

Internet Interfaces¹

sean dreilinger

November 9, 1993

¹Written for a course in Human Computer Interaction & User Interface Design at the UCLA Graduate School of Library & Information Science

Contents

1 Learning The Basics	3
2 Ease of Use	4
3 Facilities & Software	4
3.1 Networks and Databases	4
3.2 Software and Operating Environments	4
4 Requirements	5
5 User Education	5
6 Experiences	5
6.1 Penpal & Netnews	5
6.2 Three different online catalogs	5
6.3 Two different non-catalog databases	6
6.4 Gopher name location	6
6.5 Archie to locate software	6
6.6 Gopher and WAIS to locate relevant documents	6
7 Similarities	7
8 Differences	7
9 Human Factors	7
9.1 Time to learn	8
9.2 Speed of performance	8
9.3 Rate of errors by users	8
9.4 Retention over time	8
9.5 Subjective Satisfaction	9
10 Suggested Improvements	9
11 Conclusion	10
12 Figures	10

List of Figures

1 Interdependent layers in a single network connection	10
2 Five Different File Transfer Protocol Interfaces	10
3 Online Catalogs and Databases	10
4 Macintosh Client-Server Utilities: WAIS and Archie	11
5 Crash Landings	11

1 Learning The Basics

How long did it take you to learn each of the basic capabilities (ftp, gopher, archie, etc.)? For my *first* experience with each type of software, it took me about five minutes to learn the concept and twenty minutes to learn the basic capabilities and features for that particular interface. When approaching the same software utility on another operating system or environment, I already knew its major function(s) and focused on comparing the way to accomplish those functions within a different interface. For example, one of the basic capabilities of ftp software is ‘*getting*’ binary files from a remote computer. The way user accomplishes this with a command–line ftp interface is much different than the way a user ‘*gets*’ the *same* file from the *same* remote computer using a graphical interface for ftp.

My first experience with *gopher* software was with the University of California at San Diego’s *Infopath* Campus–Wide Information System (CWIS). If the basic capability of gopher is navigating Menus, I don’t think I’ll ever get it—I’ll always get lost between the first menu and the menu I am seeking. I was quite happy to discover that gophers offer a ‘bookmark’ feature to find an obscure item quickly and repeatedly. I already knew that most gopher clients can make telnet connections to other systems and that they can transfer and save text files. I was pleased to learn the graphical gophers are capable of transferring a variety of other file formats, first loading them into memory and viewing them on the client computer, and then *optionally* saving any desired item.

Archie searches on the GSLIS–LAN are a value–added version of the DOS Archie client. One capability that is still beyond my grasp is the option to search for filenames using a *regular expression* search. Without this knowledge, a user can do little more than make a wild stab at a filename, and hope that it turns up some results. The Macintosh Interface was much friendlier to both me as a user, and to the remote sites it connected with. It offered the capability to search multiple filename–strings at once—which is great for someone like me, doing haphazard, best–guess searches.

I’m not sure that WAIS (Wide Area Information Server) software *has* any ‘basic’ capabilities. When demonstrated by an experienced user, the potential of this sort of client–server software is amazing. Information from all sources, in all formats, from all around the world, is delivered to the user’s desktop. In practice, I was not so lucky. The first two Microsoft Windows clients, WAISman 3.0 and jWAIS, locked up and crashed so frequently that they have been removed from the GSLIS–LAN. The remaining Client, WAIS 2.2 for Windows, also tended to lock up before it could return results from any remote computers. WAIS via the OAC’s gopher client performed reliably, but its information format was limited to textual information—there was no capability to transfer graphics, formatted documents, or any other binary files. Only the Macintosh WAIS software was installed, configured, and designed well (and lucky?) enough to live up to its promise.

A variety of electronic mail interfaces are used to keep in touch with my penpal, Ka_Wai Ho in Australia, as well as other friends. The mail interfaces vary widely in their ease of use and features. While the OAC’s *ben* software was difficult to learn because it has no concept (every action is a *gamble*), other software for electronic mail is a pleasure to use and *also* difficult to learn all of the clever features—software like *elm* for UNIX and *Pegasus Mail* for DOS.

I’ve been using two Usenet News clients for two years now, and I’m still getting the hang of it. I also tried two additional newsreader interfaces for this project: *Trumpet* for DOS, which crashes on the GSLIS–LAN, and *Trumpet* for Windows, which I couldn’t figure out how to install. The UNIX and MVS Newsreaders are being used to ‘lurk’ in the *comp.hci* newsgroup. Perhaps I’ll get the nerve to post there later in the quarter.

Once I had tried an Internet software utility on one platform, learning the others was a matter of pattern recognition—comparing features in the present interface with those I had learned from previous interfaces. How does one escape from the OAC’s telnet software? How does one escape from the TNVT220 telnet software on the GSLIS LAN? How does one escape from the TN3270 telnet software in Microsoft Windows? The same task in each interface requires a different action by the user.

2 Ease of Use

How easy or difficult were they to learn? The first interface to any software utility or system was the most difficult and time consuming to learn. Some of the difficulty with different software and systems was due to circumstances beyond my control—the WAIS for Windows software continually crashed the entire computer. The Windows and Macintosh Gopher clients were not fully or properly configured. Several remote library catalogs did not respond to a telnet ‘open’ request.

Top Prize for an easy-to-learn-and-use Interface goes to the *Rapid Filer* for LAN Workplace ftp utility. This graphical ‘front end’ to ftp can connect to a remote site on autopilot, login as an anonymous user, and display both remote and local files in an easy to understand ‘folders and documents’ iconic representation. The user can then drag-and-drop these icons to transfer files from one site to another, while the *Rapid Filer* software takes care of the binary/ASCII file type determination, file re-naming, and other hassles.

Bottom Prize for the most-cryptic-and-elusive Interface goes to the WAIS for Windows version 2.2. The process is too confusing. all of the interfaces attempted to communicate with the user through inconsistent icons. Many icon-based graphical software packages offer ‘bubble help,’ or they automatically display the function of each icon in the application’s ‘status bar’ when the mouse cursor floats above the icon in question. Our network setup prevents a user from saving a collection of WAIS servers into a file, so every Windows WAIS search at GSLIS requires the user to begin with an additional search *for the WAIS servers themselves!*

3 Facilities & Software

Indicate which interfaces and facilities you used

3.1 Networks and Databases

FACILITIES:

GSLIS LAN, IBM Ring
 GSLIS LAN, Macintosh Ring
 OAC, telnet3270 connection
 OAC, dial-in TVI950 connection
 UCSD sdcc LAN, local terminal
 UCSD sdcc LAN, dial-in VT220 connection
 Invisible College (my bedroom network)

REMOTE DATABASES:

Melvyl
 Nova Net Library System (Nova Scotia)
 ETHICS (Sweden)
 Dialog
 Melvyl COMP
 Melvyl CC

3.2 Software and Operating Environments

INTERFACES	TELNET	FTP	EMAIL	ARCHIE	GOPHER	WAIS	NEWS
DOS	TNVT220, TN3270	Novell ftp	Pegasus Mail, cc:Mail	GSLIS- LAN batch file to run Archie	gopher	–	Trumpet
MS WIN- DOWS	TN3270	Rapid Filer for LAN Work- place	Pegasus, WinPMail, cc:Mail	–	hGopher, gopher	jWAIS, WAISman 3.0, WAIS 2.2	Trumpet
MACINTOSH	NCSA telnet TN3270	Fetch	–	Archie Beta ver .9	turbogopher	WAIS	–
VAX (OAC)	telnet	User ftp	tso mail, ben	–	OAC go- pher via InfoUCLA	WAIS via InfoUCLA	rn
UNIX	Linux tel- net X3270	Linux ftp	Linux mail, elm, pine, xmh	–	UCSD’s In-foPath gopher	–	rn

4 Requirements

What expertise seems to be required to use each of these systems and capabilities? *Domain Knowledge* of computers and internetworking is very helpful for understanding and using the current generation of Internet software and systems. *An understanding of the information being presented or retrieved* is also important. It does a user no good to enter an Archie search and get back 25 perfect ftp sites, directories, and filenames in a text file, if they do not understand: 1. what a file name is. 2. what a text file is 3. how to view or print a text file. 4. how to obtain the software listed in the Archie text file using an ftp utility.

In order to understand email addresses, telnet addresses, and ftp addresses, a user needs to understand the basic concept of internetworking—that all Internet interfaces and utilities will connect their desktop box to one or more machines located across the room or across the world. In order to capture and save valuable information, the user must understand the concept of files and directories—during a gopher session, a user may *view* a file without saving it to any local machine. In order to do this, the user must understand how to transfer, name, and how to access and manipulate the information again later. In order for the user to conduct a successful information search, they must understand the search logic of the system or software—does it accept Boolean operators? Can I use regular expressions in my search query? Will this utility simply match the first string I enter?

5 User Education

What training will be required for general usage? Ideally, some sort of license should be created. The same way we have a driver's license to share the roadway, a medical license to practice medicine, or a law degree to practice law. The licensing process could be handled by the D.I.S. (Department of Internet Surfing) in each major metropolitan area. Licensees would be tested for familiarity with one computer operating system, network and internetworking knowledge, and, of course, netiquette! Each general licensee would be cautioned against using capital letters in email, sudden disconnections from remote computers, and surfing during 'rush hours' on the net. A second license or training program should be established for the people who set up, configure, and maintain installations of networking software. Its one thing to *use* these systems and utilities. Just wait until you're trying to *hook them up* to the ethernet cable in you living room two years from now! (Good Luck!)

6 Experiences

Discuss your experiences in performing each of the tasks... How useful were each of the resources and facilities for term paper development? Be Specific.

6.1 Penpal & Netnews

I've exchanged electronic mail with my buddy Ka_Wai Ho in Australia, and she offers moral support, but no academic input for my project. As a Usenet News Reader, I've been 'lurking' in the comp.hci group, but have yet to actually post a message asking for advice or information. The information available in this newsgroup has led me to a complete book on user interface design available via ftp. When my paper topic can be better expressed by a short paragraph, I hope to solicit advice from this group of Internet colleagues.

6.2 Three different online catalogs

I attempted to log in to four remote catalog systems, and succeeded with three: 1. The NovaNet Library System in Nova Scotia, Canada (NOVANET.DAL.CA). This was a telnet connection to a library catalog. The interface

was very well thought out—the searches were conducted in a quick and logical fashion. Each screen featured a menu of popular commands, plus options for more advanced searching and display of information. One ironic point was that the first screen had a ‘Dialing in from home? Come see us at the reference desk!’ message. That doesn’t make much sense for people dialing in remotely, and makes even less sense for a user logging in from another country! 2. Eidgenoessische Technische Hochschule Zuerich (Swiss Federal Institute of Technology, Zurich, Switzerland) (ETHICS3270.ETHZ.CH). This was a TN3270 connection to Europe. My first foreign–language interface experience! I chose to use the German menus, and was surprised how much of it I understood. I think my domain knowledge of bibliographic records and information technology was helping me out here. I was really thrilled when I figured out how to do truncation searches and which German words to use in Boolean searches. Finally, I logged in to System number 3. The Melvyl Catalog (MELVYL.UCOP.EDU) This is my ‘native’ library catalog. The interface is probably not acceptable by ‘novice’ HCI standards, but with a short training session, it becomes a powerful tool for locating bibliographic information. Melvyl was the only system searched that could locate accessible information for my term paper(i.e. I won’t have to hop a plane to Sweden to pick up my books).

6.3 Two different non–catalog databases

Still connected to Melvyl, I attempted to start the COMPuTer articles index, but was kicked out for not having a current U.C. Melvyl password. I reconnected to Melvyl via ORION, and the password requirement was waived. In this non–catalog database, the user can apply familiar Melvyl commands to another information source. I was able to locate several promising articles, retrieve their abstracts and full text, email the results to myself, and setup an automated, weekly search for new articles on the same topic. Very impressive.

The second non–catalog database selected for this exercise was Dialog. Dialog is an example of a for–profit information service. I logged in using the GSLIS–Lab’s Procomm Plus telecommunication software and logon script. Using knowledge from the GSLIS 220 Information Access course, I was able to perform a ‘dialindex’ search of all the computer–related databases, retrieve the text of promising articles, and logout with a total cost of under \$10. Dialog is available as a telnet connection, but the GSLIS Lab connection still has a few bugs in it.

6.4 Gopher name location

There are several utilities available on Gopher menus to help locate people and their electronic addresses. I searched for Scott Kim and for Bruce”Tog” Tognazzini’s address with no success. I tried the Netfind telnet connection, and an X.500 ‘power search,’ the database of Usenet News contributors, and several campus ‘CWIS’s, all to no avail. My only success story is that I located the address of a childhood friend, who is now a Ph.D. candidate at Harvard—we’ve been exchanging email all week!

6.5 Archie to locate software

When a file name was known, such as *psp21n.zip*, Archie was able to find the closest ftp site from which to download the file. Without an exact filename, Archie wasn’t very helpful. The Macintosh interface is preferable to the DOS client—the DOS client takes over your machine for about ten minutes while it searches for *one* filename string at a time, and you can’t do anything else while you wait. The Macintosh Archie client searches for multiple filename strings at once, and you can continue to use any other software while you wait for the Archie results. This is a good example of the benefits of *multi–tasking*.

6.6 Gopher and WAIS to locate relevant documents

I found these promising resources to be disorganized. Gopher let me connect (or try to connect) with and search one source at a time, after navigating countless and tedious menus. Gopher is powerful, but hampered because

its menus are inconsistent from site to site (and from day to day), and the software is difficult to configure. In addition, when everything else is working swell, the remote site or sites may not be working properly.

WAIS could search many computers all over the world and bring back whatever it found—but it tended to crash before the information arrived. Using a WAIS connection to Archie, I was able to locate some Icon software and the source code for the NeXTStep Operating System. I crashed every Microsoft Windows gopher and WAIS client. The gopher software locked up when it was required to start an external program, such as a file viewer or a telnet connection. The WAIS software locked up whenever it encountered a missing or incorrect server address. Not too encouraging to the novice Internet Surfer! The Macintosh WAIS client worked well, but my search words were not able to locate any useful information items.

John Houser, UCLA GSLIS Information Systems Librarian, *j*WAIS software author and recent National Public Radio interviewee, points out that many of the WAIS source servers provide access to Usenet News archives... a source that might be better examined by reading the index file of that ftp site or the FAQ (frequently asked questions) posting of that newsgroup.

7 Similarities

What did you find in common among the computer systems you entered? Most remote systems require a private or public login name, and a public or private ‘password.’ The systems that contained databases of searchable bibliographic information had similar fields in their records, such as ‘author,’ ‘title,’ and ‘subject,’ (or ‘descriptor’ in Dialog). This information was probably stored in a common format—the MARC record format or a variation of it. The systems I entered all allowed their information to be carefully searched using Boolean logic. Finally, all of these systems featured a character-based interface with the option to enter commands directly (over-riding any menus).

8 Differences

What were the most striking differences? Language was a big difference, but not a big obstacle. I was surprised how easy it was to use the German-language catalog. If front-end software such as MacMelvyl is included in this comparison, then its graphical, dialog-box style of interface is very different from the underlying command-line interface of Melvyl. In file-transfer sites, the ability to directly manipulate the files and directories with software such as *Rapid Filer* for Windows and *Fetch* for the Macintosh was completely different from the command line interface to ftp. Another, almost anecdotal difference is connections that worked versus connections that failed. It is hard to tell whether a failed connection is the fault of the user, the local interface, or the remote system. Successful connections made one interface seem more successful, while crummy connections reflected poorly upon another interface. Where successful connections were accomplished, some systems and searches yielded the desired information—and this made an interface seem more helpful than one where no useful information was retrieved. Some of these ‘differences’ might seem superstitious, but I’m sure I’m not the only user who will always return to the interface that ‘worked for me last time.’

9 Human Factors

Compare the interfaces to the human factors guidelines studied earlier. How closely do these systems follow those guidelines? Give specific examples Schneiderman’s Measurable Human Factors Goals, presented on page 18 of *Designing the User Interface*, apply to our Internet Interfaces as follows:

9.1 Time to learn

How long does it take a typical user to learn the interface? All of these interfaces are too complicated to learn in one sitting. The basic use of any of the bibliographic databases can be taught in a one-hour session, but the fine points take many months to get under control, and even then they are continually evolving—new commands and displays are added regularly to both public and commercial catalogs. The programmers of the Internet utilities assumed that their ‘typical’ user knows *a lot* about computers and inter-networking. Or, maybe the utilities were written with more savvy users in mind.

I first tried WAIS software in March of 1993. I have dabbled with it regularly, and tried in earnest to master it over the past three weeks. I have yet to retrieve a unique or especially helpful information item with this software.

9.2 Speed of performance

How quickly can this particular interface perform a benchmark set of tasks? While the command line interfaces are probably quicker on the client computer, the Internet utilities cooperate with a remote system that operates independently of the local interface. Instead of speed, why not go with a theoretically ‘slower’ interface that will allow a user to perform multiple activities—like editing a document—while they wait for search results to be returned from around the world.

The DOS Archie client is too slow. It is unacceptable to disable a computer users terminal for ten minutes, without warning or explanation, while a search is performed upon the CPU of a remote computer. The Archie client for DOS should return control of the computer to the user while it executes its search.

9.3 Rate of errors by users

What can they do wrong? Using the command line or character based interfaces—ftp, telnet3270, andarchie—I entered many incorrect commands, filenames, and cursor positions. When presented with a limited set of options, through graphical dialog boxes and perfectly named icons, I felt like everything I did was on purpose (even when I dragged an important file into the Macintosh Trash can—maybe this is a false sense of security). For these Internet utilities, I was able to work better with the graphical versions—where I could click on the desired gopher menu, drag the desired filename, and where there was no chance of typing or cursor positioning errors.

When connected to any remote system with Telnet 3270 software, the cursor must be positioned at exactly the right position on the screen for data entry. If the user presses the keyboard when the cursor is *anywhere* else on the screen, the terminal locks up and must be reset using a cryptic key combination. I’ve witnessed this (and *heard* the distinct ‘NOT HERE’ error chirp) over and over again in the GSLIS Lab—a mistake made by both experienced and novice users alike.

9.4 Retention over time

How well can a user remember the interface after an hour/day/week/year? Because I’ve tried so many interfaces and systems for this assignment, my head is swimming with telnet escape key combinations, WAIS server addresses, foreign language equivalents for Boolean searching, common gopher menus, etc. I have to look up almost everything at this point, or rediscover it when I can. The systems and tools most ingrained in my memory are the oldest ones—command line utilities and the Dialog and Melvyl systems.

An example of an interface whose features are difficult to recall is Gopher. The user must be able to navigate a long series of complicated menus to reach the desired information source—something akin to jumping through a dozen hoops for a dog biscuit. If the average Gopher user is not aware of the bookmark feature (or cannot

save bookmarks on a network), they must remember a long series of menus in order to find the same information during repeated use.

9.5 Subjective Satisfaction

My subjective satisfaction with the Internet interfaces is based upon three things:

1. the information I succeeded in getting
2. how quickly I could obtain it
3. the aesthetic presentation of the interface

The information available ‘out there’ and its successful retrieval isn’t really a function of the interface. Neither is the speed—when a lot of users are on the Internet, it slows down. When a lot of programs are running on the client or the host computer, it slows down. But these are aspects that I or any other user might attach to their subjective (dis)satisfaction with one particular interface.

I am not entirely satisfied with the Internet Utilities and Interfaces I have tried. My goal as a serious user is to obtain information. During this exercise, I found satisfaction *from* the information I found *in spite of* the interface. Some of the internet utilities was created before Human–Computer Interaction was a consideration in computer programming. Other software is written by individuals as a hobby or small part of their job. They might not be trained in User Interface Design, might not have time to construct an ideal interface, or might have to maintain compatibility with older, inconsistent interfaces or commands.

10 Suggested Improvements

What user interface improvements can you suggest? *Integration!* The function of the older software is highly specialized—telnet controls one remote computer but cannot exchange files. Ftp exchanges files with one remote computer but cannot perform any other tasks in to the remote account. Reading Usenet news allows a user to learn about new telnet, ftp, gopher, and WAIS sites, yet offers no facility to connect with these systems.

A working improvement would be to link the existing utilities into some sort of hypertext package: I read Usenet news, and when I see a new file I would like to get via ftp, I highlight the filename in the newsreader software and press enter (or click the mouse). The hyper–link will invoke ftp to retrieve the file, and the ftp–link will call a decompression program to restore the file to an executable or viewable state. If the operating system can deal with multiple ‘threads,’ then I’ll still be reading the news while all of this takes place in the background.

An ideal implementation of the ‘integrated Internet utilities’ would re–write all of them to communicate with an advanced protocol, such as Z39.50. This change would allow a variety of integrated interfaces, for various computer platforms and user groups—and it would work *better* because the software would be designed to work together from day one. (It all seems to be delicately patched together at present). This integration and adherence to a common protocol would benefit online catalogs and databases too—imagine being able to use the same software package to log in to Dialog, Melvyl, the OAC and ORION. Imagine finding a great document in Melvyl, and issuing a ‘transfer’ command to ‘check out’ a binary copy of the book, image, etc—with all of its original formatting, tables, and photos. (There might be a few copyright infringements here, but you get the concept...).

WAIS is a great idea, but it needs a stronger way to search for information, perhaps by Boolean logic or regular expression. It needs an online tutorial, and an interactive wizard or agent script that could walk novice and intermittent users through a search, step by step. The Windows WAIS client is especially hampered by the fact that it crashes when it encounters any single host computer that is not up and running (instead of skipping to the next remote server or even just quitting the search—the whole computer crashes and the user loses any unsaved data).

11 Conclusion

What is your overall assessment of Internet interfaces from a human factors perspective? Today, we're doing an awfully good job with some very old hardware and software. Small utilities that were 'experimentally' designed at one institution have been widely adopted and are being used by thousands of people every day. Online databases designed to serve one small community are now sharing their information with a global community. Technological advances, such as High-speed ISDN phone lines and the NREN network in America will allow Human Factors experts to redesign interfaces and utilities with greater freedom, and for different user groups.

With the existing utilities and interfaces, only a highly motivated (and patient) user can take full advantage. Until the new technology and interfaces arrive, I believe the best way to empower users is to motivate and educate them—first share the concepts of computers and computer networks, then give them an enthusiastic introduction to the interfaces and information available across the Internet.

12 Figures

Complicated Connections: When a student in the gslis lab connects to Melvyl as an option on the ORION menu, look at the layers of software communication involved. The local user inputs on a keyboard and sees the output of Melvyl in Oakland, CA. look at all those layers of communication that can be messed up!

Figure 1: Interdependent layers in a single network connection

Figure 2: Five Different File Transfer Protocol Interfaces

Figure 3: Online Catalogs and Databases

Figure 4: Macintosh Client-Server Utilities: WAIS and Archie

Figure 5: Crash Landings