.65$). In row 5, users are also significantly faster ($W_{22} = 4.17$, $p = .0003$) in Control condition ($M = 3.40$, $SD = 1.07$) than in Cloud condition with low prediction ($P- : M = 5.60$, $SD = 1.65$). More detailed results further suggest that:

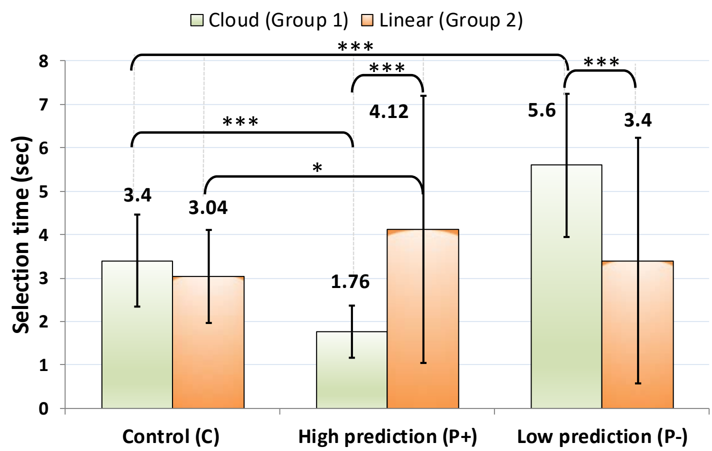

Fig. 15. Selection time for all conditions (normal mean, error bars with a confidence interval of α = .05)

- Cloud with low prediction and target on first screen of the main list ($P-$: M = 4.98, SD = 1.76) is significantly faster (W_{22} = 4.14, p = .00003) than Control with target on first screen (M = 2.42, SD = .83).
- Cloud with low prediction and target on second screen of the main list (M = 6.24, SD = 1.63) is significantly faster (W_{22} = 3.56, p = .0003) than Control with target on second screen (M = 4.40, SD = 1.78).
- In Control condition, users are also significantly faster (W_{22} = 4.14, p = .00003) when the target is on the first screen (M = 2.42, SD = .83) than when it is located on the second screen (M = 4.40, SD = 1.78).

However, row 7 reveals that no significant difference (W_{14} = .20, p = .84) can be detected between Control condition (M = 3.04, SD = 1.07) and Linear condition when prediction is low ($P-$: M = 3.40, SD = 2.84). More detailed results further suggest the following absences of significance:

(1) No significance (W_{14} = .91, p = .36) between Linear with low prediction and target on first screen of the main list ($P-$: M = 3.55, SD = 3.42), Control with target on first screen (M = 2.11, SD = .53).
(2) No significance (W_{14} = 1.14, p = .25) between Linear with low prediction and target on second screen of the main list (M = 3.25, SD = 2.35) and Control with target on second screen (M = 3.97, SD = .86).
(3) However, users belonging to the Control condition are significantly faster (W_{14} = 3.40, p = .0006) when the target is in the first screen (M = 2.11, SD = .53) than when it is located on the second screen (M = 3.97, SD = .86), which is of course normal since any navigation to the second screen will inevitably increase the selection time.

Fig. 16 plots the average selection time of the Cloud menu as a function of the displacement [65], calculated as the distance in pixels between the central point and the item selected with interpolation. The dashed black line represents a trend line for the selection time, which seems comparable to small radial menus (SPARSE in [71]) represented by a red dotted line in Fig. 16.

***H_{11}* is supported**. When prediction is high, users find the target item among 6 predicted items in the Cloud Menu and the Linear Menu faster than in Control condition when the target belongs to a list of 16 items, which somewhat normal since item selection is operated with a smaller amount of time included in a smaller surface, which is in line with [28].

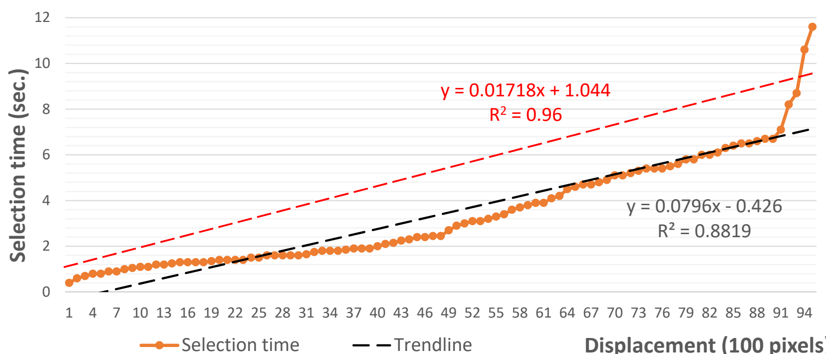

Fig. 16. Selection time as a function of displacement.

***H_{21}* is supported**. When prediction is high, users are faster in Cloud menu than in Linear condition. This result can be justified by the fact that circular form of the Cloud facilitates exploration of a number of predicted items assigned to 6, which is higher than the typical 3 items found in other adaptive split menus. The user attention is distributed to different sides of the Cloud in contrary to the Linear condition when the user has to browse all items one by one. Consequently, by transitivity, Cloud Menu is faster than Linear menu, which is in turn faster than Control condition, which is the most important conclusion that demonstrates the effectiveness and efficiency of Cloud Menus.
***H_{31}* is supported**. This is justified by the fact that user attention is often attracted by the item in the center of the Cloud Menu because it is consistently located and it is always emphasized by a larger font size compared to other predicted items in the Cloud. Therefore, the central item is always the most preeminent among all menu items and its position is very predictable, even it requires some scanning. Suggested conclusions are:

(1) Prediction is crucial, reduces navigation time, search time especially when menus are long.
(2) Increasing the amount of predicted items is not penalizing the interaction because it is still better than the non-adaptive menu provided that this amount is not prohibitive. This requires another study to determine the threshold for this positive effect.
(3) The prediction window as a circular word cloud may be an important factor making the Cloud Menu efficient. It is better than the Linear Menu because of its impact on user perception and usability of word clouds [42, 57].
(4) Linear menu is competitive to scan items (one eye fixation, one column), but becomes surpassed by a cloud menu (many eye fixations, multiple directions) when the amount of items increases the length of the split area.

***H_{41}* and *H_{51}* are partially supported**. Results showed that, overall when prediction is low, Control condition is faster than Cloud Menu and no significant difference is found between the Linear and Control conditions. A cloud menu subsumes two costs: the visual search time of an item and the checking time of an item. When prediction is high, the predicted item appears in the prediction window, its visual searching time is limited by the amount of items and the surface and its checking time is null. But when the prediction is low, the predicted item does not appear in the prediction window. The end user then starts by scanning the area several times to ensure that the target item is really not there: the checking time becomes superior to the visual searching time. Conversely in a Linear menu, the checking time remains equal to zero because the end user ensures that all items are processed when scanning items vertically. Conveying six predicted items increases the probability to present the target item to the end user with respect to three predicted items only.
***H_{61}* is supported**. Users can have direct access to target when this latter is on first screen contrary

to when it is on the second screen, in this case scrolling is required that can justify the fact that selection time is shorter on first screen than on second screen.

Error Rate. Error rate for all conditions is reported in the fourth large column of Table 6. Error rates are assessed as equivalent ($Z = .22$, $p = .82$) in Control condition both in G1 ($M = .17$, $SD = .29$) and G2 ($M = .33$, $SD = .62$) as reported in row 1 of Table 6. Overall, there is no significant difference observed ($Z = 1.47$, $p = .14$) in terms of errors between the Cloud condition ($M = .95$, $SD = 1.20$) and Linear condition ($M = 1.93$, $SD = 2.12$) as indicated in row 2. In Cloud Menu, when prediction is high and item target is located in the center of the Cloud ($P+ : M = .04$, $SD = .21$), errors are less frequent ($W_{22} = 3.44$, $p = .0005$) than when the target is located in the periphery ($M = 1.87$, $SD = 2.40$). In row 3, we observe that errors are significantly less frequent ($W_{22} = 3.04$, $p = .002$) in Control condition ($M = .17$, $SD = .29$) than in Cloud Menu ($M = .96$, $SD = 1.20$), which may be due to tally errors in the periphery of the Cloud. When prediction is low in Cloud Menu ($P- : M = .74$, $SD = 1.74$), there is no significant difference ($W_{22} = 1.22$, $p = .22$) with respect to the Control condition ($M = .17$, $SD = .29$) as reported in row 5. Similarly, errors are less frequent ($W_{14} = 2.54$, $p = .01$) in Control condition ($M = .33$, $SD = .62$) than in Linear condition ($M = 1.20$, $SD = 2.11$) as reported in row 6 when prediction is high. When prediction is low in row 7, it is even more significant: errors are less frequent ($W_{14} = 3.54$, $p = .007$) in Control Condition than in Linear case with low prediction.

H_{71} is supported. When target is one of the 6 predicted items in Cloud and Linear Menu, the visual search time and selection time are reduced which reduce errors number.

H_{81} is supported. In the periphery, some items can be close together, which suggests that a large number of tally errors can be generated. Contrarily to a target located in the center, locations, directions, and font size are factors minimizing tally errors. Items predicted inside the cloud can be laid out in a more optimized way by calculating the distance between items and the center, and distributing them, e.g., by considering their semantic relation. Further investigation is required to improve this aspect, by laying these items out depending on their position in the static menu.

H_{91} is supported. In the case of low prediction, participants did not rely on the Cloud Menu and used the button to make it disappear. The target is in the static menu, like in Control condition, which suggests that there is no difference between Cloud Menu and Control condition.

5.2.9 Experiment Overview. The experiment for the cloud menu corroborates several findings from Lohmann et al. [57]:

- *Item font size*: items with a large font size attract more user attention than with a small font size (an effect influenced by other parameters, such as item length, item position, and item neighboring). According to this study, recognition for items with a larger font size was significantly higher than items with a smaller font size: 83%, 73%, and 59% respectively for the three largest. The item font size is the linear menu remained constant but could also benefit from the same effect: to become a size-changing menu if the selection area changes.
- *Scanning*: cloud menus have been proved as an efficient adaptive split menu for small screens because participants tend to scan menu items rather than reading them, which accelerates their processing time.
- *Centering*: menu items located in the middle of the cloud attract more user attention than tags near the borders (an effect influenced by layout). H_{31} and H_{81} are two supported hypotheses that confirm this finding.
- *Position*: the upper left quadrant receives more user attention than the others, but we did not exploit this effect since menu items are positioned so that they can point to their original position in the static menu, even when they are off-screen.

6 DESIGN GUIDELINES

6.1 Design Guidelines for Graphical Adaptive Menus

By combining results from the preference analyses and the controlled experiment, some design guidelines could be devised on which form of graphical adaptivity could be preferred:

- **(G1) Privilege spatial stability.** When the amount of predicted items is low with respect to displayed items, such as in a ratio of 3/7, the spatial stability should be first property to be respected, which concur with [39]: the most preferred graphical adaptive menus emerging from the analyses all adhere to this property. They should be position-invariant and orientation-invariant, which is also important for gestures [79]. Consequently, adaptivity should be conveyed by format, preferably by changing the value, but not the color or the texture. When the amount of predicted items increases, perhaps with the amount of items in the static menu, it is no longer possible to rely on spatial stability only. The split menu with replication comes as the first preferred option followed by cloud menus afterwards. The first enlarges the menu size while the second does not.
- **(G2) Maintain physical stability if possible.** Among all menus satisfying spatial stability, those also satisfying physical stability should be considered first. Graphical adaptive menus satisfying both properties come before those which are size and/or shape variant. Size-invariance also comes before shape-invariance.
- **(G3) Foster value-changing menus.** Among all properties involved in format stability, the value variance is largely accepted in many forms, mainly because it does not affect spatial and physical stabilities. Color-invariance and texture-invariance should be always preserved: graphical adaptive menus not satisfying these properties were always ranked low and assessed negatively.
- **(G4) Relax temporal stability only if spatial stability is maintained.** Motion-variant menus are very much depreciated. The only cases were they were still appreciated were when spatial stability is maintained. The ephemeral menu [34] is a representative example, while twisting or rotating menus are not.

6.2 Design Guidelines for Cloud Menus

A cloud menu represents an adaptive split menu for which some design guidelines can be devised:

- **(G5) A cloud menu should be used for a substantive static menu.** The main idea behind using a tag cloud in information retrieval is that the relevance of a document must be determined with respect to a set of documents before this document actually appears. Similarly, the main idea behind using a tag cloud as an adaptive split menu is that the probability of selecting a menu item among other items in that menu should appear before the menu is entirely browsed and displayed. It does not make sense to produce an adaptive split menu such as the cloud menu for an initial (static) menu containing only a few items, even on a small screen. In our experiment, the half of the menu is visible: 8 items among 16.
- **(G6) A cloud menu should not hold more than 6 items.** So far, adaptive split menus have been mainly explored for large screens (e.g., laptop, desktop, large monitors, wall screens). Even under these conditions, 3 or 4 items were recommended [32, 34] as the maximum threshold. The results of the experiment conducted suggest that this threshold could be upgraded up to 6, even on a small screen.
- **(G7) A cloud menu should not exceed 3 levels of prediction.** Beyond this threshold, the end user is likely to be no longer able to make any difference between the three levels. This is somewhat consistent with the 3 levels of emphasizing recommended in usability guidelines

[60]. Other coding schemes could augment this representation, but may also increase the cognitive load as opposed to reinforcing the same data.

- **(G8) A cloud menu should be located as close as possible to the static original menu.** When an adaptive split menu is located too far away from its original menu, there is a risk of losing the semantic or physical relationship between the static and the predicted parts. This is consistent with existing recommendation [71, 72] to minimize visual displacement between the various regions.
- **(G9) The items of a cloud menu should be located as much as possible to point to their corresponding (static) items.** When both are on the same display, they should be positioned on the same line (plain red arrow in Fig. 13). When a predicted item refers to an off-screen location, it should also be positioned to indicate this situation (dotted red arrow in Fig. 13). This guideline is applicable to any adaptive split menu, especially for normal screens.
- **(G10) A cloud menu on a small screen can be superimposed.** While large screens can accommodate another (close) location for displaying the prediction window, a small screen is unable to satisfy the same constraint. Thus, a superposition avoids creating another parallel menu like in the traditional split menu, with the risk of oscillating between the two. It also preserves spatial, physical, and format stability (since the static menu is left untouched), but not temporal stability.
- **(G11) A cloud menu should optimize its circular layout.** Since shape was elicited in the focus group as the first variable to manipulate for materializing a cloud, other parameters, like color, texture, animation should be left out. Instead, the circular layout could be optimized based on [71], with only one item per line and only one background and foreground color. We did not play with transparency like alpha blending.

7 CONCLUSION AND FUTURE WORK

In this article, we first reviewed a large set of graphical menus, some of them supporting adaptivity, some other being candidates for expressing some adaptivity depending on how the menu items change based on their prediction. Observing the wide and deep variety of these menus, a systematic exploration of a design space, based on Bertin's eight visual variables, was conducted to classify these menu techniques against four stability properties. The exploration of this design space allowed us to describe any graphical adaptive menu, to compare two or more of them, and to generate unprecedented new ideas. An exploratory study was conducted to determine the end users' preference for the eight visual variables based on menus prototyped based on Findlater's menu. The results of this exploratory study enabled us to focus on shape-changing and value-changing menus, suggested as the most preferred visual styles.

Cloud menus, a particular type of shape-changing menu, was further investigated to determine its impact of effectiveness, efficiency, and satisfaction. The Cloud Menu consists of a linear list for the static menu superimposed by a prediction window materialized as a circular word cloud with three prediction levels (Fig. 13): any item with high prediction ($probability = 80\%$) is located in the center of the cloud highlighted with large font size, any item with medium prediction ($60\% \leq probability \leq 80\%$) is presented in the periphery with a decreasing size font and a larger distance from the center depending on the probability (the lower the prediction, the more far and the smaller the item becomes), and any item with low prediction ($probability \leq 60\%$) is displayed only in the static menu. The empirical study conducted on cloud menus suggests that they reduce item selection time and error rate when prediction is correct without penalizing it when prediction is incorrect, compared to two baselines: a non-adaptive static menu and an adaptive linear menu. These findings reinforce once again the need for an accurate prediction [39]. From this study, a set of design guidelines for graphical adaptive menus and cloud menus are elaborated.

Further work could be elaborated at different levels of investigation.

At the level of the cloud menu. Several variables have been frozen as constant to facilitate the conducting of the controlled experiment. These variables be released to determine their effect on selection time, error rate, and subjective user satisfaction. Regarding the prediction window, the radius of the cloud menu, initially aligned with the screen length, could be reduced to minimize occlusion; the amount of predicted items, between three and six, could be revisited in parallel with the font size to determine when the amount of predicted items becomes too large to be effectively perceived; more or less prediction levels could be tested; other shapes used for tag clouds could be considered than a circle, although considered as the simplest one. The size of menu items is known to determine some efects especially on small screens [2, 80]. Regarding the static part, the amount of menu items per menu, initially set to sixteen (Fig. 10), could increase; the amount of groups of related items, initially set to four, could also vary; the depth of the menu could range from one to three or more, thus enabling a cloud menu to benefit from the step-by-step-menu [17].

At the level of graphical adaptive menus. Other forms of graphical adaptive menus revealed in the preference analyses delivered in this paper suggest further investigation and comparison. These experiments could also involve other measures, such as a menu performance model [28], the Fitt's Law or other models to predict the selection times. These models should cover the motor and cognitive spaces of menu selection and rely on the various steps for adaptation found in adaptation cycles such as in the ISATINE framework [58] or the LPA-PDA cycle [19]. Bertin's classification of visual variables has been revisited and extended by Carpendale [27]: *grain* is a new visual variable where any color or value, except the extremes, can support variations in grain; *pattern* is another variable expressing how repetitive use of shape variations may result into a patten; and a revisited definition of texture, expressed as any material characteristic (e.g., steel, metal, wood). When visual attributes are manipulated in an animation, new variables are analyzed, such as: display rate, order, duration, frequency, rate of change, synchronization. These additional variables constitute a interesting starting point for extending the design space of graphical adaptive menus. They were not considered in this paper because the hypothesis was to stick to Bertin's initial definitions as a starting point.

At the level of multimodal adpative menus. All adaptivity forms discussed in this paper assume that the graphical modality is predominant. Adaptivity could be conveyed by other means than via the visual channel. Even if the vast majority of signals processed by our human brain emanate from our visual channel, other channels could be investigated, such as tactile, touch, haptics, sound [83], and sonification. Relying on channels that do not require any synchronization with the visual channel, e.g., the haptic and the sound channels, avoid overwhelming the visual channel, when a cognitive overload occurs. Alternate modalities are particularly welcome in eye-free or hand-free conditions [67]. For instance, a haptic surface could be employed to let the end user feel the prediction level or mid-air haptics could convey vibro-tactile feedback without touching anything. Investigating these other modalities, such as in Polymodal menus [18], require identifying and defining another set of variables relevant for each modality: e.g., intensity, timbre and register for sound or swiftness for haptics. But so far, apart from the general multimodal interfaces offering modalitiy alternatives to the end user [12, 31], multimodal adaptive menus are in their infancy [13].

ACKNOWLEDGMENTS

The authors would also like to warmly thank the anonymous referees for their valuable comments and helpful suggestions. They also thank all the participants involved in the studies and experiments. We also thank Mathieu Zen for implementing the A/B testing system used in the preference analysis. Jean Vanderdonckt was supported by the Wallonie Bruxelles International (WBI) program.

REFERENCES

[1] David Ahlström, Andy Cockburn, Carl Gutwin, and Pourang Irani. 2010. Why It's Quick to Be Square: Modelling New and Existing Hierarchical Menu Designs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1371–1380. https://doi.org/10.1145/1753326.1753534

[2] Khalid Al-Omar and Dimitrios Rigas. 2009. The Effect of Size of Personalised Menus on User Satisfaction. In *Proceedings of the 11th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering (MMACTEE'09)*. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 322–327. http://dl.acm.org/citation.cfm?id=1949006.1949062

[3] Liat Antwarg, Talia Lavie, Lior Rokach, Bracha Shapira, and Joachim Meyer. 2013. Highlighting items as means of adaptive assistance. *Behaviour & Information Technology* 32, 8 (2013), 761–777. https://doi.org/10.1080/0144929X.2011.650710 arXiv:https://doi.org/10.1080/0144929X.2011.650710

[4] Leena Arhippainen, Tapani Rantakokko, and Marika Tähti. 2005. Navigation with an Adaptive Mobile Map-Application: User Experiences of Gesture- and Context-Sensitiveness. In *Ubiquitous Computing Systems*, Hitomi Murakami, Hideyuki Nakashima, Hideyuki Tokuda, and Michiaki Yasumura (Eds.). Springer, Berlin, Heidelberg, 62–73. https://doi.org/10.1007/11526858_6

[5] Siddhartha Asthana, Pushpendra Singh, and Amarjeet Singh. 2013. Exploring Adverse Effects of Adaptive Voice Menu. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 775–780. https://doi.org/10.1145/2468356.2468494

[6] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *Comput. Surveys* 49, 4, Article 60 (Dec. 2016), 41 pages. https://doi.org/10.1145/3002171

[7] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: Interactive Optimization of Menu Systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 331–342. https://doi.org/10.1145/2501988.2502024

[8] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 169–178. https://doi.org/10.1145/1166253.1166280

[9] Michel Beaudouin-Lafon. 2004. Designing Interaction, Not Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 15–22. https://doi.org/10.1145/989863.989865

[10] Benjamin B. Bederson. 2000. Fisheye Menus. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 217–225. https://doi.org/10.1145/354401.354782

[11] Jacques Bertin. 1967. *Sémiologie graphique*. Mouton/Gauthier-Villars, Paris, France.

[12] Matthias Bezold and Wolfgang Minker. 2010. A Framework for Adapting Interactive Systems to User Behavior. *J. Ambient Intell. Smart Environ.* 2, 4 (Dec. 2010), 369–387. http://dl.acm.org/citation.cfm?id=2021081.2021084

[13] Matthias Bezold and Wolfgan Minker. 2011. *Adaptive Multimodal Interactive Systems*. Springer, Berlin, Germany. https://doi.org/10.1007/978-1-4419-9710-4

[14] François Bodart, Anne-Marie Hennebert, Jean-Marie Leheureux, Isabelle Provot, Jean Vanderdonckt, and Giovanni Zucchinetti. 1996. *Key Activities for a Development Methodology of Interactive Applications*. Springer London, London, 109–134. https://doi.org/10.1007/978-1-4471-1001-9_7

[15] Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2016. Automated Evaluation of Menu by Guidelines Review. In *13th International Conference on Human Computer Interaction, RoCHI 2016, Iasi, Romania, September 8-9, 2016.*, Adrian Iftene and Jean Vanderdonckt (Eds.). Matrix Rom, 11–21. http://rochi.utcluj.ro/proceedings/en/articles-RoCHI2016.php

[16] Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2016. A Comparison of Shortcut and Step-by-Step Adaptive Menus for Smartphones. In *HCI 2016 - Fusion! Proceedings of the 30th International BCS Human Computer Interaction Conference, BCS HCI 2016, Bournemouth University, Poole, UK, 11-15 July 2016 (Workshops in Computing)*, Shamal Faily, Nan Jiang, Huseyin Dogan, and Jacqui Taylor (Eds.). BCS. http://ewic.bcs.org/content/ConWebDoc/56904

[17] Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2016. A Design Space for Engineering Graphical Adaptive Menus. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '16)*. ACM, New York, NY, USA, 239–244. https://doi.org/10.1145/2933242.2935874

[18] Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2017. Polymodal Menus: A Model-based Approach for Designing Multimodal Adaptive Menus for Small Screens. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 15 (June 2017), 19 pages. https://doi.org/10.1145/3099585

[19] Sara Bouzit, Gaëlle Calvary, Joëlle Coutaz, Denis Chêne, Éric Petit, and Jean Vanderdonckt. 2017. The PDA-LPA design space for user interface adaptation. In *11th International Conference on Research Challenges in Information Science, RCIS 2017, Brighton, United Kingdom, May 10-12, 2017*, Saïd Assar, Oscar Pastor, and Haralambos Mouratidis (Eds.). IEEE, 353–364. https://doi.org/10.1109/RCIS.2017.7956559

[20] Sara Bouzit, Denis Chêne, and Gaëlle Calvary. 2014. From Appearing to Disappearing Ephemeral Adaptation for Small Screens. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design (OzCHI '14)*. ACM, New York, NY, USA, 41–48. https://doi.org/10.1145/2686612.2686619

[21] Sara Bouzit, Denis Chêne, and Gaëlle Calvary. 2015. Evanescent Adaptation on Small Screens. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (OzCHI '15)*. ACM, New York, NY, USA, 62–68. https://doi.org/10.1145/2838739.2838749

[22] Ralph Allan Bradley and Milton E. Terry. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* 39, 3/4 (1952), 324–345. http://www.jstor.org/stable/2334029

[23] Robert Bridle and Eric McCreath. 2005. Predictive Menu Selection on a Mobile Phone. In *Proceedings Workshop W7 on mining spatio-temporal data (ECML/PKDD'2005)*. 327–329. https://cs.anu.edu.au/~Eric.McCreath/papers/mining05.pdf

[24] Robert Bridle and Eric McCreath. 2006. Inducing Shortcuts on a Mobile Phone Interface. In *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI '06)*. ACM, New York, NY, USA, 327–329. https://doi.org/10.1145/1111449.1111526

[25] Nicolas Burny, Suzanne Kieffer, Nathan Magrofuoco, Jorge Luis Perez Medina, Paolo Roselli, and Jean Vanderdonckt. 2018. Feedup, Feedback, and Feedforward in Curve Mid-Air 3D Gestures. In *Proceedings of CHI '18 Workshop on Mid-Air Haptics for Control Interfaces*, M. Giordano, O. Georgiou, B. Dzidek, L. Corenthy, Jin Ryong Kim, S. Subramanian, and Stephen Brewster (Eds.). http://hdl.handle.net/2078.1/198679

[26] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (2003), 289–308. https://doi.org/10.1016/S0953-5438(03)00010-9

[27] M. S. T. Carpendale. 2001. *Considering Visual Variables as a Basis for Information Visualisation.* University of Calgary, Faculty of science, Calgary, Canada. https://doi.org/10.5072/PRISM/30495

[28] Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A Predictive Model of Menu Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/1240624.1240723

[29] Ph. Courcoux and M. Semenou. 1997. Preference data analysis using a paired comparison model. *Food Quality and Preference* 8, 5 (1997), 353 – 358. https://doi.org/10.1016/S0950-3293(97)00004-9 Third Sensometrics Meeting.

[30] Charles-Eric Dessart, Vivian Genaro Motti, and Jean Vanderdonckt. 2012. Animated Transitions Between User Interface Views. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 341–348. https://doi.org/10.1145/2254556.2254623

[31] Bruno Dumas, María Solórzano, and Beat Signer. 2013. Design guidelines for adaptive multimodal mobile input solutions. In *15th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '13, Munich, Germany, August 27 - 30, 2013*, Michael Rohs, Albrecht Schmidt, Daniel Ashbrook, and Enrico Rukzio (Eds.). ACM, 285–294. https://doi.org/10.1145/2493190.2493227

[32] Leah Findlater and Joanna McGrenere. 2004. A Comparison of Static, Adaptive, and Adaptable Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 89–96. https://doi.org/10.1145/985692.985704

[33] Leah Findlater and Joanna McGrenere. 2008. Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1247–1256. https://doi.org/10.1145/1357054.1357249

[34] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. 2009. Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1655–1664. https://doi.org/10.1145/1518701.1518956

[35] Sebastian Fischer and Stephan Schwan. 2008. Adaptively shortened pull down menus: location knowledge and selection efficiency. *Behaviour & Information Technology* 27, 5 (2008), 439–444. https://doi.org/10.1080/01449290701497095 arXiv:https://doi.org/10.1080/01449290701497095

[36] Jérémie Francone, Gilles Bailly, Eric Lecolinet, Nadine Mandran, and Laurence Nigay. 2010. Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-free Selection for Expert Mode. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI '10)*. ACM, New York, NY, USA, 173–180. https://doi.org/10.1145/1842993.1843025

[37] Yusuke Fukazawa, Mirai Hara, Masashi Onogi, and Hidetoshi Ueno. 2009. Automatic Mobile Menu Customization Based on User Operation History. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, Article 50, 4 pages. https://doi.org/10.1145/1613858.1613921

[38] Krzysztof Z. Gajos and Krysta Chauncey. 2017. The Influence of Personality Traits and Cognitive Load on the Use of Adaptive User Interfaces. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI '17)*. ACM, New York, NY, USA, 301–306. https://doi.org/10.1145/3025171.3025192

[39] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '06)*. ACM, New York, NY, USA, 201–208. https://doi.org/10.1145/1133265.1133306

[40] Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. 2008. Predictability and Accuracy in Adaptive User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1271–1274. https://doi.org/10.1145/1357054.1357252

[41] Saul Greenberg and Ian H. Witten. 1985. Adaptive personalized interfaces - A question of viability. *Behaviour & Information Technology* 4, 1 (1985), 31–45. https://doi.org/10.1080/01449298508901785 arXiv:https://doi.org/10.1080/01449298508901785

[42] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. 2014. Word Cloud Explorer: Text Analytics Based on Word Clouds. In *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*. IEEE Computer Society, 1833–1842. https://doi.org/10.1109/HICSS.2014.231

[43] Tobias Hesselmann, Stefan Flöring, and Marwin Schmitt. 2009. Stacked Half-Pie Menus: Navigating Nested Menus on Interactive Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, 173–180. https://doi.org/10.1145/1731903.1731936

[44] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceeding of the CHI '99 Conference on Human Factors in Computing Systems: The CHI is the Limit, Pittsburgh, PA, USA, May 15-20, 1999.*, Marian G. Williams and Mark W. Altom (Eds.). ACM, 159–166. https://doi.org/10.1145/302979.303030

[45] 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. http://dl.acm.org/citation.cfm?id=2835658.2835662

[46] Anthony Jameson, Silvia Gabrielli, and Antti Oulasvirta. 2009. Users' Preferences Regarding Intelligent User Interfaces: Differences Among Users and Changes over Time. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. ACM, New York, NY, USA, 497–498. https://doi.org/10.1145/1502650.1502734

[47] Iyad Khaddam, Sara Bouzit, Gaëlle Calvary, and Denis Chêne. 2016. MenuErgo: Computer-aided Design of Menus by Automated Guideline Review. In *Actes de la 28ième ConféRence Francophone sur l'Interaction Homme-Machine (IHM '16)*. ACM, New York, NY, USA, 36–47. https://doi.org/10.1145/3004107.3004130

[48] Iyad Khaddam and Jean Vanderdonckt. 2011. Flippable User Interfaces for Internationalization. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '11)*. ACM, New York, NY, USA, 223–228. https://doi.org/10.1145/1996461.1996524

[49] Kyungdoh Kim, Robert W. Proctor, and Gavriel Salvendy. 2009. Menu Design in Cell Phones: Use of 3D Menus. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part III: Ubiquitous and Intelligent Interaction*. Springer-Verlag, Berlin, Heidelberg, 48–57. https://doi.org/10.1007/978-3-642-02580-8_6

[50] Thomas Kühme, Uwe Malinowski, and James D. Foley. 1993. Facilitating interactive tool selection by adaptive prompting. In *Human-Computer Interaction, INTERACT '93, IFIP TC13 International Conference on Human-Computer Interaction, 24-29 April 1993, Amsterdam, The Netherlands, jointly organised with ACM Conference on Human Aspects in Computing Systems CHI'93, Adjunct Proceedings*, Stacey Ashlund, Kevin Mullet, Austin Henderson, Erik Hollnagel, and Ted N. White (Eds.). ACM, 149–150. https://doi.org/10.1145/259964.260160

[51] John Lamping, Ramana Rao, and Peter Pirolli. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 401–408. https://doi.org/10.1145/223904.223956

[52] Talia Lavie and Joachim Meyer. 2010. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies* 68, 8 (2010), 508 – 524. https://doi.org/10.1016/j.ijhcs.2010.01.004

[53] Dong-Seok Lee and Wan Chul Yoon. 2004. Quantitative results assessing design issues of selection-supportive menus. *International Journal of Industrial Ergonomics* 33, 1 (2004), 41 – 52. https://doi.org/10.1016/j.ergon.2003.07.004

[54] H. Lee, Y. Choi, and Y. Kim. 2011. An adaptive user interface based on Spatiotemporal Structure Learning. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. 923–927. https://doi.org/10.1109/CCNC.2011.5766642

[55] Howard Levene. 1960. Robust tests for equality of variances. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, Ingram Olkin and Harold Hotelling et al. (Eds.). Stanford University Press, Palo Alto, CA, USA, 278–292.

[56] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22, 140 (1932), 55–. http://psycnet.apa.org/record/1933-01885-001

[57] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff. 2009. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. In *Human-Computer Interaction – INTERACT 2009*, Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 392–404. https://doi.org/10.1007/978-3-642-03655-2_43

[58] Víctor López-Jaquero, Jean Vanderdonckt, Francisco Montero Simarro, and Pascual González. 2007. Towards an Extended Model of User Interface Adaptation: The Isatine Framework. In *Engineering Interactive Systems - EIS 2007 Joint Working Conferences, EHCI 2007, DSV-IS 2007, HCSE 2007, Salamanca, Spain, March 22-24, 2007. Selected Papers (Lecture Notes in Computer Science)*, Jan Gulliksen, Morten Borup Harning, Philippe A. Palanque, Gerrit C. van der Veer, and Janet Wesson (Eds.), Vol. 4940. Springer, 374–392. https://doi.org/10.1007/978-3-540-92698-6_23

[59] Xiaoyang Mao, Yuji Hatanaka, Atsumi Imamiya, Yuki Kato, and Kentaro Go. 2000. Visualizing Computational Wear with Physical Wear. In *Proceedings of the 6th ERCIM Workshop "User Interfaces for All" (UI4All '00)*, Pier-Luigi Emiliani and Constantine Stephanidis (Eds.). CNR-IROE, Pisa, Italy, 12. http://ui4all.ics.forth.gr/UI4ALL-2000/files/Long_papers/Mao.pdf

[60] C. Marshall, C. Nelson, and M.M. Gardiner. 1987. Robust tests for equality of variances. In *Applying Cognitive Psychology to User- Interface Design*, Design guidelines (Ed.). Wiley & Sons Ltd, Chichester.

[61] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Patina: Dynamic Heatmaps for Visualizing Application Usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3227–3236. https://doi.org/10.1145/2470654.2466442

[62] Mitsuhiro Matsumoto, Ryozo Kiyohara, Hidenori Fukui, Masayuki Numao, and Satoshi Kurihara. 2008. Proposition of the context-aware interface for cellular phone operations. In *2008 5th International Conference on Networked Sensing Systems*. 233–233. https://doi.org/10.1109/INSS.2008.4610872

[63] J. Mitchell and B. Shneiderman. 1989. Dynamic Versus Static Menus: An Exploratory Comparison. *SIGCHI Bull.* 20, 4 (April 1989), 33–37. https://doi.org/10.1145/67243.67247

[64] Jakob Nielsen and Jonathan Levy. 1994. Measuring Usability: Preference vs. Performance. *Commun. ACM* 37, 4 (April 1994), 66–75. https://doi.org/10.1145/175276.175282

[65] Lauren Norrie and Roderick Murray-Smith. 2016. Investigating UI Displacements in an Adaptive Mobile Homescreen. *Int. J. Mob. Hum. Comput. Interact.* 8, 3 (July 2016), 1–17. https://doi.org/10.4018/IJMHCI.2016070101.oa

[66] Jungchul Park, Sung H. Han, Yong S. Park, and Youngseok Cho. 2007. Usability of Adaptable and Adaptive Menus. In *Usability and Internationalization. HCI and Culture*, Nuray Aykin (Ed.). Springer, Berlin, Heidelberg, 405–411. https://doi.org/10.1007/978-3-540-73287-7_49

[67] Martin Pielot, Anastasia Kazakova, Tobias Hesselmann, Wilko Heuten, and Susanne Boll. 2012. PocketMenu: Non-visual Menus for Touch Screen Devices. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 327–330. https://doi.org/10.1145/2371574.2371624

[68] Antoine Ponsard, Kamyar Ardekani, Kailun Zhang, Frederic Ren, Matei Negulescu, and Joanna McGrenere. 2015. Twist and Pulse: Ephemeral Adaptation to Improve Icon Selection on Smartphones. In *Proceedings of the 41st Graphics Interface Conference (GI '15)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 219–222. http://dl.acm.org/citation.cfm?id=2788890.2788929

[69] Dimitrios Raptis, Nikolaos Tselios, Jesper Kjeldskov, and Mikael B. Skov. 2013. Does Size Matter?: Investigating the Impact of Mobile Phone Screen Size on Users' Perceived Usability, Effectiveness and Efficiency.. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 127–136. https://doi.org/10.1145/2493190.2493204

[70] Anne Roudaut, Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2009. Leaf Menus: Linear Menus with Stroke Shortcuts for Small Handheld Devices. In *Human-Computer Interaction – INTERACT 2009*, Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler (Eds.). Springer, Berlin, Heidelberg, 616–619. https://doi.org/10.1007/978-3-642-03655-2_69

[71] Krystian Samp and Stefan Decker. 2011. Visual Search in Radial Menus. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer, Berlin, Heidelberg, 248–255. https://doi.org/10.1007/978-3-642-23768-3_21

[72] Andrew Sears and Ben Shneiderman. 1994. Split Menus: Effectively Using Selection Frequency to Organize Menus. *ACM Trans. Comput.-Hum. Interact.* 1, 1 (March 1994), 27–51. https://doi.org/10.1145/174630.174632

[73] Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. ACM, New York, NY, USA, 173–182. https://doi.org/10.1145/2370216.2370243

[74] Benjamin L. Somberg. 1987. A Comparison of Rule-based and Positionally Constant Arrangements of Computer Menu Items. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*. ACM, New York, NY, USA, 255–260. https://doi.org/10.1145/29933.275639

[75] Erum Tanvir, Jonathan Cullen, Pourang Irani, and Andy Cockburn. 2008. AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-down Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1381–1384. https://doi.org/10.1145/1357054.1357270

[76] Theophanis Tsandilas and m. c. schraefel. 2005. An Empirical Assessment of Adaptation Techniques. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 2009–2012. https://doi.org/10.1145/1056808.1057079

[77] Jean Vanderdonckt. 1999. Computer-Aided Design of Menu Bar and Pull-Down Menus for Business Oriented Applications. In *Design, Specification and Verification of Interactive Systems'99, Proceedings of the Eurographics Workshop in Braga, Portugal, June 2-4, 1999*, David J. Duke and Angel R. Puerta (Eds.). Springer, 84–99. https://doi.org/10.1007/978-3-7091-6815-8_7

[78] Jean Vanderdonckt, Sara Bouzit, Gaëlle Calvary, and Denis Chêne. 2018. Cloud Menus: A Circular Adaptive Menu for Small Screens. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 317–328. https://doi.org/10.1145/3172944.3172975

[79] Jean Vanderdonckt, Paolo Roselli, and Jorge Luis Pérez Medina. 2018. !FTL, an Articulation-Invariant Stroke Gesture Recognizer with Controllable Position, Scale, and Rotation Invariances. In *20th International Conference on Multimodal Interaction, ICMI 2018, Boulder, CO, USA, October 16-20, 2018*. 353–364. https://doi.org/10.1145/3242969.3243032

[80] Shelly Welch and Si-Jung Kim. 2013. Determining the Effect of Menu Element Size on Usability of Mobile Applications. In *Design, User Experience, and Usability. Web, Mobile, and Product Design*, Aaron Marcus (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 740–749.

[81] Janet L. Wesson, Akash Singh, and Bradley van Tonder. 2010. Can Adaptive Interfaces Improve the Usability of Mobile Applications?. In *Human-Computer Interaction*, Peter Forbrig, Fabio Paternó, and Annelise Mark Pejtersen (Eds.). Springer, Berlin, 187–198. https://doi.org//10.1007/978-3-642-15231-3_19

[82] Michelle Wiebe, Denise Y. Geiskkovitch, and Andrea Bunt. 2016. Exploring User Attitudes Towards Different Approaches to Command Recommendation in Feature-Rich Software. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16)*. ACM, New York, NY, USA, 43–47. https://doi.org/10.1145/2856767.2856814

[83] Pavani Yalla and Bruce N. Walker. 2008. Advanced Auditory Menus: Design and Evaluation of Auditory Scroll Bars. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '08)*. ACM, New York, NY, USA, 105–112. https://doi.org/10.1145/1414471.1414492

Received May 4, 2018; revised August 18, 2018; accepted October 14, 2018