

A Process for Anticipating and Executing Icon Selection in Graphical User Interfaces

**David M. Lane
S. Camille Peres
Anikó Sándor
H. Albert Napier**

Psychology Department, Rice University

This article presents a system for predicting the icon a user will select from an icon toolbar, based on command use frequency and mouse trajectory. The system differs from previous systems in two important ways: First, the prediction system does not initiate any action. Instead, it predicts where the mouse is moving and subtly “suggests” a command for the user to verify. Second, the system takes into account the relative likelihood of commands being used when making its predictions. Initial testing suggested that a system that only predicted the most frequently used icon choices was better than one that predicted all choices. A study with 12 test users using a mouse and 10 using a trackpad found substantial benefits of this “limited prediction system.” The system resulted in a mean reduction in time to issue a command of 41% for trackpad users and 25% for mouse users. Trackpad users, but not mouse users, preferred the prediction system to the traditional way of pointing. These results suggest that a prediction system such as the one described here has the potential to reduce the time and effort required to issue commands. The utility of the system appears to be especially great in laptops and other devices that use trackpads as their primary pointing devices.

1. INTRODUCTION

In recent years, icon toolbars have become very popular, and are now the method of choice for most users. In a recent study (Lane, Napier, Peres, & Sandor, 2004), we found that among experienced users of a word-processing program, 65% favored issuing commands from an icon toolbar compared to 25% who favored the use of

We thank William Ryan, Daniel Treiman and Jeremyah Payne for their work on the prototype, and Kijana Knight for helping collect data. This research was partially supported by the Center on the Management of Information Technology and Jones Graduate School of Management at Rice University.

Requests for reprints should be sent to David M. Lane, Psychology Department—MS 25, Rice University, P.O. Box 1892, Houston, TX 77251-1892. E-mail: lane@rice.edu

pull-down menus and only 10% who favored some other method, such as keyboard shortcuts. Despite the popularity of the icon toolbar, its use has drawbacks. First, it is not as efficient as keyboard shortcuts. Second, as larger display sizes become the norm, the distance traversed, and therefore the time to reach the icon toolbar, will increase. One approach to improving user efficiency is for trainers to encourage the use of keyboard shortcuts rather than icon toolbars. However, because it is easier to learn how to use icon toolbars than to memorize a set of keyboard shortcuts, this approach is unlikely to have a substantial impact. An alternative is to seek ways to make using icon toolbars more efficient. In this article, an approach to improving the efficiency of icon toolbars is described based on a system that anticipates the icon that is going to be chosen as the cursor approaches the icon toolbar.

There are two reasons to expect that such a system will improve user efficiency: Relatively few commands account for a substantial portion of command use, and attempts to predict the intended destination of the cursor based on its trajectory have been successful.

Relatively few commands have been found to account for a sizeable proportion of command usage in a spreadsheet program (Napier, Batsell, Lane, & Guadagno, 1992) and in a word processor (Linton, Joy, & Schaefer, 1999). In both studies, the five most frequent commands accounted for about 50% of total command usage. Not only are some commands issued much more frequently than others, but users tend to repeat recently issued commands (Greenber & Whitten, 1993) or issue commands in predictable sequences (Guadagno, Lane, Batsell, & Napier, 1990). In the latter study, two predictions of a user's next command choice were made based on the user's last two commands. Approximately 85% of the commands issued were one of the two predicted commands.

A number of algorithms have been developed to predict the target of the cursor's movement based on the cursor's trajectory (Murata, 1995, 1998; Oirschot & Houtsma, 2000; van Mensvoort & Oirschot, 2001). Each of these approaches has been relatively successful. Their success derives, in part, from the fact that cursor trajectories tend to be relatively straight and thus predictable (Oirschot & Houtsma, 2000).

Murata (1995, 1998) conducted a series of studies that used the cursor's trajectory for predicting the target toward which the cursor was moving. The prediction method was based on repeated calculations of the angle between the cursor movement vector and the vector that connected the current cursor position and the center of each target. This prediction algorithm was incorporated into an interface that would "jump" the cursor to the target. The system was tested on a task in which the user moved the cursor to one of five targets. The system was very successful, reducing pointing time by about 25%. The system's accuracy varied as a function of the position of the target being predicted with the targets on the ends and in the middle of the display having a much higher prediction accuracy rating than the other two. As would be expected, accuracy was higher when the targets were more spatially separated. Perfect accuracy was achieved in the targets on the ends and in the middle when 30×30 pixel targets had a space of 50 pixels between them.

2. ITERATIVE DEVELOPMENT OF A SYSTEM FOR PREDICTING ICON SELECTION

Because a command issued in error by a command-prediction system would be very disconcerting to a user, we feel that such a system would have to be extremely accurate. Although we can only guess the accuracy rate that users would find minimally acceptable, we suspect it to be above 95%. To achieve a high accuracy rate, our system differs from previous systems in that it does not initiate any action. Instead it predicts where the cursor is moving and subtly “suggests” a command for the user to verify. The expectation is that giving the user final say about whether a command should be issued would decrease errors and increase user acceptance. Our system also differs from previous systems by taking into account the relative likelihood of command usage when making its predictions.

The system begins with a preliminary assessment of the likelihood of each icon being chosen based on its frequency of use determined either from previous data on typical users or from data collected on the individual user in question. As the user moves the cursor toward the icon toolbar, the system revises its assessment of the likelihoods based on the direction of the cursor’s movement. The icon with the highest likelihood is indicated by a subtle visual change, such as a framed border (see Figure 1).

If the user wishes to issue the command associated with the icon, he or she pushes the mouse button. Therefore the system never initiates an action on its own, it only predicts where the mouse is going and suggests commands for the user to verify.

2.1. Prototype 1

This prototype allowed the user to perform a simple icon-selection task. At the beginning of a trial, an icon was shown at the bottom of the screen matching one of the icons in an icon toolbar at the top of the screen. The user mentally located the icon in the icon toolbar, clicked on the icon at the bottom of the screen to initiate the trial, and then selected that icon on the toolbar. The icon toolbar had 17 icons with the letters A through Q; these icons were 5 mm × 5 mm (see Figure 2). A target icon was displayed at the bottom of the window in a centered position. A trial started when a participant clicked on the target icon at the bottom of the screen, and the task was to



FIGURE 1 Icon *H* is highlighted, indicating that it has been predicted. In actual use, the highlight border is red.

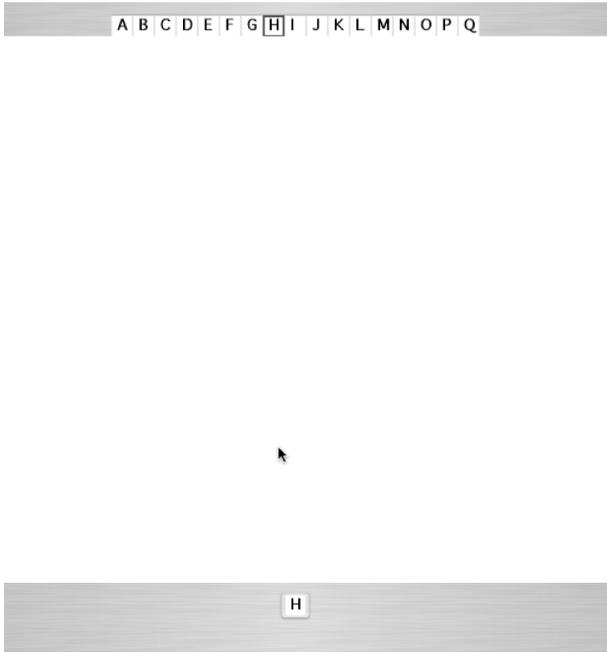


FIGURE 2 A screen from the prototype of the prediction system. The target icon is highlighted on the toolbar.

select the matching icon on the toolbar. The trial finished when an icon from the toolbar was selected.

When the prediction system is active, the predicted icon is continually updated and indicated visually as the cursor is moved toward the icon toolbar. The user clicks when the predicted icon matches the target icon.

The predicted icon is determined as follows. Each of the icons is initially assigned a value (V_i) related to the probability that it is the icon the user intends to select. The V s are not true probabilities and can be thought of as pseudoprobabilities. The initial value of this parameter can be used to indicate which icon is a "high-frequency" icon. The V s are updated every time the mouse moves a distance of d pixels. The icon with the highest V is selected as the predicted command.

The updating of the values is calculated as follows: Define p_i as the proportion of the distance the cursor has traveled to icon i since the last updating. Define $V_{i,j}$ as the value of icon i after update j . If icon i is not currently the predicted icon, then

$$V_{i,j+1} = V_{i,j} p_i$$

If icon i is currently the selected icon, then the updated probability is

$$V_{i,j+1} = kV_{i,j} p_i$$

where k is less than 1. The k parameter is used to reduce the value of an icon that is currently selected but not chosen by the user (with a mouse click) on the assumption that because the user did not click the mouse, it is less likely to be the intended

target. Because it would be confusing to have the predicted icon change rapidly, the predicted icon cannot change in less than t msec.

Although the system is designed to avoid errors by requiring the user to verify the command before it is executed, an error can occur if the user decides to verify a prediction, but the highlighted icon changes in the time between the initiation and execution of the user response. Therefore, any response occurring in fewer than q msec after the predicted icon changes is considered a confirmation of the previously predicted icon.

The implementation of the system was written in Java and run on Apple® Macintosh® eMac™ computers using the Microsoft® Internet Explorer Web browser. The screen size was 21 in., and the screen resolution was 1024×768 . To perform the task, the user sat at a distance of approximately 600 mm from the computer screen, making the visual angle for the 5×5 mm icons about 0.5° .

Because of the large number of parameters, it would have been impractical to attempt a comprehensive exploration of the parameter space. Therefore, each of the authors experimented with different parameters and together discussed what seemed to be best. Based on the results from this informal testing, we decided to use the parameters shown in Table 1 in subsequent tests with the initial values for V of the high-frequency icons set at three times higher than the values of the low-frequency icons.

After observing a few users, we found that when moving toward a target icon, the users often started by moving the mouse vertically until an icon close to the target icon was selected, and then they moved the mouse horizontally to highlight the target icon. The prediction system could not originally handle this type of behavior. For example, if the user moved the cursor horizontally to the right to indicate that he or she wanted to select the icon directly to the right of the predicted icon, the system would mistakenly predict that the user wanted to choose an icon on the extreme right.

2.2. Prototype 2

To avoid the problem of horizontal movements, the system was modified so that it would attempt to determine whether the user was in “lateral move mode” by considering how close to a horizontal direction the cursor was moving. When the system judged that the user was in this mode, the system would change the selected

Table 1: Parameters Used for Prediction System

<i>Parameter</i>	<i>Description</i>	<i>Value</i>
d	Distance in pixels the mouse must move before the V is updated	25
k	Used to reduce the value of an icon that is currently selected but not chosen by the user	0.9
q	Number of ms after the predicted icon changes before any response is considered confirmation of the newly predicted icon	200
t	The amount of time before the predicted icon can change	0

icon horizontally one icon at a time. Recall that once an icon was predicted, it stayed the predicted icon for at least t msec. This modification made the prediction system's behavior much less erratic.

Pilot testing of Prototype 2 found that although the system was easy to use and worked smoothly, it did not appreciably decrease the time to select an icon. We therefore concluded that Prototype 2 was not a workable system.

2.3. Prototype 3

We suspected that choosing among 17 icons was too complicated a task for the prediction system to accomplish well enough to result in a substantial decrease in selection time. As an alternative, we considered whether a less ambitious system would work better. Our thinking was that because a small number of commands accounts for a large percentage of command usage, a system that was only active for a few commands but was very accurate in the prediction of these commands could be better than a system that attempted to predict each of many possible commands. In this system, which we call the limited-prediction system, the prediction system was only active for 4 of the 17 icons. The 4 active items were spaced on the icon toolbar to optimize the performance of the prediction system. Based on the high accuracy achieved by Murata (1998) when his targets were spaced, we expected a high degree of accuracy in the limited-prediction system, as it also used widely spaced targets.

A pilot test of Prototype 3 was conducted with 6 test users: Four were tested on a computer with a mouse, and 2 were tested on a computer with a trackpad. Users selected icons from an icon toolbar using three different methods: (a) the full prediction systems (all 17 icons were potentially predicted), (b) the limited-prediction system, and (c) a no-prediction system (the users selected icons using the traditional method of moving the mouse cursor to the target icon to select it). In the limited-prediction system, there were 4 icons that could be predicted by the system. These icons were gray so the user would know which ones the system would predict. When a gray icon was the target icon, the user could move the mouse toward the icon, the icon would be highlighted with a red box, and the user could click the mouse to select the icon. In this system, the 4 predicted icons were spaced on the bar so they would not be close together, thus ensuring accurate predictions with the algorithm. In practice, when the user moved the mouse in the direction of the predicted icon, the prediction algorithm was highly accurate.

For trackpad users, response times in the limited-prediction system ($M = 1,149$ msec) were substantially shorter than both the control (no-prediction) condition ($M = 1,419$ msec) and the full-prediction system ($M = 1,349$ msec). For mouse users, however, neither the full-prediction system nor the limited-prediction system resulted in shorter times than in the control condition.

3. EVALUATION OF A SYSTEM FOR PREDICTING ICON SELECTION

Encouraged by the results with Prototype 3 for the trackpad users, we conducted a more formal evaluation of the efficacy of the limited-prediction system. The proce-

ture was the same as in the testing of Prototype 3 except that the full-prediction system was not included.

3.1. Method

Test users. Twenty-two students from Rice University participated in the study, 12 using the mouse and 10 using the trackpad. The mouse users were undergraduate students and received course credit for participation. The trackpad users were MBA students and volunteered to participate. We chose to use MBA students for the trackpad because they all had laptops and were experienced in using a trackpad. For both groups, test users used their preferred pointing device for the computer (mouse or trackpad).

Apparatus/algorithm. The same computer and algorithm used in the evaluation of Prototype 3 were used here.

Design. Each test user was presented with two methods of selecting icons from the toolbar: no-prediction system and limited-prediction system. In the no-prediction system, test users selected the target icon with the traditional method of moving the pointer to the target icon and clicking on it. In the limited-prediction system, the system only anticipated movement toward 4 of the 17 icons. These test icons were highlighted in gray so test users would know which ones the system could predict. They were in the 3rd, 7th, 11th, and 15th positions on the icon tool bar. The other 13 icons were selected using the traditional method. For both the prediction and four-prediction conditions, test users were given feedback on their selection for every trial.

To assess the ease with which the prediction system could be learned by users already familiar with the task, all test users performed first in the no-prediction system and then in the limited-prediction system. We were not concerned about practice effects because our pilot studies showed performance asymptoted quickly. The results of this study confirmed that test users had asymptoted by the end of the block of practice trials.

Test users in the mouse condition completed 185 trials in the control (no-prediction) and then 185 trials in the limited-prediction system condition. Test users in the trackpad condition completed 125 trials in each condition. The number of trials was lower for the trackpad users because it took longer to point using a trackpad, and we did not want the experimental session to last so long that the test users would lose interest. As a result, for both types of users the experiment lasted for about 30 min. The test users were given instructions on how each of the conditions would work and were told that the first few trials in each condition would be considered practice.

Additional measures. After the experiment, test users completed a questionnaire on their subjective impressions of the different methods of selecting icons.

Specifically, they were asked which method they preferred, which method they thought was the fastest, and which method they would be most likely to use in the future (if available).

3.2. Results

Due to technical difficulties, the time data for 1 test user in the mouse condition and 2 test users in the trackpad condition were lost, leaving 11 and 8 test users, respectively, in those groups. The preference data for these test users were intact.

For analysis purposes, trials with response times longer than 3,000 msec and the first 20 trials of both the control and limited-prediction system were excluded, and the remaining trials were divided into two blocks: practice and experimental. The practice block contained 82 trials in the mouse condition and 52 trials in the trackpad condition; the experimental block contained 83 trials in the mouse condition and 53 trials in the trackpad condition.

To assess whether test users improved with practice on this task, performance on test items in the first half of the practice block was compared with performance in the second half. The mean time for each test user was computed for the first and second halves of the practice block for both the mouse and trackpad, resulting in four scores per user. A 2×2 (Device \times Half) analysis of variance (ANOVA) revealed a significant effect of device (mouse was faster) but no evidence of an effect of half, $F(1, 17) = 0.01, p = .911$, or a Device \times Half interaction, $F(1, 17) = 0.01, p = .93$. Therefore, there is no evidence that users were improving after the first half of the practice block.

Mean times for each test user were computed for each combination of device (mouse, trackpad), block (practice, experimental), system (on, off) and icon type (test, control) for correct trials, resulting in 16 scores per user. Table 2 shows the means and standard deviations of these means and the proportion of error trials as a function of condition. An inspection of Table 2 reveals that the times for the test icons were substantially lower in the limited-prediction system than in the no-prediction system and that the difference was considerably larger for the trackpad than for the mouse.

A $2 \times 2 \times 2 \times 2$ (Device \times Block \times System \times Icon Type) ANOVA was conducted to test the significance of these differences. The effects of device, $F(1, 17) = 105.10, p < .001$; system, $F(1, 17) = 79.90, p < .001$; and the Device \times System interaction, $F(1, 17) = 18.15, p < .005$, were significant. Further, the main effect of icon type, $F(1, 17) = 196.25, p < .001$; the Icon Type \times Device interaction, $F(1, 17) = 38.41, p < .001$; the Icon Type \times System interaction, $F(1, 17) = 237.47, p < .001$; and the Icon Type \times System \times Device interaction, $F(1, 17) = 77.69, p < .001$, were significant. Neither the effect of block nor any of the interactions with block were significant.

Because the effects differed as a function of icon type (test or control), separate analyses were done for these two types. For the test icons, the effects of device, $F(1, 17) = 98.95, p < .001$; system, $F(1, 17) = 161.88, p < .001$; and the Device \times System interaction, $F(1, 17) = 44.70, p < .001$, were significant. Neither the effect of block nor any of the interactions with block were significant ($p > .15$) in all tests. For the control icons, the effect of device, $F(1, 17) = 99.53, p < .001$, was significant. No other effects were significant ($p > .08$).

Table 2: Mean Response Time in Milliseconds as a Function of Block, Icon Type, and System Status

	Test Icons			Control Icons		
	On	Off	Off-On	On	Off	Off-On
Practice block						
Mouse						
M	882	1,147	264	1,199	1,208	9
SD	153	171	156	193	189	116
Error rate	0.045	0.007	-0.038	0.002	0.002	0.000
Trackpad						
M	1,172	2,061	889	2,128	2,047	-81
SD	148	262	297	196	241	178
Error rate	0.028	0.004	-0.024	0.011	0.000	-0.011
Experimental block						
Mouse						
M	830	1,102	272	1,120	1,152	32
SD	118	132	70	157	143	91
Error rate	0.030	0.016	0.014	0.016	0.002	-0.004
Trackpad						
M	1,192	2,028	836	2,073	2,059	-14
SD	185	226	276	263	328	144
Error rate	0.010	0.005	-0.005	0.011	0.000	-0.011

The box plots shown in Figure 3 show the percentage decrease in time due to the prediction system for the test icons in the experimental block. The percent decrease was calculated individually for each test user as the mean time across trials when the system was off minus the mean time across trials when the system was on divided by the mean time across trials when the system was off. The mean decrease was 25% (95% confidence interval [CI]: 21%–29%) for the mouse and 41% (95% CI: 32%–50%) for the trackpad. This difference was significant, $t(17) = 4.21, p < .001$. It is apparent from Figure 3 that the variability in proportion decrease was greater for the trackpad than for the mouse.

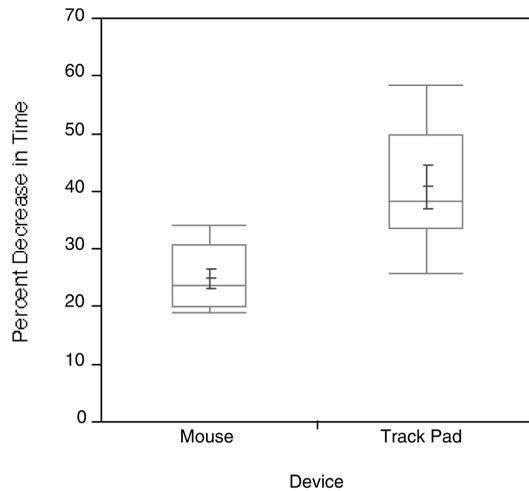


FIGURE 3 The proportional decrease in time due to the prediction system for the test icons in Block 2 as a function of device. The vertical line within each box represents the mean \pm the standard error of the mean.

As can be seen in Table 2, the error rates were low. Most important for present purposes was the finding that in the experimental block when the prediction system was on, the error rate was low for the mouse condition ($M = 0.03$) and very low for the trackpad condition ($M = 0.01$).

At the conclusion of the experiment, test users were asked to indicate which pointing method they preferred and which method they thought resulted in the faster speed. Figure 4 shows clearly that the trackpad users thought the limited-prediction system was both faster than and preferred to the traditional pointing method. The mouse users, however, perceived no time saved and preferred the traditional pointing method to pointing with the prediction system. The differences between the mouse and trackpad users were significant for both preferences, $\chi^2(1, N = 22) = 7.25, p = .007$, and perceived speed, $\chi^2(1, N = 22) = 4.03, p = .045$.

3.3. Discussion

Our initial goal of predicting any icon a user will select proved to be overly ambitious: the full-prediction system appeared to be of little value. However, the limited-prediction system substantially reduced selection time for both mouse and trackpad users. Nevertheless, the time saved for the mouse users was apparently not enough for them to want to adopt the system, as indicated by their postexperimental response. The trackpad users, on the other hand, saw a greater time savings and strongly preferred having the system on to having it off. This finding is a good example of a disassociation between preference and performance measures and suggests that, although the mouse is not a perfect device, it works well, or is “satisficing,” to use Simon’s (1956) term. It appears that a device must

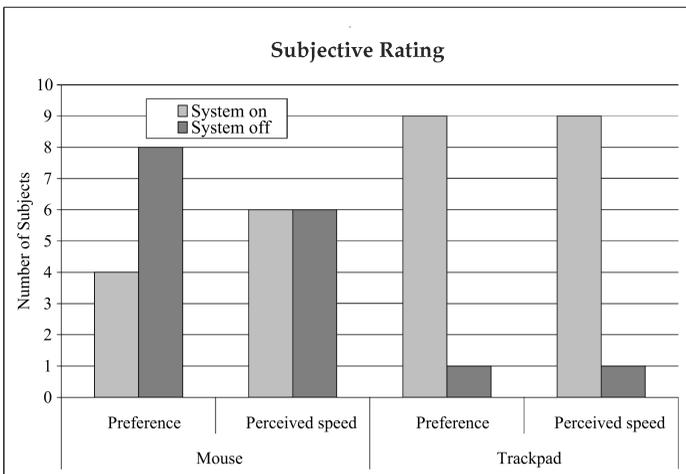


FIGURE 4 Ratings of preference method and perceived speed as a function of pointing method and system status.

save a great deal of time for users to prefer the prediction system to the traditional method. As for trackpad users, the time savings and preference ratings show that the implementation of the prediction system would be welcome. Considering the increasing screen sizes of laptops, the benefits from a prediction system such as described in this article could become even more sizable.

The accumulated time saved for trackpad users could be considerable. For example, assume a worker issued 120 commands an hour using a trackpad and that 50% of these were predicted by the system. From our data, the average time saved on a predicted command would be approximately 30% of 2 sec, or 0.6 sec. Multiplying by 60 predicted commands per hour gives a savings of 36 sec per hour. Because 36 sec is 1% of 1 hr, this means 1% of the worker's time could be saved. To convert this time savings into dollars, simply compute 1% of an employee's salary. Thus, the yearly savings for a professional making \$40,000 per year would be \$400 (U.S. Bureau of Labor Statistics, 2002). Moreover, the satisfaction of using a system that makes one's job easier is an important benefit.

Saving time is not the only benefit of a method that decreases the time spent issuing commands. Extended use of pointing devices can cause repetitive strain injury, and by reducing the time spent using a point device of any sort (mouse, trackpad, etc.), the occurrence of these types of injuries may be reduced with the use of the prediction system (Wickens, Gordon, & Liu, 1998).

It should be noted that the trackpad users were MBA students and the mouse users were undergraduates, and there were more trials per block for the latter group than for the former group. We feel it is very unlikely that much if any of the performance differences between these groups is attributable to these facts. However, it is possible that they had a nontrivial effect on the subjective responses. Therefore, future research using comparable classes of users would be informative.

We found no evidence that users were improving with practice after the first set of trials, and therefore we believe that an explanation of our results in terms of practice effects is untenable. Nonetheless, future research should take steps to avoid possible confounding of experimental and practice effects. Further, it should be kept in mind that we used a very small sample. However, the value of the system is apparently sufficient to be shown clearly even with such a small sample.

The design of an icon bar to be used with the prediction system should be created with care. It is especially important to consider the trade-off between the most logical ordering of icons and the ordering that would work best with the prediction system. For example, it is logical to group icons with similar functions. However, this would only allow one of the icons in the group to be predictable by the system.

It is important to note that the prototype presented here used a left click of the mouse to "click-on" the icon and that this method would probably not be practical for most software programs. In future applications, the prediction system could be implemented in a variety of alternative methods; for instance, one of the programmable buttons on a mouse could turn on the system so when the user moves the mouse, the movement would be interpreted as issuing a command. Similar implementations could be used on a trackpad as well.

In the prototypes used here, the initial relative likelihoods of the commands were determined in advance and only roughly mirrored actual command frequen-

cies. In a real-world implementation of the system, the initial likelihoods could be based on data from typical users or could be dynamically updated from the user in question. Naturally, the latter would be better, but whether the gain would be sufficient to justify the costs is an open question.

REFERENCES

- Greenber, S., & Whitten, I. (1993). Supporting command reuse: Mechanisms for reuse. *International Journal of Human-Computer Interaction*, 39, 391-425.
- Guadagno, N., Lane, D. M., Batsell, R. R., & Napier, H. A. (1990). The predictability of commands in a spreadsheet program. *Interacting with Computers*, 2, 75-82.
- Lane, D. M., Napier, H. A., Peres, S. C., & Sandor, A. (2004). Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *International Journal of Human-Computer Interaction*, 18, 133-144.
- Linton, F., Joy, D., & Schaefer, H.-P. (1999, June 20-29). *Building user and expert models by long-term observation of application usage*. Paper presented at the 7th International Conference on User Modeling, Bauff, Canada.
- Murata, A. (1995, July 9-14). *Discussion on method for predicting targets in pointing by mouse*. Paper presented at the Proceedings of the Sixth International Conference on Human-Computer Interaction, Tokyo, Japan.
- Murata, A. (1998). Improvement of pointing time by predicting targets in pointing with a PC mouse. *International Journal of Human-Computer Interaction*, 10, 23-32.
- Napier, H. A., Batsell, R. R., Lane, D. M., & Guadagno, N. (1992). Knowledge of command usage in a spreadsheet program: Impact on user interface design and training. *DATA BASE*, 23, 13-21.
- Oirschot, H. K.-v., & Houtsma, A. J. M. (2000). Cursor trajectory analysis. *LNCS Haptic Human Computer Interaction*, 127-134.
- Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, 63, 129-138.
- U.S. Bureau of Labor Statistics. (2002). *National compensation survey: Occupational wages in U.S.* Washington, DC: Author.
- van Mensvoort, K. M., & Oirschot, H. K.-v. (2001). *Cursor trajectory prediction using a genetic algorithm*. Retrieved October 8, 2002, from <http://www.koert.com/work/cursorpredictor/>
- Wickens, C. D., Gordon, S. E., & Liu, Y. (1998). Upper-extremity cumulative trauma disorders. In P. McGeehon (Ed.), *An introduction to human factors engineering* (pp. 340-346). New York: Addison-Wesley.