Bubbling Menus: A Selective Mechanism for Accessing Hierarchical Drop-Down Menus

Theophanis Tsandilas
Department of Computer Science
University of Toronto
fanis@dgp.toronto.edu

m.c. schraefel
School of Electronics and Computer Science
University of Southampton
mc@ecs.soton.ac.uk

ABSTRACT
This paper introduces bubbling menus, a new design for cascading drop-down menus. Bubbling menus combine the bubble cursor [10] with directional mouse-gesture techniques to facilitate the access of certain items in a menu, such as frequently selected items. Through an extensive iterative design process, we explore bubbling menus in the context of adaptive and customizable user interfaces. Unlike other adaptation and customization techniques such as split menus, bubbling menus do not disrupt the original structure of menus and enable the activation of menus far from a menu bar. Results from two evaluation studies presented in the paper show that bubbling menus provide an effective alternative to accelerate menu selections tasks.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Experimentation, Human Factors

Keywords: Adaptive/-able user interfaces, customization, cascading menus, mouse gestures.

INTRODUCTION
When users interact with a user interface (UI), they frequently repeat tedious actions such as pressing a tool button, selecting a menu item or moving through a deep hierarchical structure. Furthermore, only a small number of the commands in complex software are frequently used. Previous work by McGrenere and Moore [17] on Microsoft Word showed that only 21.5% of the first-level functions of the application were used by more than half of the users and only a 3.3% functions were used in a regular basis by more than three quarters of the users. Hotkeys can help users to access items faster but require them to remember several key sequences. As Lane et al. [14] report, even experienced users fail to use such shortcuts effectively. Also, hotkeys cannot be used to access items in dynamically evolving components, such as bookmark lists.

This paper introduces bubbling menus, a new design of cascading pull-down menus that accelerates the selection of *hot menu items*. By "hot", we refer to menu items that are either frequently selected or their selection relates to a certain context of interaction such as editing a picture. According to our design, users can use single-stroke mouse gestures to switch to an alternative view, in which the activation area of hot items is enlarged. The new design uses the bubble cursor [10], a dynamically resizable cursor, and several motion-aware techniques to increase the activation area of hot items, improve motor control and facilitate menu selection. The advantage of bubbling menus over existing techniques, e.g., split menus [7, 21], that use automation or customization mechanisms to boost the selection of a small subset of items, is the fact that their application does not affect the original structure of menus. This property makes the proposed design particularly useful for expert users, who, having memorized a menu structure, can select widgets with directional gestures. At the same time, it does not prevent novice users from learning the position of items in a menu. Besides, unlike other designs, bubbling menus let users activate pull-down menus out of the strict boundaries of a menu bar (see Figure 1).

The paper presents two user studies that explore strengths and limitations of bubbling menus. The first study tests an adaptive version of the design against traditional menus on their ability to improve targeting performance. The second study evaluates an extended, customizable version of bubbling menus applied to the menu structure of Microsoft Word. Bubbling menus are contrasted with customizable split menus [7, 21], extended to support nesting. Based on the results of the two studies, we conclude with recommendations about appropriately using bubbling menus.

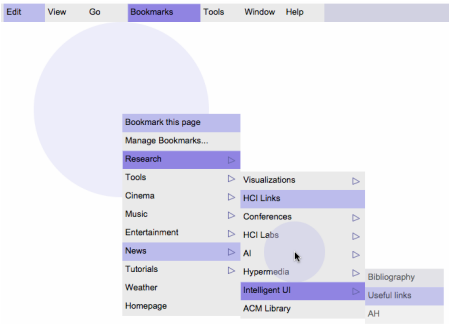


Figure 1. Overview of bubbling menus. Bubble cursors are used to select blue (hot) items in an alternative view, activated through mouse-dragging gestures. Blue items are specified by either a manual or an automatic customization mechanism.

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.

RELATED WORK

Menu Selection

Various models have been proposed to predict selection times in menus [13, 20]. These models consider two factors affecting performance in menu selection: visual search and pointing. Visual search depends on the ordering of menu items as well as user expertise. If items are sorted, e.g., alphabetically, search time can be predicted by Hick's Law [13], which states that the time to locate an item is a logarithmic function of the menu size. When menus are not alphabetical, users have to scan them in a linear fashion to locate an item. Alternatively, if users have memorized the position of items in a menu, search time is constant.

Pointing time can be predicted by Fitts' law [8], which states that the movement time *(MT)* needed to acquire a target is a logarithmic function of the ratio between the target distance *D* and the target width *W*, known as the task's Index of Difficulty (*ID*), which is measured in *bits*. According to Fitts' law, menu items that appear further down the menu have a greater *ID*. Not taking into consideration constraints in the shape of the motion trajectory, Fitts' law cannot accurately predict movement time in nested menus. If the cursor has to be *steered* along a tunnel, movement time is better modeled by the steering law [1]. According to this law, movement time is determined by the ratio between the tunnel distance *d* and the tunnel width *w*. The steering law has been used to model selection times in cascading pull-down menus [2, 23]. Ahlstrom [2] described menu selection tasks as compounds of vertical and horizontal motions, where vertical motion is modeled as a Fitts' law pointing task and horizontal motion is modeled as a steering task. Based on this model, Ahlstrom applied "force fields" by adapting the visual motion of the cursor to decrease the distance-width ratio. Kobayashi and Igarashi [11], on the other hand, suggested that submenus should pop up at the position where horizontal motion occurs so that the steering distance is minimized. Both approaches assumed that traditional menus require users to perform perfect steering motions to keep menu folders open. In fact, menu behaviour in modern operating systems is different. Figure 2 demonstrates the activation of menus in Mac OS X. After a submenu has been activated, its items can be pointed with a single motion. The width of the constrained steering motion is considerably wider than the width of the selection, and therefore, selection is faster.

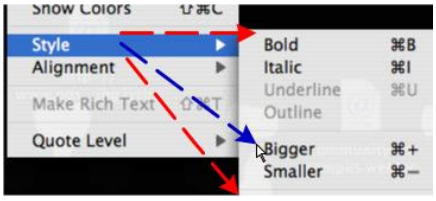


Figure 2. Selection in nested menus (Mac OS X). Motion is constrained within the triangular area outlined by the outer (red) arrows. When the cursor exits the boundaries of the selection ("Style"), for a brief time window, the user can move the cursor towards the submenu without causing the selection to change.

Finally, Cockburn and Gin [6] showed that menu selection times could be reduced if enlarging the activation area of menu folders and removing any delays before and after displaying submenus. Such delays (1) prevent fluttering, (2) force users to target the correct folder before initiating a steering motion, and (3) allow for optimal diagonal movements as shown in Figure 2. As the activation area of a menu folder grows, targeting and steering become easier, and the need for such delays disappears.

Improving Pointing Performance

Several interaction techniques have used Fitts' law to improve pointing either by decreasing the distance *D* [3] or by increasing the width *W* [5, 19, 23]. Other approaches have tried to decrease the *ID* of target acquisition tasks by dynamically adjusting the control-display (C-D) gain [4]. Finally, Grossman and Balakrishnan [10] introduced the bubble cursor (see Figure 3), a dynamically resizable cursor. The bubble cursor has two main strengths: (1) it provides continuous visual feedback making the selection of targets predictable; and (2) it makes maximum use of the free space. As free space is not equally allocated to all the targets, the success of the bubble cursor highly depends on the density of the targets and their position in space.

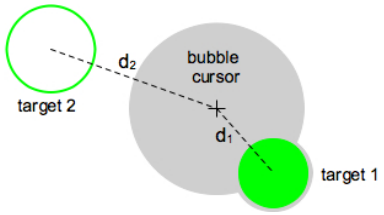


Figure 3. Bubble cursor. The size of the cursor dynamically changes as the cursor moves and selects the target within the closest distance ($d_1 < d_2$).

Adaptive and Customizable Menus

Information about selection patterns can be used to facilitate selection in menus. In split menus [21], for instance, a number of frequently selected items are moved to the top of the menu. Sear and Shneiderman [21] showed that depending on the distribution of selection frequencies, split menus can improve average selection times. The original design of split menus assumed that selection frequencies are a priori known and remain constant. In real environments, however, selection patterns may vary among users and change as user interaction and experience evolve over time. Adaptive menus in Microsoft Office made use of evolving selection patterns, but their success has been questioned [18]. Besides, Findlater and McGrenere [7] compared a static, an adaptive, and a customizable version of split menus and showed that the adaptive version was the slowest.

DESIGN GOALS AND REQUIREMENTS

The overall objective of our work was to develop a new design of hierarchical drop-down menus that accelerates menu selection. We required that the design should be able to support both user-driven customization and automated adaptation. Another requirement underlying our approach was to view adaptation as an alternative mode of interaction that could be deliberately activated by users based on

their needs. We hypothesized that if users could foresee whether and when adaptation helped or hindered their task, they would adapt their actions accordingly. We concentrated on experienced users, who, being aware of the structure of menus, can make selections without searching. As discussed in the previous section, in such cases, assistance primarily depends on improving pointing times. To eliminate any danger of destroying mental models of users regarding a menu structure, we differentiated from previous work on personalized menus [21], requiring that any adaptation should not affect the original structure of menus.

MULTIMODE POINTING

Respecting our requirement that adaptation should be available to users as an alternative mode of interaction, we explored designs of menus that support two separate views: a default static view, identical to the traditional view of menus, and an alternative view in which the selection of hot items is boosted by expanding their activation area. The advantage of this approach is that the decision of whether to adapt a menu or not is made by users depending on their evolving goals. On the other hand, its application requires that switching between views is quick so that benefits coming from adaptation are not canceled by the cost of view switching. Furthermore, it assumes that users can predict or remember whether a menu item is "considered" as hot so that they can switch views effectively. Note that remembering whether an item is hot or not is a binary decision. Therefore, it is less demanding than the use of keyboard shortcuts, which requires the remembering of a key sequence. Nevertheless, we were interested in exploring the viability of our approach in situations in which hot items are automatically chosen by an intelligent mechanism. In such cases and unless the intelligent mechanism makes perfect decisions, users have to deal with the uncertainty about whether an item has been classified as hot or not.

A simple technique that we used to relax this problem was to subtly highlight hot menu items by using a distinctive background colour. By colouring items, we intended to help users identify hot items before making any decision about switching views and, at the same time, facilitate visual search. Unfortunately, this approach cannot eliminate the problem of uncertainty. Users do not receive any visual feedback until a menu opens. Any decisions have to be made after the menu opens, possibly causing delays. According to extensive experimental work in Cognitive Psychology on response times, the cognitive cost of two-choice decision-making in response to simple visual stimuli is roughly 150-200 ms [16].

To assess the cost of mode switching and decision-making on pointing performance, we conducted a preliminary experimental study with 12 participants. The experiment tested whether benefits would be possible if users were given the chance to expand a small number of hot targets by using mode switching techniques such as dragging the mouse or pressing a modifier key. Tasks were designed as simple Fitts' Law pointing tasks on a 2-D plane. Expecting that decision-making could be partially performed in parallel with the movement of the mouse and knowing that benefits from target expansion can happen even if the target expands late in the movement [19, 23], we hypothesized that such as a *multimode-pointing* approach would be beneficial. The results, however, revealed that logarithmic benefits coming from expansion were counterbalanced by the constant costs of mode switching and decision-making even when the task *ID* was decreased by 2-3 bits. The results were discouraging but not conclusive. The experiment tested a worst-case scenario where the level of expansion was unpredictable and mode switching was expensive compared to the difficulty of the task. Selection in cascading menus may include multiple pointing tasks as well as steering motions. Moreover, cognitive costs originating from the unpredictability of target expansion could be possibly reduced by supporting appropriate visual feedback.

BUBBLING MENUS

Rather than applying target expansion to improve pointing performance, we iterated through various designs that use the bubble cursor [10]. The bubble cursor uses all the space between targets to expand their activation area, and as a result, it can result in maximal benefits. Besides, its selection mechanism is based on geometric proximity, a quantity that can be easily perceived by users. Considering the results of the preliminary study, we integrated the bubble cursor into a gesture-based interaction model that deals with costs of mode switching and improves steering in cascading menus. This section presents the design that formed the basis of our first evaluation study.

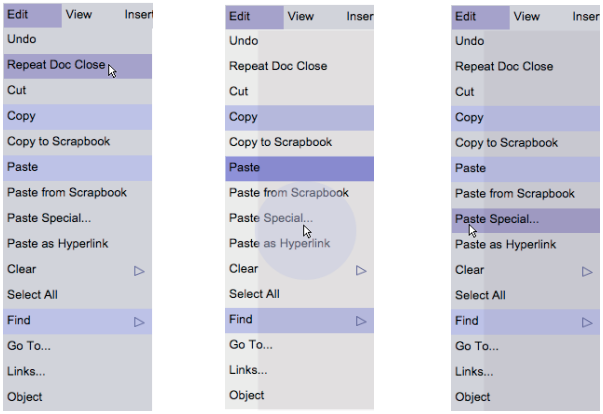


Figure 4. Bubbling menus. (a) Default view - A small number of hot items are highlighted with a light-blue color. (b) Alternative view - The user drags the mouse and a bubble cursor selects highlighted items. (c) When the cursor moves to the left sub-area of the menu, the bubble cursor disappears.

Visual and Interaction Design

Our first design of bubbling menus examines user interaction after menus are activated by pressing a menu label. As Figure 4 demonstrates, the design allows users to switch between two menu views: a default view (see Figure 4 (a)) that behaves as a regular menu; and an alternative view (see Figure 4 (b)), in which a semitransparent bubble cursor selects hot menu items. A light-blue colour is used to highlight hot items in both these views. Interaction with a bubble cursor is extended beyond the geometric boundaries of

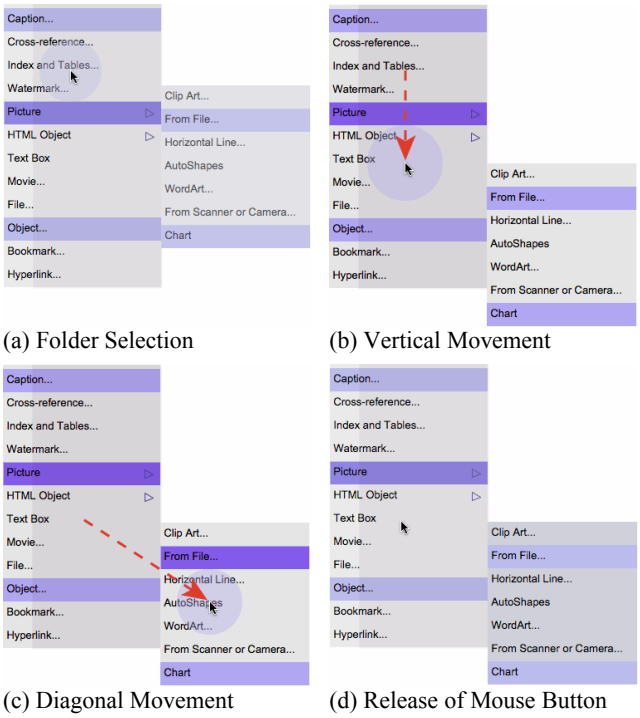
menu boxes. The bubble cursor can select a menu item even if its center is in the space that surrounds the menu. Switching between views is entirely controlled by mouse gestures. The alternative view of a menu is activated by dragging the mouse. In a typical course of interaction with the alternative view, the user presses on the menu label and without releasing the button, drags the mouse towards the goal item. A selected item can be activated by releasing the mouse button. Consequently, a sequence of actions press-drag-release is sufficient for the selection and activation of a hot menu item. The alternative view can be activated by dragging the mouse in any position within the menu. Dragging has to exceed a minimum distance, e.g., 20 pixels, before the menu switches to its alternative view.

A difficulty that we faced during the design stage was coming up with a quick mechanism that would enable users to switch an adapted menu back to its default view. The preliminary experiment stressed the importance of such a mechanism in minimizing errors and reducing the cognitive load of decision-making. We dealt with this issue by using free space surrounding a menu for switching views: while the space beneath and right to the menu is used to select items with the bubble cursor, the space above and left to the menu is reserved for error correction. If the user drags to the latter portion of the space and releases the button, the menu returns to its default view. An additional quicker mechanism is demonstrated in Figure 4 (b-c). The area of adapted menus is split into two distinct sub-areas. The right dark sub-area is dedicated to the selection of hot items with the bubble cursor. The bubble cursor disappears when the mouse enters the left sub-area (see Figure 4 (c)). Within this sub-area, any item can be activated by simply releasing the mouse button over the item's boundaries.

Clearly, benefits from the use of the bubble cursor occur only if hot items are sparsely spread along menus. Real data on usage patterns [7, 17] show that such an assumption is reasonable.

Submenus

The bubble cursor can be used to access items in any level of nesting in a menu hierarchy. Interaction with the bubble cursor moves forwards and backwards to subsequent levels of nesting. To accelerate interaction with submenus, and in addition to the use of a bubble cursor, we made a several changes to the interaction model of regular cascading menus. First, the enlargement of the activation area of menu items allowed us to remove the delay that follows the selection of a folder before its submenu is displayed. Second, we enhanced motor control by allowing submenus to follow the movement of the cursor, floating along the vertical direction. Third, we added simple interactions that allow users to switch from an adapted view to the normal view of a submenu, avoiding errors. As opposed to Cockburn and Gin [6], who removed all the delays associated with submenus, we only removed the delay after the placement of the mouse over a folder's area and before the display of the associated submenu. Following the example of Mac OS X (see Figure 2), we further enhanced motor control by allowing for direct diagonal steering movements.



(a) Folder Selection (b) Vertical Movement

(c) Diagonal Movement (d) Release of Mouse Button

Figure 5. Interaction with menu folders in bubbling menus. The user drags the mouse. Arrows show the direction of mouse movement.

Figure 5 summarizes the interaction with the alternative view of cascading bubbling menus. As soon as the bubble touches a menu folder, the associated submenu appears. The user can detect which submenu items are highlighted and decide whether to continue the motion towards the submenu or release the mouse button to activate the submenu in its regular non-adapted view (see Figure 5 (c-d)). Figure 5 (b) shows that submenus float along the vertical position of the bubble cursor's center. When the mouse moves towards the submenu though, the submenu freezes so that the user can target items without being disturbed by additional movements.

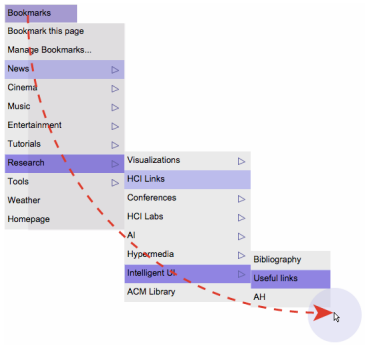


Figure 6. A nested item in a bookmarks menu is selected with a single-stroke gesture.

The above interaction model eliminates time spent with regular menus opening a menu folder by halting the cursor or clicking on it. As Figure 6 demonstrates, users can select hot items in any level of a menu hierarchy with uninteruppted single-stroke mouse gestures without having to be accurate in their movements. As oposed to first-level

menus, submenus do not provide any mechanism for selecting non-highlighted items while the user drags the mouse. Spliting submenus into sub-areas could not be applied without sacrificing the effectiveness of the design. Error correction is solely based on backtracking: the user can drag the mouse to the left of a submenu to move interaction to the previous level of nesting.

EXPERIMENTAL STUDY

To test whether adaptive versions of bubbling menus would improve targeting performance, we conducted an experimental study. As the success of an adaptive/-able UI depends on the *accuracy* of its prediction mechanism [22], the experiment examined how benefits and costs of bubbling menus balanced under both high and low accuracy levels. Here, we define accuracy as the proportion of tasks in which the path of the goal menu item has been successfully highlighted, so items can be selected by mere dragging.

Experimental Conditions

The experiment compared bubbling menus against regular static menus. Interaction with static menus followed the interaction model supported by modern operating systems:

- A menu item was selected after releasing the mouse button over its boundaries; i.e., an item could be selected by either clicking or by dragging and releasing.
- A submenu was activated by either clicking on its parent item or by keeping the cursor over its boundaries for a short period of time (300 msec).
- Items within a submenu were selected with respect to the interaction model demonstrated in Figure 2. The time window in which motion was constrained by the triangular area shown in the figure was set to 400 msec.

Participants were free to use the selection strategy that best fitted their past experience. The same interactions were supported by the default view of bubbling menus with the exception that dragging for more than 20 pixels activated their adaptive views.

Apparatus

The experiment was conducted on a PowerBook G4 12-inch laptop with screen resolution 1024x768 and 768 MB RAM, running Mac OS X 10.4.4. A USB mouse was used as input device. The experimental software was implemented in Java 1.4.2.

Participants

Six female and ten male volunteers, 24-35 years old, participated in the experiment. All the participants had experience interacting with pull-down menus and a mouse.

Task

Participants completed a series of menu selection tasks. For each task, they had to select an item that appeared either at the second or third level of nesting. Selections were made from four different menu categories. As shown in Figure 7, a label "CLICK ME" guided the selection (not necessarily by clicking) of menu items. A task started as soon as the participant pressed on the label of the menu and finished when the goal item was successfully selected.

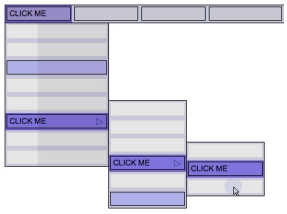


Figure 7. Demonstration of the experimental task.

The experimental trials were structured in blocks of 35 randomly ordered tasks. These 35 tasks were variations of the ten base tasks shown in Figure 8. Base tasks were cases where system suggestions were perfect, i.e., suggestions always included the "CLICK ME" item. In their variations, suggestions were imperfect. For the static menus, there were no suggestions, and as a result, variations were identical to the base tasks. Tasks 1-5 required the selection of a second-level item. There was one variation for each of these tasks in which suggestions were wrong at the second level, and one variation where suggestions were wrong from the first level. Tasks 6-10 required the selection of a third-level item. For each of these tasks, there were three variations in which suggestions were wrong from the third, the second, or the first level. Suggestions were constrained as follows: menus with 1-4 items had one suggestion; menus with 5-9 items had two suggestions; and menus with 10-14 items had three suggestions. A menu did not include any suggestions if its parent item was not suggested.

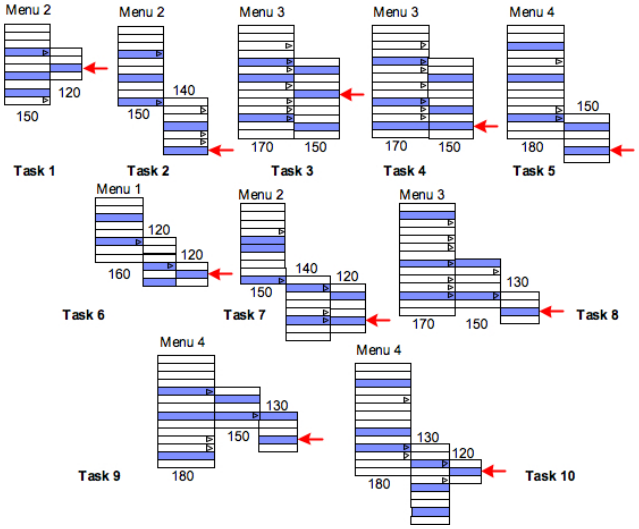


Figure 8. Base experimental tasks. Goal items are marked by an arrow. Numbers specify the width of menus in pixels. The height of items was 30 pixels.

Design and Procedure

A mixed factorial design was used. Accuracy of menu suggestions was treated as a between-participants factor. Participants were split into two groups with equal sizes. Each group was exposed to a different accuracy level. For the first group, the base tasks and all their variations appeared with the same probability, and as a result in only 10 out of the 35 tasks (28.6%) of each block suggestions were perfect. For the second group, the base tasks accounted for 30 out of the 35 tasks of each block, so accuracy was 85.7%.

Real menu selection data on one-level menus reported by Findlater and McGrenere [7] show that selection patterns allow for predictions with accuracy levels over 90% even if simple heuristics such as selection frequencies are used and a small number of items, e.g. 3 out of 15, is suggested.

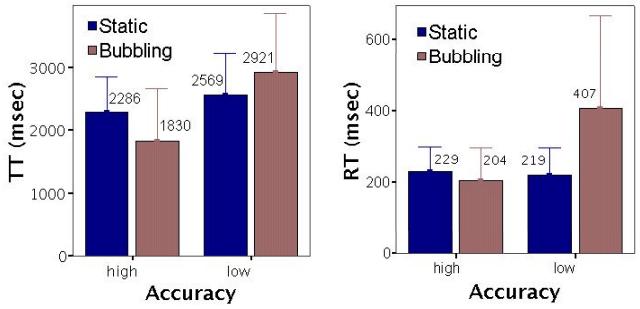
Tasks were randomly ordered within each block. In other words, we simulated a worst-case scenario of adaptive behaviour according to which participants could not predict how menus were adapted before starting a task. Each participant was exposed to both bubbling menus and static menus. The design can be summarized as follows:

> 2 accuracy conditions (*low*, *high*) × 8 participants × 2 techniques (*static*, *bubbling*) × 4 blocks × 35 tasks = 4480 trials in total.

As bubbling menus was a new technique to which participants were not previously exposed, we tried to minimize the learning curve by including training sessions adapted to the individual needs of each participant. Also, for both techniques, before the four main blocks, we added an extra block not included in the analysis. For this block only, participants were instructed to start completing the tasks slowly but accurately and accelerate as soon as they felt confident. For the four main blocks, they were asked to complete the tasks as fast as possible avoiding errors. Each participant completed the experiment in one session lasting 45-60 minutes. The order in which the two menu techniques were presented was balanced among participants.

Measures

We analyzed the total time *TT* to complete a task as well as the response time *RT* measured from the beginning of the task until the cursor entered the menu. We also conducted a separate analysis for the total time ($TT_{perfect}$) needed to complete tasks for which system suggestions were correct.



(a) Mean Total Times (b) Mean Response Times

Figure 9. Overall results. Error bars represent standard deviations.

Results

Errors due to wrong item selections or accidental collapses of first-level menus were removed from our analysis. Error rates were 1.68% for static menus and 3.41% for bubbling menus. Although higher, the error rate for bubbling menus is reasonable if taking into consideration the novelty of the technique and the unpredictability of suggestions. Besides, this error rate was considerably lower (2.59%) for the high-accuracy condition compared to its value (4.22%) for the low-accuracy condition.

Figure 9(a) demonstrates mean times as measured for the two accuracy conditions. Bubbling menus improved mean selection speed by 20% when accuracy was high. However, they reduced mean performance by approximately 14% when accuracy was low. An ANOVA analysis with accuracy treated as a between-participants variable and technique, block, and task treated as repeated measures showed that the main effect of accuracy on *TT* was statistically significant ($F_{1,14}$=35.308, p<.0001). Its interaction effect with technique was also significant ($F_{1,14}$=36.846, p<.0001). A post hoc comparison using Bonferroni's adjustment showed that bubbling menus significantly improved performance (p=.0003) in high accuracy and significantly decreased performance (p=.002) in low accuracy. No significant learning effects were found, i.e., the main and interaction effects of the block variable were not found to be significant. The poor performance of bubbling menus in the low-accuracy condition can be partially explained by the inflation of response times as shown in Figure 9(b). The results indicate that participants who experienced a low accuracy tended to start moving the mouse after thinking about whether to drag the mouse or not. A deeper, however, analysis of the results showed that response times were not uniformly distributed among these participants. Different participants followed different strategies, which explains the great variance shown in the figure. Figure 10 shows that some participants exposed to the low-accuracy condition did not get any benefit from the use of the bubbling menus even when suggestions were perfect.

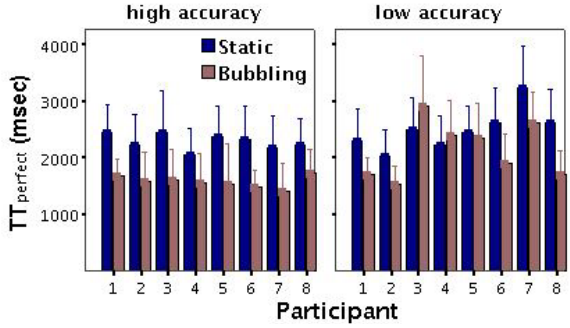


Figure 10. Performance for each of the sixteen participants when suggestions were perfect.

As shown in Figure 11, low accuracy did not only hurt overall performance. It delayed menu selections even when suggestions were perfect. This result is consistent with previous research on adaptive user interfaces [22], indicating that as user trust over automation [15] declines, correct adaptations become less effective. An ANOVA analysis applied on the ten base tasks of each block showed a significant main effect of accuracy ($F_{1,14}$=9.68, p=.008), and a significant main effect of technique ($F_{1,14}$=34.27, p<.0001) on *TT*. Their interaction effect was not observed to be significant though ($F_{1,14}$=3.23, p=.094). As shown in Figure 11, bubbling menus were particularly effective when targets appeared in 3rd level menus. Selections in 3rd level menus involve two steering motions, and therefore, the strengths of the technique became more apparent.

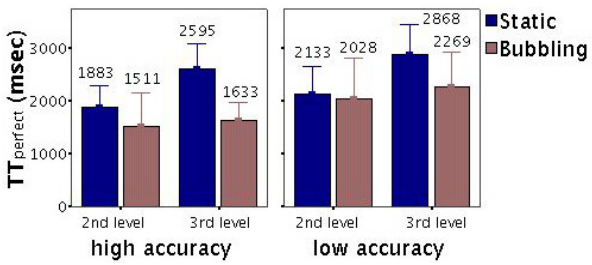


Figure 11. Performance for perfect suggestions shown separately for 2nd and 3rd level menus.

Subjective User Feedback

Most participants stated that bubbling menus helped them access suggested items faster and commented that they would use the new design as long as suggestions were generally accurate. Several participants were very enthusiastic about their use. On the other hand, one participant was very negative. Exposed to the low-accuracy condition, he felt that interaction with bubbling menus added unnecessary complexity to a rather simple task. Another participant, also exposed to the low-accuracy condition, mentioned that sometimes, she had to "be attentive to too many details", which was "time-consuming". Note, however, that the experiment tested a worst-case scenario where suggestions were completely unpredictable. As a participant commented "if this was an actual system I used, it is very likely that I would know which items are highlighted and which are not. In that case I would choose my strategy before starting the action". Several participants disliked the fact that splitting menus into sub-areas was only applied to first-level menus. Although they found the mechanism useful, they felt that it was not applied consistently.

Supplementary User Study

Results by Cockburn and Gin [6] suggest the hypothesis that removing delays from the interaction with submenus might improve performance even if activation areas were not increased. If this was true, someone could attribute gains shown by our results to the elimination of such delays. To clarify this issue, we conducted a small study with four participants that compared regular menus as tested in our main experiment against menus with no delays. We used the same experimental setup but kept only two blocks per condition. The results rejected the above hypothesis. They indicated that merely removing delays deteriorates motor control and can severely hurt the performance of some users. Furthermore, all the four participants preferred the original version of menus that preserved delays.

EXTENDED DESIGN

We have shown that bubbling menus result in performance benefits even when suggestions are unpredictable as long as the accuracy of suggestions is relatively high. Trying to maximize the benefits of our approach, we extended the design so that selection gestures can start far from a menu bar. The extended design is demonstrated in Figure 12. Hot menu items can be selected with single-stroke gestures starting from any position on the screen. Again, dragging the mouse to activate the bubble cursor is optional, initiated by users based on their own intentions and needs.

Figure 12. The extended design of bubbling menus. (a) When the user drags the right mouse button, a bubble appears that selects highlighted menu categories. A preview of the corresponding menu follows the cursor as the cursor's center moves towards the menu bar. (b) The user moves the mouse downwards while dragging. The position of the menu freezes and a second bubble allows for the selection of highlighted items within the menu. (b) Alternatively, the user can release the button to activate the default view of a menu.

As shown in Figure 12, the new design makes use of two bubble cursors activated at different stages. The outer bubble cursor selects highlighted menu categories from a menu bar. The nested bubble cursor selects items within menus like in the initial design. The first bubble cursor is activated after dragging the mouse for a small distance (20 pixels) while pressing the right button. This interaction enables its direct application to a wide range of applications in Microsoft Windows, where contextual menus are activated after the right mouse is released. In Mac OS X, contextual menus are activated when the right button is pressed. However, its integration would be feasible with minor changes, e.g., by adding a brief delay before the activation of contextual menus, so that sensing a dragging motion is possible.

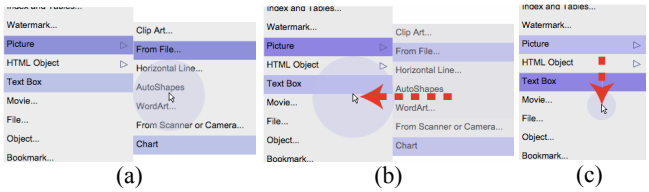


Figure 13. Improved backtracking. (a) The user enters a submenu while dragging. (b) If moving to the back level, the bubble selects the activated folder item ("Picture") rather than selecting the nearest highlighted item ("Text Box"). (c) The user slightly moves the mouse vertically to update the selection.

The outer bubble cursor is enhanced with menu previews that follow the movement of its center. Such previews are always fully visible even when dragging starts near the bottom of the screen. Expecting that menu previews would reduce the need for error corrections and taking into consideration participants' feedback, we decided to remove the left sub-area of first-level menus and keep the same interaction model for all the levels of nesting. We also improved the backtracking mechanism. Some participants complained that sometimes, an active submenu unexpectedly

collapsed when they backtracked, planning to select a non-highlighted item within the submenu. Our solution, shown in Figure 13, addresses this problem by freezing the selection of a folder item when moving back to a previous level.

EVALUATION

A usability study was conducted to evaluate the extended design of bubbling menus. This user study provided mostly qualitative data about strengths and weaknesses of our approach in a more realistic setting. Participants performed common menu selection tasks using a simulation of the menu structure of Microsoft Word 2004 for Mac (MSWord). Rather than testing an adaptive version of bubbling menus, we evaluated a customizable version, in which users manually highlighted items. The evaluation had three main objectives: (1) to establish useful criteria and strategies of customization/adaptation based on needs of various users; (2) to test how users would take advantage of bubbling menus when interacting with a familiar menu structure; and (3) to collect feedback about the usability and potential of our approach.

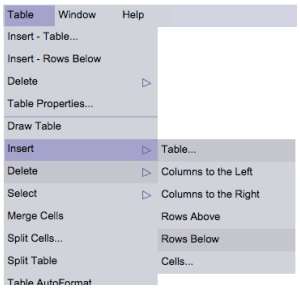


Figure 14. Our version of cascading split menus. Hot items can be copied to the top area of the menu from any level of nesting. The original copy of these items is slightly grayed.

Techniques

To better satisfy our third objective, we contrasted our design against a design of customizable split menus. The version of split menus that we tested (see Figure 14) extends the original design of split menus [21] by permitting the placement of submenu items in the top section of customized menus. Following the suggestion of Gajos et al. [9], we changed the original design so that items are copied rather than moved to the top section. This approach is less intrusive as it does not prevent users from accessing the original structure of menus. Finally, we used a customization mechanism similar to the one proposed by Findlater and McGrenere [7], but instead of using virtual buttons attached to the menu, users pressed the Page Up/Page Down keys while hovering over an item to control the item's position. Bubbling menus were customized by pressing the spacebar key, which caused items to get highlighted or return to their original state.

Apparatus and Participants

The apparatus of the previous study was used. Six volunteers participated. The background of participants is as follows: two Ph.D. students in Computer Science (females, 29 and 30), a Master's student in Architecture (male, 24), a professional engineer and programmer (male, 33), a high-school teacher in Physics (male, 31), and a civil servant (male, 39). All the participants were users of MSWord.

Procedure

At first, participants were given to complete a questionnaire about their familiarity with MSWord and about strategies that they used to activate commands in office applications. Then, participants were presented the menu structure of MSWord and were asked to freely explore it for 2-3 minutes. To test their familiarity with the menus, the experimenter asked them to locate and activate specific commands. After this step, participants were introduced to the two techniques. Order of exposure to the techniques was balanced among participants.

For each technique, the following procedure was followed. Initially, participants were explained the selection and customization mechanism supported by the technique. Then, they were asked to customize the menus based on their personal needs while thinking aloud. Lastly, they were asked to complete two tasks. The first task acted as a training session, allowing participants to develop their customization and selection strategies. Each task had three steps. First, participants completed 38 menu selection trials without using the technique. The order of trials simulated sequences of common commands needed to complete realistic tasks such as inserting and formatting pictures and tables. Second, participants were asked to customize the menus according to their experience from the first step. They were also asked to justify their customization strategy. Third, they were asked to complete 50 menu selection trials. 94% of these trials asked for items appearing in the first step with a similar frequency. 70-75% of the trials were first-level menu selections and the rest were second-level menu selections. To start a trial, participants placed the cursor over a small red box, appearing at various positions on the screen. Participants were instructed to follow strategies that would best facilitate their tasks without rushing. The use of the customization and selection mechanisms supported by the techniques was optional. At the end, participants were given a questionnaire to evaluate the two techniques and rank them against traditional menus. Evaluation sessions lasted from 80 to 120 minutes.

Results

Familiarity with MSWord menus varied across participants. Two participants (Architecture student and civil servant) spent time searching before locating several commands. The rest could select nearly all the commands required by the tasks without any searching. The following paragraphs summarize our results. The summary is based on data recorded in log files, notes taken by the experimenter during the sessions, and answers given to the questionnaires.

Customization Strategies

Although frequency of use was the primary criterion of customization for all the participants, customization strategies varied greatly among them. For split menus, the maximum number of items copied to the top section of a menu ranged from four up to seven items. Participants' comments indicated four distinct strategies used to sort

items within the top section of a split menu: (1) sorting items according to the frequency of their selection from top to bottom; (2) preserving the original order of items; (3) grouping items based on logical relationships, e.g., keeping Cut, Copy and Paste together; and (4) ignoring order. Coping nested items to the top section was a common strategy. According to two participants, replicating frequently selected items that were either nested or appeared near the bottom of a long menu was particularly useful.

For bubbling menus, the maximum number of highlighted items in a menu ranged from four to ten. Proximity between menu items did not seem to determine the customization patterns of participants. Neighbouring items such as Undo, Cut, and Copy were all highlighted as they were all used. A participant explained that he chose to highlight all the items belonging to frequent command sequences. This strategy helped him to easily remember how items had been customized and minimized the need for switching between different selection techniques. Various strategies were used for customizing the menu bar. Two participants highlighted all the menu categories as long as they were selected at least once throughout the task. Other participants did not highlight menu categories if they did not contain a minimum number of frequently selected items, e.g., more than two items. A participant observed that highlighting both the File and Edit menus reduced the effectiveness of the bubble cursor. He explained that File was an important menu but its items were less frequently selected. Therefore he preferred highlighting only the Edit menu.

Finally, two participants stated that if hotkeys were available, they would use the customization mechanisms only for commands that were not usually selected by hotkeys.

Selection Patterns

The average probability that a goal item had been included in the top section of a split menu was 68%. In bubbling menus, the average probability that a goal menu category or a goal menu item had been highlighted was 87% and 73%, respectively. Error rates due to incorrect selections were 1.7% for split menus and 2.6% for bubbling menus.

Surprisingly, one participant (civil servant) did not use any dragging gestures to interact with the bubbling menus. He explained that the use of bubbling menus increased the mental load required to complete selection tasks. He kept, however, customizing the menus because, as he explained, highlighting improved visual search. The other five participants used the technique heavily. On average, in 80% of the trials, users activated the outer bubble cursor. Also, in 70% of the trials, the goal command was selected with the bubble cursor. Besides, results indicate that the five participants remembered how menus had been customized and used the bubbling menus selectively. More specifically, the probability that highlighted goal items were not selected with the bubble cursor was only 4%. Also, the probability that a bubble cursor was falsely activated to select a non-highlighted goal item under a non-highlighted menu category was 12%. Overall, for these five participants, the alternative view of bubbling menus was falsely activated in approximately 7% of the total number of trials. The same participants "missed" to activate the bubble cursor in approximately 3% of the total number of trials.

Preferences

Three participants, a computer scientist, the Physics teacher, and the professional engineer, ranked bubbling menus as their first choice, split menus as their second choice, and traditional menus as their last choice. According to the first participant, bubbling menus "allow eyes-free selection and gracefully deal with more items than split menus". The second participant stated that bubbling menus "are not very easy to learn but when you do learn they are very fast in use". He also noted that "you don't have to be very accurate with the mouse", as the activation area of menu items is larger than in normal menus. The third participant commented that bubbling menus would be more appropriate for expert users. He explained that they better supported selection speed, whereas split menus might be more appropriate for menu-browsing tasks as opposed to goal-oriented tasks.

The second computer scientist and the civil servant ranked split menus as their first choice followed by bubbling menus. The former explained that split menus were faster than traditional menus and "less problematic to control" than bubbling menus. She noted, however, that bubbling menus did "not require accurate motor control" and if she had "mastered" the technique, bubbling menus might have been ranked as a first choice. Finally, the Architecture student ranked split menus as his first choice followed by traditional menus. He found that bubbling menus were sometimes "confusing" stating that "speed is the strength of the mechanism, but it needs awareness".

Participants were asked whether they would prefer a different version of split menus were items would be moved instead of being copied to the top section. Five out of the six participants preferred our version of split menus because it supported memorization and allowed them to ignore the top section. On the other hand, the sixth participant observed that copying instead of moving menu items overloaded menus with redundant information.

CONCLUSIONS AND FUTURE DIRECTIONS

We have presented bubbling menus, a new design for pull-down cascading menus. Bubbling menus accelerate the selection of a subset of menu options by increasing their activation areas. We have presented two user studies that evaluated the proposed design. The first study experimentally tested a version of adaptive bubbling menus. Bubbling menus were shown to significantly accelerate the selection of nested menus when adaptation accuracy was high (approximately 86%). On the other hand, performance deteriorated when accuracy became low. Nevertheless, the experiment tested a worst-case scenario, in which adaptation was unpredictable. Besides, participants were instructed to select highlighted items with the bubble cursor as frequently as possible. As participants were uncertain about how menus had been adapted, they had to spend time deciding about their selection strategies and frequently switch

between interaction modes. In real environments, usage patterns change slowly and semi-static predictable adaptation schemes could be applied. Bubbling menus do not disrupt the order of items in a menu and do not enforce the use of dragging gestures. We hypothesize that in low-accuracy environments, during, for instance, the learning stage of a classification mechanism, users would choose not to activate the adaptive view of menus. As soon as they anticipated that the system had learnt their selection patterns, they would (optionally) take advantage of the technique according to their needs. As shown by our second evaluation study, users can activate the bubble cursor selectively or even ignore it when they feel that it hinders their task.

Results of the second user study have indicated that bubbling menus may be more appropriate for expert users who having memorized the structure of menus can make selections with quick single-stroke gestures. We expect that if customization remains constant over time, expert users can use bubbling menus in a fashion similar to using marking menus [12]. On the other hand, we recognize that some participants found that bubbling menus were harder to use than traditional and split menus. A common difficulty that participants encountered was using the backtracking mechanism to cancel the bubble cursor. We have started exploring more intuitive view-switching mechanisms such automatically canceling the bubble cursor when the cursor's center is halted over a menu option. We believe that such mechanisms can reduce the cognitive load associated with decision-making and error correction.

As future work, we are particularly interested in applying the approach to pen-based interfaces, where the absence of a keyboard disallows the use of hotkeys. We also plan to extend the approach to other UI widgets such as contextual menus and toolbars. We envision desktop environments where users smoothly switch between different contexts of interaction, i.e., different customized views of the elements of a UI, without disrupting the structure of the space.

REFERENCES

1. Accot, J. and S. Zhai (1997). Beyond fitts' law: Models for trajectory-based HCI tasks. *ACM CHI*. 295-302.
2. Ahlstrom, D. (2005). Modeling and improving selection in cascading pull-down menus using fitts' law, the steering law and force fields. *ACM CHI*. 61-70.
3. Baudisch, P., E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger (2003). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. *INTERACT*. 57-64.
4. Blanch, R., Y. Guiard, and M. Beaudouin-Lafon (2004). Semantic pointing: Improving target acquisition with control-display ratio adaptation. *ACM CHI*. 519-525.
5. Cockburn, A. and A. Firth (2003). Improving the acquisition of small targets. *British HCI Confer*. 181-196.
6. Cockburn, A. and A. Gin (2006). Faster cascading menu selections with enlarged activation areas. *Graphics Interface*. 65-71.
7. Findlater, L. and J. McGrenere (2004). A comparison of static, adaptive, and adaptable menus. *ACM CHI*. 89-96.
8. Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6): 381-391.
9. Gajos, K.Z., M. Czerwinski, D.S. Tan, and D.S. Weld (2006). Exploring the design space for adaptive graphical user interfaces. *ACM AVI*. 201-208.
10. Grossman, T. and R. Balakrishnan (2005). The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. *ACM CHI*. 281-290.
11. Kobayashi, M. and T. Igarashi (2003). Considering the direction of cursor movement for efficient traversal of cascading menus. *ACM UIST*. 91-94.
12. Kurtenbach, G. and W. Buxton (1993). The limits of expert performance using hierarchic marking menus. *Proceedings of InterCHI `93*. 482-487.
13. Landauer, T.K. and D.W. Nachbar (1985). Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width. *ACM CHI*. 73-78.
14. Lane, D.M., A.H. Napier, C.S. Peres, and A. Sandor (2005). The hidden costs of graphical user interfaces: The failure to make the transition from menus and icon tool bars to keyboard shortcuts. *International Journal of Human-Computer Interaction*, 18(2): 133-144.
15. Lee, J.D. and K.A. See (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46 (1): 50-80.
16. Luce, R.D. (1986). Response times and their role in inferring elementary mental organization: Oxford University Press.
17. McGrenere, J. and G. Moore (2000). Are we all in the same "Bloat"? *Graphics Interface*. 187-196.
18. McGrenere, J., R.M. Baecker, and K.S. Booth (2002). An evaluation of a multiple interface design solution for bloated software. *ACM CHI*. 163-170.
19. McGuffin, M. and R. Balakrishnan (2002). Acquisition of expanding targets. *ACM CHI*. 57-64.
20. Norman, K.L. (1991). The psychology of menu selection: Designing cognitive control at the human/computer interface: Alex Publishing Corporation.
21. Sears, A. and B. Shneiderman (1994). Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, 1(1): 27-51.
22. Tsandilas, T. and m.c. schraefel (2005). An empirical assessment of adaptation techniques. *ACM CHI, Extended Abstracts*. 2009-2012.
23. Zhai, S., S. Conversy, M. Beaudouin-Lafon, and Y. Guiard (2003). Human on-line response to target expansion. *ACM CHI*. 177-184.