A Predictive Model of Menu Performance

Andy Cockburn
Computer Science
University of Canterbury
Christchurch, New Zealand
andy@cosc.canterbury.ac.nz

Carl Gutwin
Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
gutwin@cs.usask.ca

Saul Greenberg
Computer Science
University of Calgary
Calgary, Alberta, Canada
saul@cpsc.ucalgary.ca

ABSTRACT

Menus are a primary control in current interfaces, but there has been relatively little theoretical work to model their performance. We propose a model of menu performance that goes beyond previous work by incorporating components for Fitts' Law pointing time, visual search time when novice, Hick-Hyman Law decision time when expert, and for the transition from novice to expert behaviour. The model is able to predict performance for many different menu designs, including adaptive split menus, items with different frequencies and sizes, and multi-level menus. We tested the model by comparing predictions for four menu designs (traditional menus, recency and frequency based split menus, and an adaptive 'morphing' design) with empirical measures. The empirical data matched the predictions extremely well, suggesting that the model can be used to explore a wide range of menu possibilities before implementation.

Author Keywords

Menus, Hick-Hyman Law, Fitts' Law, performance modelling, adaptive behaviour.

ACM Classification Keywords

H5.2 [User Interfaces]: Interaction styles.

INTRODUCTION

Menus are one of the primary controls for issuing commands in graphical user interfaces. Commensurate with their importance, there has been a great deal of research into alternative designs, including marking menus [23], keyboard shortcuts [19], split menus [31], adaptive menus [10], tracking menus [12], cascade improvements [2], and fisheye menus [4].

While the empirical work evaluating menus is strong, there has been less theoretical work attempting to predict the efficiency of alternative designs. When models have been proposed, they have been based on the Fitts' Law pointing requirements and thus ignore the time taken to find the item in the menu [2], or they ignore the users' skill development by addressing only expert [25] or novice [5] behaviour. These problems arise because the theories have not adequately taken the *Hick-Hyman* Law into account. This law models human decision time as a function of the information content conveyed by a visual stimulus [16, 20]. Although other models such as Fitts' Law have been rigorously studied in HCI, the Hick-Hyman Law has failed to gain momentum despite its value for modelling the time taken to choose a command or action.

In this paper, we propose a model of menu selection time that incorporates both the Hick-Hyman and Fitts' Laws, and that integrates a transition from novice to expert performance. As we will see, the model accommodates a variety of alternative menu designs, both adaptive and non-adaptive. We conducted an experiment to calibrate several parameters in our model, and we then used the model to predict performance with four menu types: traditional menus, recency-based split menus (widely used in commercial software), frequency-based split menus, and a novel 'morphing' design. Finally, we conducted an experiment to compare the predictions with actual performance. The empirical data matched the model's predictions extremely well.

This experience shows the model's potential for comparing existing menu designs, and for predicting the expected performance of new menu designs. For example, the 'morphing' menu to be described in this paper is an intuitively appealing design: it reduces pointing time for frequent items by gradually enlarging them. However, our model predicted that morphing menus would give very little advantage over traditional menus; this prediction was subsequently confirmed by an empirical analysis of its use. These and other results suggest that designers could use the predictive model to evaluate a larger number of menu designs without having to implement them or carry out extensive user trials.

BACKGROUND

Hick-Hyman and Fitts' Laws

Our model integrates two well-established laws of human behaviour. The Hick-Hyman Law [16, 20] describes human decision time as a function of the information content conveyed by a visual stimulus. Fitts' Law [11] describes the movement time taken to acquire, or point to, a visual target. Both Hick-Hyman and Fitts' Laws are derived from information theory [33], where the

Cockburn, A., Gutwin, C. and Greenberg, S. (2007)
A Predictive Model of Menu Performance. Proc ACM CHI'07 Conference on Human Factors in Computing Systems. (Apr 28-May 3,San Jose,CA,USA). 10 pages

information content H of an event, measured in bits, is inversely proportional to its probability – likely events have low information content; unlikely ones, high. The formula for information content is: $H=log_2(1/p)$, where p is the probability of the event.

The Hick-Hyman Law, then, states that the time T to choose an item is proportional to its information content, giving $T=a+b\times H$, where a and b are empirically derived constants. When the user chooses between C equally probable alternatives, the Hick-Hyman Law can be re-written as $T=a+b\times log_2(C)$.

Similarly, Fitts' Law is based on the amount of information that a person must transmit through their motor system to move to an item – small, distant targets demand more information than large close ones, and consequently take longer to acquire. The movement time MT taken to acquire a target is modelled by the formula $MT=a+b\times ID$, where the 'index of difficulty' $ID=log_2(A/W+1)$, and A is the amplitude of movement and W is the target width.

Fitts' Law has been heavily used in HCI research, largely because many expert tasks require low-level object selection of graphical screen elements or of physical keys. However, the Hick-Hyman Law has failed to gain widespread use. Seow [32] provides a recent analysis and review of Hick-Hyman and Fitts' Laws in HCI, observing that while Fitts' Law has been robustly applied across many experiments, attempts to model behaviour with the Hick-Hyman Law have been less successful. For example, Soukoreff and MacKenzie [35] tried to fuse Hick-Hyman and Fitts' Law to predict performance with visual keyboards, but their later empirical work refuted the model [27]. Sears et al. [30] claim that Soukoreff and MacKenzie's model failed because it used the Hick-Hyman Law to predict visual search time, which is not modelled by the law – Hick and Hyman's original experiments timed *decision time* in response to clear visual stimuli.

Hoffmann and Lim [17] also attempted to fuse Hick-Hyman and Fitts' models, with limited success in an abstract decision plus pointing task. However, their model is not directly applicable to typical computer use.

Menu studies and performance models

Although scarce when compared to the extensive developmental and empirical work on menus, there has been some notable work on menu performance models, which we review here.

GOMS/KLM. Card, Moran and Newell [8] provide the seminal work on user interface modelling. John and Kieras [21] present an excellent update review of GOMS and KLM research in HCI, including models of expert menu use (e.g. [25]). GOMS and KLM models are limited for two reasons: they are confined to expert performance of routine tasks, and they use a 1.35s average time for mental preparation, which is crude compared to the precision accessible through the Hick-Hyman law.

Evidence of Fitts and Hick-Hyman together. Sears and Shneiderman [31] present a simple model of menu performance. They observed that selection times degraded logarithmically with menu length for frequently selected items, but linearly with infrequent ones. Although they did not use the Hick-Hyman Law to explain the observation, we conjecture that their participants were moving from linear visual search with unfamiliar items to Hick-Hyman decision times as the locations were learnt. Landauer and Nachbar [24] did observe that expert performance in hierarchical full-screen menu selections is well described by Hick-Hyman and Fitts components applied in series. Their tasks explicitly emphasised decision time over visual search by having well practiced participants select an integer or word from among 4096 candidates. Each menu presented between 2 and 16 sub-divided ranges of integers or alphabetically ordered words, requiring navigation through between 3 and 12 levels of hierarchy. Although seminal work, the deep hierarchies analysed, the total ordering of menu labels and the full-screen menus are all highly dissimilar to the menus used now, more than two decades after their study.

Visual search and spatial memory. Prior work has shown that users quickly form and rely on spatial knowledge of menu item location [22, 34]. Consequently, for expert users there is little visual search involved in menu selection, reducing the task time to Hick-Hyman decision time plus Fitts' pointing time (both log-based, consistent with the observations of Sears and Shneiderman [31]). Novices, however, who have not yet formed a spatial model must visually search the menu for the target item. While there have been conflicting theories and cognitive models of how users search menus – randomly [7], linearly [26], and in parallel [18] – eye-tracking data supports a predominantly top-to-bottom search order [6]. Regardless of how users conduct their search, there is consistent empirical evidence that novices' search time is linear with menu length [18, 29, 31].

Zipf distribution. A model of menu performance must accommodate the fact that the frequency of item selection is non-uniform. It is well established that the frequency of command use follows a Zipfian distribution [9, 10, 13, 15] – that is, a power-law function $P_n \sim 1/n^a$, where P_n is the frequency of the n^{th} ranked word, and a is close to 1. To quickly verify the distribution of menu selections, we analysed Microsoft Word menu-use data from two sources: our own records (one user, three months) and records from Findlater and McGrenere (one user, 20 weeks) [9, 10, 13, 15]. Regression analysis showed that both datasets are strongly Zipfian for 'File' and 'Insert' menu selections (R^2>0.92), and 'Format' menu selections fit reasonably well (Findlater and McGrenere R^2=0.69, ours R^2=0.83).

ACT-R/PM. Byrne [5] and Anderson, Matessa and Lebiere [3] use the ACT-R/PM cognitive architecture to predict performance in random menu contents. Their models, however, are predominantly concerned with how users visually search for menu targets.

While this previous work hints at the relation between the Fitts and Hick-Hyman Laws, they do not address the user's transition from novice to expert behaviour – that is, that search time in terms of menu length [24] shifts from linear to logarithmic as expertise increases.

A NEW MODEL OF MENU PERFORMANCE

We propose a model of menu performance that integrates both the time to find the item and the time to move to the target as components of performance. It also incorporates the user's gradual transition from linear visual search to logarithmic decision time as they gain expertise.

Given a menu with multiple items, our model predicts selection time T_i for each item *i*, based on the sum of the decision/search time T_{dsi} and the Fitts' Law pointing time T_{pi} for that item:

$$T_i = T_{dsi} + T_{pi} \qquad \textit{Equation 1.}$$

As we are ultimately interested in predicting the average performance of entire menu widgets, rather than just individual selections, we generalise the model to average performance (T_{avg}) using:

$$T_{avg} = \sum_{i=1}^{n} p_i T_i \qquad \textit{Equation 2.}$$

where p_i is the probability of item *i* being selected (further described below), and *n* is the number of menu items. In the next sections, we detail the calculation of the two components of the model, T_{pi} and T_{dsi}.

Item pointing time – T_{pi}

The pointing time component for each item, T_{pi}, uses the traditional Fitts' Law model:

$$T_{pi} = a + b\log_2(A_i / W_i + 1) \qquad \textit{Equation 3.}$$

where A_i and W_i are the amplitude of movement to and width of target item *i*, and *a* and *b* are constants (discussed further in the Calibration section below).

Item decision/search time – T_{dsi}

When users first encounter a menu they must visually search for the target item regardless of the particular menu design. As they become more experienced with the menu they can increasingly rely on spatial location memory, reducing task time from linear search to Hick-Hyman decision, but only if the interface allows them to do so through predictable and stable item placement.

Our model calculates the decision/search time (T_{dsi}) by interpolating between a linear visual search-time component (T_{vsi}) and a logarithmic Hick-Hyman decision time (T_{hhi}) component:

$$T_{dsi} = (1 - e_i)T_{vsi} + e_i T_{hhi} \qquad \textit{Equation 4.}$$

where e_i is the user's expertise with that item, from 0 (complete novice) to 1 (complete expert). The equation reflects the fact that as users become familiar with the items in the menu, their visual search time decreases towards zero, and the Hick-Hyman decision time dominates.

Expertise

We use two factors to model the user's expertise e_i with menu item *i*: t_i, the number of previous trials (selections) of the item and *L*, the 'learnability' of the interface.

$$e_i = L \times \left(1 - 1/t_i\right) \qquad \textit{Equation 5.}$$

Expertise e_i ranges from 0 to 1, with 1 indicating that the user is entirely expert. *L* also takes values in the range 0 to 1, with 1 representing an entirely learnable menu representation – that is, the items do not change locations or positions. The value of *L* can be estimated for different interfaces by calculating one minus the average distance that items move as a proportion of half of the total menu length – e.g., random items will on average move half of the menu length *l* per selection, hence *L=1-0.5l/0.5l=0.*

This measure of expertise is important because it is used to model the users' transition from novice visual search to expert Hick-Hyman decision, and also because it shows that some menu designs can never reach Hick-Hyman performance levels because of poor learnability.

Note that this model of expertise is limited in that it makes no allowance for the number of items in the menu; in reality, users are likely to need more trials to become expert with longer menus (we will consider this in our later discussion of future work).

Item probabilities

Our model assumes that all items are initially equally probable ($\forall i$, $p_i=1/n$), but that their probabilities are continually modified to reflect the number of times each item is selected ($\forall i$, $p_i=t_i/t_t$, where t_t is the total number of selections in the menu). Thus, the model can reflect Zipfian or other frequency distributions.

Visual search time – T_{vsi}

We assume that when the user is inexperienced, the visual search time for each item is linear with the total number of items *n* (consistent with prior work [18, 29, 31]), and that the search time is negligible when the user is expert. Thus, we model the visual search time for each item with:

$$T_{vsi} = b_{vs} n + a_{vs} \qquad \textit{Equation 6.}$$

where a_{vs} and b_{vs} are empirically determined constants.

Hick-Hyman decision time – T_{hhi}

Once expert, decision time is dependent on the entropy of each item $H_i=log_2(1/p_i)$. Hence, decision time for each item is given by:

$$T_{hhi} = b_{hh} H_i + a_{hh} \qquad \textit{Equation 7.}$$

where a_{hh} and b_{hh} are empirically determined Hick-Hyman decision constants. See the Calibration section for estimation of these constants.

Parallel versus serial search and decision

Although the model assumes that T_{dsi} and T_{pi} occur in series, this will often not be the case: for example, if the cursor is used to track the current visual search region, then the visual search and targeting activities will occur in parallel. In these cases, however, the logarithmic pointing component T_{pi} is likely to be small in comparison to the linear search time T_{dsi} (discussed further in future work).

Modelling multi-level menus (e.g. cascades)

Equation 1 predicts single-level menu selections. When users need to navigate through l multiple levels to select an item i, we generalise the formula as follows:

$$ML_i = \sum_{j=1}^{l-1} (T_j + sc_j) + T_l \qquad \textit{Equation 8.}$$

where sc_j is the 'steering cost' associated with navigating from one menu level to the next. For the average efficiency of a multi-level menu widget, Equation 2 for T_{avg} is modified by replacing T_i with ML_i. Ahlström also proposed adding Fitts' and steering components to model cascade menu selections [2], and for traditional cascade menus the steering cost sc_j can be modeled using the steering law [1]: $sc=a+b(A/W+1)$.

Modelling split menus

Split menus copy regular menu items into an extra region at the top for fast access. Their performance is predicted by treating the menu as consisting of two separate menus: the split region (T_{split}) and the regular menu region (T_{reg}). Equation 1 is modified as follows:

$$T_i = splitDecision + p_{i\_split} T_{i\_split} + (1 - p_{i\_split}) T_{i\_reg} \text{ where}$$

$$splitDecision = (1 - e_{split}) T_{hh2} \qquad \textit{Equation 9.}$$

and where e_{split} is the user's expertise with the contents of the split region of the menu. The *splitDecision* term models the time required for the user to decide between the split region of the menu and the regular region. T_{hh2} is the time to decide between the two regions, calculated from b_{hh}+a_{hh} in Equation 5 (the information entropy H for a two way decision is 1). This model of the split decision process assumes that expert users (i.e., those highly familiar with a stable design) do not decide between regions; instead they only decide between the items located in the correct split region by drawing on their knowledge of the item's absolute location in the menu, as modelled by T_i, further described below.

The model for T_i assumes optimal decision-making – that users always select items from the split when they are available there and from the regular menu when they are not in the split. P_{i_split} represents the probability that the item is in the split, and can be calculated from knowledge about the menu design and the distribution of selections. Performance within the two parts of the menu (the split region T_{i_split} and the regular region T_{i_reg}) is modelled using the original formulation of Equations 1-8.

Note that the learnability of the split region (substituting L_{split} for L in Equation 5) is likely to be lower than that of the regular region (L_{reg}) due to item movement in the split. Also, certain split designs will prohibit the calculation of expertise using Equation 5, because the exact number of trials with each item at each split location will be unknown. In such cases, a fixed value can be used for e_{split} that is determined either by the designer's understanding or by a simulation of the menu design. For example, in the prediction of recency-based split menus below, we used the value 0.2 for e_{split}, based on a simple programmatic simulation of the split menu that showed approximately a one-in-five probability of items remaining in a stable location in the split.

CALIBRATING THE FORMULAE

We conducted an initial study to calibrate the main undefined parameters: Fitts' a and b, visual search a_{vs} and b_{vs}, Hick-Hyman a_{hh} and b_{hh}. The study analysed Fitts' target acquisition in menus, and two menu designs that represent opposite extremes in terms of menu learnability: *static* menus, in which items are never relocated; and *random* menus, in which every item is randomly located in the menu on each successive posting. Static menus were tested with two datasets: an unfamiliar dataset (based on country names) that allows us to examine how quickly users learn their item locations, and a familiar set from Microsoft Office that represents 'optimal' use.

Method

Eight participants took part in the study and performed a number of single-level menu selections. All participants were right-handed graduate computer science students and thus competent users of mice, screens and computers. They were all familiar with Microsoft Office applications.

Each participant initially completed a Fitts' Law block of tasks involving pointing to cued menu items. These tasks allow us to calibrate the Fitts' Law parameters a and b; and with these values determined, we can then calculate the decision/search time (T_{dsi}) by subtracting pointing time from the overall acquisition time (Equation 1). In the Fitts' block, participants were instructed to identify a single item displayed in the menu, and then select it as quickly and accurately as possible. Only the target item was labelled in the menu; all others items were blank.

Software measured the time from leaving the menu button to correctly clicking on the target.

The menu used in the Fitts' Law block was sixteen items long (each item was 130×22pixels, with the same size being used throughout the study). The item in the first menu location was labelled "1", 2nd location "2", and so on up to item "16". The entire block consisted of ten selections of each of the following items, presented in a random order: 1, 4, 8, 12, and 16 – 50 selections in all.

The participants then completed a series of menu selections in twelve different conditions covering three menu conditions (static+unfamiliar, static+familiar, and random menus) and four menu lengths (2, 4, 8 and 12 items). The participants completed all three conditions in a random order with each length before proceeding to the next longest length.

In the static+unfamiliar and random conditions, the menu was populated with randomly selected country names (not alphabetical) from a list of 177 countries. Each country name was only used in one condition to avoid transfer effects. In the random condition, the menu items were randomly relocated each time the menu was posted. In the static+familiar condition, each menu of length *n* contained the first *n* items from the list '*New, Open, Close, Save, Save As, Save As Webpage, Page Setup, Print Preview, Print, Send To, Properties, Exit*', based on the contents of Microsoft Office 'File' menus.

Trials within each condition were presented in a series of blocks: seven blocks with the static conditions, three with random. Each block consisted of *n* menu selections: one for each of the *n* items in the menu, presented in a random order. The seven blocks in the static conditions were intended to reveal any transition from visual search to spatial knowledge of the menu.

Participants began each trial by clicking on a 'Menu' button, which posted the menu and immediately displayed the name of the target alongside. Software logged all mouse movements and timed the task from the initial display of the target.

Apparatus

The experiment ran on a Compaq nx9010 laptop computer (P4, 2.6GHz) with a 15-inch 1400×1050-pixel display. Input was collected through a Logitech three-button mouse set to a constant control-display gain.

Design

Fitts' Law models were generated for each participant from their Fitts' block data. These models were then used to calculate a Fitts' pointing time T_{pi} for each trial with the three menu conditions. The dependent measure T_{dsi} was calculated by subtracting the Fitts' time from the total selection time using $T_{dsi}=T_i-T_{pi}$ (from Equation 1).

Data from each of the three menu conditions (static+unfamiliar, static+familiar, and random) were analysed separately, using regression analyses across *menu length* (2, 4, 8 or 12 items) and *block* (1 to 7 for both static conditions, and 1 to 3 for random). Data from the random condition were used to calibrate visual search parameters (Equation 6); data from static+familiar were used to calibrate Hick-Hyman parameters (Equation 7); and data from the static+unfamiliar condition were used to inspect the learning model's accuracy (Equation 5).

Calibration Study Results

Fitts' Parameters a and b

Averaged across all participants, data from the Fitts' Law block were well modelled by Fitts' Law ($R^2=0.93$, $p<.01$), with $T_p=0.37+0.13ID$ (*a*=0.37s, *b*=0.13sec/bit).

Evidence supporting the Decision and Search Time model

Figure 1 plots the mean measured decision/search time against the number of items, and it shows best-fit regression lines for each condition. It clearly shows that the trends in T_{dsi} times across number of items in the menu conform to our model's predictions. In the random condition, search time linearly increases with the number of items, and the static+familiar condition closely adheres to the logarithmic Hick-Hyman predictions. In Figure 2, the static+unfamiliar condition shows users making a transition from linear visual search to logarithmic Hick-Hyman performance. Note that the T_{dsi} differences between conditions are relatively large – any model failing to account for the transition between novice and expert performance will be inaccurate.

Visual search time (T_{vsi}) parameters a_{vs} *and* b_{vs}

In the random condition, the participants were dependent on visual search alone to find their targets – there was no way for them to anticipate the location of each item, equating to zero expertise in our model ($T_{dsi}=T_{vsi}$ when expertise is zero, Equation 4). Data from the random

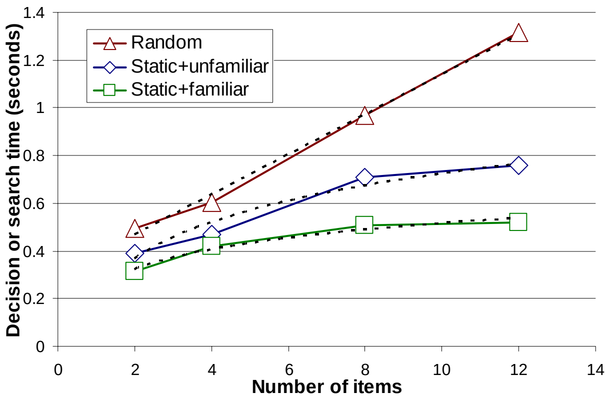


Figure 1 Mean decision/search time (T_{dsi}) for the three menu conditions. Dashed lines show regression models: linear for random; logarithmic for static+familiar and static+unfamiliar.

condition were therefore used to parameterise the visual search constants in Equation 6.

An analysis of variance of performance over blocks showed significant improvement between blocks 1 and 2 ($F_{2,14}$=4.9, p<.05, apparent in Figure 2), presumably because there is an additional cost in reading the menu labels for the first time. To avoid incorporating this one-off cost in the calibration of visual search we removed block 1 from the visual search analysis. Our model is therefore based on visual search for items that have been previously read, which avoids conflating familiarity effects with visual search costs. The model of increasing expertise is further described below.

Regression analysis shows an excellent linear relationship between T_{dsi} and the number of menu items (Figure 1): $T_{dsi}=T_{vsi}=0.08n+0.3$, R^2=0.99, giving values of 0.08s/item for b_{vs} and 0.30s for a_{vs}.

Hick-Hyman decision parameters a_{hh} and b_{hh}

Once completely expert, Equation 4 predicts that decision/search time T_{dsi} depends solely on the Hick-Hyman formula in Equation 7. As intended, our participants were most expert when using the static+familiar condition, so we used this data to parameterize Equation 4.

Block 1 was slower than the others (Figure 2), presumably due to reading and familiarization effects. As Equation 7 only concerns expert performance we removed block 1 from this analysis. Regression analysis produced a strong model with $T_{dsi}=T_{hhi}=0.08log_2(n)+0.24$, R^2=0.98, giving values of 0.08s/bit for b_{hh} and 0.24s for a_{hh}.

Expertise – Equation 5

Determining the ‘correct’ formulation of Equation 5 is complex because the rate of gain in expertise will be influenced by several factors, including the nature or the dataset contained in the menu. For example, memory chunking suggests that a menu containing three groups of four related items will be more memorable than one containing twelve unrelated items [28].

Data from the static+unfamiliar condition represents an entirely learnable menu design (L is 1 because items do not move), but a ‘hard’ data-set because the items contained in the menu are semantically unrelated country names. Data from the static+familiar condition also represents an entirely learnable interface, but with an ‘easy’ data-set because it is familiar from Microsoft Office and chunked.

Having calibrated the visual search and Hick-Hyman parameters of Equations 6 and 7 above, we calculated the cross-block learning prediction for T_{dsi} using Equations 4 and 5; as shown in Figure 2. The line shows the predicted rate of transition from visual search to Hick-Hyman decision for static menus – from initial values determined by the random condition in blocks 2-3, towards Hick-Hyman decision times determined by the latter blocks of the static+familiar condition.

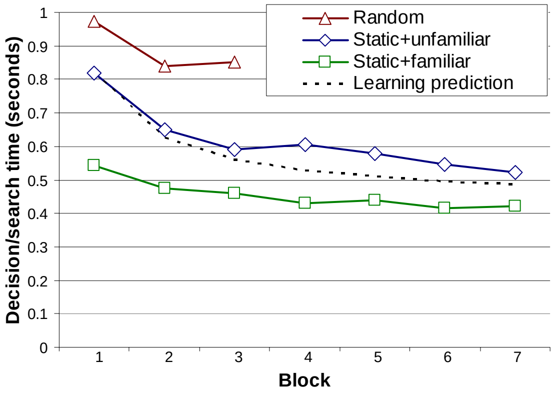


Figure 2. Mean decision/search time for the three menu conditions across blocks of trials (based on the mean of all menu lengths per block).

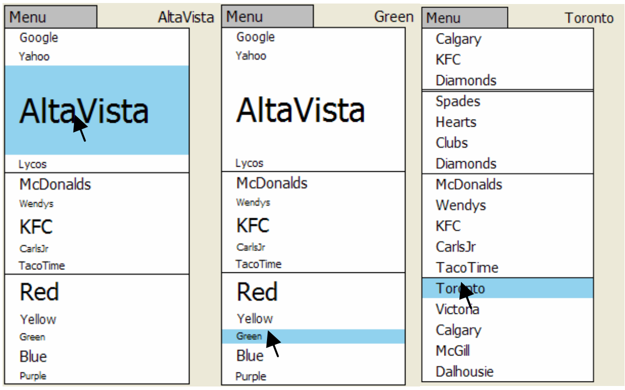
EVALUATING THE MODEL WITH REAL DESIGNS

We evaluated the model by comparing its predictions with empirical measures. Four different types of menus were used: traditional menus, morphing menus, recency- and frequency-based split menus. The evaluation also uses a realistic Zipfian distribution of selections.

Menu designs

Four menu types were implemented, modelled and evaluated. Traditional menus are the baseline. Recency-based split menus were included because they are widely used in desktop interfaces, and frequency-based split menus were used because of prior favourable results [31]. The split menu types were visually identical (Figure 3c), using a three-item split region at the top of the menu, containing the three most recently or frequently selected items. Both split menu variants left the main menu region unaltered – items were not removed from the main menu when added to the split region.

Both recency and frequency based split menus have been previously evaluated elsewhere [10, 31]; to further test the predictive capability of our model we also included a new and intuitively appealing menu design, which we call ‘morphing menus’. The design intention is to minimise the Fitts’ Law pointing requirement for high-frequency items, while maintaining both the user’s ability to form a spatial understanding of item location and the designer’s intended item order. Figures 3a/b show a morphing menu in use during the experiment. The high frequency items have large motor-spaces, (e.g. ‘AltaVista’, Figure 3a) while the infrequently used items are small (e.g. ‘Green’, Figure 3b). High frequency items ‘borrow’ motor-space from infrequent items. Mouse-over enlargement is used to aid visual identification of items that are displayed in the minimum size. Importantly, this enlargement does not move the text of the item under the cursor. This avoids ‘hunting’ effects caused by fisheye distortion [14].



(a) Morphing (b) Morphing (c) Split

Figure 3. Morphing and split menus. The target is shown next to the menu button when posted.

To help users learn item locations, 'morphing' avoids abrupt item enlargement. Initially all items are of equal probability and size. After each selection, the probability of each item is recalculated: $\forall i$, $p_i=t_i\,/t_t$. A damping function is used: $t_t=n\times damping$, where n is the number of items and *damping* is a small integer value (higher values cause slower adaptation). t_i is initialised to *damping*, and is incremented with each selection of item i.

To reiterate, high frequency probable items 'borrow' motor-space from improbable ones. The amount of menu space available for reallocation is n times the difference between the minimum and maximum menu heights (set to 10 and 22 pixels): $sharedHeight=n\times(normalHeight–minHeight)$. The height of each menu item h_i is then calculated with $h_i=minHeight+(p_i\times sharedHeight)$.

Method

The experiment began with a Fitts' Law calibration block of tasks, identical to that at the start of the calibration experiment – the menu was long enough to display 16 items, but only the target item was displayed; and the participants made five selections of the 1st, 4th, 8th, 12th and 16th menu item, with logs recording the time from the cursor leaving the menu button to selecting the target.

The menu selection tasks followed, with each task involving selecting an item from a 14 item menu. Each target item was cued by showing the name of the item alongside the menu button, but the target was not revealed until the menu was posted. Logs recorded the time from posting the menu to selecting the item; hence, task times included the decision/search time.

The eighteen male and female participants (none of whom participated in the earlier experiments) used all four menu types. An incomplete Latin square controlled menu type order. Each participant completed five blocks of tasks with each menu type before proceeding to the next.

Each block of tasks consisted of 45 menu selections, with a Zipfian distribution of selection frequency. The selection counts for the items were 14, 8, 5, 4, 3, 2, 2, 1, 1, 1, 1, 1, 1, and 1 (Zipfian R^2=0.99). The correspondence between item location in the menu and item frequency was determined by a one-off random process, and re-used across all menu conditions. This was necessary to avoid a menu-type bias due to differences in the location of the frequently selected items: for example, if the most frequent item was at the top of the menu in one condition, it would create an artificially low Fitts' pointing requirement. Hence, all conditions in all blocks used the following frequency distribution across the 14 menu locations (from top to bottom): 2, 1, 14, 1, 4, 1, 5, 1, 1, 8, 2, 1, 3, 1. Participants received no instruction on the frequency of menu item selections or on the location of frequent items. They were simply informed that their tasks involved selecting menu items as quickly and accurately as possible.

The contents of the menu remained constant across the blocks with each particular menu type, but changed across menu types – the intention being to allow participants to develop expertise with each menu type independently. All menus contained three groups of related items split by a thin line (see Figure 3): one group of four items (top), then two groups of five items. For each participant with each menu type, the menu contents were randomly constructed by selecting one group of four items and two groups of five items. No group of items was reused in any other condition. Example groups include car manufacturers (Toyota, Honda, Ford, Mazda, Subaru) and furniture (Table, Chair, Desk, Lamp, Bed).

Design

We designed the study both to test the accuracy of our prediction model, and to compare the performance of the four different menu designs. For the accuracy analysis, we report percentage differences and regression analyses between predicted and actual values. For the comparison part of the study, we analysed selection time data using a 4×5 RM-ANOVA for factors *menu type* (traditional, morphing, recency-split, frequency-split) and *block* (1-5).

RESULTS

The theoretical predictions for traditional and morphing menus were calculated exactly as described by the equations above (spreadsheet at www.cosc.canterbury.ac.nz/~andy/publs/chi07/). The split menu predictions required additional parameterisation as follows. For frequency-based splits, the value of L_{split} in blocks 1 and 2 were set to 0.6 and 0.9 to account for the small amount of item movement within the split prior to the item locations becoming stable due to the Zipfian item distribution; the value was 1 in all other blocks. For recency-based splits, we used a general parameter for e_{split} (value 0.2) to replace Equation 5 which requires *a-priori* knowledge of item location that is unavailable due to the split's behaviour.

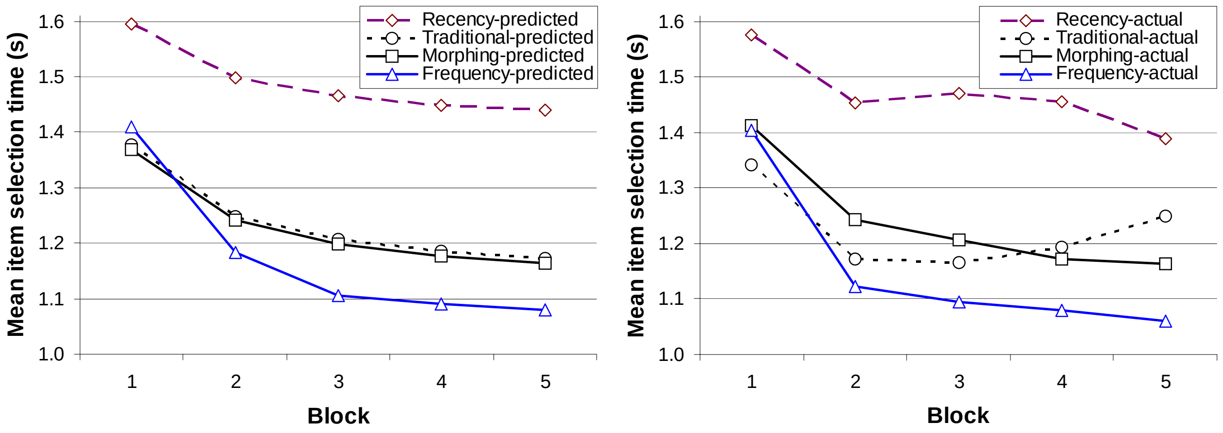


Figure 4. Predicted performance (left) and actual performance (right) (note the baseline at 1.0s).

Prediction Accuracy

Empirical data (Figure 4, right) matched the predictions extremely well. The percentage difference between the cross block empirical and predicted means range are all under 2% for each of the designs: 1.1% for traditional, 0.8% for morphing, 1.3% for recency-based splits, and 1.8% for frequency-based splits. Regression analysis of the predicted cross-block mean for each design against the empirical mean shows a strong relationship that is close to unity: $pred=1.02\times emp-0.01$, $R^2>0.99$. Similarly, regression of the predicted cross-design mean for each block against with the empirical mean is also close to unity: $pred=0.98\times emp+0.03$, $R^2=0.96$.

As Figure 4 suggests, regression analysis of predicted values against empirical ones for each design across blocks shows a strong relationship for morphing ($R^2>0.99$) and frequency-splits ($R^2=0.98$), but weaker ones for recency-splits ($R^2=0.84$) and traditional menus ($R^2=0.52$). The poor fit for traditional menus is caused by the participants' degrading performance after block 3, which we believe was due to a boredom effect caused by their familiar non-adaptive behaviour.

It is important to note that the disparity between the predicted and actual data is relatively small when compared to the differences between some of the menu designs. For example, frequency-based splits were predicted to be 21.8% faster than recency-based ones and empirical measures showed this prediction to be accurate, with an actual value of 21.5%.

It is also important to note that the fundamental premise of the model – that users will migrate from linear search strategies to logarithmic Hick-Hyman choice – accounts for a substantial effect in performance measures, particularly when a design fully supports it. For example, the model predicts a 24% improvement between blocks 1 and 5 for frequency-based splits, supported by an empirical difference of 25%; recency-based splits were predicted to allow less cross-block improvement (11%) due to their spatial instability, and again the prediction was supported (12%). Previous models based purely on expert performance or purely on visual search fail to account for this marked improvement.

Comparison of menu techniques

Analysis of variance of actual performance showed a significant main effect for *menu-type* ($F_{3,51}=118.8$, $p<.001$). Frequency-splits were fastest (mean 1.15s), then traditional and morphing (1.22 and 1.24), and recency-splits slowest (1.47s). A post-hoc Tukey HSD of 0.11s showed significant pair-wise comparisons between recency splits and all others, but not between the other conditions. These results support earlier work [9,29] comparing traditional and split menus. As anticipated, there was a significant main effect for *block* ($F_{4,68}=142.8$, $p<.001$), with pair-wise differences ($p<.05$) between blocks 1-2 and 2-3, but none thereafter. Finally, there was a significant *type*×*block* interaction ($F_{12,204}=14.9$, $p<.001$), evident in the slowdown with traditional menus.

DISCUSSION

These studies confirm the primary hypothesis of the model as predicted by Hick-Hyman and Fitts' Laws: performance with spatially stable menus migrates from novice behaviour that degrades linearly with menu length due to visual search time, through to expert behaviour that degrades logarithmically with menu length. Our calibrated model accurately predicted several types of menu behaviour, both adaptive and non-adaptive.

In the following sections we discuss issues for consideration in further work: reflections on details of the model, on the model as a tool for design, and on the value of predictive performance models in UI design.

Reflections on the model itself

Although the model is accurate for all four menu types, it is notable that it markedly underestimated the performance improvement between blocks 1 and 2. A possible explanation for this is that the second experiment used menus consisting of semantically similar groups or 'chunks' [28], while the static+unfamiliar condition used in the calibration experiment did not. Consequently, the participants could gain 'experience' with menu items before they had encountered them: for example, it is reasonable to expect that previous selections of ‘Ford’, ‘Honda’ and ‘Mazda’ would help the user select ‘Toyota’ for the first time because of its association with the group of cars. Our current model fails to account for this chunk-based incidental learning between tasks.

Despite this weakness in the learning model of Equation 5 (further discussed below) it is notable that the participants rapidly became expert when the designs allowed them to do so. This highlights the importance of modelling learnability because certain designs inhibit the transition to expert performance. For example, the model accurately predicted that expertise would account for a 24% improvement with frequency-split menus, compared to only 11% for the recency design currently in wide use.

Improvements and limitations

We are continuing to develop and test the model in three ways: better modelling of expertise, more work with cascading menus, and work with other menu types.

We are now developing variants of Equation 5 to better model increasing expertise. In particular, we want to account for three factors influencing the gain in expertise: the number of menu items (long menus will be harder to learn than short ones); the number of semantic chunks in the menu; and incidental learning across chunk items.

In future work we will empirically test the model of cascading menu selections (Equation 8). We will also test the model’s success with menus that have different forms than the rectangular pull-downs evaluated to date. For example, we believe that pie menus will be accurately modelled – like traditional menus, they are spatially constant (items lie in a stable direction from the cursor), and they have both a decision and pointing component.

Certain menu designs, however, will be harder to model. Bederson’s Fisheye Menus [4], for example, dynamically adjust the location and size of the items as the cursor approaches. This causes undesirable ‘hunting effects’ [13] because items are displayed away from the motor-space that activates them, but users can ‘lock’ the fisheye to ease the problem. Modelling user performance with this design is challenging because the cost of the hunting effect is unpredictable – it depends on how often the lock is used. Despite this challenge, the model will predict relatively poor learnability (L) for Fisheye Menus due to the low spatial constancy of items.

Scrolling menus are another challenge for the model. We suspect that their multi-part interaction (targeting then dragging the scroll-thumb, visually searching for an item, and finally targeting it) will overwhelm the Hick-Hyman decision time whenever scrolling is necessary. Again, we will continue to empirically investigate these issues.

Reflections on the model’s use in the design process

The model was used to predict performance for several menu designs. Perhaps the most interesting case of these is morphing menus – an intuitively appealing design with (seemingly) a sound basis in Fitts’ Law, but one that the model correctly predicted would provide almost no performance improvement.

First, the example of morphing menus shows the value of a predictive model – that in situations where many designs are possible and many designs appear to have potential, a model can provide an objective view.

Second, the model is also valuable in explaining why designs do or do not perform well. In the case of morphing menus, there were two reasons for the design’s lack-lustre performance. First, morphing menus only focus on pointing time, ignoring decision time; this limits the possible improvement that can be realized. Second, morphing menus increase some items’ size at the expense of others, in order to take advantage of the distribution of selections. However, less-frequent items will take longer, and the model showed clearly that the overall benefit from the larger items would be small.

It is also possible that performance with morphing menus could be improved by changing the parameters of use (e.g., the item frequencies). This raises an additional use for the predictive model – as part of a simulation system that can test a range of parameters, looking for the local performance maxima for a particular design.

Further uses of this and similar models

Although we have focused on menu selection, the model should apply equally well to any interaction that involves a choice decision followed by a pointing task. This type of interaction is very common in user interfaces, for example selecting an item on the desktop, or in a folder, toolbar or button-panel. Empirical work is needed to validate the model’s use in these domains.

Finally, it is important to empirically determine whether the model is successful with menu selections in real tasks, rather than experimentally controlled selections.

CONCLUSIONS

Menus are one of the primary controls for issuing commands in user interfaces. There are many different menu designs, but most constrain the user to predictable patterns of behaviour that are amenable to theoretical analysis. We presented a model that integrates the Hick-Hyman and Fitts’ Laws to predict the efficiency of alternative menu designs. The model accommodates variable item probabilities, the users’ increasing expertise, and adaptive and non-adaptive behaviours. We tested the model by comparing its predictions for four different menu designs with empirical performance measures. The predictions were accurate – within 2% of empirical data.

The model is important for two reasons. First, from a theoretical standpoint, there has been surprisingly little work in HCI to combine the fundamental Hick-Hyman and Fitts' Laws. Our success will hopefully stimulate further work on modelling tasks that include both decision and pointing components. Second, the model should aid researchers and developers working on alternative menu designs, of which there have been many in recent years. Indeed, the authors of this paper hotly debated the efficacy of the morphing design prior to its implementation, modelling, prediction and testing. The model will allow other designers to rapidly test alternative schemes without the labour costs of implementation and empirical evaluation.

REFERENCES

1. Accot, J. and Zhai, S. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. *Proc. CHI'97*, 295–302.
2. Ahlström, D. Modeling and Improving Selection in Cascading Pull-Down Menus Using Fitts' Law, the Steering Law and Force Fields. *Proc. CHI'05*, 61-70.
3. Anderson, J., Matessa, M. and Lebiere, C. ACT-R: A theory of higher level cognition and its relation to visual attention. *JHCI*, *12* (4). 1997. 439-462.
4. Bederson, B. Fisheye Menus. *Proc. UIST'00*, 217-225.
5. Byrne, M. ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *IJHCS*, *55*. 2001. 41-84.
6. Byrne, M., Anderson, J., Douglass, S. and Matessa, M. Eye Tracking the Visual Search of Click-Down Menus. *Proc. CHI'99*, 402-409.
7. Card, S. Visual search of computer command menus. in Bouma, H. and Bouwhuis, D. eds. *Attention and Performance X*, LEA, London, 1984, 97-108.
8. Card, S., Moran, T. and Newell, A. *The Psychology of Human-Computer Interaction*. LEA, 1983.
9. Ellis, S. and Hitchcock, R. Emergence of Zipf's Law: Spontaneous Encoding Optimization by Users of a Command Language. *IEEE Tran. Man & Cybernetics*, *16*, 1986. 423-427.
10. Findlater,L. and McGrenere, J. A comparison of static, adaptive, and adaptable menus. *Proc. CHI'04*, 89-96.
11. Fitts, P. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *J. Experimental Psych.*, *47*. 1954. 381-391.
12. Fitzmaurice, G., Khan, A., Pieke, R., Buxton, M. and Kurtenbach, G. Tracking Menus. *Proc. UIST'03*, 71-79.
13. Greenberg, S. and Witten, I. Supporting Command Reuse. *IJMMS*, *39*. 1993. 353–390.
14. Gutwin, C. Improving Focus Targeting in Interactive Fisheye Views. *Proc. CHI'02*, 267-274.
15. Hansen, S., Kraut, R. and Farber, J. Interface Design and Multivariate Analyis of Unix commands. *ACM TOIS*, *2* (1). 1984. 42-57.
16. Hick, W. On the rate of gain of information. *J. Experimental Psychology*, *4*. 1952. 11-36.
17. Hoffmann, E. and Lim, J. Concurrent manual-decision tasks. *Ergonomics*, *40* (3). 1997. 293-318.
18. Hornof, A. and Kieras, D. Cognitive Modeling Reveals Menu Search is Both Random and Systematic. *Proc. CHI'97*, 107-114.
19. Howes, A., Payne, S. and Woodward, A. The trouble with shortcuts. Proc. *CHI'00*, 267-268.
20. Hyman, R. Stimulus information as a determinant of reaction time. *J. Experimental Psych.*, *45*. 1953. 188-196.
21. John, B. and Kieras, D. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM TOCHI*, *3* (4). 1996. 320-351.
22. Kaptelinin, V. Item Recognition in Menu Selection: The Effect of Practice. *Proc. InterCHI'93.*, 183–184.
23. Kurtenbach, G., Sellen, A. and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *JHCI*, *8* (1). 1993. 1-23.
24. Landauer, T. and Nachbar, D. Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth and width. *Proc. CHI'85*, 73-78.
25. Lane, D., Napier, H., Batsell, R. and Naman, J. Predicting the Skilled Use of Hierarchical Menus with the Keystroke-Level Model. *JHCI*, *8* (2). 1993. 185-192.
26. Lee, E. and McGregor, J. Minimizing user search time in menu retrieval systems. *Human Factors*, *27* (2). 1985. 157-162.
27. MacKenzie, I. and Zhang, S. An Empirical Investigation of the Novice Experience with Soft Keyboards. *Behaviour & Info. Tech.*, *20*. 2001. 411-418.
28. Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, *63*. 1956. 81-97.
29. Nilsen, E. *Perceptual-motor control in human-computer interaction*. Report 37, The Cognitive Sci. & Machine Intelligence Lab., Univ. Michegan, 1991.
30. Sears, A., Jacko, J., Chu, J. and Moro, F. The role of visual search in the design of effective soft keyboards. *Behavior and Info. Tech.*, *20*. 2001. 159-156.
31. Sears, A. and Shneiderman, B. Split menus: Effectively using selection frequency to organize menus. *ACM ToCHI.*, *1* (1). 2004. 27-51.
32. Seow, S. Information Theoretic Models of HCI: A Comparison of the Hick-Hyman Law and Fitts' Law. *JHCI*, *20* (3). 2005. 315-352.
33. Shannon, C. and Weaver, W. *The mathematical theory of communications*. Urbana: Univ. Illinois Press, 1949.
34. Somberg, B. A comparison of rule-based and positionally constant arrangements of computer menu items. *Proc. CHI+GI '87*, 255-260.
35. Soukoreff, R. and MacKenzie, I. Theoretical Upper & Lower Bounds on Typing Speed Using a Stylus and Soft Keyboard. *Behavior & Info. Tech*, *14*. 1995. 370–379.