LBL-15311
Preprint c.2

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Engineering & Technical Services Division

RECEIVED
LAWRENCE
BERKELEY LABORATORY

FEB 18 1983

LIBRARY AND
DOCUMENTS SECTION

To be submitted for publication

THOUGHTS ON USABILITY

David F. Stevens

December 1982

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 6782.

LBL-15311 c.2

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Thoughts on Usability*

David F. Stevens

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

December 1982

*This work was supported by the U.S. Department of Energy under contract No. DE-AC03-76SF00098.

Thoughts on Usability[1]

David F. Stevens
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

December 1982

1. Introduction.

This note is founded in the belief that a high level of usability is fundamental to the success of an information services facility.[2] In general terms, an information service is usable if the barriers it places between users and their effective use of the system are few and easily penetrable. Some of the characteristics of a high level of usability are

a) ability to do the job

b) it is easy to discover what tools are available

c) it is easy to teach oneself to use those tools

d) the full power of the system is available to any authorized user

e) it is available when the user wants to use it

f) the users' dealings with the facility are characterized by feedback and follow-through

g) the users are given freedom from inconsistency, freedom from irritation, and freedom from interruption

h) deviations from any of the above are accomplished with grace

---

[1]This work was supported by the U.S. Department of Energy under contract No. DE-AC03-76SF00098.

[2]Time was that the principal information processing department was reasonably universally known as EDP, ADP, or just plain DP. We now have departments that supply and support Word Processing Systems, Text Processing Systems, Information Management Systems, Data Base Management Systems, Office Automation Systems, Executive Information Systems, CAD/CAM Systems, et cetera. The general principles (if not necessarily all the fine detail) espoused herein apply to all of these entities, and since they all may be characterized as *information services* I shall use the term "information services facility" (or simply "facility") throughout this note as the generic term of reference.

i) and, finally, there is no sense of "us" versus "them".

The achievement of a high level of usability is complicated by the fact that usability is not a consistent goal, either across the user population or even for one user across time. As an example, consider the language of the user interface. Novice or occasional users are comfortable with recognizable words and phrases (such as *write*, *quit*, and *insert after*), while expert users become impatient with such fully-specified long forms. Conversely, experienced users, especially those who spend a lot of their time writing or keying-in such material, are grateful for one- or two-letter contractions and abbreviations, while the neophytes find such secret codes to be immensely confusing. With each passing day, direct upper-management usage of information services becomes more common, and upper-level management usage tends to fall into the "occasional", rather than the "expert", category. You are thus faced with the following dilemma: A system employing only long forms penalizes those who use it most, while one allowing only short forms confuses and frustrates its most influential users.

A similar dilemma arises with respect to the protection of novices, who are capable of doing themselves (or their projects) unrecoverable damage unless the system provides means of restraining them: The same safety belts that protect the novices also deny experienced users access to useful and powerful constructs.

Until we have systems that are capable of adapting themselves automatically to the user's level of competence we will be faced with such dilemmas. In general, one should try to tailor a system to the abilities and interests of the greater part of the user population, but without totally abandonning those users who are either less familiar or more experienced than the average. Your users have

work enough and problems enough of their own; do not make them do your work and solve your problems as well. They want the tools you can provide, but only if they can use them effectively *without* having to learn large amounts of information services esoterica along the way.

2. Can Do

The most fundamental characteristic of a usable system is that it can do the job a user has in mind, in a reasonable time and at reasonable cost. Mere ability is not sufficient: almost any system can handle almost any problem if the problem is presented in a suitable manner and if the system stays alive long enough to finish. Usability thus rests with the rather fuzzy concept of *reasonability*. It is generally understood that reasonability is both task- and environment-dependent. (One expects it to take longer to convert a table of sales data into a pie-chart than to delete a comma from a memo; display of that pie-chart can proceed at a more leisurely pace than display of the vital signs of a patient under the knife in an operating room.) It is equally important to realize that reasonability is also expectation-dependent. (Ten-second response-time can be marvelous to a user who is accustomed to thirty seconds, but abysmal to one who is accustomed to one second, *even if the operations involved are totally different*.)

This question of expectation is bound up with a complete understanding of cost as it relates to usability. In the first place, users are not concerned with *your* costs, only with their own. In the second place, they are likely to count costs that you are inclined to overlook. In particular, they are going to consider their own lost time as a cost, and to consider time spent learning (and unlearning and relearning and updating their learning of) your systems as time lost. An important aspect of usability, then, is how the abilities of the system are

translated into the tools the users see.

- Do you provide comprehensive guides for the translation of their requirements into system languages, or do you provide task-oriented dialogue?
- Do you provide a single, well-defined way to perform each task, or do you allow users with different backgrounds and preferences a choice of tools?
- Do you provide instructions on how to structure the work to make effective use of your tools, or do you provide tools that fit the work?

This last is rather a tricky point, for the existence of new and powerful tools can -- and should -- shape the work to some extent. But the measure of a system is in the doing of the work, and not in the exercising of the tools. You must not only provide tools, they must be suitable tools, and the users must know what and where they are and how to use them.

3. Easy to discover what tools are available...

In its general form, this is an unsolved problem, largely because unless you know what there is you can't find out what is there: it is very difficult to provide meaningful current indices to tools. In some instances the name of a program is a good clue to its function (***mail***, ***readnews***, ***kill***), but others (***biff***, ***grep***, ***awk***, ***wall***) defy analysis by the uninitiated. (These examples are all drawn from a UNIX[3] environment.) Many information services environments are so rich in new terminology that neophytes are unable to make any sense out of the directories that are provided. A common problem is the provision of an index that includes all the system-specific words *but only those*: unless you *already* know all the magic words you can't find what you need.

[3]UNIX is a trademark of Bell Laboratories. It well may become the most familiar system for most Computer Science graduates in the last half of this decade.

For example, standard UNIX documentation includes a kwic-like permuted index containing entries for all commands. There are also entries for each of the principal words in a one-line description of each command. The problem is that the describer's words do not necessarily match the users' guesses. The command to create a new directory is "mkdir"; it is listed under "make" but not under "create"; the usual commands to display or list a file are "cat" and "more"; neither is listed under either "display" or "list". Thus, even this rather sophisticated tool is relatively useless for a UNIX novice.

Until the fundamental problem is solved (which requires the development of extremely clever "**HELP**" facilities and query-and-dialogue systems) you have to rely upon education.

4. ...and to teach oneself to use them.

Much, perhaps even the bulk, of ones technological education takes place in corridor conversation, but that tends to be ad hoc, disjoint, quite specific (the question is answered, but useful generalizations are rare), easily forgotten, and a burden on the rest of the staff. It is important to supplement this highly informal and unstructured mechanism with a more formal and structured one providing continuity and a permanent reference. The preferred medium is a series of introductory documents and tutorials spiced with suggestive examples. These examples should include not only the obvious topics (editing, getting hard copy, running programs, reading mail) but also some of the less familiar characteristics, tools, terminology, tendencies, and tactics of the system in question. Any useful list of topics to be covered is of necessity quite system-specific, but some generalizations are possible:

- Existing "entry-level" documents are not really entry level. They usually assume, for example, that a terminal is *on* and *connected to the system*.

For installations where that is not always the case (as, for instance, when a terminal can be connected to more than one system), a *pre-primer* that takes the novice through the preliminary steps is very valuable. The contents of a pre-primer are system- and installation-dependent, of course, but should include

—how to tell whether or not the terminal is on

—how to turn it on

—how to connect to a system

—how to interpret system messages and indications

—what to do and whom to call when it doesn't work as described

- Most systems have elements of vocabulary and style that distinguish them from other systems with similar functions. (Some interactive systems are menu-driven, some are dialogue-driven, for instance.) Thanks to increasing computer literacy, the majority of the users of many of today's systems have had some exposure to computers, though not necessarily in the forms and languages you are supporting. Elementary documentation of the user interface should assume users who are familiar with *other styles* as well as those who are true beginners.
- Most terminal time is spent doing some form of editing or other generalized text processing, and these services are among the most system-idiosyncratic. The learning curve for these products can be steepened considerably by providing good, user-work-oriented tutorials in addition to reference manuals.
- Most installations develop over time both a feeling for the most useful products for their particular applications, and also sets of home-grown tools and packages to simplify their life. This folklore should be codified and documented to the extent possible without interfering with its

development.

- And even though not everything you would like to have is available or in the best possible form, every installation has a reasonable amount of helpful documentation scattered about the place. Make a reading list for newcomers.

5. Full Power...

There are two general classes of barrier that can exist in an information services facility between its users and the services they desire to use: barriers to knowledge and barriers to access. Some barriers – those mandated by law or corporate policy, or inherent in the systems themselves – are beyond your power (or authority) to breach. The rest you should work to eliminate, when you cannot eliminate them, you should provide the users ways to get around or get over them.

Barriers to knowledge are (conceptually, at least) easier to eliminate than are barriers to access, for all that is necessary is the education of the users. Of course, this simplistic statement hides a number of problems. The most obvious is the difficulty of providing adequate and appropriate educational tools. Vendors usually provide reference manuals from which an information services professional can deduce promising approaches to users' problems, but seldom provide the several levels of user documentation that are desirable. And where local supplements exist, they are often constructed by, and in practice are only usable by, the local hackers. Another problem is the rate of change, of the facility as a whole as well as of its individual systems. This problem is particularly severe now as many organizations are adopting new classes of information services (office automation, departmental mini-computers, local area networks, ...); not only are there more kinds of services to explain, but also

there are more kinds of users in need of the explanations.

The most difficult of the educational problems may be inherent in the system itself, in that it is so complex and hard to master that users are unwilling to expend the time and effort necessary to achieve "full-power" competence. (There is no royal road to JCL.[4]) One approach that is occasionally taken in such cases, enclosing the unfriendly system in an ergonomic cocoon (*cf.* catalogue procedures), almost always carries as a consequence the loss of power and generality. (In other words, in avoiding a barrier to knowledge one can introduce barriers to access.) One must therefore take care to ensure that the cocoon is not *all*-encompassing.

Barriers to access admit of no simplistic solution, even on the conceptual level. A few examples may serve to illustrate this:

- Access to certain machines, software, or data is limited to one department, or to certain times of the day.
- Not all the machines I need are on the same network, or use compatible systems.
- There are no ports available when I try to connect.
- Access exists but its quality is too low to permit me to do any useful work. (This can come about through bottlenecks, interrupts, poor response time, preference to local users (when I am remote), ...)

Some of these problems are political, most are technical; they are all solvable if the facility has the interest to discover them and the will (and the authority) to solve them.

A final caution: the achievement of a reasonable state of full-power availability is no guarantee of the continuance of that state. Systems change out from under

[4]With apologies to Euclid.

the user-friendly bridges you build; access quality degrades under increasing load; (perceived) system complexity increases with the rapid growth in the user community that many organizations are now experiencing. It requires frequent modification and repair to keep the bridges open, constant updating to maintain access quality at acceptable levels, and sustained effort to generate and maintain the documentation required to make the facility self-teaching.

6. ...when the Users want it.

The only hours of system operation that are of interest to me as a user are the hours when *I* wish to use it. A usable system tailors its operational hours to the habits of its users rather than expecting them to tailor their work habits to the availability of system resources. For some organizations, prime time is therefore sufficient, for the users all keep regular hours. For others, especially those with users in several time zones, or with a significant proportion of night people among the user population, extended or all-day availability is necessary. Users do not keep track of downtime per se, but they tend to keep track of the number of times they are rebuffed when they try to access a system; ***how much*** downtime may be less important than ***when it occurs***. The "rules" for maintaining adequate and friendly access are fairly obvious:

- Try to limit downtime to the times of least user impact.
- Keep it to announced hours.
- Explain unscheduled downtime (in advance, whenever possible).
- Know the causes of access failure, and provide alternatives for unreliable links.
- When you *do* go down during user hours, come back up as quickly as possible.

To come back up quickly you need quick and accurate detection of the down condition and rapid and effective response by those charged with correcting problems, whether they be in-house or vendor personnel. One common mistake (with respect to coming back up quickly) is to give some level of question-answering responsibility to those who are also responsible for fixing systems errors. To do this is to ensure that their effectiveness at *both* jobs is lessened just when it is most necessary (i.e., when a system has just crashed).

Remote users are particularly subject to, and frustrated by, access failures. (Remote access failure is often accompanied by a complete lack of information, response, or even acknowledgement.) In addition to working towards more reliable communications, then, you should also do your best to provide alternate paths, and to to provide good follow-up services when your remote users report such failures.

7. Feedback and follow-through

Feedback and follow-through are particularly important because they address the *human* relationships between the users and the staff of the facility. (By contrast, the discussion to this point has been primarily concerned with the *technical* relationships between the users and the information services you provide.) Feedback and follow-through play a major part in determining your users' perceptions of how interested you are in their welfare and how receptive you are to their suggestions. Failure to provide feedback and follow-through can undermine any level of technical excellence insofar as the users' *opinion* of a facility is concerned.

Feedback and follow-through rest upon a common foundation: dialogue. They both imply the cooperative exchange of information between a facility and its

users. That, in turn, implies more than just message traffic in each direction. The classic picture of the standard impersonal data center includes lots of (unanswered) requests, suggestions, comments, and complaints from the users and lots of (unanswerable) directives, demands, and pronouncements from the center. There is plenty of traffic in each direction, but little communication.

An effective feedback loop provides a mechanism by which the users can offer suggestions, comments, requests, and complaints and receive responses. Furthermore, if the mechanism is *public* it provides broader feedback as well as ample motivation for you to follow through in a friendly manner. (You *must* follow through if you wish the feedback to continue.)

Following through does not mean that you have to *adopt* all suggestions, but it does mean that you need to *answer* them all in a helpful and informative way, and that you must, in fact, do what you say you are going to do, in the time scale you said you were going to do it. (Users have long memories for your promises; you would do well to match them in this respect.)

One of the characteristics of a usable facility is that it adapts to the changing needs of its users. Continuing dialogue is a powerful aid to timely and appropriate adaptation, but it must include not only your users' present experience and their hopes for the future, but also your planned adjustments to both. This dialogue provides the feedback you need to design the future, but that design will never be realized unless you follow through.

8. Freedom from inconsistency, irritation, and interruption

If there were a User Bill of Rights it would include the right to be free of inconsistency, irritation, and interruption. (The fact that "irritation" is extremely subjective doesn't make it less important, just harder to measure.)

Usability is some non-linear inverse function of the frequency with which a user encounters any of these three. Both inconsistency and interruption are sources of irritation, of course, but they are important enough to consider separately, and not all irritations are due to inconsistency or interruption. (Consistent loss of data, for example, or uninterrupted snarkiness on the part of the staff of a facility are both quite irritating.)

Inconsistencies arise from a failure to define, know, or respect conventions and standards. Some examples from real information services facilities that I have encountered include

- the existence and use of both ANSI and EBCDIC files
- the use of both labelled and unlabelled tapes
- the use of "**R**" as a control card option to refer variously to "**READ**", "**RING**", and "**REWIND**" with respect to tape
- the use and non-use of the **RETURN** key to signal completion of command entry in different utilities
- the use of the **RETURN** key for cursor control as well as to signal completion of command entry in the same utility
- different versions of various languages and utilities on different machines in the same room

Consistency allows users to generalize with confidence from what they already know; inconsistency forces them to learn every detail.

As noted above, irritation may arise from inconsistency or interruption, but there are other sources as well:

- inaccuracies in documentation
- lack of necessary (or desired) information
- unnecessary job steps

- gratuitous repetition
- overly complicated interfaces
- too-frequent changes
- obstructive procedures
- unfriendly people
- environmental factors (such as temperature, noise, cold drafts, glare in offices)
-
-
-

Many sources of irritation are of small importance individually, but they aggregate into a user's opinion of the facility, and the way in which you respond (or not) to them strongly affects a user's view of your attitude.

Interruption is perhaps the worst of these three evils for it has the greatest effect on productivity. Many of the tasks that users of an information service undertake have a relatively large set-up time (establishing the proper mental context and computational environment, loading data, preparing files) and an interruption can force the user to repeat that preparation. Besides being a waste of time and effort, it provides renewed opportunity for error and is a source of personal frustration and stress. It is worth expending significant effort to avoid interruptions. Downtimes should take place when they cause the fewest interruptions and at advertised times. Ideally, that means outside of the normal workday; if that is not possible, then at the beginning of the work day. (Since machines are not gracious enough to break in accordance with a schedule there are certain to be some unscheduled outages, of course, but if any individual user is affected as much as once a week then your reliability is not acceptable.)

9. Grace

It is inevitable that you fail from time to time to live up to the high standard assumed herein. The manner and style in which you handle those failures can have as great an impact on your user community as the failures themselves. You should recognize problems promptly and provide immediate symptomatic relief. When the problem is brought to your attention by the users you should respond to *their* concerns. (Explaining that they don't really understand how the system is supposed to work is *not* responding to their concerns.) When you discover a problem before they do, you should tell them what it is, what you are doing about it, and how to cope in the interim until it is resolved. You should not attempt to hide bad news from the users. You should institute plans to overcome critical shortages and recurrent failures.

It is important that your concern for your users be evident in your response to problems, that you respond quickly, and that they *see* what is being done to help the situation.

10. Us vs. Them

One of the hallmarks of an effective group is the sense of community that develops among its members. This effect is accentuated when the group is distinguished from other groups in a parent organization by its technological specialization, for such specialization gives the group an environment, and even a language, that is foreign to those who have no contact with, or who are users (rather than developers and providers) of, the technology. In such a situation it is very easy for the group to pass from a sense of unity ("us") to a sense of difference, distance, and even conflict ("us" vs. "them"). This in turn leads to the development of an adversary relationship between the information services

facility and its users.

The job of an information services facility is to assist all of its users to accomplish *their* tasks as effectively, comfortably, and efficiently as possible. The proper performance of that job requires cooperative, rather than adversarial, relationships between the facility and its users. A properly directed sense of usness contributes to good morale. In the case of an information services facility, that sense of usness should be directed towards the betterment of the lot of the users.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720