XX

Polymodal Menus: A Model-based Approach for Designing Multimodal Adaptive Menus for Small Screens

SARA BOUZIT, Orange Labs, Université Grenoble Alpes
GAËLLE CALVARY, Université Grenoble Alpes
DENIS CHENE, Orange Labs
JEAN VANDERDONCKT, Université catholique de Louvain

This paper presents a model-based approach for designing Polymodal Menus, a new type of multimodal adaptive menu for small screen graphical user interfaces where item selection and adaptivity are responsive to more than one interaction modality: a menu item can be selected graphically, tactilely, vocally, gesturally, or any combination of them. The prediction window containing the most predicted menu items by assignment, equivalence, or redundancy is made equally adaptive. For this purpose, an adaptive menu model maintains the most predictable menu items according to various prediction methods. This model is exploited throughout various steps defined on a new Adaptivity De-sign Space based on a Perception-Decision-Action cycle coming from cognitive psychology. A user experiment compares four conditions of Polymodal Menus (graphical, vocal, gestural, and mixed) in terms of menu selection time, error rate, user subjective satisfaction and user preference, when item prediction has a low or high level of accuracy. Polymodal Menus offer alternative input/output modalities to select menu items in various contexts of use, especially when graphical modality is constrained.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability

KEYWORDS

Adaptation, adaptive menu model, model-based approach, polymodal menu, multimodality, prediction window

ACM Reference format:

Sara Bouzit, Gaëlle Calvary, Denis Chêne, and Jean Vanderdonckt. 2017. Polymodal Menus: A Model-based Approach for Designing Multimodal Adaptive Menus for Small Screens. *ACM Conf. on Eng. Int. Computing Systems*. Article ? (June 2017), 8 pages.
DOI: ?/?

1 INTRODUCTION

Menus in Graphical User Interfaces (GUIs) still represent today one of the most frequently used techniques for interacting with any web site or application, thus explaining why a significant body of research and development [3,22] has been devoted to optimizing its usage. Two adaptation forms are primarily investigated [6,31]: *adaptable menus* [15,31], respectively *adaptive menus* [15,27,29], are subject to any adaptation controlled by the end user, respectively the system. Adaptive menus typically present the end user with predicted items so as to speed up the item selection without searching for them in deep and wide menus, especially for feature-rich software where the amount of items becomes important. The ability to adapt a UI is often heralded as a desirable function of an interactive system because it should allow to adapt the system to user's need for her ultimate benefit [21]. While regarded as essential, adaptation does not make much use of other modalities of interaction, particularly on small screens where menus are prevalent [7,28].

In neurology, *polymodality* [6] refers to the ability of a single receptor of being responsive to stimuli coming from multiple modalities, such as light, sound, temperature, taste, pressure, and smell. A *polymodal receptor* denotes a sensory receptor that is responsive to more than one sensory modality or sub-modality (Merriam-Webster dictionary-http://www.merriam-webster.com/medical/polymodal). For instance, our human nocireceptors send pain signals to our brain because they are responsive to temperature, pressure, air, chemical conditions. By analogy to neurology, we hereby define *Polymodal Menus* as adaptive menus for graphical user interfaces where both the rendering of predicted items (for adaptivity) and the menu item selection inside or outside the prediction window can be ensured by more than one interaction modality.

With the continuous expansion of mobile applications running on an ever-increasing variety of mobile devices, new adaptivity techniques are required [21,32] that consider constraints imposed by these devices such as a moderate computational power, a limited set of interaction techniques and a reduced screen resolution [14]. This last constraint negatively affects the graphical navigation through several screens or pages, lists like phone setting or address books, and menus, especially when lengthy [22]. Since selection of a menu item requires a navigation time and a visual search time that are depending on the number of items that can be displayed in the UI, small screen devices, such as smart phones, are particularly affected [9,28,31].

End users wish to interact with UIs that exhibit an increasing intelligence adapted to their interactive task. UI adaptation remains essential to ensure the user experience and to introduce the right UI changes at the right time. However, the quality of adaptation is proportional to the quality of prediction [28]. If the prediction algorithm is powerful enough, adaptation brings its earnings by accelerating and facilitating user interaction [17]. Since the quality of prediction can never be guaranteed, the following questions are stated: how to convey the predicted items without disturbing the initial menu and how to efficiently select a menu item inside this prediction window when prediction is correct without penalizing the selection outside the window when prediction is incorrect. A second goal is to assess to what extent relying on another modality than graphical could impact the selection. In order to address the aforementioned challenges, this paper investigates *Polymodal Menus* (Fig. 1), a new type of multimodal adaptive menu where item selection and prediction window for adaptivity are responsive to more than one interaction modality or several modalities considered together. The remainder of this paper is structured as follows: Section 2 reviews work related to adaptive menus and multimodal menus, Section 3 presents a model-based approach for designing Polymodal Menus based on an Adaptive Menu model throughout a new design space, Section 4 reports on a controlled experiment conducted to compare four conditions of Polymodal Menus, i.e., graphical, gestural, vocal, and mixed, in terms of quantitative and qualitative measures. Section 5 concludes the paper, namely by presenting some future avenues to this paper.

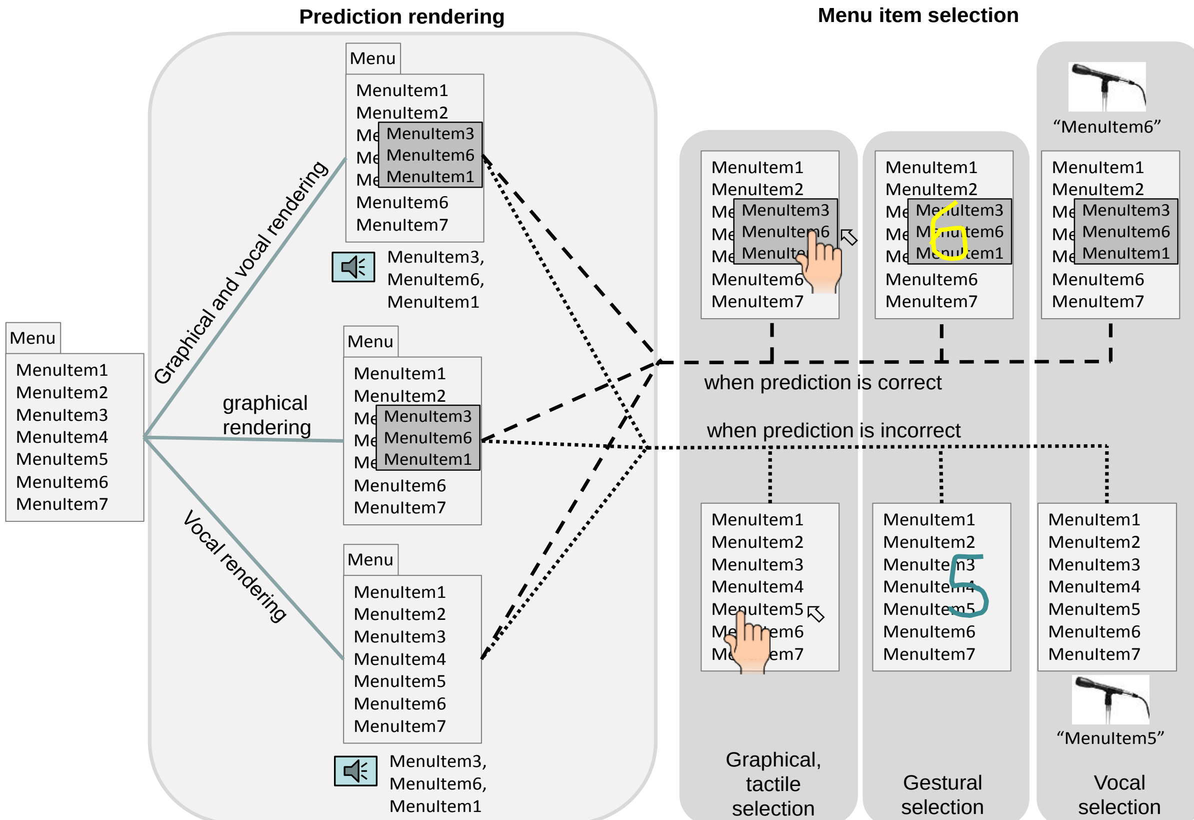

Fig. 1. The possible modalities for prediction rendering and menu item selection in Polymodal Menus.

2 RELATED WORK

Adaptive menus demonstrated a better performance [32] in terms of item selection time over non-adaptive menus [27] and even a better performance and user satisfaction on small screens than on large screens [14]. This performance increase is at the price of a high understanding [28] and predictability [9,11] of menu items and accuracy in which items are predicted [17]. Lack of predictability may damage the user subjective satisfaction, which is a risk for any adaptive UI [21] as well as for menus [17], because it may disorient end users and provoke inconsistency between different states of the same adaptive GUI. End users often feel the loss of control induced by adaptivity [19] as opposed to adaptability. Adaptability fosters predictability since the adaptation remains under the end user's control, but adaptive GUIs are recognised for diminishing the work load that would be otherwise required by adaptability [19]. Polymodal Menus represent a new type of adaptive menu overlapping two areas of research: adaptive menus and multimodal menus, which are discussed in the next sub-sections with a focus on menus.

2.1 Background on Adaptive Menus

Adaptive interfaces exhibit a certain amount of potential benefits at a certain cost [21,31]: the gain expected from promoting predicted items to speed up selection is unfortunately sometimes counter-balanced by the end user's need to constantly adapt to the altered menu layout, especially for novice users who hesitate between alternate areas. The cost/benefit ratio becomes positive when the adaptivity can reduce the menu selection time, such as in a hierarchical menu [18] or when limited screen resolution induces long scrolling [22]. Other revealed 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 instability is provoked by altering the initial menu [16]. Four menu stability properties [8] could be preserved when the layout of the initial menu is left unchanged after adaptivity [16].

Dynamic menus [3] sort items by decreasing order of frequency, depending on the end user's actions. *Split menus* [27] separate a graphical menu into two parts: a *prediction window* (a topmost area promoting a small list of predicted items, typically 2-3) and a *static menu* (a second area containing non-predicted items). Since split menus induce such a separation, *Split menus with replication* [18] were introduced to re-establish spatial stability: the adaptive part is simply added on top of the initial menu without altering it.

Smart menus 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 [4]. Smart menus track how often an end user invokes each item, in order to predict frequently and recently used items.

Morphing menus [11] increase, respectively decrease, the font size of each menu item if its prediction is high, respectively low. While morphing menus preserve spatial stability, they facilitate selection of accurately-predicted items, but not low-prediction items.

Adaptive Activation-Area Menus (AAMU) [29] enlarge the activation area for predicted items, which dynamically resizes itself depending on the level of prediction, thus 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.

Temporal menus [23] display 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 a smartphone is not straightforward because all items cannot be displayed on a single screen.

Ephemeral menus [16] are an adaptive 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 500 ms remaining items appear gradually. This approach suffers from the same problem of temporal menus and items cannot be displayed on a single screen.

In-Context Disappearing (ICD) Menus are tailored to smartphones [9]: at opening the initial menu, the end user finds a superposition of the full menu 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. *Out-of-Context Disappearing (OCD) Menus* make the inverse [9]: at opening, the prediction window is immediately displayed with the predicted items; after a delay of 500 msec [16], the complete menu gradually replaces the prediction window.

Other forms of menus represent candidates for adaptivity [3], but they are aimed to be adapted at design-time to some predefined conditions, such as accessibility or structure optimisation [18] and are not aimed at supporting adaptivity at run-time. Run-time adaptive menus do not address the trade-off between usable adaptivity and false predictions [9]: inadequate prediction may lead to user error and frustration.

In conclusion, all known adaptive menus remain monomodal since they exclusively exploit the graphical modality for adaptivity. An exploratory study [8] suggests that no adaptive menus could actually satisfy the four stability properties. Previous studies show that prediction displayed through a modal window seems to be a good candidate [32].

2.2 Background on Multimodal Menus

SketchedMenu [4] automatically adapts the creation of a UI object, like a menu, by sketching its location, size, and orientation, but the menu, once created, is not adaptive.

PocketMenu [26] changes 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 is useful for visually disabled users or eye-free interaction.

Ma *et al*. [24] introduced a pen-touch/speech-enabled Web browser for browsing web visual hierarchical menus. This system enables keypad-based and spoken input, and allows for spoken audio output. Results from a user study that compared this multimodal menu-based interface with standard graphical UIs on small devices indicate that the resulting UI requires fewer button presses to find information and was subjectively preferred by end users. Both performance and preferences were improved by the alternative modalities.

Sonically-enhanced UIs, where sound is added to the initial graphical UI, and their advantages have also been explored [12]: many different widgets, such as menus, buttons, scroll bars, progress bars, have been augmented with non-speech sound, called *earcons*, resulting in lower error rates and decreased task completion times [12]. Compared to a visual menu, a sonically-enhanced menu on smartphone demonstrated a significant reduction in task completion time and subjective workload, as well as an improvement of the overall menu usability [32].

Audio feedback has also been used for item selection, to indicate that the cursor has reached a target [1,12]. Akamatsu *et al*. [1] found out that such audio feedback reduces the time spent over the target, as the sound made users react faster. Cockburn & Brewster [12] found out that the addition of audio feedback reduced mean selection times by 4.2%, but that combining sound with other feedback modalities does not assure further improvements. Akamatsu *et al*. [1] and Cockburn & Brewster [12] used tactile vibrations to indicate that the mouse cursor hovered over a target. They both discovered that tactile vibration could improve the performance in certain contexts of use although they reported that the vibration could make the user miss small targets. Therefore, multimodal feedback provides ample opportunities for adaptivity [6,31].

Haptically-enhanced UIs [25], where haptic feedback is added to the initial UI, also offer potential benefits, more for single targets than for multiple targets, unless a particular adjustment is added. Multimodality is also demonstrated attractive for input on small devices, particularly in eyes-free condition [33] when the end user no longer wants to rely on the visual modality to select a menu item [32], which may be suitable for blind people or without any visual contact with the mobile device, similarly to the PocketMenu [26].

In conclusion, either menus are adaptive, but they remain monomodal, exclusively exploiting the graphical modality, or menus exhibit some multimodality, but considered as an additional or an alternative interaction modality to graphical modality either for input or for output, but never for supporting adaptivity. While all the aforementioned adaptive menus have provided some degree of adaptation or adaptivity, none is as adaptive as we are proposing: enabling menu adaptivity by using different input modalities for predicted item selection. Further, none has examined which modality might be used to improve adaptivity with respect to existing adaptive graphical-only adaptive menus. Multimodality has never been examined as a way to support menu adaptivity.

3 A MODEL BASED APPROACH FOR POLYMODAL MENUS

3.1 Initial Design of Polymodal Menus

A focus group was initially conducted with 2 UI designers, 1 human factors expert, 1 researcher as moderator, and 12 end users to determine potential forms for adaptive menus on smartphones in a qualitative way: the

discussion focused on adaptivity flow, suitability of adaptive menus on smartphones, quality of adaptivity, impact of adaptivity, and ease of use. Based on this exploratory study, as well as on related work reported in Section 2, the four most frequently quoted requirements with high importance were elicited for the Polymodal Menus: (**R1**) provide an adaptive menu that improves user interaction when item prediction is accurate without depreciating it when prediction is inaccurate, (**R2**) produce a prediction window with a maximum amount of four predicted items, (**R3**) maintain all four stability properties on the initial menu, and (**R4**) be sensitive to three interaction modalities (i.e., graphical, vocal, and gestural).
In order to satisfy R1-R3, a Polymodal Menu consists in two parts: the *initial menu*, the static part containing all the menu items from which the end user can select, which is left untouched throughout the whole interaction, and the *prediction window*, the adaptive part containing predicted items. The *prediction rendering* is hereby defined as the interaction modality used to convey the prediction window: graphical (the prediction window is displayed as a window superimposed to the initial menu [7]) or vocal (the items of the prediction window are issued vocally by text-to-speech synthesis). The *item selection* is hereby defined as the interaction modality used to select an item either from the prediction window when the prediction is correct or from the initial menu when the prediction is incorrect: graphical (the item is selected by pointing: a finger tap on a touchable surface or a pointing with a stylus), vocal (the item is captured by speech recognition), or gestural (the item is selected by a multi-stroke gesture drawn on the device screen). These three modalities have also been selected because sight, hearing, and touch respectively represent 80%, 10%, and 3% of the signals handled by the human brain [30] and the most frequent ones [1].
Considering the different techniques for composing the prediction rendering (output) and the selection modality (input), the CARE properties for multimodal interaction [13] are stated as follows in order to satisfy (R4):

- *Assignment*: a specific abstract event can be triggered exclusively using a specific item selection, only one of the three included.
- *Equivalence*: the same abstract event can be triggered through different item selections. The user can select one among them for completing the selection.
- *Redundancy*: an abstract event is triggered providing the same selection through more than one modality (e.g., pointing plus vocal confirmation of the same selection).
- *Complementarity*: two or more item selections must be used for triggering the abstract event, but none of them is able to complete the change individually. It is not implemented because of interaction complexity.

Possible interaction paths with Polymodal Menus are (Fig. 1): the prediction can be rendered graphically (the prediction window is displayed superimposed to the initial menu) and/or vocally (the items belonging to the prediction window are issued vocally), thus supporting assignment, equivalence, or redundancy, but not complementarity. Depending on the prediction is correct (i.e., the item to be selected belongs to the prediction window, graphically depicted by a dashed path in Fig. 1) or incorrect (i.e., the item to be selected does not belong to the prediction window, graphically depicted by a dotted path in Fig. 1), three item selections are possible: graphical/tactile (the target item is selected by pointing), vocal (the target item is captured by speech recognition), or gestural (the item is captured by a gesture indicting the item position in the prediction window) or mixed. When prediction is incorrect, the end user can come back to the (untouched) initial menu graphically (two push buttons $\boxed{-}$ and $\boxed{+}$ were added to send the prediction window backward or bring it forward under the end user's control or press [Escape] to return to the initial menu), vocally (the user pronounces "Escape") or gesturally (the user issues a "X" gesture on the surface area). Similarly, assignment, equivalence, and redundancy are supported, but not complementarity.

3.2 Adaptive Menu Model for Polymodal Menus

We hereby define a model for characterizing adaptive menus that could be used in principle for any adaptive menu. Each menu item is firstly characterized by the following attributes:

a. A *menu label*, which contains the textual label of the menu item, along with the “&” character representing the mnemonic of the menu item. The character after this delimiter is underlined.
b. The *mnemonic*, which is the character to be selected by combining it with the “Alt” key instead of selecting it by pointing, e.g. “Alt + 3 for “Save”. A mnemonic should always be chosen among the real letters of the label, preferably those that are actually pronounced.
c. The *menu activation status*, which specifies whether the item is by default activated or deactivated.
d. The *menu shortcut*, which defines the sequence of keys to be pressed for directly accessing the menu item, which consists of normal keys, i.e., A, B, C, ..., X, Y, Z, 1, 2, ...9, 0, F1, F2, ..., F11, F12, Del, Ins, ... and control keys, i.e., « Crtl », « Alt », « Shift ».
e. The *menu attachment type*, which specifies whether a menu item is related to displaying a sub-menu (for instance, a pull-down menu or a cascading menu), to opening a dialog box or a secondary window, or to triggering directly a semantic function of the application. When a menu item is itself attached to another menu, it points to the corresponding menu identification.
f. The *contextual help message*, which specifies the message to be conveyed when the menu item is activated.

Second, each menu item is characterized by a series of *Usage vectors UV=(User, Device, PredictionMethod)* where *User* denotes a pointer to a user model, *Device* denotes a pointer to a device model, and *PredictionMethod* is of the form *PredictionMethod=(prediction type, value)*, such as:

- (MFU, frequency) where MFU stands for Most Frequently Used and frequency holds the percentage of the menu item frequency usage (e.g., 10%).
- (MFA, frequency) where MFA stands for Most Frequently Activated and frequency hold the percentage of the menu item activation (e.g., 15%). Note that a menu item could be activated (selected), but finally not used. Therefore, MFA is always greater or equal than MFU.
- (MRU, timestamp) where MRU stands for Most Recently Used and timestamp denotes the last time this item was used.
- (MRUn, amount) where MRUn stands for Most Recently Used n times and amount denotes how many times a menu item has been accessed after the previous one.
- (DOI, percentage) where DOI stands for Degree of Interest and percentage denotes the percentage of general interest for the *User* on her device *Device*.
- (TOI, percentage) where TOI stands for Topic of Interest and percentage denotes the percentage of specific (topical) interest for the *User* on her device *Device*.
- …

Finally, an adaptive menu model is defined as *AMM= (menuID, menu_items, usages)* where *menuID* is the menu identifier (e.g., an integer), *menu_items* is a list of menu items as defined above, and *usages* is a list of UV as defined above. Note that there could be as many UV as desired when a particular menu item could be considered for multiple prediction methods. When a menu item is attached to another menu, it also specifies the *menuID* of another AMM recursively.

4 A DESIGN SPACE FOR ADAPTIVE USER INTERFACES

Adaptive user interfaces have already been subject to a model-based approach [23] and to model-driven engineering [2], as well as for evaluation purposes [20]. Basically, an Abstract User Interface, describing a user interface without making any reference to a particular technological space and interaction modality, is mapped onto a Concrete User Interface, describing a user interface without making any reference to a particular technological space, but for a particular modality. Lopez *et al*. [23] rely on connectors to ensure this mapping, while Akiki *et al.* [2] rely on model-to-model transformations. While these approaches are compliant with the principles of Model-Driven Engineering (MDE), they do not discuss how adaptivity could be decomposed into a series of steps and sub-steps, during which an adaptive user interface model could be exploited. Adaptation is primarily expressed in terms of properties [8] or functions that are subject to adaptivity [20]. But the menu items themselves should be subject to adaptivity according to prediction methods.

In order to address these challenges, this subsection is aimed at defining a design space for adaptive UIs based on the psychological cycle Perception-Decision-Action (PDA) borrowed from cognitive psychology. Any interactive task involves three activities: *perception* when the end user has to perceive the basic properties of the context of use (e.g., the user model, the device model, and the environment model to be considered in the adaptation [10]), inkling the UI, *decision* when the end user has to make a choice between alternative options, and *action* when a decision has been taken to perform a particular action. The end user follows a series of PDA cycles with the UI (Fig. 2) until the task is completed. Similarly, the system perceives what the user is actually doing (e.g., through sensors, probes, user-generated events), decides what to do next (e.g., based on a task planner or a dialog controller), and execute required actions. End-user interaction with the systems is consequently depicted as a loop of two PDA cycles, one for the user and one for the system.

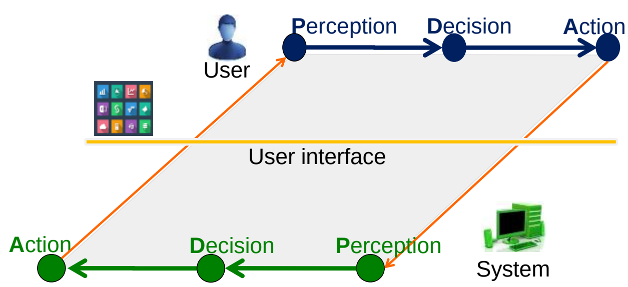

Fig. 2. Perception-Decision-Action user-system cycles.

In order to characterize adaptivity on top of this figure, a second Learning-Decision-Prediction (LDP) cycle intersects the two previous ones at the level of decision (Fig. 3): *learning* when the system exploits available data from end user interaction to learn what the end user did (e.g., action history, previous errors, traces), *decision* when the system may trigger adaptivity (e.g., through an adaptation engine based on adaptation rules), and *prediction* when the system infers future actions from previous actions (e.g., by relying on various prediction methods as captured in the adaptive menu model). Similarly, the end user also adapts herself to the system according to another LDP cycle. After each interaction, the user creates a mental model about the system, creates some knowledge resulting from prior experiences with the system functions and paths allowing to achieve an objective, then creates adaptation techniques to improve her interaction. Finally, the user is also able to predict part of whole of the adaptive system behavior. The combination of the two PDA and LDP user-system cycles gives rise to a new design space for adaptive UIs (Fig. 4) that expresses the adaptivity behavior, including its predictability and accuracy, as analyzed in [17] and already initiated in [20]. If a prediction method produces a wrong item in the prediction window, this will generate an incorrect prediction (1 on Fig. 4), that will be decided (2) and applied (3), thus inducing an extraneous cognitive load in the head of the end user (4) who reacts by deciding (5) to undertake other actions (6) to overcome the problem. This extraneous behavior is itself captured by the system (7) and learnt by it (8). If a prediction method produces an accurate item in the prediction window, the same cycle will be followed, but with a positive output as opposed to a negative output in reaction to an inadequate prediction.

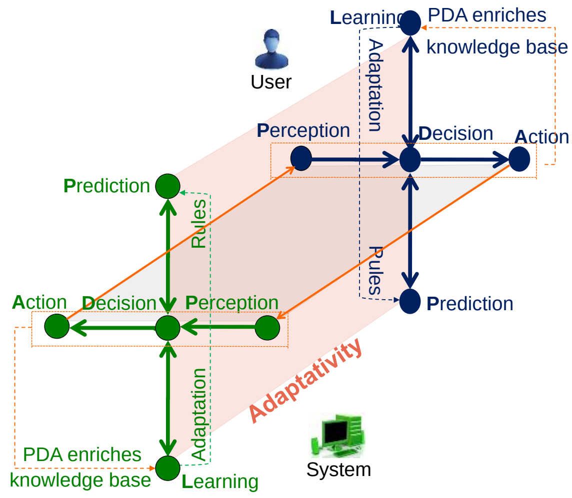

Fig. 3. Learning-Decision-Prediction user-system cycles.

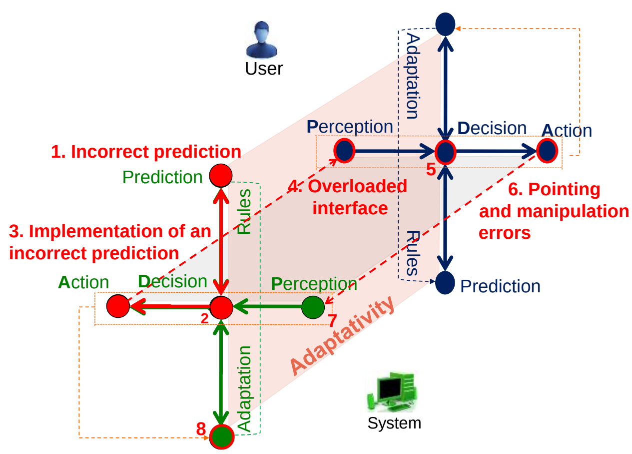

Fig. 4. Using the design space for adaptive user interfaces.

5 MODEL BASED APPROACH FOR PRODUCING POLYMODAL MENUS

Based on the design space for adaptive UIs represented in Fig. 4, a model-based approach for producing Polymodal Menus has been devised, structured into seven steps (Fig. 5) [20], and implemented in Java for Eclipse based on Android Software Development Kit.

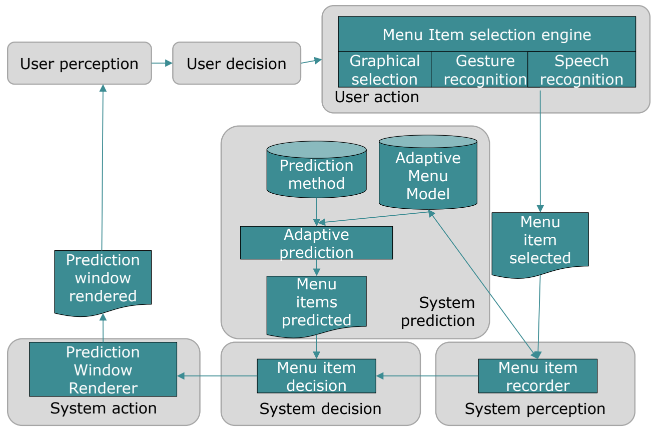

Fig. 5. Model-based Approach used for Polymodal Menus.

Step 1. System prediction. For any menu subject to polymodality, an *Adaptive Menu Model* is continuously maintained throughout the life cycle. For any prediction method (e.g., MFU), a usage vector is maintained. *Adaptive prediction* is then performed based on the prediction method and the model in order to determine predicted items. This module performs various prediction methods [11,28] or any combination of them, through a weighted function. This module accesses the corresponding usage vectors to update their values. For instance, a newly selected item will update its frequency in MFU and MFA, its timestamp in LRU, etc.

Step 2. System decision. Predicted items are then classified in *Menu item decision* into one level of prediction as recommended in [15]: (1) any item with high prediction (frequency≥ 80%); (2) any item with a medium prediction (60% ≤ frequency ≤ 80%); (3) any item with low prediction above a tailorable threshold (threshold ≤ frequency ≤ 60%).

Step 3. System action. A *Polymodal Menu* consists of a linear list for the static menu superimposed by a prediction window that is produced by the *Prediction Window Renderer*, based on the Adaptive Menu Model. For instance, a graphical rendering may be generates as follows: any first-level item is located at the top of the prediction window, any medium-level item is presented in the centre of the prediction window, any low-level item is displayed at the bottom. For the vocal rendering, Text-to-speech synthesis is implemented with Android Text2Speech module, where each predicted menu item is encoded as a formatted ASCII string generated by template filling based on the *Adaptive Menu Model* without having any internal semantic representation (for instance, the item labelled "Rotate view" is considered syntactically only). When several items are predicted, a single utterance is produced with the menu items separated by a small pause (e.g., "Rotate, [pause], View, [pause], Options"). Graphical and vocal renderings could be combined.

Step 4. User perception. The prediction window is rendered to the end user, who is perceiving it according to the chosen interaction modalities. This step is outside the system.

Step 5. User decision. Based on her experience, the end user decides which menu item should be selected, either in the prediction window if the prediction is accurate or outside if the prediction is incorrect. This step is outside the system.

Step 6. User action. The *Menu Item selection engine* is a fusion engine of signals coming from the three sources where the end user can act: graphical/tactile by pointing, gesture recognition, and speech recognition. Speech recognition is ensured by Android Speech Recognizer based on a grammar containing the list of predicted items with no other intelligent speech recognition. Gesture recognition is implemented based on Android Gesture Library and Android Gesture builder, thanks to which initial gestures were defined and added to the Android recognizer. For instance, the menu item labelled "Rotate" will automatically select the initial letters of the label name, here *R*, *O*, *T*, etc. to feed the gesture recognition until no ambiguity remains. Menu items selected are then serialized in a message with timestamp to be passed to the *Menu Item Recorder* module and so forth.

5.1 User Study

Polymodal Menus support various prediction renderings and menu item selections (Fig. 1), thus representing a significant amount of possible combinations of interaction paths. In order to identify desirable paths, the controlled experiment envisioned here is aimed at comparing three prediction renderings and three item selections in the context of smart phones, which are typical devices constrained by screen resolution.

5.2 Methodology

There are two factors (independent variables): the within-subject menu type and the prediction accuracy level. The prediction accuracy level is *high*, respectively *low*, when the target item to select belongs, respectively does not belong, to the prediction window, however it is rendered. The four menu conditions to be compared are therefore:

1) *Graphical menu*: the prediction rendering is graphical-only and so does the item selection by assignment. The prediction window can be closed by pressing the [Close] button.
2) *Gestural menu*: the prediction rendering is graphical-only and the item selection is gestural by assignment. A "3" gesture is drawn for selecting the third item in the prediction window or in the initial menu after a "X" closed the prediction window.
3) *Vocal menu*: the prediction rendering is vocal-only and so does the item selection by assignment. Once the utterance of the prediction window is completed, the *Polymodal Menu* initiates voice recognition. The end user hears a microphone trigger similar to the "Ok Google" micro in Google vocal searching and is prompted to give an answer. If the user is looking for the item "France" in the prediction window "Germany, France, Pecan", a "France" utterance validates the end user's choice.
4) *Mixed menu*: the prediction rendering is graphical and vocal while the item selection is graphical, vocal, and gestural, all by redundancy (which subsumes equivalence).

5.3 Hypotheses

The hypotheses formulated for this user study are:

H1. Speed

- High prediction accuracy level

No significant difference between Graphical and Gestural conditions. When prediction is correct, item selection in Gestural condition is achieved by drawing "1", "2" or "3". These three gestures are assumed affordable to remember and to produce. This action should be as efficient as pointing in Graphical condition. No significant difference should exist.

Graphical and Gestural conditions will be faster than Vocal condition. The time elapsed from the vocal pronunciation by the system of predicted items to the recognition of the user response may be longer than in Gestural condition when users draws any digit, e.g., "1", and will be also longer than pointing in Graphical condition. A gesture takes some time to produce, although not conditioned by any selection since the gesture is acquired on the menu entire surface.

No significant difference between Mixed condition and the three other conditions (Graphical, Gestural, Vocal). In mixed condition, the end user is free to use any modality for validating prediction. So, the user can point the target directly as well as drawing “1”, “2”, “3” gestures or response using voice modality, which minimizes the difference between mixed condition and other conditions.

- Low prediction accuracy level

Vocal will be faster than Graphical, Gestural and Mixed conditions. When prediction is wrong, the UI in Vocal condition is not overloaded by a useless prediction. The end user interacts directly with the full list of items without any hindrance. On the contrary, for Graphical, Gestural and Mixed conditions, the user has to close prediction window and to search the item in the full list of items.

No difference between Graphical condition and Gestural condition. In both Graphical and Gestural conditions, when prediction is incorrect, the end user has to close the prediction window and search the target in the full list of items. The user closes the prediction window by selecting “Close” in Graphical condition or closes it by drawing “X” in Gestural condition. This action is simple to achieve and will be similar in terms of time with selecting “Close” button.

Mixed condition will be slower than Graphical and Gestural conditions. In Mixed condition, the user may choose among different modalities (pointing, gesture, voice) for closing the prediction window, which may be confusing and slow down interaction compared to Graphical and Gestural conditions.

H2. Errors

- High prediction accuracy level

Errors will be less frequent in Vocal condition than in Graphical, Gestural and Mixed conditions. When prediction is correct in Vocal condition, the user vocally selects the item without making tallies in the UI which reduce errors compared to Graphical, Gestural and Mixed conditions where the user can point or draw in the UI.

Errors will be less frequent in Gestural condition than in Graphical condition. When prediction is correct in Graphical condition, the end user should precisely point the item in the prediction window, as opposed to issuing a gesture at any location, in any size in the Gestural condition, provided that it is properly recognized.

- Low prediction accuracy level

Errors will be less frequent in Vocal condition than in Graphical, Gestural and Mixed condition. When prediction is wrong in Vocal condition, the user does not need to do anything for closing the prediction window since it is verbalized: the user is immediately on the full list of items, which may reduce errors. As opposed to Graphical, Gestural and Mixed conditions, an interaction step is added as soon as the end user closes the prediction window for accessing the full list of items, which may generate extraneous actions, including but not necessarily errors.

Errors will be more frequent in Mixed condition than in Graphical and Gestural conditions. When prediction is wrong, the user closes the prediction window with different modalities in Mixed condition, then searches for the target item in the full list of items. This may be confusing compared to Graphical and Gestural conditions.

H3. User Preference

- High prediction accuracy level

Vocal condition will be preferred over Graphical, Gestural and Mixed conditions. In Vocal condition, user interacts with the system by producing an utterance of the target. In Gestural and Graphical condition, the user should precisely draw a digit, e.g., “1”, “2” or “3” or point to the target. This behaviour is similar when the user relies on gesture or pointing in Mixed condition. For this reason, Vocal condition may be preferred because it reduces perceptive and cognitive efforts and relies on an alternate channel.

- Low prediction accuracy level

Vocal condition will be preferred over Graphical, Gestural and Mixed conditions. No prediction window needs to be closed when prediction is wrong. Contrarily to Graphical, Gestural, and Mixed condition where the interface is overloaded by graphical prediction rendering, the Vocal condition could become the preferred option.

5.4 Task

Users were asked to perform a sequence of item selections. For each condition, i.e., graphical, gestural, vocal, and mixed, the user has to select "Start" button for starting the test (which also starts the chronometer), a message appears indicating the target item to be selected. Then the menu appears. In all conditions, the item name remains displayed at top of the screen as a reminder. The user has to select the requested target in order to move to a new selection (new task). If the user makes a wrong selection, an error message appears inviting user to find the requested target. When the user succeeds in selecting the right target, a new message appears specifying the name of the new target to select. A thank you message informs the user when the test is complete.

5.5 Quantitative and Qualitative Measures

Two response (dependent) variables were measured: 1) speed selection (menu item selection time) that was measured by the time taken from opening the menu until final selection of requested target (in seconds); 2) task completion, based on recorded error rates. Speed selection and error rates have been selected because they have been estimated as critical factors determining the performance of adaptive menus [11,17] and are representative variables for various forms of adaptivity to be controlled [7,14,15].

5.6 Apparatus

Android-based Google Nexus smartphones were used in this experiment. System data were recorded in a database: selection time (in msec), scrolling time (in msec), and error rate.

5.7 Participants

Thirteen subjects participated in this experiment. They are aged between 21 and 58 (μ=29.38, SD=11). All participants were regular smartphone users and they were recruited in other departments of our organization through a mailing list.

5.8 Procedure

Each participant performed the task in a controlled environment. Prior to the task, each participant was welcomed, signed the consent form, and filled in a short questionnaire on their profile and background. After the questionnaire was completed, the researcher demonstrated the usage of Polymodal Menus to participants and explained the principle of each condition without mentioning the two prediction accuracy levels. The participants were then given a short training period (5 minutes) during which ten item selections were completed in a dedicated pre-test. The list of items used in this pre-test was different from those used in the test conditions. Findlater *et al*. [15] test menu used for their experiment on *ephemeral menus* today becomes a reference test for adaptive menus: the initial menu contains 4 groups of 4 related items (i.e., England, France, Germany, Spain – Venus, Mercury, Jupiter, Saturn – Cabernet, Chardonnay, Merlot, Shiraz – Almond, Pecan, Pistachio, Walnut) and the prediction was defined as follows: Venus=80%, Spain, Shiraz=70%, Pecan, Cabernet, Pistachio=60%, all other items having the same normal probability to be selected. The rationale for choosing 4 items in the prediction window is based on an interval of a minimum of 3 items [16] and a maximum of 4 [14]. Beyond 4 items, the ratio # predicted items/# visible items becomes prohibitive as the prediction window almost overlaps the initial menu on small screens.

Similarly to the Findlater test menu, four menus of 16 items were defined and divided into 2 screens, each one containing 8 items. The menus used in the four conditions were different as well as the sequences of target items to select which were randomly produced by a pseudo-random generator. In each condition, users were asked to select 20 items: 10 items when prediction is correct and 10 items when prediction is wrong. In summary, the design was as follows:

13	participants	×
20	target items (10 correct and 10 incorrect)	×
4	conditions (*graphical*, *gestural*, *vocal* and *mixed*)	
=1040	item selections in total	

After each test, each participant was also asked to fill in a post-task questionnaire composed of 9 items extracted from the IBM Computer Satisfaction Usability Questionnaire (CSUQ), an empirically-validated questionnaire benefitting from a 0.94 reliability coefficient related to usability and a high internal cohesion. Each question was measured using a 5-point Likert scale (1= strongly disagree, 2=disagree, 3= neutral, 4=agree, 5= strongly agree) and was phrased positively in an assertive way as follows:

Q1: Overall, I am satisfied in learning this menu
Q2: This menu is easy to use
Q3: I successfully completed the task required
Q4: I felt comfortable using this menu
Q5: I felt effective in learning how to use this menu
Q6: I felt efficient in using this menu
Q7: The item organization on the screen is clear
Q8: The user interface of the menu is user-friendly
Q9: Overall, I am satisfied in using this menu

Six items were intended to measure perceived ease of use (Q1, Q2, Q4, Q7-9) and three items to measure perceived usefulness (Q3, Q5, and Q6). Items were randomly ordered.

5.9 Analysis

After each participant, the questionnaire, ratings and ranking data was added into a database, along with the results of the experiment. The data was entered in an anonymous format so the participants could not be identified. The data was analysed by a dedicated Microsoft Excel sheet combining graphs and inferential statistics.

5.10 Results

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

Speed

Obtained results (Table 1) show that when prediction is correct, graphical condition is significantly faster than the other three conditions (Gr.: M=1.461, SD=0.366, Ge.: M=2.969, SD=0.9, $W(24)=3.18$, $p<0.005$), (Gr.: M=1.461, SD=0.366, V.: M=4.438, SD=0.876, $W(24)=3.18$, $p<0.005$), (Gr.: M= 1.461, SD=0.366, Mixed:

M=4.184, SD=1.154, *W*(24)=3.18, $p<0.005$). Next, users were revealed faster in Gestural condition when prediction is correct compared to Vocal and Mixed conditions (G.: M=2.969, SD=0.9, V.: M=4.438, SD=0.876, *W*(24)=3.04, $p<0.005$), (G.: M=2.969, SD=0.9, Mixed: M=4.184, SD= 1.154, *W*(24)=2.48, $p<0.05$). No significant difference exists between Vocal and Mixed condition when prediction is correct (Vocal: M=4.438, SD=0.876, Mixed M=4.184, SD=1.154, *W*(24)= -0.63, $p > 0.5$).

Table 1. Results for Selection time

Menu	Average	Standard Deviation	W	p-Value
Graphical (P+)	1.461	0.366	3.18	0.005
Gestural (P+)	2.969	0.9		
Graphical (P+)	1.461	0.366	3.18	0.005
Vocal (P+)	4.438	0.876		
Graphical (P+)	1.461	0.366	3.18	0.005
Mixed (P+)	4.418	1.154		
Gestural (P+)	2.969	0.9	3.04	0.005
Vocal (P+)	4.438	0.876		
Gestural (P+)	2.969	0.9	2.48	0.05
Mixed (P+)	4.418	1.154		
Vocal (P+)	4.438	0.876	-0.63	0.5
Mixed (P+)	4.418	1.154		
Graphical (P-)	4.1	0.886	3.18	0.005
Gestural (P-)	6.484	1.643		
Graphical (P-)	4.1	0.886	-2.76	0.01
Vocal (P-)	3.115	0.883		
Graphical (P-)	4.1	0.886	3.06	0.005
Mixed (P-)	6.577	1.978		
Gestural (P-)	6.484	6.484	-3.18	0.005
Vocal (P-)	3.115	0.883		
Gestural (P-)	6.484	1.643	0.42	0.5
Mixed (P-)	6.577	1.978		
Vocal (P-)	3.115	6.577	3.06	0.005
Mixed (P-)	0.883	1.978		

However, when prediction is wrong, Vocal condition is the fastest one compared to all Graphical, Gestural, and Mixed conditions. When prediction is incorrect, Graphical condition is better than Gesture and Mixed conditions (Gr.: M=4.1, SD=0.886, Ge.: M=6.484, SD=1.643, *W*(24)=3.18, $p< 0.005$), (Gr.: M=4.1, SD=0.886, Mixed: M=6.577, SD= 1.978, *W*(24)=3.06, $p<0.005$). There is no significant difference between Gestural and Mixed conditions when prediction is wrong. In Graphical, Gestural, and Mixed conditions when prediction is correct, users were faster than when prediction is wrong. Users were faster in Vocal condition when prediction was wrong than when prediction was correct.

Errors

When prediction is correct, errors occur slightly less frequently in Gestural and Vocal conditions than in Graphical and Mixed conditions (Gr.: M=0.17, SD=0.39, Ge.: M=0, SD=0, Vocal: M=0.08, SD=0.29, Mixed: M=0.33, SD=0.65) but no really significant difference was captured (Table 2).

Table 2. Results for Error rate

Menu	Average	Standard Deviation	W	p-value
Graphical (P+)	0.17	0.39		
Gestural (P+)	0	0		
Vocal (P+)	0.08	0.29		
Mixed (P+)	0.33	0.65		
Graphical (p-)	0.33	1.73	2.50	0.05
Gestural (P-)	2.08			
Gestural (P-)	2.08	1.73	2.67	0.01
Vocal (P-)	0	0		
Gestural (P-)	20.8	1.73	1.92	0.1
Mixed (P-)	0.75	0.96		
Vocal (P-)	0	0	2.26	0.05
Mixed (P-)	0.75	0.96		

In Graphical condition, there is no significant difference in terms of errors between the two cases when prediction is correct or not (Gr. with correct prediction: M=0.17, SD=0.39, Gr. with wrong prediction: M=0.33, SD=0.65, $W(24)=1$, $p< 0.5$). The same observation holds for Vocal condition: there is no significant difference between a correct and an incorrect prediction. The same result holds for mixed condition when prediction is correct or not. However, in Gestural condition, users tend to produce more errors when prediction is wrong than when prediction is correct. When prediction is wrong, errors are less frequent in Graphical condition than in Gestural condition. In the same case when prediction is wrong, errors are also less frequent in Vocal condition than in Gestural condition. Errors are also less frequent in Mixed condition than in Gestural condition when prediction is wrong. But errors in Mixed condition occur more frequently than in Vocal condition when prediction is wrong. In summary, when prediction is correct, the four conditions were almost similar in terms of errors; when prediction is wrong, users did fewer errors in Graphical and Vocal conditions.

User Preferences

Data analysis of questionnaires showed that users found that Graphical condition facilitates their interaction compared to three other conditions. Slight difference was captured between Gestural and Vocal conditions. There is no significant difference between Gestural and Mixed conditions. No significant difference was captured between Vocal and Mixed conditions. There is no significant difference in terms of user preferences.

5.11 Discussion

H1. Speed

- High prediction accuracy level

No significant difference between Graphical and Gestural conditions. **Not supported**. When prediction is correct, users pointed easily the item in the prediction window due to reduced navigation and visual search times. In Gestural condition, the user selects a target in prediction window by producing a digit gesture, which takes more time than pointing.

Graphical and Gesture conditions will be faster than Voice condition. **Supported**. In Vocal condition, the time taken for recognition and treatment of user response is longer than pointing and gesturing.

No significant difference between Mixed condition and the three other conditions (Graphical, Gestural, Vocal). **Partially supported**. There is no significant difference observed between Vocal and Mixed conditions: the time required by the speech recognition slows down interaction. In the Mixed condition, surprisingly, the user devoted more time to selecting the preferred modality and to operate it than actually performing the selection, which explains why Graphical and Gestural conditions were observed faster.

- Low prediction accuracy level

Vocal will be faster than Graphical, Gestural and Mixed conditions. **Supported**. This important result is justified by the fact that the interface is not overloaded by wrong prediction in Vocal condition and user has direct access to complete list of items. There is no voice response and no treatment delay. This may explain why users were faster in Vocal condition when prediction is wrong than when it is correct. In Graphical, Gestural and Mixed conditions, an interaction step is added to close the prediction window before moving.

No difference between Graphical condition and Gesture condition. **Not supported**. Between closing prediction window using pointing standard modality and using gesture modality, pointing stays the most beneficial modality.

Mixed condition will be slower than Graphical and Gesture conditions. **Partially supported**. Although the user could also rely on pointing in the Mixed condition, it was not considered better than in the Graphical condition: this is confusing for the user who has several modalities in the same time.

H2. Errors

There is no important difference between the four conditions in term of errors when prediction is correct: the user finds target directly on prediction window in Graphical, Gestural and Mixed condition which reduce errors. So, the prediction accuracy is important for decreasing the error rate. When prediction is wrong, errors are less frequent in Graphical and Vocal conditions: pointing reduces errors compared to gestures. In Vocal condition, user selects target using pointing on full list. The Mixed condition offers several modalities.

H3. User preferences

Apart for Graphical condition, there is no important difference between the other conditions.

6 CONCLUSIONS

This paper presented a model-based approach for designing Polymodal Menus, a new type of adaptive menu in which predicted menu items are rendered in a graphical and/or vocal prediction window and selected graphically, vocally, and/or gesturally. An Adaptive Menu Model has been defined to be exploited throughout an adaptivity design space. A user study reveals a promising interaction technique for supporting interaction with adaptive menus on smartphones. The impact of choosing one particular interaction modality either as input (graphical or vocal) or as output (graphical, vocal, or gestural) was analysed. Graphical pointing is beneficial in terms of speed and errors whether prediction is correct or not: it accelerates interaction and reduces errors rate. Gestural condition was not better than the Graphical one. Vocal condition does not accelerate interaction when prediction is correct due to system processing time. But when prediction is wrong, it does not penalize interaction because the UI did not graphically change and was not overloaded by graphical rendering. This preference depends strongly on the context of use: if the user is in a noisy context of use, vocal presentation should be avoided; in an eyes-free context, this preference may be important and beneficial. The mixed condition was not better in both cases when prediction is correct and wrong. The complete behaviour of the Polymodal Menu has been characterized on a design space consisting of two user-system cycles based on Perception-Decision-Action and Learning-Decision-Prediction. Which behaviour is offered at which step and which level of flexibility could be depicted on this design space, which could be used for any adaptable and any adaptive UI, not just adaptive menus. The only difference is that the three types of actions could be undertaken in principle by the end user, the system or by a combination of them, such as in mixed-initiative.

REFERENCES

[1] Motoyuki Akamatsu, I. Scott MacKenzie, and Thierry Hasbrouc. 1995. A Comparison of Tactile, Auditory, and Visual Feedback in a Pointing Task using a Mouse-Type Device. *Ergonomics* 38, (1995) 816–827. DOI: http://dx.doi.org/10.1080/00140139508925152

[2] Pierre Akiki, Arosha K. Bandara, and Yijun Yu. 2014. Adaptive Model-Driven User Interface Development Systems. *ACM Computing Surveys* 47, 1 (July 2014), Article #9. DOI: http://dx.doi.org/10.1145/2597999

[3] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Computing Surveys* 49, 4 (Feb. 2017), Article #40. DOI: http://dx.doi.org/10.1145/3002171

[4] Mohammed Belatar and François Coldefy. 2010. Sketched Menu: a Tabletop-Menu Technique for GUI Object Creation. In *Proceedings of the 2nd ACM Symposium on Engineering Interactive Computing Systems* (EICS'2010). ACM, New York, NY, 77–86. DOI: http://dx.doi.org/10.1145/1822018.1822031

[5] Carlos Belmonte, Juana Gallar, Laura G. Lopez-Briones, and Miguel A. Pozo. 1994. Polymodality in Nociceptive Neurons: Experimental Models of Chemotransduction. In *Cellular Mechanisms of Sensory Processing*, NATO ASI Series, 79 (Jan. 1994), 87–117. DOI: http//dx.doi.org/10.1007/ 978-3-642-78762-1_6

[6] Matthias Bezold and Wolfgang Minker. 2011. *Adaptive Multimodal Interactive Systems*. Springer, Berlin

[7] 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* (OzCHI'2014). ACM, New York, NY, 41–48. DOI: http://dx.doi.org/10.1145/2686612. 2686619

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

[9] Robert Bridle and Eric McCreath. 2005. Predictive Menu Selection on a Mobile Phone. In *Proceedings of Workshop W7 on mining spatio-temporal data* (ECML/PKDD'2005). Retrieved May 21st, 2017 from http://users.cecs.anu.edu.au/~Eric.McCreath/papers/ mining05.pdf

[10] Eduardo Castillejo, Aitor Almeida, and Diego López-de-Ipiña. 2014. Modelling Users, Context and Devices for Adaptive User Interface Systems. *International Journal of Pervasive Computing and Communications* 10, 1 (2014), 69–91. DOI: http://doi.org/10.1108/IJPCC-09-2013-0028

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

[12] Andy Cockburn and Stephen Brewster. 2005. Multimodal Feedback for the Acquisition of Small Targets. *Ergonomics* 48, 9 (Feb. 2007), 1129–1150. DOI: http://dx.doi.org/10.1080/00140130500197260

[13] Joelle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May and Richard M. Young. Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. In *Proc. of IFIP TC 13 Int. Conf. on Human-Computer Interaction* (INTERACT'95). Chapman & Hall, London, 115–120. DOI= http://dx.doi.org/10.1007/978-1-5041-2 896-4_19.

[14] Leah Findlater, and Joanna McGrenere. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2008). ACM Press, New York, NY, 1247–1256. DOI= https://doi.org/10.1145/1357054.1357249

[15] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. 2009. Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2009). ACM Press, New York, NY, 1655–1664. DOI= http://dx.doi.org/10.1145/1518701.1518956

[16] Krzysztof Z Gajos, Mary Czerwinski, Desney S Tan, and Daniel S Weld. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proc. of the ACM Working Conf. on Advanced Visual Interfaces* (AVI'2006). ACM Press, New York, NY, 201–208. DOI=http://dx.doi.org/10. 1145/1133265.1133306

[17] Krzysztof Z Gajos, Katherine Everitt, Desney S Tan, Mary Czerwinski, and Daniel S Weld. 2008. Predictability and accuracy in adaptive user interfaces. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2008). ACM Press, New York, NY, 1271–1274. DOI= https://doi.org/10.1145/1357054.1357252

[18] Mikhail V Goubko, and Alexander Varnavsky. 2016. Users' preference as a criterion for hierarchical menu optimization. In *Proc. of ACM Symposium on Engineering Interactive Computing Systems* (EICS'2016). ACM Press, New York, NY, 305–310. DOI= https://doi.org/10.1145/2933242. 2935875

[19] Francesca Gullà, Silvia Ceccacci, Michele Germani, and Lorenzo Cavalieri. 2015. Design Adaptable and Adaptive User Interfaces: A Method to Manage the Information. In *Proc. of Italian Forum 2014 on Ambient Assisted Living*. Biosystems & Biorobotics Series, vol. 11. Springer, Berlin, 47–58. DOI=http://dw.doi.org/10.1007/978-3-319-18374-9_5

[20] Bronwin Jason, Andre Calitz, and Jean Greyling. 2010. The evaluation of an adaptive user interface model. In *Proc. of the 2010 Annual Research Conf. of the South African Institute of Computer Scientists and Information Technologists* (SAICSIT'2010). ACM Press, New York, NY, 132–143. DOI=https://doi.org/10.1145/1899503.1899518

[21] Talia Lavie, and Joachim Meyer. 2010. Benefits and costs of adaptive user interfaces. *Int. J. of Human-Comp. Studies* 68, 8, 508–524. DOI=http://dx.doi.org/10.1016/j.ijhcs.2010. 01.004

[22] Dong-Seok Lee, and Wan Chul Yoon. 2004. Quantitative results assessing design issues of selection-supportive menus. *Int. J. of Industrial Ergonomics* 33, 1 (January 2004), 41–52. DOI= http://dx.doi.org/10.1016/j.ergon.2003.07.004

[23] Victor López-Jaquero, Francisco Montero, José P Molina, Antonio Fernández-Caballero, and Pascual González. Model-Based Design of Adaptive User Interfaces through Connectors. In *Proc. of 10th Int. Workshop on Design, Specification, and Verification of Interactive Systems* (DSVIS'2003). Lecture Notes in Computer Science, vol. 2844. Springer, Berlin, 245–257. DOI=http://dx.doi.org/10.1007/978-3-540-39929-2_17

[24] Xiaochuan Ma, Paul P Maglio, and Hui Su. 2003. Multimodal Menu Interface for Mobile Web Browsing. In *Proc. of 9th IFIP TC 13 Int. Conf. on Human-Computer Interaction* (INTERACT'2003). IOS Press, Amsterdam

[25] Ian Oakley, Stephen Brewster, and Philip Gray. 2001. Solving multi-target haptic problems in menu interaction. In *Proc. of ACM Conf.*

on Human Factors in Computing Systems (CHI'2001 EA). ACM Press, New York, NY. DOI= http://dx.doi.org/10.1145/634067.634278

[26] Martin Pielot, Anastasia Kazakova, Tobias. Hesselmann, Wilko. Heuten, and Susanne. Boll. 2012. PocketMenu: non-visual menus for touch screen devices. In *Proc. of the 14th Int. Conf. on Human-computer interaction with mobile devices and services*. (MobileHCI'2012). ACM Press, New York, NY, 327–330. DOI= http://dx.doi.org/10.1145/2371574.2371624

[27] Andrew Sears, and Ben Shneiderman. 1994. Split menus: Effectively using selection frequency to organize menus. *ACM Trans. on Comp.-Human Interaction* 1, 1, 27–51. DOI=http://dx.doi.org/10.1145/174630.174632

[28] Choonsung Shin, Jin-Hyuk Hong, and Anind K Dey. 2012. Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proc. of Int. Conf. on Ubiquitous Computing* (UbiComp'2012). ACM Press, New York, 173–182. DOI= http://dx.doi.org/10.1145/2370216.2370243

[29] Erum Tanvir, Jonathan Cullen, Pourang Irani, and Andy Cockburn. 2008. AAMU: Adaptive Activation Area Menus for Improving Selection in Cascading Pull-Down Menus. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2008). ACM Press, New York, NY, 1381–1384. DOI= http://dx.doi.org/10.1145/1357054.1357270

[30] Libb Thims. 2005. On the Nature of the Human Chemical Bond. *Journal of Human Thermodynamics* 1, 5, 36–61. Retrieved April 3rd, 2017 from http://www.humanthermodynamics.com/JHT/Human-Chemical-Bond.html

[31] Theophanis Tsandilas, and M. C. Schraefel. 2005. An empirical assessment of adaptation techniques. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2005 Extended Abstracts). ACM Press, New York, NY, 2009–2012. DOI= http://dx.doi.org/10.1145/1056808.1057079

[32] Yaohua Yu, and Zhengjie Liu. 2010. Improving the performance and usability for visual menu interface on mobile computers. In *Proc. of the ACM Working Conf. on Advanced Visual Interfaces* (AVI'2010). ACM Press, New York, NY, 369–372. DOI= http://dx.doi.org/10.1145/ 1842993.1843065

[33] Shengdong Zhao, Pierre Dragicevic, Mark Chignell, Ravin Balakrishnan, and Patrick Baudisch. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *Proc. of ACM Conf. on Human Factors in Computing Systems* (CHI'2007). ACM Press, New York, NY, 1395–1404. DOI= http://dx.doi.org/10.1145/1240624.1240836

Received January 2017; revised April 2017; accepted May 2017