Effects of Frequency Sorting Towards Finding Optimal Organizations of Hierarchal File Structures

Clayton T. Stanley, Michael D. Byrne and Katherine R. Ramos

Proceedings of the Human Factors and Ergonomics Society Annual Meeting 2008 52: 945
DOI: 10.1177/154193120805201310

Effects of Frequency Sorting Towards Finding Optimal Organizations of Hierarchal File Structures

Clayton T. Stanley, Michael D. Byrne, and Katherine R. Ramos
Department of Psychology, Rice University
Houston, TX

Browsing through a hierarchy of information is a common task, but the relationship between human performance and parameters of the hierarchy are still incompletely understood. Based on Cockburn, Gutwin, & Greenberg (2007), we developed a model of this task involving two components at each level of the hierarchy: visual search time and motor movement time. A Monte Carlo simulation of the model shows how the optimal breadth of a hierarchal structure depends on both the size of the structure and how efficiently users perform the visual search. We fit a function to the Monte Carlo data which produced an optimum breadth consistent with the literature, and also shows analytically how properties like sorting can affect performance. The model indicates that the optimal hierarchy breadth depends heavily on the efficiency of the users' visual search; if the search can be sub-linear, then very broad hierarchies can be highly efficient.

People often seek information in electronically-stored databases. Two main strategies for finding such information are search queries and browsing. Although search queries are becoming increasingly more efficient and accurate at finding precisely what an individual is looking for, it is still possible that the user does not know exactly what he or she wants and therefore resorts to a browsing strategy to forage through the information database. We therefore wished to quantify how the precise arrangement (i.e., sorting files by access frequency) and structure (i.e., depth, breadth, size) of the database, as well as the user's visual search efficiency (i.e., linear, sub-linear, logarithmic) affect average access time when foraging through a hierarchy of information.

A canonical example of where this analysis applies is a folder hierarchy on a personal computer. The information to be accessed (e.g., electronic documents, pictures, papers) consists of the files that a user wishes to find efficiently in the database. The user has the option to sort files (as well as the folders containing the files) by frequency of access and wishes to know how this sorting strategy will affect average access time. Further, we wanted to examine if these effects are invariant to how the user is searching through the files. In particular, is the user scanning item by item (a linear search) or does she already know the name of the item she needs to access beforehand (a potentially logarithmic search). Answering these questions will provide guidelines to the user on how she should structure and sort files when certain environmental constraints are present.

Previous literature has involved analysis and modeling of user proficiency on particular menu subtypes (Cockburn, Gutwin, & Greenberg, 2007; Accot & Zhai, 1997; Gray & Boehm-Davis, 2000). For example, a common research question has concerned the optimal breadth (i.e., number) of items to place on the display as a function of database size and information context. If a small number of items are displayed at one time (i.e., small breadth), then the user must iterate through the process of choosing the correct item more often (i.e., large depth) to find what he is looking for, which can take time. However within each iteration, when displaying a smaller number of items at once, the user does not have to search through as many incorrect alternatives, which can save time. Thus, there should be an optimal number of items to display on the screen that minimizes average user access time to move through the hierarchy, that is, an optimum breadth. Previous research has found this optimal breadth to range between 3 and 12 depending on the user, the database size (e.g., the number of files), and the context of information she is browsing through (Sisson, Parkinson & Snowberry, 1983; Landauer & Nachbar, 1985). This study aimed to build on this literature by analyzing how sorting the selection of items by access frequency affects both user access time and this optimal branching factor. The effects of this frequency-sorting were analyzed for both linear and logarithmic (i.e., Hick-Hyman) visual search conditions.

METHODS

To analyze this breadth-depth tradeoff and effects of frequency sorting, a Monte Carlo simulation was performed where the time for each trial is determined by a sum of visual search times and motor movement times required to browse through a hierarchy of information and find an item. Using terms relevant to our canonical example, this information hierarchy would be represented by a set of folders and enclosed subfolders while the files embedded within the folders are the items of interest.

At each level of the hierarchy, the user must visually locate the next appropriate item and then click on it. If users do not necessarily know the name of the folder enclosing a particular file and can only find the correct path by checking folder names item by item, their visual search time is modeled by a linear function indicating standard self-terminating serial search. With these assumptions, users' visual search time was modeled as:

$$t = a + bN_{cut} \quad (1)$$

where N_{cut} is the correct item's location from the top of the list.

On the other hand, if users do know the location of the folder enclosing a particular file in advance, visual search time will be proportionate to the amount of uncertainty eliminated when choosing a particular path, as argued by Cockburn, Gutwin, and Greenberg (2007). Specifically, visual search time should obey the Hick-Hyman law:

$$t = a + b\log_2 1/p_i \quad (2)$$

Here, p_i is unique for each item in the folder hierarchy and corresponds to the likelihood that the item will be selected when browsing for files.

Once located, the next item must be clicked. To model the movement time, we assumed that the effective width of the mouse movement is equal to the distance between adjacent items displayed on the screen. Therefore, the motor movement time was approximated using the Shannon formulation of Fitts' law as:

$$t = a + b\log_2(1+\#\,itemstomoveacross) \quad (3)$$

For example, if the 2nd of a set of 7 alternative subfolders was clicked previously and the file of interest is enclosed in the 5th of 7 alternative subfolders displayed currently, the number of items to move across would be 3.

These two times (i.e., visual search time and motor movement time) were summed and this process was aggregated along the entirety of the file's path determining the total access time required to find the file in the tree. This process was then repeated for every file in the tree, producing a list of access times. The entire process of building a list of access times from a random assortment of files was then repeated multiple times to help ensure an unbiased sample.

The total number of access times calculated for each hierarchal configuration (i.e., each breadth-depth pair) was 300,000. The number of breadth-depth pairs was 127.

Each access time calculated corresponds to finding a particular file in the tree. However, not all files are accessed with the same frequency. The likelihood of accessing a particular file has been shown to follow a Zipf-like power function with an adjustable parameter alpha:

$$p(r) = \frac{r^{-\alpha}}{\sum r^{-\alpha}} \quad (4)$$

Here, r is the file's frequency rank. A .75 alpha was chosen for this dataset to align with the more conservative values found for this task domain (Glassman, 1994; Nishikawa, 1998; Breslau, 1999).

Additionally, the long-term average shape of the distribution of access frequencies following Eq. 4 should be invariant to users replacing, modifying, and rearranging files. The files may shift and swap in rank (i.e., change individual access frequency), but the underlying distribution will remain unaltered. Files for each r may indeed map to different locations in the hierarchy due to this modifying. However, the Monte Carlo simulation was performed both within and across particular mappings of rs to location. Therefore, the results apply to the realistic situation where the files within the folder hierarchy are modified, altered, and rearranged dynamically, and the user accesses a large range of the enclosed files (with various frequency) over an extended period of time.

For the experimental condition, the folders and enclosed files were sorted by access frequency so that more frequently accessed items remain at the top of the user's search path. For the control condition, everything was randomly sorted.

To calculate these average access times, the coefficients for equations 1-3 were taken from Cockburn, Gutwin, & Greenberg, (2007) due to the similarity of the task. A sensitivity analysis revealed that simulation performance is relatively invariant to the precise coefficient values (within a realistic range). This therefore affords us greater confidence in the validity of the simulation results.

A Monte Carlo simulation was chosen over analytical derivation primarily due to its flexibility; simulations are more easily manipulated to test the sensitivity of results due to small changes in initial configurations. Additionally, an analytical derivation seemed intractable without imposing a few assumptions that were not required for the simulation, due to several artifacts that we were interested in modeling (i.e., access frequency of files (Eq. 4) and browsing behavior (Eq. 3)).

To determine the optimal branching factor for a particular number of files, plots of total search time vs. branching factor were produced holding the number of files in the tree constant. This process constrains the effective tree depth by the relationship $b^d = \#\,files$, which is not necessarily a whole number. The Monte Carlo simulation was only performed for integer values of breadth and depth. Therefore, to determine the average access time as a function of breadth for a constant number of files required interpolating between the calculated breadth-depth pairs from the simulation. A combination of multiple regression (i.e., fitting a plane to four points in space), planar fitting (i.e., fitting a plane to three points in space), and linear interpolation (i.e., fitting a line to two points in space) was used as a means of accurately interpolating between these simulation data.

A folder hierarchy at any one moment with a fractional depth is certainly unrealistic. However, having a folder hierarchy with an *average* fractional depth is indeed realistic. The interpolation methods discussed previously can therefore be conceptualized as finding average access time when the structure (i.e., breadth and depth) of the folder hierarchy is slightly perturbed during the simulation, giving fractional values for the long-term average breadth and depth.

The results after interpolation were then used to produce the plots of average access time as a function of branching factor (i.e., breadth or b) for a particular number of files. The error bars for these plots are the average residual error resulting from using the previously discussed interpolation methods. The error associated with the uncertainty in the mean for these simulation data was not propagated to these plots for this error is at least one order of magnitude smaller than the error due to the interpolation techniques.

One theoretically plausible function relating the users' average access time to the breadth of the folder hierarchy might be a product of the average time to choose and click a

particular item in the tree and the number of times required by the user to do so before finding the file in the tree (i.e., tree depth):

$$t(b) = (\log_b files)(Ap_1 b + B + C\log(p_2 b + 1) + D) \qquad (5)$$

Here, the first term is the tree depth and the last term contains a combination of linear and logarithmic functions of *b* aimed at modeling the average summed search and Fitts' movement time across all levels in the tree. The capitalized parameters are the coefficients for the Fitts' movement time and appropriate visual search time, while *b* is the tree breadth (i.e., branching factor), and $p_{1,2}$ are empirically determined coefficients dependent on both sorting strategy and visual search type (i.e., logarithmic or linear), and slightly on the number of files as well.

It would be a straightforward interpretation of Eq. 5, at least for the linear search condition, if $p_1 b$ reflected the average number of items searched before finding the item to select while $p_2 b$ reflected the average number of items moved across before selecting this item. However, the visual search time coefficient (p_1) and movement time coefficient (p_2) do not seem to be decoupled; rather, having both present allows the model to fit well to these data, but neither of the two fit average search and movement time components in isolation. We are still working on determining the precise relationship between these parameters and average search and movement times.

Nevertheless, the entire problem of finding optimal sorting strategies and ideal structural configurations of the folder hierarchy can be approached by finding the solution for a particular domain which minimizes the time required by a user to search for and point to an item. For this model, this is analogous to minimizing the latter product in Eq. 5 that depends on $p_{1,2}$.

To determine the optimal branching factor for both the frequency-sorted and randomly-sorted conditions when visual search is linear, the fitted function was differentiated, set equal to zero, and numerically solved for this optimal *b*. Error was propagated from the error associated with the coefficients used to fit the function to these simulation data (i.e., $p_{1,2}$), assuming independence in these coefficients.

The minimum access time is the average access time returned by the fitted function at the optimal *b*. The uncertainty in this average access time was again propagated from the error associated with $p_{1,2}$ assuming independence.

RESULTS

Plots of resulting data when visual search time is accurately modeled by a Hick-Hyman relationship are included in Figure 1:

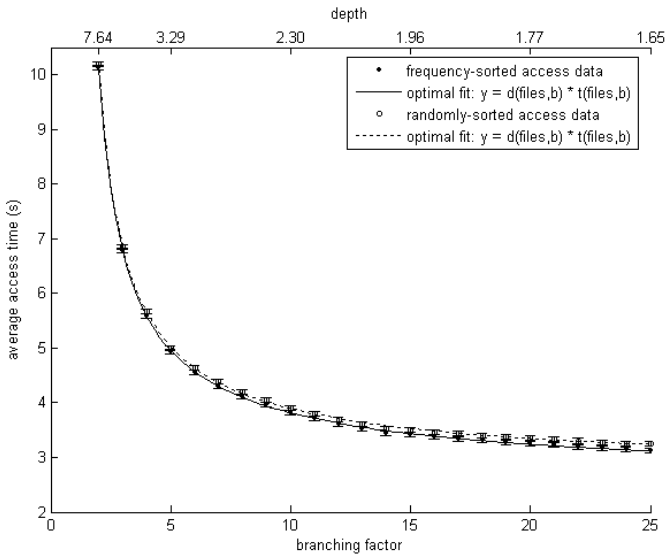

Figure 1: Plots of average access time at 200 files with a logarithmic visual search function

Both fitted functions for the frequency-sorted and randomly-sorted conditions capture the majority of the variance in average access time seen in these data ($\tilde{X}^2(24) = .188,\ p \approx 1,\ R^2(22) = .999$ & $\tilde{X}^2(24) = .206,\ p \approx 1,\ R^2(22) = .999$). Note that sorting the files by access frequency has a negligible effect in improving user access time. Also, these data do not show an optimal branching factor – rather, average access time continues to decrease as breadth (*b*) increases, although the rate of decrease slows with larger *b*.

Plots of resulting data when visual search time is linear are included in Figure 2.

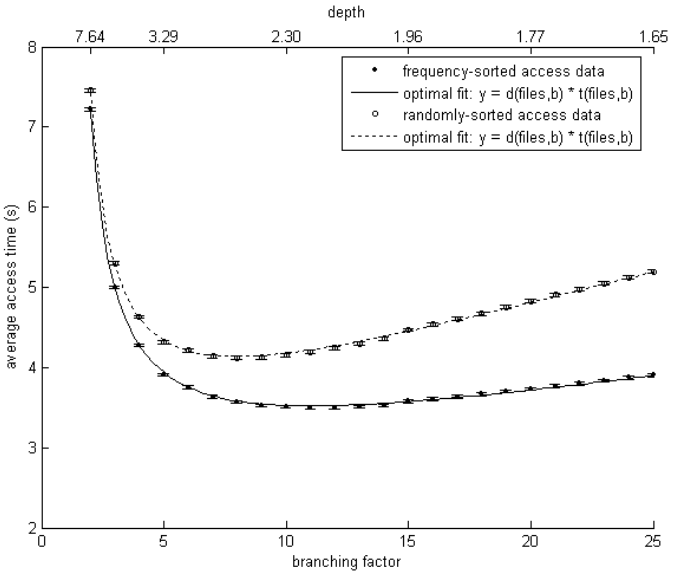

Figure 2: Plots of average access time at 200 files with a linear visual search function

The fitted functions again capture the majority of variance seen in these data ($\tilde{X}^2(24) = .96,\ p = .5,\ R^2(22) = .999$ & $\tilde{X}^2(24) = .91,\ p = .5,\ R^2(22) = .999$ for the frequency-sorted and randomly-sorted conditions). Note that sorting the files by

access frequency here has a relatively large effect in improving user access time. Also, these data do show an optimal branching factor for both conditions. Finally, note that frequency sorting the files reduces the curvature of the fitted function. This consequently allows for a larger amount of slack in branching factors still producing relatively efficient access times.

For linear visual search, plots of optimal branching factor and minimum average access time as a function of the number of files are included in Figure 3. For both panels of Figure 3, these data were fit to a logarithmic function using the Levenberg-Marquardt nonlinear least squares fitting algorithm. Qualitatively, the fitted functions capture the main logarithmic component seen in these data:

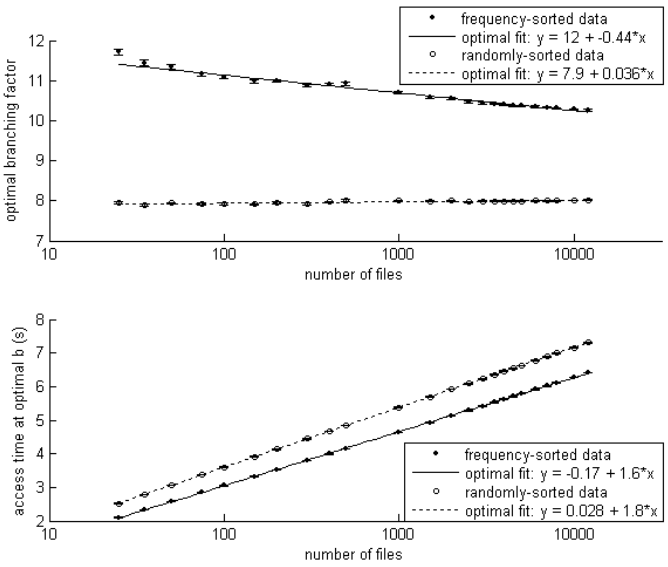

Figure 3: Top: Plots of optimal branching factor as a function of number of files (log scale). Bottom: Plots of minimum access time vs. number of files (log scale)

Quantitatively, for the frequency-sorted condition, both statistical tests would allow one to make the case that a purely logarithmic function might not capture all the variance present across the number of files ($\tilde{X}^2(24) = 3.24$, $p < .001$, $\tilde{X}^2(24) = 2.17$, $p = .003$ for the top and bottom plots). However, there are strong correlations for these fits ($R^2(22) = .969$, $CI_{95} = .927,.986$ & $R^2(22) = .999$, $CI_{95} = .999,.999$ respectively). For the randomly-sorted condition, there is no conclusive evidence to argue that a purely logarithmic function is not a reasonable model for these data ($\tilde{X}^2(24) = 1.08$, $p = .5$, $\tilde{X}^2(24) = .057$, $p \approx 1$ for the top and bottom plots). Additionally, the correlations for these fits are strong as well ($R^2(22) = .754$, $CI_{95} = .511.887$ & $R^2(22) = .999$, $CI_{95} = .999,.999$). Although a purely logarithmic function may not predict the entirety of variance in these plots for the frequency-sorted condition, correlations for both conditions show that there is at least a strong logarithmic component of trend present for all.

Sorting the files by frequency of access aims to reduce the intercept in the bottom plot while increasing the intercept in the top plot. Consequently for the bottom plot, these data show that average access time decreases when sorting by access frequency regardless of the number of files in the hierarchal structure. Averaging across all the data points, this sorting method produces a $14 \pm 1\%$ improvement in access time efficiency. For the top plot, these data show that sorting files by access frequency increases the optimal number of items to place at any level in the hierarchal tree regardless of the number of files in the tree. The optimal branching factor for a set of files sorted by frequency ranges between 12 and 10, while this branching factor for a random assortment of files is centered around 8.

DISCUSSION

The results of our model both complement the current literature studying access time for hierarchal databases and expand on previous findings by quantifying how the precise arrangement (i.e., sorting files by access frequency), likelihood distribution of files (i.e., Eq. 4), and visual search efficiency (i.e., linear, logarithmic) affect average access time. We will discuss the relationship between previous work and ours, and end with the key results found for this study; specifically how visual search efficiency and precise arrangement of files affect average access time.

MacGregor, Lee, and Lam (1986) also used a two-stage model of access time when browsing; their model includes a decision component and a response component. This model is similar to our two-component model: their decision time and response time map to our visual search and pointing time respectively. MacGregor et al. also included a computer response time (i.e., the time required by the computer to respond to the user's request) in the response component that was not modeled here.

However, even without modeling this component, our results are consistent with MacGregor et al.'s main finding: Optimal breadth varies inversely with the time to reach a decision and directly with the time to make a response. Consider our Eq. 5; the stronger the linear component (relative to the logarithmic component), the sooner the parabola bends upward (see Fig. 2) and the smaller the optimal breadth. We modeled MacGregor et al.'s response time component as a Fitts' movement. Therefore, if response time is increased, the logarithmic component of average access time will increase, relatively decreasing the strength of the linear component, and the optimal breadth will expand outward. A similar argument can be made when improving visual search efficiency (i.e., MacGregor et al.'s decision time). As this time decreases, the linear component of average access time decreases, relatively increasing the logarithmic component's strength, and the optimal breadth will again expand outward.

MacGregor, et al. essentially discussed factors—primarily target-distractor disciminability—which, in our terms, increased the linear component of the visual search, thus decreasing the optimal breadth. What our results show is the ultimate impact not only of those changes, but of positive changes such as increased visual search efficiency (i.e., logarithmic visual search). Our mathematical model therefore complements the results found by MacGregor et al. (1986), and extends their findings by quantifying not only how the

structure (i.e., depth, breadth, size) affects average access time, but also how the precise arrangement, likelihood distribution of files, and visual search efficiency contribute as well.

Miller and Remington (2004) showed that the optimal structure of the tree is dependent on the ambiguity of labels at top levels, ranging from around 8 for unambiguous to 16-32 for ambiguous titles. Having unclear labels will cause the user to search down multiple wrong paths in the directory structure and back-track recursively until the correct path is discovered. Miller and Remington showed how backwards movements when browsing can cause the optimal branching factor for the hierarchal structure to change. However, each iteration of browsing through the hierarchy is still composed of the same core components (i.e., a search and movement time). Therefore, it's plausible that altering information scent (e.g., manipulating label ambiguity) affects average access time by merely changing the search and movement slope coefficients (i.e., search and movement efficiency) present in Eq. 5. Such manipulations would alter the relative strengths of the linear and logarithmic components in Eq. 5 and consequently shift the optimal breadth.

Of course, our model does not currently backtrack (i.e., it assumes perfect accuracy of visual search at each level). However, it should be relatively straightforward to add a probabilistic failure rate to our model. We could then quantify the precise relationship between selection accuracy and the variables manipulated here (i.e., structure, arrangement, and visual search efficiency) affecting average access time, and challenge the hypothesis that imperfect information scent merely alters the search and movement slope coefficients (i.e., search and movement efficiency) present in Eq. 5.

In fact, a wide range of browsing scenarios, all with slightly different configurations and application domains, could potentially be modeled this way. Testing the generalizeability of Eq. 5 to situations with imperfect information scent is only one of a multitude of scenarios that could be studied. Our formalism should be able to capture the more global effects of any specific change in performance in any one component of the search; the hope is that doing so will wash away any slight perturbations in individual browsing trials and reveal a cleaner model of access time on a long-term average scale. This research therefore compliments previous experimental work in menu selection and hierarchal browsing. It provides a plausible means to examine the large-scale similarities and differences between various browsing scenarios using a common model to predict average access times.

The current modeling effort was primarily an examination of the effect of visual search efficiency (i.e., linear vs. logarithmic). When visual search time has a strong linear component, there is an apparent depth-breadth tradeoff and consequently an optimal branching factor producing minimum access times (i.e., 8 and 10-12 for randomly-sorted and frequency-sorted sets of icons). However, when sorting by frequency, the sensitivity of this 'sweet-spot' is not as strong (i.e., the curvature of the parabolic curve decreases). Therefore, a designer could be less concerned with hitting this optimal branching factor at every sublevel in the tree when the items are sorted by access frequency.

Therefore, when visual search time is linear, the benefits of sorting by access frequency two-fold: First, average access time is decreased by placing the items with higher likelihood of access at the top of the user's search path. Second, moving slightly away from the optimal number of items to place on the screen will not increase access times as much as when the items are randomly sorted. This again will allow the designer to worry more about contextual groupings of the information and less about the exact number of items placed on the screen.

When visual search time is highly efficient (i.e., has a strong logarithmic component), the benefits are numerous: First, average access time will decrease due solely to the fact that search times are sub-linear. Second, the 'sweet spot' for optimal breadth is extremely large and spans primarily outward (see Fig. 1). Consequently, breadth can be chosen over depth as much as possible, and a designer structuring the information hierarchy can be less concerned about maintaining a set number of items at each sublevel in the tree. Finally, a designer may not need to be as concerned about how the information in the tree is sorted (e.g., by access frequency), since such sorting strategies do not seem to affect average access time. However, if the sorting strategy is the cause of logarithmic search times (e.g., sorting alphabetically when the user knows the name of the file he's looking for), this freedom to move between sorting strategies is no longer available.

REFERENCES

Accot, J., & Zhai, S. (1997). Beyond Fitts' Law: Models for trajectory based HCI tasks. *CHI 97*, *1*, 295-302.

Breslau, L., Cao, P., Fan, L., Phillips, G., & Shenker, S. (1999). Web caching and Zipf-like distributions: Evidence and implications. *IEEE Infocom, 1*, 126-134.

Cockburn, A., Greenberg, S., & Gutwin, C. (2007). A predictive model of menu performance. *CHI 2007*, *1*, 627-636.

Gray, W. & Boehm-Davis, D. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology Applied*, *6*, 322-335.

Hornof, A. J. (2004). Cognitive strategies for the visual search of hierarchal computer displays. *Human Computer Interaction*, *19*, 183-223.

Landauer, T., & Nachbar, D. (1995). Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width. *CHI 95*, 73-78.

Larson, K., & Czerwinski, M. (1998). Web page design: Implications of memory, structure, and scent for information retrieval. *CHI 98*, 25-32.

MacGregor, J., Lee, E., & Lam, N. (1986). Optimizing the structure of database menu indexes: A decision model of menu search. *Human Factors*, *28*, 387-399.

Mackenzie, I., & Soukoreff, R. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' Law research in HCI. *International Journal of Human Computer Studies, 61*, 751-789.

Mackenzie, I. (1992). Fitts' Law as a research and design tool in human-computer interaction. *Human Computer Interaction*, *7*, 91-139.

Norman, K. (1990). *The psychology of menu selection: Designing cognitive control at the human/computer interface (Human/Computer Interaction)*. Norwood, NJ. Ablex Publishing Corporation.

Prischepa, V. (2004). An efficient web caching algorithm based on LFU-K replacement policy. *Proceedings of the Spring Young Researcher's Colloquium,* 1-5.

Remington, R. (2004). Modeling information navigation: Implications for information architecture. *Human Computer Interaction*, *19*, 225-271.