GUI vs Command Line

“Once you’ve used a display editor, you’ll never want to go back to a line editor,” (Schneiderman). Overall, using Graphical Interfaces tend to speed up overall process of the project. Display of the final draft of documents, visible cursor action to the user, cursor motion through physically obvious and intuitively natural means, labeled buttons for actions, immediate display of results, rapid action display and simple “undo” actions are all tough command line tasks, if they are offered at all. This is where GUI may come in beneficial.

A full display allows viewing of each sentence in a context which simplifies reading and scanning documents. This is very inefficient when using one line a time when using line editors. I’d be like seeing the world through a narrow cardboard tube (Schneiderman).

Labelled buttons for actions can be one of the most appreciated add-ons to a GUI. Having clearly labelled buttons with hardwired commands such as INSERT, DELETE, BOLD, LOCATE are night and day differences. These act as a permanent menu selection display, constantly reminding the user of the available features that are offered. This can obviate the need for memorizing of commands to get the same results.

Rapid feedback is very handy when performing lengthy projects. When a executable button is activated, the cursor does at it is told on display for the user to see immediately. Deletion, insertion, and manipulating of the text can be tricky when running a command-based system, where the operator would be required to print or display command before the results of a change can be seen.

Mistakes in entering text happen maybe once in every five words. With a visual cursor, these can be corrected by backspacing and overstriking. Simple changes can be made by moving the cursor to the misspelled/grammatically incorrect word to be selected and overwritten with the intended correction. This easily reversibility reduces user anxiety over making mistakes.

In conclusion, graphical user interfaces may be a big advantage towards overall user productivity. Performance time of display editors is only half that of line-oriented editors, and since display editors also reduce training time, the evidence supports the enthusiasm of display editor developers.