



Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY

Information and Computing
Sciences Division

RECEIVED
LAWRENCE
BERKELEY LABORATORY

JAN 30 1987

LIBRARY AND
DOCUMENTS SECTION

Submitted to EDP Performance Review

MORE CAUTIONARY APHORISMS FOR USER-ORIENTED
COMPUTER MANAGEMENT

D.F. Stevens

October 1986

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.*



LBL-22254
^{c2}

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.

More Cautionary Aphorisms for User-Oriented Computer Management*

Winnow all my folly, folly, folly and you'll find

A grain or two of truth among the chaff.

W. S. Gilbert [1]

Aphorism is better than none.

Ryan Anthony [2]

INTRODUCTION

This is the second set of "cautionary aphorisms" relevant to the management of computing collected by the author. The first set[†] consisted primarily of variations on the theme that

0.0: A human user is a co-processor of largely unpredictable behavior. [3]

The present collection is concerned less with the strictly human condition than with the interface between humans and computers. Four aspects of this often turbulent boundary are considered: Ordinary people using computers, enthusiasts (I would say "hackers" but that the term has become wholly pejorative), the introduction of new systems, and power tools.

As a prelude to looking at the relationship between ordinary people and computer *systems* it is appropriate to spend a moment considering the

relationship between ordinary people and computer *people*. The salient characteristic of that relationship has been discussed often and at length, in general literature as well as in the technical literature. It should therefore come as no surprise that it was expressed in memorable form by one of the giants of literature more than 150 years ago:

0.1: Computer people are like Frenchmen: Whatever you say to them they translate into their own language, and it immediately becomes something different. [4]

There being no computer people around at the time, Goethe actually referred to mathematicians, of course; furthermore, the statement is not limited to mathematical disciplines, for it is in some sense true of any profession that possesses a rich and vigorous jargon. But few other professions have as pervasive an impact on everyday life as computing, and only medicine rivals computing in the reluctance of its practitioners to revert to ordinary language when speaking to ordinary people. It is not only that we create new words (bit, byte, Fortran, *et al.*), but also that we insist on treating non-words as if they

* This work was supported by the U.S. Department of Energy under contract No. DE-AC03-76SF00098.
 † *Some Cautionary Aphorisms for User-Oriented Computer Management*. The original version was published in Information Processing 80, the proceedings of IFIP Congress 80, North Holland, 1980. A substantial revision was published under the title *The customer's always right, right?* in *Information Resource Management*, August, 1983.

were real words (DASD and EBCDIC are well-known examples), and giving real words unreal meanings:

0.2: *Computing is the only discipline in which we consider adding a wing to the building to be maintenance.* [5]

A second traditional complaint that ordinary people have about their dealings with computing people is the lack of sympathy they receive when they run afoul of the dehumanizing aspect of traditional computing systems. It is both ironic and distressing that the computer, which has the capability to personalize and individualize the user interface, has so often been used as an excuse for the de- and im-personalization of computer-driven systems. Ordinary people have been driven to the conclusion that

0.3: *The computer establishment is to the user as the dog is to the tree.* [6]

The current hope is that the spread of popular computing, which is putting a computer on everyone's desk, or at least in everyone's future, will remove some of the distance between computer people and ordinary people. The information center, where many ordinary people today are introduced to computing, is seen by some in the computing establishment as a positive step in this direction. The information center must become a friendly and familiar place, however, if it is to contribute to a lessening of the confusion and frustration that ordinary people associate with computing.

That it has not yet done so is attributable to several factors, including a lack of suitable introductory and explanatory material, the continuing arrogance and thoughtlessness of the computing community (many of whom still believe that the world must adapt to the computers rather than they to the world), and the fact that ordinary people approach computers and computing in far different ways than computer people do. Therein lies the subject of the next section.

ORDINARY PEOPLE USING COMPUTERS

Ordinary people have different expectations for computing systems than computing people do. Ordinary people react to computing systems differently than computing people do. Things that are friendly and obvious to a computer person may be quite mysterious and threatening to an ordinary person. Computers are not the center of an ordinary person's worklife, even when their use is essential. These observations suggest that four elements of the interface between ordinary people and computer-assisted systems are of particular interest to us here: strong *task orientation* on the part of ordinary people, *rapidity of change* in computer-associated environments, a few persistent *universal problems*, and *the ill-tempered interface*.

Task Orientation

Computer people have an unusually strong desire to learn about the fundamentals of the systems they use [7]. Ordinary people are less driven to

learn; when they ask for the time, they do not want to be told how to build a clock.

1.0: *Performance is more important than knowledge; ego is more important than enlightenment. [8]*

When ordinary people have to use a computer or a workstation to accomplish a task, they want to be told how to do the task, and they don't expect or want to be told how to format a disk or create a macro library along the way. Since

1.1: *Text processing is about text, it shouldn't have to be about programming, [9]*

and ordinary people, unlike computer people, do not enjoy the programming aspects of some of today's text-processing systems.

It takes a certain amount of intelligent effort to perform any task with a computer. This includes both *task effort* and *system effort*. *Task effort* is applied directly to the task in hand and demands *task intelligence*; *system effort*, on the other hand, applies to the selection and proper application of a suitable (sub)system for the task, and demands *system intelligence*. In the past it was necessary that the actual user of a computing system (as distinguished from the beneficiary or the end user) supply both the task effort and the system effort, through a fairly complex set of interfaces. Some modern systems are being introduced under the premise that they are suitable for use by their end users and beneficiaries, i.e., by ordinary people. Ordinary people are not interested in acquiring and maintaining system-intelligence;

they prefer to concentrate on the exercise of their task-intelligence. If these systems are to succeed, then, they must supply all, or nearly all, the system intelligence required for the tasks to which they are put. In short,

1.2: *It takes a smart system to serve an ordinary person well. [10]*

The Quicksand of Change

Computer people welcome the excitement of the continual introduction of new technology that characterizes computing, and they are continually surprised that ordinary people are less than enthusiastic when constant change is demanded in the way they are expected to do things. They (the computer people) forget that

1.3: *Attitudes do not change at the same rate as technology, [11]*

and that, in the world of ordinary people,

1.3.1: *Established technology tends to persist in the face of new technology. [12]*

As your grandfather might have said, rather wistfully, "The old ways are the good ways." If the old ways work, no reason is seen to adopt the new.

Ordinary people have invested time and energy in developing their skills, and are not always willing to sacrifice them on the altar of technology. In using computer-assisted systems the problem has been not so much that new skills have to be learned once, but that those new skills are so soon outdated, and another (incompatible, as often as

not) set of new skills has to be learned, and another, and another, *ad infinitum*. This becomes more critical at higher levels in the corporate hierarchy, in systems that purport to support high-level executive management. One thing that a senior executive does *not* need is a system that must be (re)learned each time he tries to use it.

1.4: *If it isn't self-evident, it doesn't belong in the executive suite. [13]*

1.4 is in marked contrast to the creed of the true computer person:

1.4.1: *If it can be understood, it isn't finished yet. [14]*

Persistent Perils

There are a few problems that have been with us since well before computers came upon the scene, but which have adopted computer dress for the computer era. Two of the most familiar are:

1.5: *Otherwise intelligent people will (still!) accept arrant nonsense if it is backed by a computer printout. "Garbage in, gospel out." [15]*

1.6: *People vary what the system lets them vary and ignore everything else. [16]*

1.5, of course, is a restricted form of McLuhan's "the medium is the message". It used to be that anything printed in capital letters on 14×11-inch paper with sprocket-holed tearstrips was treated as divine revelation. Today, it is the output of the laser printer that carries the aura of authority.

The reader of a rough draft prepared on a laser printer is only with great difficulty persuaded that it is not a final, smooth copy. The finished appearance of the document, properly formatted and displayed in sharp, clear, readable type, belies the fact that it is merely a working draft. (Perhaps we should call this the tyranny of proportional spacing.) Whatever the reason, it remains the case that the credibility of a statement depends as much upon presentation (i.e., form and medium) as upon content.

There are almost as many versions of 1.6 as there are of Murphy's original Law. It expresses mankind's nearly universal tendency to take a sanity-preserving desperation measure and, through laziness, transform it into a system-preserving narrowness of view. It was formerly most seen in bureaucratic or big-business contexts, but is now moving into the computing milieu. Other formulations, in rough chronological order, include:

1.6.1: *If it ain't official, it didn't happen. [17]*

1.6.2: *If you don't have a birth certificate [Social Security number, driver's license, . . .], you don't exist. [18]*

1.6.3: *If it was made in Detroit, it must be a car. [17]*

1.6.4: *If you can't do it on/within the system, you can't do it. [18]*

1.6.5: *In the electronic workplace, people's knowledge of events is limited to the ways in which they are represented by*

the machine and the ways in which they can alter those representations.
[16]

1.6.6: *The real world is what's on the screen.*
[17]

Two other long-standing problems experienced by ordinary people when dealing with computers is their (the computers') literal-mindedness and their (the people's) unrealistic expectations.

With respect to the first of these, ordinary people are used to dealing with other ordinary people, who, albeit with occasional irritation, are willing to compensate for some imprecision of utterance, and who tend to question unreasonable input. When an ordinary boss asks an ordinary secretary to destroy all the files, the usual response is something like "Are you sure you mean *all* of them?" When an ordinary boss asks ordinary *unix*[™] to destroy all the files the response is the system prompt for the next command *after* all the files have been destroyed. The problem is that

1.7: *Machines follow instructions better than people do. Even stupid instructions.* [19]

They persist in doing what we ask instead of what we thought we asked, or intended to ask, or meant. (See also 2.3.3.)

In the early days of computing, computing people rather enjoyed the priesthood that was thrust upon them and the awe in which they and their machines were held. They allowed and encouraged ordinary people to acquire a perception of computers and computer-driven

systems as mysterious but immensely powerful servants, capable of essentially anything they decided to undertake. They have generally failed to teach ordinary people anything about the *fundamental limitations* of computer-assisted systems (one of the most fundamental of which is the narrow view of the computer people who build the systems). The result is that ordinary people have trouble distinguishing the routinely possible from the truly revolutionary. They have no basis from which to judge the claims of the system sellers (we will return to this topic in the discussion of new systems, below) and so they expect computers and computer-driven systems to work exactly according to specifications. Computer people have learned, through bitter experience, to expect somewhat less. The parents among you will recognize this as an expensive form of the Christmas Rule, which is most often found in its GNF (Gumperson Normal Form):

1.8: *The more desirable the toy [system], the sooner it breaks.* [20]

but which has also surfaced as the Axiom of Adult Disillusionment:

1.8.1: *Children expect their toys to work; adults expect them to break.* [21]

In the context of the present discussion, this can better be worded as

1.8.2: *Ordinary people expect the (computer) systems they use to work as advertised; computer people are surprised if they do so.* [18a]

Faulty Interface

It has already been noted that ordinary people are less enthusiastic than computer people about constant change in the way they must work. This remains true even when that change results from the introduction of clever new features. Rather than revelling in the brilliance of the concept, ordinary people are more likely to be bothered or confused by the new interface.

1.9: For every subtle feature on a workstation there are several annoyed users who wonder what it is and how it works. [22]

Because they don't understand this fundamental difference between themselves and ordinary people, computer people tend to treat the situation as simply another documentation problem, to be cured by adding yet another incomprehensible manual to the poor user's already overwhelming stack of incomprehensible manuals. This approach has a fundamental flaw, with an obvious corollary:

1.9.1: You do not improve a poor interface by carefully documenting its idiosyncracies. [22]

1.9.2: Fix the problem, not the manual. [23]

All of this -- the unwanted rapidity of change and the person-height stack of unreadable manuals -- is done in the name of "user friendliness". The problem is that computers are inherently unfriendly; it is possible to tame them, but only a certain few will ever find them truly friendly:

1.10: Computers are like cats; dogs are user-friendly. [24]

This section would not be complete without noting that when, despite all the hurdles, ordinary people actually start to use computer systems, we quickly discover that they are exactly like computer people in at least one respect:

1.11: Human impatience remains constant at about 3 seconds. [25]

This, of course, is just a computer person's version of a more general truth:

1.11.1: At any level of traffic, any delay is intolerable. [26]

NEW SYSTEMS

The problems with new systems are numerous and, in contrast to the systems themselves, well documented. Nevertheless, there are a few lessons that bear repeating in today's atmosphere of rapid introduction of new systems to ever-broadening spectra of users, including many of those ordinary people we have just been considering. The first of these is the primary, but often forgotten, Law of the Introduction of New Systems:

2.0: New systems create new problems. [27]

If we are extraordinarily lucky a new system will solve the problems it was intended to solve, and the new problems will be no more intransigent than the old ones. But there are three particular problems that frequently arise with new systems

whose clientele includes new or naive users: *exaggerated expectations*, the lure of *upward compatibility*, and *instant obsolescence*.

Expectations

The problem of exaggerated expectations was alluded to in the discussion of 1.8, above; as was noted there, it stems in part from the aura of mystery and infallibility that surrounded computing in the early days, but it derives much of its current strength from a failure of ordinary people to truly understand that

2.1: *Vendors are to vend, [28]*

and not to be one's friend and mentor, or even to render impartial expert advice. Asking a vendor for advice is like asking the barber if you could use a haircut; you *know* what the answer is going to be [29]. More specifically,

2.1.1: *Vendors oversell expectations and understate the costs and energy necessary to make the system work as advertised. [8a]*

Ordinary people tend to forget that almost any system will work under the right circumstances, and that vendors have taken a great deal of care to ensure that the on-site trial occurs under the "right" circumstances. They (ordinary people) understand that in the automotive world nothing breaks until *after* the warranty period, but they fail to transfer this experience into their new working environment. Unlike art, demonstrations do not imitate life. This provides us with one of those rare instances when perfect

performance is actually a confirmation, rather than a refutation, of Murphy's Law:

2.1.2: *The deficiency will never show itself during the dry run. [30]*

Upward Compatibility

One of the most oversold of virtues is "upward compatibility". Upward compatibility makes it possible for vendors to continue to make money from ratty old programs that should be junked, just by allowing them to be run at horrendous cost in emulation mode on later systems. Thanks to this undeserved immortality, many primitive programs that should have been strangled in infancy have persisted into decrepit old age. In other words,

2.2.1: *"Upward compatible" means we get to keep all our old mistakes. [31]*

At least one practitioner takes an even more astringent view of the joys of upward compatibility, contending that

2.2.2: *Upward compatibility is really bug-for-bug compatibility. [32]*

The System is Always Out-of-Date . . . even when it's new

Another old lesson to bear in mind in the new era has been expressed rather plaintively about evolution:

2.3: *I was designed by a process I don't understand to live in a world that no longer exists. [33]*

It has been expressed often and pungently, and in a number of forms. The following selection carries us from the initial desire for a new system, through its acquisition and implementation, to the final realization that it is unsuitable, inadequate, or both:

- 2.3.1: *Needs are a function of what other people have. [34]*
- 2.3.2: *People ask for what's available rather than what they need. [35]*
- 2.3.3: *People don't know what they want until you give them what they ask for; they don't discover what they need until you give them what they want. [36]*
- 2.3.4: *How can I know what I think until I see what I say? (I don't know what I really want until I see what I do when I have freedom of choice.) [16]*
- 2.3.5: *The systems we install are designed for the current user community, not for the system-experienced user community. [37]*
- 2.3.6: *The people we got the system for no longer exist. [37]*
- 2.3.7: *By the time you learn to use the system you don't want it any more. [37]*

POWER TOOLS

It has often been stated in the trade press that the new "power tools" are just what ordinary people

need to help them tame the computer and start getting useful work from it. These statements usually overlook the double- or triple-edged nature of power tools [38]. It is certainly true that power tools wielded by expert hands can vastly speed the development process, but it is also true that they can quickly and seriously damage both the wielder and the object under construction. They are also no guarantee of quality:

- 3.1: *It's just as easy to use powerful tools to build a bad system quickly as to build a good one. [39]*
- 3.1.1: *Applications generators are the microwave ovens of the programming world: They get it done faster, but don't make it any better. [39]*

After all,

- 3.1.2: *Necessity is not the mother of good design. [40]*

The embarrassment of riches now facing most computer users creates another problem: Which tool should be used for which purpose? More problems result from having several ways to do something than from not having any way at all, and the wrong choice of tools can make even the simplest of tasks into an exercise in frustration. People will use a familiar tool even though it's unsuitable rather than risk an unfamiliar one. And they will complain about the results, even though

- 3.2: *It's not very efficient to eat your soup with a fork. [41]*

Another problem with power tools is their environmental specificity. They have been carefully tailored to work well in the intended environment, and they often don't work as expected, if at all, when moved to a new one. This is due not only to anti-transfer devices designed to prevent piracy, but also to a general property of programs that is surprisingly poorly-understood, considering its importance and prevalence. Despite the best efforts of their authors,

3.3: Other people's programs work only in other people's computers. [42]

These are all rather general observations; in addition, specific power tools present specific problems. The next three items are but a tiny sampling of the many tool-specific comments that have appeared in the past few years:

3.4: The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offense. [43]

3.5: APL is a write-only language. [44]

3.6: An open public teleconference is an intellectual flea market. [45]

(Some readers may scoff at the inclusion of COBOL as a power tool. They forget how great a leap forward it was. A power tool remains a power tool even when it becomes obsolete.) The new interface language that has the technophiles excited is, of course, *natural* language. They reason that because nobody has to learn it, English will make everyone a successful and contented user. Unfortunately,

3.7: If you allow people to program in English you will discover they don't know English. [46]

People will program no better in English than they did in Fortran (and for many of the same reasons).

ENTHUSIASTS

(I.e., those who have just discovered the power -- and pleasures -- of computing)

So far we have been primarily concerned with ordinary people; there is another group of people who can cause problems for the computer manager, and that is the enthusiasts. Their range of interest is often quite narrow, but their knowledge can be quite deep in the chosen area, and they are often tempted to interpret their detailed knowledge as a sophisticated understanding of the underlying system. They combine the fervor of the newly-converted with the energy of a hyperactive five-year-old, and their moral sense (alas) is also often that of the five-year-old. They are quite adept at discovering the tricks of their new trade and putting them to unexpected, and perhaps inappropriate, uses. For enthusiasts, ingenuity is its own reward [8a], and need have no legitimate purpose:

4.1: It's more fun to play with a new tool than to learn how to think right. [47]

If asked to justify this activity, enthusiasts will assert that they are investigating their new tools, developing their knowledge, and gaining experience. Unlike ordinary people, enthusiasts

do enjoy computing for its own sake, and are curious about how these powerful new tools work. They enjoy investigating interesting tangents [48], and are not always suitably circumspect about observing either the technical limitations or the administrative constraints that should apply to their activities. Whereas, as we have noted above, ordinary people expect systems to work exactly to specifications, enthusiasts expect to exploit the exceptions. To enthusiasts,

4.2: *Computer security (and the copyright laws) are about as useful as bones in a potato. [49]*

4.3: *Moral issues tend to melt in the heat of a juicy problem to solve. [36]*

After all,

4.4: *It's easier to get forgiveness than permission. [50]*

It is tempting to speculate whether enthusiasts are of weaker moral fibre than ordinary people to begin with, or whether exposure to the computer has made them so. While expert opinion on this topic differs (see 4.5.2 and 4.5.3 below), it seems to be universally recognized that computers are, indeed, a source of power:

4.5: *He who has the stuff in his computer has the power. [51]*

4.5.1: *Manage the data and you have controlled the environment. [40a]*

4.5.2: *Power corrupts. [52]*

4.5.3: *Power attracts the corruptible. [53]*

The viewpoint of the enthusiasts themselves is quite clear, however:

4.6: *That which we call sin in others is but experimentation in ourselves. [54]*

One last thought about enthusiasts. They are often accused of charging off in all directions without adequate planning. The accusers are bound by the chains of tradition to the paper-trail mentality of modern American bureaucracy. They fail to understand that

4.7: *Planning is an unnatural act that few do willingly and none do well. [24]*

In fact,

4.8: *Planning is cost-defective when the consequences of error are trivial. [37]*

The instant response of the PC, allowing the user to see the consequences of his actions immediately, often renders detailed planning unnecessary, and, in fact, counterproductive.

On the other hand, some caution is recommended:

4.9: *The ability to iterate is not a license to be sloppy. [3]*

CAUTIONARY AFTERWORD

It is the nature of aphorisms that they take the form of general rules. General rules have the disadvantage that there are special cases to which they do not apply; unfortunately,

5.0: *Everything in real life is a special case.*

[55]

REFERENCES & SOURCES

Most of the aphorisms here presented are quasi-universal truths. As such, they exist in many versions. Where I have encountered an existing version that suits my purpose, I have used, and acknowledged, it. Where my taste and style have been better served by a new formulation, I have not hesitated to supply it; in these cases the reference points to the version that was the immediate inspiration for its inclusion in this work, and thus not to a literal source. Some of the aphorisms have been wrenched out of context, and many have suffered some mutilation in my quest for appropriate wordings. That they remain pungent and pertinent in the face of this treatment is ample demonstration of their value as general principles. I apologize neither for the mutilation nor for the fact that the bulk of the paper consists of the wisdom of others. There is, after all, nearly as much merit in recognizing wisdom as in producing it. (And while creativity may be great, plagiarism is quicker. [56])

I have made extensive use of several earlier treasuries of wisdom. To conserve space, they are given below in abbreviated fashion, to wit:

Expl = *The Official Explanations*, Gordon Dickson, Dell, 1980

Lights = *Are Your Lights On?*, Gause and Weinberg, Ethnotech, 1977

Rules = *The Official Rules*, Gordon Dickson, Delacourt, 1978

St. Pat's = *St. Patrick's Almanack*, the "Open

Channel" column in the March '86 Issue of *Computer*

Syst = *Systemantics*, John Gall, Quadrangle, 1975

- [1] *The Yeomen of the Guard*, 1883.
- [2] Law Law #1 in *Expl*.
- [3] J. D. Gould and C. Lewis, "Designing for usability: Key principles and what designers think", *CACM*, March, 1985.
- [4] Goethe, quoted in *St. Pat's*.
- [5] Jim Horning, quoted in *St. Pat's*.
- [6] Oral comment at the UC Management Institute, 1979 (originally referring to Administration and Faculty).
- [7] This is one of the principal findings of J. D. Couger's continuing research into what makes computer people tick. See *Motivating and Managing Computer Personnel*, Couger and Zawacki, John Wiley and Sons.
- [8] The second half comes from J. Seymour, *Coping with computer egos* (an AMA Management Briefing), 1984; the first half is an obvious variation on the first.
- [8a] J. Seymour, *op.cit.*
- [9] F. J. Grant, "The downside of 4GL's", *Datamation*, July 15, 1986

- [10] L. M. Branscomb and J. C. Thomas, "Ease of use: A system design challenge", *IBM Systems Journal*, 1984 #3.
- [11] K. Buechner, BNR.
- [12] Blaauw's Law in *Rules*.
- [13] D. F. Stevens, "Supporting the literate user", *EDP Performance Review*, April, 1984.
- [14] Smith's Second Principle of Bureaucratic Tinkertoys in *Rules*.
- [15] The tag end of 1.5 entered into the public domain right on the heels of the original *GIGO*. Various forms of this Great Truth have appeared in such diverse sources as Variation 9 in D. F. Stevens's "A look at white-collar personal computing: Theme and variations", W. H. Agor's "The logic of intuition: How top executives make important decisions", *Organizational Dynamics*, Winter 1986, and J. Shore's *The Sachertorte Algorithm*.
- [16] K. Weick, "Cosmos vs. chaos: Sense and nonsense in electronic contexts", *Organizational Dynamics*, Autumn, 1985.
- [17] The Fundamental Law of Administrative Workings (FLAW) and its corollaries, in *Syst*.
- [18] Common experience, becoming (alas) more common. A pithier version was seen in an old country store by P. Howard: "If we ain't got it, you don't need it."
- [18a] While not yet common, cynicism resulting from computer experience is no longer truly rare. A. E. Housman had a rather different field of endeavor in mind when he wrote the following, but the optimist might do well to bear his implied advice in mind: "The thoughts of others were light and fleeting, of lovers meeting, or luck, or fame; Mine were of trouble and mine were steady, so I was ready when trouble came." It should also be noted that while computer people as *users* may expect systems to fail, computer people as *suppliers*, on the other hand, expect them to work.
- [19] This well-known law has seen several formulations. The most dramatic is due to W. W. Jacobs, in his story *The Monkey's Paw*, while the pithiest is the (folkloric) "Pay attention to what I mean!" The version of 1.7 is due to J. Seymour, *op. cit.* [8]
- [20] A direct application of the generalized Gumperson's Law (*Rules*).
- [21] R. Byrne, quoted in *ComputerWorld*, October 15, 1985.
- [22] D. V. Morland, "Human factors guidelines for terminal interface design", *CACM*, July, 1983.
- [23] This is, of course, derived from the tendency of vendors to fix the manual instead of the bug, i.e., to redefine the specifications to fit the delivered product, rather than correcting the product to conform to the specifications

- that persuaded you to buy it in the first place.
- [24] Oral comment at the Fifth Conference on EDP Performance Management, 1984.
- [25] W. Spencer, oral comment during a product announcement.
- [26] Bruce-Briggs Law of Traffic in *Rules*.
- [27] Another remarkably broadly-based observation. In this pure form it is one of the first laws noted in *Syst*. Slightly different versions occur in *Lights* and as the Cliff-Hanger Theorem, and Epstein's, Peers's, and Severeid's Laws in *Rules*.
- [28] Derived by analogy from the title of the children's book *A hole is to dig*.
- [29] First Law of Expert Advice in *Rules*. This has found more formal exposition in Kharasch's Law of Institutional Expertise (also quoted in *Rules*): "The expert judgement of an institution, when the matters involve continuation of the institution's operations, is totally predictable, and hence the finding is totally worthless." Successful venditure is necessary to the continuation of the vendor's institutional existence; *Q.E.D.*
- [30] Boyle's Law #3 in *Rules*.
- [31] Dennie Van Tassel in *St. Pat's*.
- [32] An anonymous contributor to *St. Pat's*.
- [33] Donald Symons, in an interview in *Psychology Today*, February, 1981.
- [34] The Joneses Criterion, misnamed *Jones's Principle in Rules*.
- [35] A variant of the Joneses Criterion, derived from F. J. Grant's observation (sidebar to [9]) that "if all reporting combinations can be easily obtained, all reporting combinations will be requested."
- [36] *Lights*.
- [37] D. F. Stevens, *More cautionary aphorisms for user-oriented computer management*.
- [38] The sole exception I have encountered is Michael Hammer, in "The OA mirage" (*Datamation*, February, 1984): "Power users require power tools on which a novice might hurt himself."
- [39] B. Stahl, "The trouble with application generators", *Datamation*, April 1, 1986.
- [40] An interpretation of a principle stated in K. Christoff, "Building a 4th generation environment", *Datamation*, September 15, 1985. Just because you have to get from A to B in a hurry doesn't qualify you to design a jetliner.
- [40a] K. Christoff, *op. cit.*
- [41] A. Metz, "A rebuttal: *unix* realities", *DEC Professional*, January, 1984.

- [42] First Law of Gardening in *Rules*.
- [43] E. Dijkstra, "Truths that might hurt", *Sigplan Notices*, February, 1982. (He also takes on PL/1, FORTRAN, APL, JCL, and the entire mystique of technical/scientific computation.)
- [44] Roy Keir in *St. Pat's*. He goes on to say by way of explanation that "I can write programs in APL, but I can't read any of them."
- [45] S. R. Hiltz and M. Turoff, "Structuring computer-mediated communication systems to avoid information overload", *CACM*, July, 1985.
- [46] "[O]ne of the standard laws of computer programming", according to K. Christoff, *op.cit.* [40]; this is also No. 8 (of 8) of the laws of computer programming listed in *SICPLAN Notices*, Vol. 2, No. 2.
- [47] D. Gries, "Bridging the software gap", *Datamation*, February 15, 1985.
- [48] Oral comment at the Seventh Conference on EDP Performance Management, 1986.
- [49] J. Zimmerman, "PC security: So what's new?", *Datamation*, November 1, 1985.
- [50] Variously referred to as *Burt Reynolds's Rationale* (oral comment at the Fourth Conference on EDP Performance Management, 1983) and the *Jesuit Principle* (in *Expl*).
- [51] Boies, Gould, Levy, Richards, and Schoonard, "The 1984 Olympic Message System -- A case study in system design", *IBM Research Report RC 11138 (#50065)*, May 2, 1985.
- [52] Acton's Law in *Rules*.
- [53] This might be called Acton According to Herbert; it is from an interview with Frank Herbert in *Psychology Today*, October (?), 1984.
- [54] Emerson's Insight in *Rules*.
- [55] Business Maxim #5 in *Expl*.
- [56] Clay's Conclusion in *Expl*.

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.

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

