



The Next UI Breakthrough: Command Lines

Want to know what I think the next UI breakthroughs will be? Here are two, both of which can be considered a return to fundamentals:

1. Command line languages
2. Physicality: the return to physical devices, where we control things by physical movement; turning, moving, and manipulating appropriate mechanical devices

This column discusses the return to command line interfaces. In a future column I will talk about the return to physical controls: physicality.

COMMAND LINE LANGUAGES

Once upon a time, computers were controlled by commands. The interaction paradigm was based on a control language; users entered commands and respective arguments on the line following the (invariably) blinking cursor.

Then graphical user interfaces (GUI) appeared, replacing arbitrary memorized commands with direct manipulation, where actions are performed by moving objects on the screen and selecting from menu displays. GUI has served us well, replacing the need to memorize geeky instructions with the ease of selecting from the visible items on the screen. But GUIs work well only when the number of alternative items or actions is small. When the number of items reaches the level typical of today's complex operating systems, applications, and the information spaces of the Internet, the GUI does not scale well. Even searching one's email records is tedious with a GUI, and when it comes to photographs and music, it's extremely difficult. The Internet, of course, cannot be navigated using just the visible structure.

What is to replace the GUI? Ah yes, journalists are constantly asking me that question, hoping I will speak of virtual reality implants by which we fly effortlessly through hyperspaces, finding just the items of interest, then immediately packaging and caressing them to do our bidding in reports, diagrams, and instant insights of wisdom. Well, the answer is much simpler, and it's already here: search. The real surprise, though, is that search engines have evolved into answer engines, controlled through a modern form of command line interface.

We navigate the Internet by typing phrases into our browsers and invoking our favorite search engine. But more and more, we type in commands, not search items. All the major search engines now allow typed commands, bypassing any intermediate Web pages to

directly yield answers. Consider these three examples, each for a different search engine.

- Google: The phrase "define: embodiment" provides the definition.
- Yahoo!: The phrase "time in Nagoya" provides the time (3:13 a.m. Friday, when I tried it last).
- Live.com: The phrase "cars in China" returns with "15 per 1000 people."

Even though these three services are called search engines, they are in fact becoming answer services controlled through their command line interfaces. The control language seems more ad-hoc than systematic, and the language forms are still spotty and idiosyncratic, but it is nevertheless a form of command line interaction. Each service uses a different language structure, and the three examples all work differently on the different services, or more accurately, fail to work except on the service for which it was developed.

These modern command languages have some major virtues over the ones in the past. They are tolerant of variations, robust, and exhibit slight touches of natural language flexibility. As a result, the requirement of strict adherence to syntax and form that characterized the earlier control line languages is not required. If an illegal command format is entered, the system gracefully retreats from the status of answer service to that of search engine, often returning pages that are of direct relevance.

The same power has now appeared within email and operating systems. Google's email system, Gmail, eliminates hierarchical menu structures for storing messages and instead provides a powerful search mechanism: search line interaction rather than GUI. Moreover, the recipient of the email can readily attach labels to individual messages to simplify retrieval and to allow different emails to be grouped together by common labels. Labels are the search engine's equivalent to folders. And Google is not alone. Microsoft's desktop search within the Vista operating system and Office 2007 provides similar power for files within the computer and Outlook email (although in this case, search is used as a supplement to file structures, so it is possible to organize and retrieve items through a combination of retrieval with the aid of search terms or by visually traversing the hierarchical file structure).

Both Apple and Microsoft now thoroughly integrate search into their operating systems. Both have pervasive search folders so that once a folder returns the results of a search, it can be saved and referred to as needed, with the contents dynamically changing

as the items in the computer change. Both have a rich command line format. Thus, to find this article in my Microsoft Vista file system, I enter a command into the search bar:

"command line folder:interactions type:doc"

The request finds all Microsoft Word files in any of the folders on my system with the word "interactions" in the folder name and the words "command line" in the contents or title. Similarly, to find my latest email exchange with the editors of <interactions> about the schedule for submitting manuscripts, I replace the search query with:

"schedule folder:interactions type:outlook"

I use these examples to illustrate the transformation of search into a truly functional command line interface. At the moment, the power of command lines within the computer is primarily as an aid to search rather than a means of executing actions and applications (although there is some ability to do so). When it comes to finding documents, these control structures really are advanced. We are looking for answers, not intermediary documents. Navigating our way through thousands of hierarchically organized files is a necessary nuisance. Now I seldom try to navigate my file structure, even though it is very efficient and quite well-structured. Why not? Because I have a half gigabyte of data in my mail files, 700 folders and 6,000 files in my documents folder, and 4,500 photographs—far too much to search through visually. And now, with powerful search command language, plus labels and keywords that can be added to files, emails, and photos, it is possible to go directly to the desired item—or at least, get very close. The old command line interface is dead: Long live the new, much-improved command line.

The new command lines are far more flexible and robust than those of the past. Word order is not critical; often, we can use synonyms or even related terms. Spelling accuracy is not even required, for the

systems can use their own knowledge to correct spelling errors, or at least, to suggest spelling variants. Search is never anything I want to do. I don't want to search—I want to know something. I want an answer engine, not a search engine. The new command line interfaces still have a way to go. They have appeared serendipitously, as system developers slowly expanded the capabilities of search tools. But now it is time to recognize them for what they are—and for how much better they could become.

The power of command line interfaces has not been lost on developers. Several companies provide powerful tools for adding commands to your favorite operating systems. Some systems already allow free input. When using Microsoft's Outlook calendar system, the dates can be entered in English syntax: "two weeks from Tuesday," for example, yields the correct date. What a relief from the formal, fixed, (and secret) format so many systems require for dates. One correspondent begged me to suggest the switch to natural language interfaces as the natural outcome of command line languages. Sorry, natural language is intended for natural people who share what psycholinguists call "common ground"—for language in isolation is ambiguous, confusing, and non-specific. But where the task is well-defined, natural language wins—as in specifying the date as "one week from tomorrow."

Command line interfaces. Once that was all we had. Then they disappeared, replaced by what we thought was a great advance: GUIs. GUIs were—and still are—valuable, but they fail to scale to the demands of today's systems. So now command line interfaces are back again, hiding under the name of search. Now you see them, now you don't. Now you see them again. And they will get better and better with time, mark my words. This is my prediction for the future of interfaces.

Isn't progress wonderful? ◆

Disclaimer: Microsoft is a client of the Norman Nielsen group.

© ACM 1072-5220/07/0500 \$5.00

ABOUT THE AUTHOR Don Norman wears many hats, including cofounder of the Nielsen Norman Group, professor at Northwestern University, and author; his latest book is *Emotional Design*. Some of the material in this column is from his forthcoming book, *The Design of Future Things* (in preparation). He lives at www.jnd.org.