Example applications of TA methods

Frank E. Ritter, 15 may 2013

I present two example tasks here, and analyze them with several techniques to illustrate the use of these techniques.

12.7.1 Adding a signature to emails

Adding a signature to an email is a common task that is performed by most users numerous times a day. Consider the task of signing the email shown in Figure 12.3 with "Cheers,<CR>Frank". The email application being used here, Eudora, offers two possibilities. Other mailers will often offer very similar choices of typing vs. using a command. The first is to just type this phrase into the email.

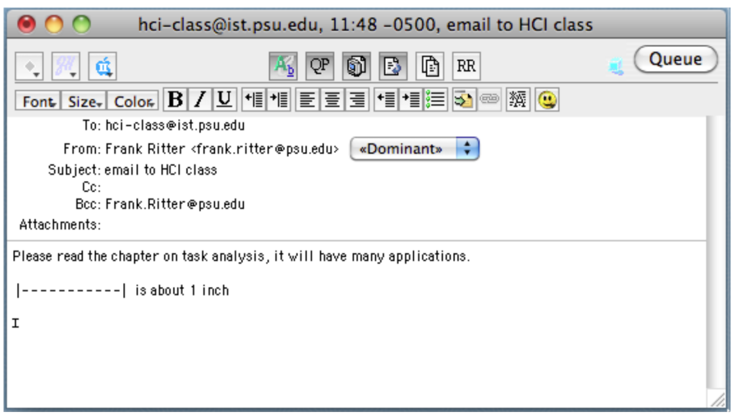

Figure 12.3 An email system waiting for a signature to be included. A one inch mark is included in the email for reference. The I-bar indicates where the mouse is positioned and where the user is typing in. (Screenshot by Ritter.)

The second way is to use the mouse, as shown in Figure 12.4. The mouse is moved from the I-bar cursor location to the signature icon below the middle dot in the menu bar, the mouse button is pressed, and then the mouse is dragged down to the "Informal" signature menu item and the mouse button is released.

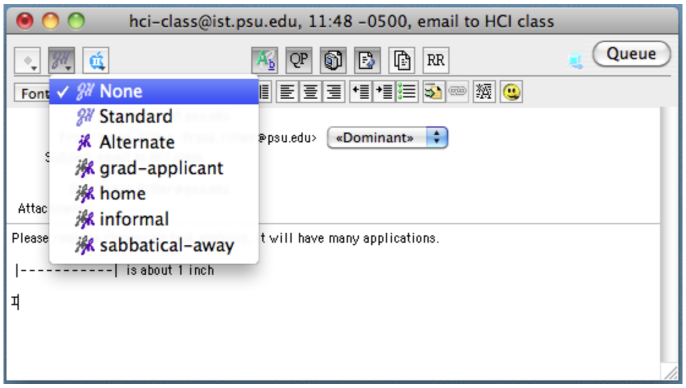

Figure 12.4 Entering the signature using the mouse. The user has moved the mouse from its position in Figure 12.3 and clicked on the signature icon. (Screenshot by Ritter.)

A simple GOMS analysis would note these two methods descriptively, as shown in Table 12-7. Although the analysis omits many details, it can be created very quickly. This level of analysis could be helpful when writing a manual, but does not provide much guidance to help the user choose between strategies for doing the task. It, like a logical analysis, would suggest that novice users will need to be able to map the task actions to the commands in the interface, and thus the interface should label its icons clearly and have its names match the user's task labels as closely as possible. It does look like the mouse option may take more actions, but they are smaller steps.

Table 12-7. A simple GOMS analysis of signing email.

Task sign email.

Method Type. Type the string "Cheers,<CR>Frank"

Method Mouse. Move the mouse to the signature button, select the "Informal" option.

Table 12-8 provides an initial KLM analysis of this task. This analysis uses the approximate times from Card, Moran, and Newell's book (1983, Ch. 8). This quick analysis suggests that the key press method will be faster than the mouse/icon based approach by about 1 s. The absolute difference is small, but the relative difference, about 20% is quite large. If the user carries out this task 30 times a day, they could save 30s a day by using the faster method (this adds up to about two hours over a year). This level of analysis is very quick to perform—less than 10 min. in this case—and the time spent would be recouped in four weeks if the results were applied to just one user.

Table 12-8. An initial KLM analysis of signing email.

Method keypress. Type the string "Cheers,<CR>Frank"

Assume the hands are on the keyboard.
No system response time.
Average typist

1 mental operator to retrieve the whole string, 1.35 s
Cheers,7 characters, 0.28 s * 7 = 1.96 s
<CR> 1 character = 0.28 s
Frank 5 characters, 0.28 * 5 = 1.4 s
Total: 4.99 s

Method mouse. Append the string "Cheers,<CR>Frank"

Assume the hands are on the keyboard.
No system response time.
Average typist

1 mental operator to initiate this task = 1.35 s
Move the hands from the keyboard to the mouse = 0.40 s
Move the mouse to the signature button = 1.1 s
Click on the signature button, 1 Mop and 1 click, 1.35 + 0.28 = 1.63
Move the mouse to the "Informal" option, 1.1 s
Release, 1 click = 0.28 s
Total: 5.86 s

A more accurate version of the KLM analysis, like that shown in Table 12-9, would take about twice as long to calculate, but would yield more accurate results. It requires more detailed information, such as sizes of targets and expected locations of the mouse to initiate mouse moves. The typing times have decreased. This is partially due to taking account of more detailed knowledge: this string is not a hard string to type. There is also likely to be some effect of assuming a faster typist. The more detailed predictions appear to be based on faster typists, as there are no times that are as long as the average typist time (0.28 s per keystroke) used in the rough analysis. Also note that the more detailed analysis found that the first analysis omitted the keystrokes required to get the capital C and the capital F (by pressing the Shift key).

Table 12-9. A more accurate KLM analysis of signing email, using detailed keypress times from Card, Moran, and Newell (1983, Figure 2-15).

Method keypress. Type the string "Cheers,<CR>Frank"

Assume the hands are on the keyboard.

No system response time.

Average typist

Using keystroke time of 0.28 (shift) and from Figure 2-15, CM&N.

1 mental operator to retrieve the whole string, “Cheers,CR Frank” 1.35 s

Using numbers from Figure 2-15

8 characters (shift) 0.28 + 0.153 (C) + 0.120 (h, alternative hand) + 0.128 (e) + 0.185 (e, same finger) + 0.149 (r, same hand) + 0.196 (s, same hand) + 0.164 (, alternative hand) = 1.375 s

<CR> 1 character = 0.28 s

Frank 5 characters, 0.28 (default shift) + 0.135 (F, alternative hand) + 0.215 (r, same finger) + 0.180 (a, same hand) + 0.132 (n, alternate hand) + 0.162 (k, same hand) = 1.104s

Total: 4.109 s

Method mouse.

Assume the hands are on the keyboard.

No system response time.

Average typist

1 mental operator to initiate this task = 1.35 s

Move the hands from the keyboard to the mouse = 0.40 s

Move the mouse to the signature button. distance = 1.66 inch, width = 0.25 inch. time = 100 ms/bit * $\log_2$ (1.66/0.25 + 0.5) = 0.28 s

Click on the signature button, 1 Mop and 1 click, 1.35 + 0.28 = 1.63

Move the mouse to the "Informal" option, time = 100 ms/bit * $\log_2$ (0.92/0.17 + 0.5) = 0.256 s

Release, 1 click = 0.28 s

Total: 4.2 s

The results from the more detailed analysis of the mouse method suggest that the two methods are more alike than the approximation suggested they were. Based on the more detailed analysis, the major differences remaining between these two methods are their sensitivities and residual results. The key press method is sensitive to typing speed. If the user is a faster typist than we assumed, they may prefer this method. Slower typists may wish to use the mouse method. The mouse method included more mental operators than the typing method. If this assumption is incorrect, the mouse method will be faster than the key press method. The mouse method also is immune to keystroke errors (but not to mousing errors). Finally, the mouse method leaves the hand on the mouse, which if the user is going to click on the "Queue" button to send the mail, makes the mouse method a little bit faster (because there is no need to move the hand to the mouse), but slower if they are going to type something.

These analyses suggest that these two methods of performing the task are fairly similar, and that the user will be able to choose pretty freely between them without penalty (if they can remember them!). Analysis will not always provide a simple answer that one method is faster than another. The relationship between methods may be more complex, and the conclusions may include caveats. It also answers the question of why some people are not sure which method is best: the reason that they appear equally fast is because in most cases the times for the two methods really are similar. Also, it reminds us to explain both methods to the user.

12.7.2 Keystroke accelerators

The analyses in Section 12.7.1 have some further lessons for interface design. If we use the analysis of the email signing task (Table 12-8, 12-9) to look at a cut command (Table 12-10), it suggests that a cut keypress (C-X) would be much faster than a cut command with a mouse.

This analysis suggests that keystroke accelerators (such as C-x or Command-X on the Mac) can be a little bit faster to a lot faster (like, 4x faster) than the equivalent mouse and menu commands. The pioneering work by Card et al. (1983) noted large differences in strategies that are possible, including how users searched for text (search, as used by experts, vs. moving down line by line, used by novices, which can be much, much slower).

How to help users find and use the faster methods remains a problem. Several studies have noted that users don't always use the most efficient approach (Fu & Gray, 2004; Lane, Napier, Peres, & Sándor, 2005). The analyses used here and also those in these studies show that keystroke driven iterations are faster than GUIs. So, "why do people use GUIs?" remains an interesting question. It may be related to feeling of knowing, practice rates, expected vs. actual task frequency, or as recently suggested, the ability to relearn interfaces (Kim & Ritter, accepted pending revisions).

So, one way to make interfaces faster to use (presumably allowing more work or play to be done, and also reducing working memory load and the effect of distractions) is to provide keystroke commands and help users learn them (an accelerator that is not used does not make the interface faster). A simple way to teach users is to put accelerators on menus. Word, for example, does this but somewhat inconsistently (many are shown, numerous are provided but not shown, and you can create your own, which are not shown, but must be remembered). Another way is to provide users assistance in finding and using more efficient approaches is making the trajectory of expertise visible to users (Lajoie, 2003). Indeed, your knowing now that experts use search rather than scrolling to find items in a document might help you be faster.

Table 12-10. A KLM analysis of cutting a piece of text using detailed keypress times from Card, Moran, and Newell (1983, Figure 2-15). Both methods assume that the text to be cut has been selected.

Method keypress. Type the string "C-X"

Assume the hands are on the keyboard.
No system response time.
Average typist
Using keystroke time of 0.28 (shift for control) and from Figure 2-15, CM&N.

1 mental operator to retrieve the whole string, 1.35 s
"C-x", Using numbers from Figure 2-15
2 characters (Control or Command) 0.28 + 0.153 (x) = 0.433 s

Total: 1.783 s

Method mouse.

Assume the hands are on the keyboard.
No system response time.
Average typist

1 mental operator to initiate this task = 1.35 s
Move the hands from the keyboard to the mouse = 0.40 s
Move the mouse to the Edit menu. distance = 6 inch, width = 0.25 inch. time =

$100 \text{ ms/bit} * \log_2 (6/0.25 + 0.5) = 100 \text{ ms} * 4.6 = 0.460 \text{ s}$

Click on the Edit menu, 1 Mop and 1 click, 1.35 + 0.28 = 1.63
Move the mouse down the Edit menu to "Cut". distance = 1 inch,

width = 0.25 inch. time = $100 \text{ ms/bit} * \log_2 (4/0.25 + 0.5) = 0.404 \text{ s}$

Release the mouse button. = 0.28 s

Total: 4.524 s

References

Card, S. K., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Fu, W.-T., & Gray, W. D. (2004). Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. *Cognitive Science, 28*(6), 901-935.

Kim, J. W., & Ritter, F. E. (accepted pending revisions). Learning, forgetting, and relearning for menu- and keystroke-driven tasks: Relearning is important. *Human-Computer Interaction*.

Lajoie, S. P. (2003). Transitions and trajectories for studies of expertise. *Educational Researcher, 32*(8), 21-25.

Lane, D. M., Napier, H. A., Peres, S. C., & Sándor, A. (2005). The hidden costs of graphical user interfaces: The failure to make the transition from menus and icons tool bars to keyboard shortcuts. *International Journal of Human Computer Interaction, 18*(2), 133-144.