“Finding and Reminding” Reconsidered

Scott Fertig, Eric Freeman and David Gelernter

Research Report YALEU/DCS/RR-1090
September 1995

This report was supported in part by AASERT grant # F4-9620-92-J-0240.
It appeared in the *SIGCHI Bulletin,* vol. 28, no. 1, January 1996.

“Finding and Reminding” Reconsidered

Scott Fertig, Eric Freeman and David Gelernter

INTRODUCTION

In the July 1995 SIGCHI Bulletin, Deborah Barreau and Bonnie Nardi rightly point out that “every computer user spends enormous time and effort in filing and finding of electronic files, yet there has been very little research on the subject.” To this end, Barreau and Nardi have investigated electronic filing and finding practices of the users of common desktop systems to determine “the factors affecting individual decisions to acquire, organize, maintain, and retrieve information.” While we applaud their efforts to study the most basic aspects of user/computer interaction, we believe they draw the wrong conclusions from their own research. Our goal in this paper is to explain why.

From two studies, with a total of 22 subjects (four DOS users, one Windows 3.1 user, one OS/2 user and 16 Macintosh users), they noted the following similarities among all the users:

1. A preference for location-based search for finding files (in contrast to logical, text-based search);
2. The use of file placement as a critical *reminding function*;
3. The use of three types of information: *ephemeral*, *working* and *archived*;
4. The “lack of importance” of archiving files;

and further conclude that these similarities represent fundamental user practices and preferences that are independent of operating system and level of experience.

We believe that conclusion three gives us a useful categorization of the user’s information space and previous studies have reported consistent findings [6]. Conclusions one, two and four however, are artifacts of the narrow scope of the systems studied rather than general statements of the way users acquire, organize, maintain and retrieve information. Both studies focus on the common desktop metaphor which favors certain types of interaction over others. In this light, the reported patterns are unsurprising because the user interfaces for the Macintosh, Windows and OS/2 platforms are close relatives.[1] We believe we are doing more than commenting on three minor points of their work; rather we are suggesting a more fundamental problem with their analysis that is analogous to concluding that radio listeners of the 1920s preferred headphones for listening, despite the fact that radios with speakers had not yet been invented. Or studying stereo owners of the 1950s and concluding that there was “a lack of importance” of high-fidelity systems because the vast majority of people listened to poor-fidelity record players. Today, we know that people prefer high-fidelity. We believe future research should broaden the scope of analysis and consider not just current practice but other possibilities.

In this article we comment briefly on Barreau and Nardi’s analysis, pointing out where and why we think they have drawn the wrong conclusions. We then mention a few systems that use different non-desktop interaction metaphors that should be included in future studies of this type.

ANALYSIS

Preference for Location-based Search

Barreau and Nardi describe location-based search as the process whereby a user “takes a guess at the directory/folder or diskette where she thinks a file might be located, goes to that location, and then browses the list of files or array of icons in the location till she finds the file she’s looking for. The process is iterated as needed.” The alternative, as described by Barreau and Nardi, is logical finding, where a text-based search of keywords and filenames is used to search for files. This functionality was provided by the Macintosh “Find” and DOS “whereis” utilities in their studies.

Barreau and Nardi conclude that users prefer to find files by using location-based cues over text-based search approaches. They hypothesize that users may prefer location-based searching because it “more actively engages the mind and body and imparts a greater sense of control.” They further hypothesize that users dislike text-based search because they have to“[sit] there waiting for the computer to return a list of files that may or may not be relevant.” Barreau and Nardi also found that filenames were used for the purpose of “jogging the memory” rather than for the purpose of search. However, they report that if users could not find a file within a couple of tries they then turned to the “find” feature to search for it.

First, note that location-based finding is nothing more than a user controlled “logical search.” In location-based finding the user searches the file collection relying on mnemonic aids and his memory of past events to lo-

[1] The legal wrangling over “look and feel” reflects this fact.

cate a file. This scheme is not without faults; it can be error prone and time consuming. Barreau and Nardi pointed out in their own study that a user could not find a file that had been created a mere several hours earlier and remarked, "What did I call that file?"

It is entirely possible that their subjects preferred location-based search because it was the lesser of evils: if other search methods are slow, difficult, or only operate on file names (not contents) then location-based search may not seem so bad. Moreover, "whereis" and "Find" are hardly state of the art in logical search. More recent systems provide incremental indexing of file contents and significantly reduce search time while increasing accuracy [7, 4]. Inclusion of these better search techniques into current systems could sway results toward logical search.

We don't argue that screen layout and organization based on conceptual locations isn't useful. In many cases it can help the user maintain a sense of context about his workspace. We do argue that using virtual location as a basis for organizing information and personal document collections is often as *hit-or-miss* as a logical search mechanism. Location-based search has many problems: How do we maintain file collections over long periods of time? How well does it work when more than one user is involved? The way information is used changes over time—how well does the location-based scheme handle this? What about scalability? We believe that location-based search is only possible when users don't archive, or give up using archived information. If archiving is unsupported or difficult then other search mechanisms become less important and relying on short-term cues, such as location, is possible. Barreau and Nardi have dismissed archiving, questioning the "supposed coming information overload." By contrast, we have no doubt the problem is *already* here. We examine archiving further below.

Reminding

Malone [6] was one of the first to point out the importance of reminding in our paper-based systems and suggests their inclusion in computer-based systems. Yet today, software systems provide little support for reminding. While a number of time management, scheduling, and "todo" list applications have come to market, they don't represent an integrated effort in providing users with this basic capability.

Barreau and Nardi observed that computer users often use a file's location as a critical reminding function. For instance, at the end of the day a Macintosh user may leave files on his desktop as a reminder of work to be done the next morning. Other users left electronic mail messages in their in-box to remind them of meetings. Like Barreau and Nardi, we believe reminding is an important capability that software systems should support. Unlike Barreau and Nardi, we find the use of location-based storage an unsatisfying, easily undermined method of creating reminders. Moreover, we see the use of location for reminding as a simple coping strategy for lack of anything better. The desktop metaphor has no semantic notion of location-based reminding, and, as the authors point out, this reminding technique amounts to a "behavioral trigger" that reminds users to take some action when they observe files in certain locations. In summary, location-based reminding amounts to an *ad hoc* user convention and its problems are obvious: there is no way to insure that a reminder actually reminds you; lack of sufficient screen real estate; inapplicability to long-term tasks; bad fit to collaborative work, etc.

Archiving

Barreau and Nardi claim that "old information is generally not useful" and so there is a "lack of importance of archiving files." While we concede that over time, old information is *generally* less likely to be valuable, situations occur when old information is essential. We can all recall times when we needed information we threw away a week, a month or a year ago. In fact, Cook's work [1] has shown that archiving information can be critical in an organizational setting.

Barreau and Nardi found that users in their studies did not archive or rely on archived information. Once again we believe these findings are artifacts. Consider the "cart before the horse" explanation—that is, if archiving information is so difficult that it deters users from archiving (and this is what Barreau and Nardi have observed), then users obviously will not depend on archived information. This leads us to wonder how users would use old information if it were convenient to store and access. If software systems handled archiving and retrieval more conveniently we might find that old information is reused more often. The underlying problem is that location-based storage and archiving are conflicting goals. Location-based storage assumes a small information collection (basically what the user can remember) and does not scale to large collections of information. But information is not always needed in the same way (and thus, not in the same location) it was originally. Archived information is often needed in a context that is different from the one in which it was created, and in a different location.

The desktop and file&folder metaphor were created so that users could relate their computer-based systems to the paper-based systems they were used to. Yet paper-based systems are first and foremost archiving systems. They accommodate ephemeral and working information but the state of the art in both these areas still seems to be a messy desktop.

NEW METAPHORS

As we have pointed out, we believe that Barreau and Nardi's findings are mostly artifacts of the desktop and file&folder metaphor. The desktop metaphor was created on analogy to our paper-based world. Our computer-based systems can do better. There are many emerging systems that go beyond our traditional file systems and user interfaces that are ripe for study. Here we briefly mention three of them: (1) the dynamic queries of Shneiderman (2) the virtual directories of the MIT Semantic File System, and (3) our own system, Lifestreams, which uses a time-based metaphor and fast logical searches to organize, monitor, find and summarize information.

Dynamic Queries

Shneiderman's dynamic queries [8] combine direct manipulation and database visualization to allow a user to rapidly filter information through the use of visual components such as sliders and buttons. User manipulation results in visual feedback within 100ms, allowing him to quickly perceive patterns in the data. Visual queries have been applied to a number of domains such as geographic database systems, movie databases, and educational applications. Visual queries have also been implemented in the form of a Unix directory browser [5]. Shneiderman *et al* found that, with the browser, user queries could be "answered more rapidly because users can filter out irrelevant information and visually scan the remaining information." The location-based alternative (i.e., using the Unix from the command line)[2] "requires more time because users must visually scan a much larger set of information." The browser work is a first step, and as Shneiderman *et al* point out, more work needs to be done integrating visual queries into our day-to-day applications.

We believe visual queries are a promising method of locating information in a file system. Shneiderman reports that the "enthusiasm users have for dynamic queries emanates from the sense of control they gain over databases." As we have mentioned, Barreau and Nardi made similar statements about location-based systems; obviously there is no sense of location in Shneiderman's system, yet users report similar feelings.

Semantic File System

The MIT Semantic File System [4] provides associative access to a file system via *virtual directories.* Using native directory commands (such as `ls` and `cd`), virtual directory names are interpreted as associative queries. The results of a query are computed via an automatically indexed set of attributes (field/value pairs). This index is generated by a number of *transducers* that map files of specific types (e.g., C files, TEXfiles, etc.) to a set of attributes.

The contribution of the Semantic File System is not the method of indexing but the ability to describe a desired view of the file system's contents. This description maps to no actual folder or directory of information but to a virtual one computed on demand. Indexing is important, however, because it guarantees acceptable response time on queries in contrast to the Macintosh "Find" and the DOS "whereis" utilities. Indexing also enables searches on a file's entire content.

The authors describe results that show reasonable performance on a realistically sized file system; precise queries are answered in the one to two second range. Their own experiments "suggest that semantic file systems can be used to find information more quickly than is possible using ordinary file systems." In contrast to Barreau and Nardi's observations of location-based finding among DOS users, users of the semantic file system create virtual locations in one step through the use of logical search.

Lifestreams

Our own work, Lifestreams, is a new model and system for managing personal electronic information. Lifestreams was first proposed in [3] and is described in [2]. Lifestreams uses a simple organizational metaphor, a time-ordered stream of documents, to replace conventional files and directories. The system acts as an electronic diary; every document you create is stored in your lifestream, as are all the documents other people send you. The tail of your lifestream contains documents from the past, starting in principle with your electronic birth certificate. Moving away from the tail and toward the present, your stream contains more recent documents such as papers in progress or the latest electronic mail you've received—other documents, such as pictures, correspondence, bills, movies, voice mail and software are stored in between. Moving beyond the present and into the future, the stream contains documents you *will* need: reminders, your meeting schedule, your todo lists.

Users organize, locate and monitor incoming information through stream filters that result in *substreams.* Substreams differ from conventional directory systems in that, rather than placing documents into fixed and rigid directory structures, they create virtual document organizations (much like semantic file systems). Users may allow substreams to persist and act as organizational structure, or—because creating and destroying substreams is inexpensive—they can be used to locate information quickly. Substreams are dynamic. Persistent substreams continue to collect new documents that match their search criteria. A substream can be summa-

[2] Barreau and Nardi describe DOS as a location-based system, we do the same (for comparison) with the UNIX file system.

rized to distill it into an overview document. The content of the overview document depends on the type of documents in the substream. For instance, an overview of a substream that holds the daily closing prices of stocks in a portfolio may contain a historical investment performance chart.

The historical nature of the stream is important. The present portion of the stream acts as a workspace, holding "working documents"; typically this is where new documents are created[3] and where incoming documents are placed. Most newly-created documents hang around in the present for some time before they become read-only and are pushed off into the past, being automatically archived in the process.

The future portion of the stream allows documents to be created in the future (unlike the paper-based world, computers can defy space and time). Allowing future creation gives us a natural method of posting reminders and scheduling information. Our system allows the user to dial to the future and deposit a document there, say, a reminder of your birthday. When your birthday arrives the note appears in the present and reminds you.

How do Lifestreams match the way people work?

Lifestreams are a metaphor for the way people work. Heavily used and recent information is stored in the present part of the stream.[4] Older information is automatically moved into the past and out of the users view. Anytime the user needs to filter out information or find older documents he can create a substream.

Locating information: Lifestreams allows the user to locate information in several ways. Ephemeral and working information is typically in the "present" part of the stream. This is actually very similar to the location based approach; the user returns to a common area to locate files. Lifestreams are more flexible, though, in the sense that the user can tailor, on the fly through substreaming, what the present part of the stream looks like. Ephemeral information such as reminders and electronic mail arrives in the present; the user is alerted when it does. Users can quickly escape information overload by working in a substream that removes such interruptions or narrows their focus to the task at hand. They can quickly search for archived information through substreaming, or set up organization categories by letting substreams persist. Unlike directories, substreams continue to collect information dynamically.

Reminding: This is an integral part of Lifestreams and built into the semantics of the model. Users create documents in the future that alert them by arriving in the present. Users can also mail "future" documents to one another (we use this functionality in our own workgroup). Future documents also act as place holders for meeting schedules and software agents can take advantage of the stream structure to assist in intelligent scheduling and reminding.

Archiving: We have mentioned two ways the desktop metaphor prevents archiving: archiving information is difficult and so is retrieving archived information. Lifestreams solves both problems: archiving is automatic because older information is pushed into the past and out of the user's view. Archived information is easily retrieved via substreaming. Moreover, users can quickly distill large amounts of archived information down into meaningful summaries.

CONCLUSION

To make general claims about "information use" in the narrow scope of today's desktop operating systems is a mistake. While Barreau and Nardi's work will be helpful for improving existing systems, their results can not be extrapolated to general statements about the way users acquire, organize, maintain and retrieve information—doing so requires the study of users in environments that include non-desktop metaphor systems. Future studies may reveal very different preferences when users are provided with a richer and more functional interaction environment.

REFERENCES

1. Terry Cook. Do you know where your data are? In *Technology Review*. MIT, January 1995.

2. Eric T. Freeman and Scott J. Fertig. Lifestreams: Organizing your electronic life. In *AAAI Fall Symposium: AI Applications in Knowledge Navigation an d Retrieval*, November 1995. Cambridge, MA.

3. David Gelernter. The cyber-road not taken. *The Washington Post*, April 1994.

4. David K. Gifford, Pierre Jouvelot, Mark Sheldon, and James O'Toole. Semantic file systems. In *13th ACM Symposium on Operating Systems Principles*, October 1991.

5. H. Lio, M. Osada, and Ben Shneiderman. Browsing Unix directories with dynamic queries: An analytical and experimental evaluation. *Proc. Ninth Japanese Symp. Human Interface*, pages 95–98, 1993.

6. Thomas W. Malone. How do people organize their desks? Implications for the design of office information systems. *ACM Transactions on Office Systems*, 1(1):99–112, January 1983.

7. Udi Manber and Sun Wu. Glimpse: A tool to search through entire file systems. Technical Report 093-34, Department of Computer Science, The Univesity of Arizona, October 1993.

[3] Documents can also be created in the future, but we disallow document creation in the past to maintain the historical perspective of the documents.

[4] Our user interface by default displays a users stream from present to past, hiding the future part of the stream.

8. Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, pages 70–77, November 1994.

BIOGRAPHY

Scott Fertig is a research scientist at Scientific Computing Associates, Inc. in New Haven, Connecticut and a research affiliate in the computer science department at Yale University. His research interests include parallel and distributed processing, artificial intelligence and database mining. Along with David Gelernter, he is the creator of the FGP software architecture that uses similarity-based reminding to discover unexpected patterns in data.

Eric Freeman is a doctoral candidate in computer science at Yale University. His research interests include the design, implementation and application of distributed systems, information systems, and programming languages. His thesis topic is Lifestreams. He received an MS and MPhil from Yale University in 1994 and an MS from Indiana University in 1991. In 1990, he was awarded a NASA Graduate Research Fellowship.

David Gelernter is a professor of computer science at Yale University. His research interests include programming languages, software ensembles, and artificial intelligence. He codeveloped the coordination language Linda. He has cowritten textbooks on parallel programming and programming language design; he is author of Mirror Worlds (Oxford, 1991) and The Muse in the Machine: Computerizing the Poetry of Human Thought (FreePress: New York City, 1993).

All authors can be reached at the Department of Computer Science, Yale University, P.O. Box 208285, New Haven, Connecticut, 06520.

ACKNOWLEDGMENT

The authors gratefully acknowledge the assistance of Elisabeth Freeman for her comments on earlier drafts.