



digital librarianship

Search Software Usability Issues



▼

“Not even the most intuitive search software can make the information content useful if the search results are not displayed in an intuitive format.”

Judging the usability of search software (or any software for that matter) is almost like critiquing art or explaining what you found so attractive in the boy or girl you had a crush on back in elementary school. It is as hard to define what makes a software product usable as it is to explain what makes a Frida Kahlo self-portrait “art.” You either experience it or you don’t. It is like charisma or charm—it’s hard to define or nail down.

Trying to Nail Down the Elusive Concepts

There are many very specific features that make a software product powerful and intelligent. I have my own 30 or so criteria that I check when evaluating search software, but almost all of them are related to specific functions, like truncation, stemming, implicit and explicit proximity between search terms, sorting by certain data elements, marking records from a set, etc. Usability is something different, something that makes a software product appealing and usable for novice and experienced users alike. It is a mixture of interface design, appearance, functionality, and more.

I don’t want to wax lyrical, so I did a quick concordance search to find out which were the most-often-used adjectives in the many software reviews I’ve written. Well, it was a no-brainer: The adjective “intuitive” won hands down. I double-checked it with a more traditional search and the results were clear. In the

318 articles I found in the Trade & Industry database under the correct spelling of my name, there were 85 that had the adjective “intuitive.” Mind you, not all of my writings touch upon software issues, so the frequency of this adjective in my software reviews is even higher.

Those who don’t like my reviews may find this as proof of how limited a vocabulary I have or how single-minded and monotonous my evaluations are. I find it as proof that intuitiveness is a key aspect in my evaluations of search software.

I knew that I often used the adjective “intuitive” to describe the essence of usability (because saying it is “usable software” would sound like a backhanded compliment or a gross understatement for software that makes itself very usable), but this excessive proportion of “intuitive” surprised me. One reason for its preponderance is exactly its omnipresence in a good software product: It’s in the interface, in the overall navigation within a database, in specifying multiple words as an exact phrase in a search term, in choosing limit options, and even in logging off from a system. It has to do with the layout, the positioning of buttons, the choice of color, the choice of words, the expected syntax or notation in a query. Let me illustrate these through some negative examples, to show you what makes a software package, or at least a software function, counterintuitive.

Color: Navigation is the most essential step in using any software. It defines

what the next steps should be. I found it very intuitive when SilverPlatter chose to display in green the button for the most logical next step in the search process. But I find it counterintuitive when Dialog uses red on its Continue button. To the best of my knowledge, in every culture red implies stopping or being cautious, rather than encouraging an action. Such symptoms may raise a—well, a red flag—about the interface.

Grayed-out buttons or labels are used to indicate when a function is not available. So it is frustrating to click on the blue abstract label in the Wiley Inter-science archive just to get the message “No abstract available.” When no full-text or PDF file is available for an article those labels don’t show up, so why give a misleading clue for nonexistent abstracts? The software could be easily programmed to not display that label if no abstract is available.

Directional Signs: Understanding signals that tell you which direction to go can be confusing, as we know from everyday traffic. Even assuming that bobbing your head is the universal symbol for a “yes” to a question can be utterly confusing if you visit Bulgaria. If you ask a local “Shall we dance?” and she bobs her head but remains seated, you get a very mixed message. The bad news for you is that, in Bulgaria, bobbing means no and horizontally shaking the head means yes.

Almost all of our friends who go through Honolulu airport have complained how misleading the sign is that points to the lane that you’re supposed to follow for dropping off your rental car. They are right. Similarly, I almost always press the wrong button in Ingenta (which otherwise has one of the best-designed interfaces overall) when browsing a journal volume by volume. To have the Next button on the left showing a back arrow and the Previous button on the right with a forward arrow, is strange even for the British, who still drive on the left side of the road. I understand that, because the volumes and issues appear in reverse chronological order, the arrows also have a reverse meaning. But it still causes cognitive dissonance, since in all other cases Ingenta uses the directional

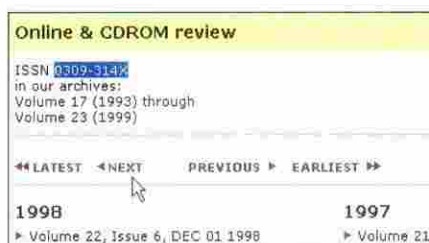


Figure 1: Arrows and labels done the unconventional way in Ingenta

signs and verbs in the traditional ways. (To put things in perspective, it is more of a problem that the ISSN in the example is wrong; it belongs to the previous title of the journal, *Online Review*—but that’s probably part of the baggage that came with acquiring UnCover.)

Syntax Conventions: This is probably the territory where you would find the most counterintuitive examples. A space between two or more words in the query usually means adjacency relation between the words, as in this example: New Mexico. In Dialog, space in the Select command implies that you are searching for New Mexico not only as an exact word pair in the given order but also as an exact descriptor, which would not retrieve the records with the descriptor New Mexico, NM. This is the most common error that students make, as it is utterly counterintuitive. The Find command, which was introduced a couple of years ago, looks for unidirectional adjacency in the basic index (second word to follow the first one in the title, abstract, descriptor, and full text), but it is not mentioned in textbooks and is little-known even by professional searchers who luckily have mastered the unusual Select New () Mexico syntax for unidirectional adjacency searching in the basic index of Dialog databases.

Similarly, it is not intuitive that the otherwise very good ProQuest software uses the ? symbol for unlimited truncation, and the * symbol for single-character truncation. In those search programs that use these symbols, it is always the other way around, and so it has been in the DOS and Windows operating systems for a very long time. It is a de facto standard for the use of these two symbols. I almost always use it incorrectly, but luck-

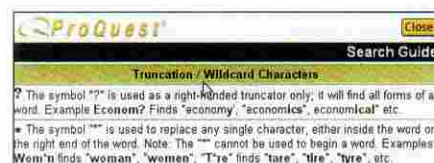


Figure 2: Unconventional use of the ? and * symbols

ily, the results (like technolo* returning no results or very few results) remind me of the counterintuitive syntax.

Misnomers: Giving a wrong name (or one that flies in the face of the common meaning) to a function is one of the most irritating of the counterintuitive features. Among its many deficiencies and misinformation, PubSCIENCE had these, too.

Limiting a search using the ALL options meant to retrieve current information (as opposed to Archive information). The text said that “All specifies information from 10 years to present.” While this may have been true in 1999, it was definitely not true in 2000, 2001, and 2002. One also wonders, why use ALL when meaning the past 10 years instead of all the coverage of several decades? Offering this ALL option as an entry in limiting by publisher (along with real publisher names followed at the end by Archive for good measure) was another highly counterintuitive idea. It’s discombobulating for all users, I guarantee.

What Good Is Software That’s Not Intuitive?

I could show you many more examples of unintuitive and counterintuitive “features” that make a software solution less usable. One or two of these would not make the software unusable, but along with other mistakes they can be fatal. Not even the most intuitive search software can make the information content useful if the search results are not displayed in an intuitive format. ▲

Péter Jacsó is associate professor of library and information science at the University of Hawaii’s Department of Information and Computer Sciences. He is also a columnist for Information Today, and a popular conference speaker. His e-mail address is jacso@hawaii.edu.