The following paper was originally presented at the
Seventh System Administration Conference (LISA '93)
Monterey, California, November, 1993

System Administration as a User Interface: An Extended Metaphor

Wilson H. Bent, Jr.
University of Southern California

System Administration as a User Interface: An Extended Metaphor

Wilson H. Bent, Jr. – University of Southern California

ABSTRACT

Many people have spent many hours searching for the right user interface. Many people have spent many hours searching for the definition of the right user interface. Many people have spent many hours decrying the wrong user interface (whatever that may be). One user interface which is often overlooked is the System Administrator.

What are the qualities of a good user interface? Some of the features most often mentioned are:

- User-friendly
- Extensible, adaptable to the user's skill level
- Gives useful error messages
- Hides underlying system's gore
- Offers easy access to helpful information
- A terrific look-and-feel

This paper will discuss each of these features with regard to the interface between the user and the system administrator.

Introduction

What is a user interface, generically? It's a layer that sits between a person (a "user") and whatever resource that person is trying to use – information, data manipulation, or access to a network. It's the way a user controls a resource. A toaster has a fairly simple user interface: a knob to set the amount of toasting, and an on-off slider. Notice that, even for a job as simple as toasting bread, it takes two controls to provide the flexibility needed.

Television sets used to get by with two controls: the channel selector and the on-off switch, which doubled as a volume control. Such technical details as the brightness and horizontal hold knobs were hidden away, and usually only needed adjusting once. As TVs have become more advanced, more flexible, and more "sophisticated," the user interface has become more complicated, so much so that even a basic model TV comes with a remote control with a couple dozen buttons.

Because of its capability of being programmed to perform a wide range of tasks, the level of complexity in a computer can be much higher than any other device. As a result, there is the possibility of a wide range of user interfaces: a spreadsheet performs a task which is very different from a word processor, and both are different from a flight simulator. However, their user interfaces are not entirely dissimilar. Programmers and system designers have, over the years, defined qualities of a good user interface. These qualities allow the users to get their jobs done with relative ease, without feeling that the computer is "getting in the way."

We're used to thinking of the term "user interface" in very narrow terms; specifically as what appears on a user's screen and how the user interacts with the computer. However, there's more to using a computer than that. Other interfaces which can make a computer easier or harder to use include the ergonomics of the workspace – and even the manuals.

As systems administrators, we ourselves are another user interface. Our function is to make resources available to users, and we can learn some lessons from the designers of graphical user interfaces.

User-Friendly

No doubt many systems administrators see the term "user friendly" with regard to their users and groan "What, *me* be friendly with *them*?" But remember, it doesn't have to be a case of us-*vs*-them, it can also be a case of client and customers. It can even be a case of peers: two people who are experts in their respective fields, one of which happens to be the field of systems administration. That's a nice, pretty-sounding side of the coin, but here's the flip side: chances are that if you are not user-friendly, it'll get back to your boss, which means that it'll get back to you.

Your attitude makes a difference. Nordstrom's department store is doing well and even expanding despite having higher prices in a time of tight budgets. Why? Because they have a widely-known policy that the customer is always right, and their staff is trained to act on it. Customers are willing to pay more, knowing that they'll get treated with respect.

There are those who argue that caution should be used when it comes to user-friendliness, and that "the customer is always right" is not always the right motto. They warn that if you promise users the world, they'll be sure to ask for it, and more. If you work too hard at solving users' problems, they will learn to see you as someone who will solve *all* their problems.

Such caution is a worthwhile thing, but be sure you've reached the point where it's appropriate to be cautious. Make sure that you've shown the users that you are capable of solving *some* of their problems before you advise them that you can't or won't solve *all* their problems. If the customers are not always right, be sure they know why they're not, and be sure they know when they *are* right. Remember, a diplomat is someone who can tell you to go to hell, and make you look forward to the trip.

You may be solving every user's problem with speed and expertise, but if you are doing it with an attitude which implies "I'm smart and you're not," you will only make the user go away not wanting to return. This may at first seem like the desired result, but in the long run it means you'll run out of users and thus put yourself out of work. The users will still exist and still have questions, they'll just find someone else to answer them.

Also, if you establish a friendly working relationship with the users, you can expect (or even ask for!) help from them at times when you need it. When you develop a good relationship with users who aren't afraid to talk to you, you're more likely to get early reports of broken stuff, and a chance to fix it before major damage is done.

Extensible, Adaptable to the User's Skill Level

No advanced user likes an interface which requires twelve button presses to delete one file. Some disk formatting programs strike me as a bit extreme: "Are you sure you want to do this (Y/N)? Are you REALLY sure (Y/N)? Press Q to quit, any other key to stop."

Similarly, the reason that GUIs are becoming so popular is that there are oodles of users at the other end of the spectrum who don't want to learn either `ls` *or* `dir`, who can't remember `del` *or* `rm`, and who can cut and paste a lot faster than they can write and read a temporary file.

It's also easy to find examples of manuals at both ends of the spectrum: many DOS programs come with a slim brochure which spends more time showing you how to insert a disk into a drive than how to use the program. Conversely, the high-end, high-price programs always seem to come with many shelf-feet of documentation but no hints on how to locate the one bit of information you need.

If a novice user asks you how to find a missing GIF file and your answer is a simple "`find . -name '*.gif' -print`", that user is not going to go away happy. Some users will benefit from a minute or two spent explaining the options to `find`, others will be better suited if you install an interactive shell script which results in running `find`. A more experienced user, however, may only need a brief reminder of some of the less-frequently-used options to `find`.

The thing to remember is that these are all valid users. They have a job to get done, and depend on you, the administrator, to help them, advise them, and set up a system for them which lets them get their job done with a minimum of fuss. Your response to their questions should match their skill level, so that you are neither insulting them by being overly simplistic, nor swamping them with your wizardly knowledge.

Journals such as the *Communications of the ACM* often publish papers on help systems and user interfaces which make use of artificial intelligence and a natural-language interface to access information. What they're trying to do is what you already do – except your interface is based on *natural* intelligence, and chances are you have a greater vocabulary than their natural-language interpreter! By engaging in an interactive dialog, an administrator can quickly focus on a user's needs and wishes, and respond in terms the user can understand. Being human has its advantages.

Some people want to learn how things work – they like to find out what makes things tick. Others are completely learning-hostile, and just want you to make them a magic bullet. Both kinds pose different challenges. The secret of user-friendliness and extensibility is: adapt. Learn to figure out the skill level of a user as quickly as possible.

A Diversion: The Systems Administrator as a Meta-Interface[1]

The real advantage of the human interface you offer to users is that you can go beyond just answering their questions. You can explore with the user such questions as "What is it that you're trying to do?" or even "Why are you trying to do that?"

If a user asks you to restore some or all of her mail file every week or so, you can solve her individual problem each time she asks. Or, you can go one step further and create a method for her to restore her own mail files, which empowers her and reduces your workload – apparently a good solution for both of you. But to take it one step further, you can explore with her why it is that she keeps having trouble with her mail: are the commands for "save" and "delete" similar, or are those two buttons in her graphical mail interface too close for comfort? Has

[1] Every serious paper should have at least one *meta*-discussion.

she uncovered a bug in the mail program? Is she receiving large mail messages which are filling her file system?

As a human interface, you can be more than just a help button.

Gives Useful Error Messages

Probably every paper on user-friendly programs ever presented at a USENIX conference has mentioned `ed`'s "?". Although `ed` is still cryptic, newer programs are sometimes quite helpful when reporting errors, although not so many that we can stop writing papers about the true-yet-unhelpful "not a typewriter" message.

Probably every paper on working with users ever presented at a LISA conference has pointed out that saying RTFM is not a good thing. Unfortunately, RTFM still gets said time and time again, so I have to speak up for the voice of reason: how about suggesting WHICH manual page to read? How about even mentioning which BOOK to read? As a Sun user, I highly recommend the AnswerBook software. Despite the fact that searching for a topic usually results in too much and occasionally too little information, it remains an easy-to-use interface to the manuals.

It's worth the time to make sure that users are informed when something doesn't work, regardless of the type of interface. If you're writing a shell script, it's more helpful to catch signals and be prepared for possible failures than to have the users confronted with core dumps. If you're referring them to another person or group, it's more helpful to explain why that group would better serve their needs than to just pass the buck. If you're helping them as they navigate a directory tree, it's more helpful to briefly explain access permissions than to just say "You can't go there." If you're on the phone with the user, it should be an absolute requirement that you say "see dee user local..." at a pace the user can understand, and when you slow down, that you keep that pained tone out of your voice!

Hides Underlying System's Gore

The joy of using computers is that they can do tasks that you or I can do, but they do them faster and without error.[2] In performing these tasks, a computer goes through many steps, but a good user interface hides that fact from the user.

The ability to cut and paste is considered essential to a GUI, but the underlying mechanics can be fairly difficult. Drag-and-drop looks equally easy from the user's standpoint, but is still not well-defined after five releases of the X Window system.

[2] That's the theory, anyway.

Even a simpler user interface such as the Bourne shell or the DOS command interpreter hides a large amount of path searching, `fork()`s, `exec()`s and `wait()`s. Under the version of UNIX I use, an 'easy' command such as `date` calls over 20 system calls and library functions, and of course the library functions themselves make other system calls...

What gore are you hiding as an administrator? Not just the internals of the UNIX system, but the internals of your administration system. When users ask for files to be restored, they're asking to be insulated from the gore of your backup scheme and tape library. If your tape library doesn't seem like gore to most of your users, either it's too small or they're as anal-retentive as you.

Your job as an interface to a fairly complicated system is to provide a method by which users may get a simple job done ("restore this file" or "print this memo") without needing to know the details. If they WANT to know the details, you should be willing and able to provide them, but even the experts usually just want to get the job done.

A Diversion into Paradigm Shifts[3]

A sure-fire way to start a conversation (or worse) among computer users is to say "The Macintosh user interface is more intuitive than MS Windows." Things will quickly degrade into comparisons of numbers of buttons on a mouse *vs* number of clicks needed to perform a task.

Yet, watch a user go from one GUI to another. How hard is it to adapt to a different window-frame decoration? How quickly can you figure out how to resize a window? Despite the frustrations, there is enough general similarity among GUIs to at least know where to start.

Administrators of systems from Sun Microsystems are discovering that a sure-fire way to start a conversation (or worse) among their peers is to say "I just switched to Solaris, and what a pain!" Things will quickly degrade into comparisons of `a.out` formats and options to the `ps` command.

Okay, so what's the point? Just as you, the system administrator, encounter problems in dealing with change in, *e.g.*, operating systems, so do the users legitimately encounter problems with changes in other, "simpler" (to you) interfaces or systems. Just because you consider those systems to be "easy to understand" or simple, doesn't mean that the user community will feel the same way. Any changes you introduce are going to require an adjustment

[3] Every serious paper should use the term *paradigm* at least once.

period for the users. Preparing them for the change will make life easier for everyone.

Offers Easy Access to Helpful Information

Another strong point of GUIs is that they often make it easy for the user to get help for a specific item or area. Two examples of this are Sun's Help system under OpenWindows, and Apple's balloon help under System 7. Such systems allow the user quick access to information about a limited topic, without having to wade through a stack of manuals.

For systems administrators, being able to provide such a service to users will not only make life easier for them, but easier for you, too. I have seen quick tutorials for both `emacs` and `vi`; having either or both on-hand, ready to give to a novice user, works wonders! It gets them going with a minimal amount of your time, and impresses the heck out of them – always a good start when dealing with novice users.

You should have readily available at all time a list of reference books. I have seen several posted to the net; an extensive one is posted regularly to `misc.books.technical` and other newsgroups.

My mother was a professional reference librarian for about 40 years, and when asked a "Do you know...?" question, her standard answer was "No, but I know where to look it up." The flip side to that, however, is that people *expected* her to look things up. Working in the music section of a large library, she was asked every day "When did Beethoven live?" If she instantly replied "1770 to 1827," the patron tended to be a bit suspicious. She found it necessary to pull out a book, flip through the pages, and refer to an authoritative source in order to convince the patron that that really was when Beethoven lived.

You should also have on hand easy-to-read information on getting started with netnews: advice on one or more news readers, and suggestions of groups to start with. If the user has a specific topic in mind, you might want to point them to one of the specialized groups, but you should think twice before inflicting one of the high-traffic groups on them. Then, of course, there are the man pages. I shouldn't have to say again how RTFM is an unhelpful suggestion, so let me just give a simple reminder: a good man page is designed for reference, not for teaching. Man pages are generally about one specific program, not the interactions of several programs. Think about the user's technical level of experience before handing them a multi-pound or multi-megabyte manual.

A Terrific Look-and-Feel

Somehow, the only times I feel as though I have the appropriate Look-and-Feel is when I'm either at a Usenix gathering or at a Grateful Dead concert. I used to work for AT&T, formerly known as The Phone Company. One aspect of that environment that never ceased to amaze me was that, despite being known world-wide for being at the forefront of communication and network technology, most of the users in my area preferred not to use the phone to ask me a question – they'd rather come to my office. This was not a small thing, since (by no doing on my part) my office was in the back corner at the far end of a long hall. I can't tell you how many times I've had a user walk down one hall, up a set of stairs, and down another hall to my room to ask me "What was that command to get the date?" Maybe they just needed the exercise.

Or maybe they preferred the look-and-feel of direct interaction with someone. Maybe the added element of voice communication, hand gestures and body language, access to but not reliance on a computer and other resources, maybe all these elements of the user interface made it worth their while to walk that distance.

A good look-and-feel doesn't mean that you have to wear a necktie, nor that you have to have scrollbars on your office door. It means having an attitude which says to users, "I am a knowledgeable person who can answer your questions." Such an attitude will inspire users to treat you as an expert, to treat you as a helpful source of information, and to treat you with respect – a treatment which is sure to improve your position in the community.

Conclusion

The key element of a well-designed user interface is that it lets the user get his job done without getting in his way. He isn't forced to jump through hoops to do a simple task, and he isn't prevented from doing a complicated task – in fact, he relies on the system to provide help of some sort at all levels.

As a system administrator, you are part of the resources available to the user. The way you interface with users affects the way they get their jobs done. With a little extra effort on your part, you can get them working at their best, and you'll all be able to take pride in the results.

Author Information

Wilson Bent recently joined the Research, Development, and Systems group of University Computing Services at the University of Southern California as a Systems Programmer. Previously, he worked for 10 years at AT&T Bell Laboratories as a programmer and administrator. Reach him via U. S. Mail at University Computing Services; University of Southern California; Los Angeles, CA 90089-0251. Reach him electronically at whb@usc.edu .