

Tension Headache #84 Users and Computers

By Celia Waker

Without a doubt we exist in a highly technological age. We approve of computerized banking, appreciate word processors, and even admire automated bibliographic searches, to say nothing of our awe for computer games. These are all services provided by computers. We use them with only slight hesitations and with little regard for the design considerations involved in creating accurate, efficient, and useful computer programs for our use.

The "tension headache" begins, however, when we neophytes must become more deeply involved in data processing. Just the vocabulary — baud, bits and bytes, floppies and flippies — could convince a person that two aspirin and going to bed may not be sufficient. Foreseeing a longer stay in data processing, perhaps involved in computerized data management systems, one might with justification fear the debilitating effects of the tension inherent between two perspectives of data processing design:

- humans control computers, determining how and what they are capable of accomplishing; and
- computers, based on their technical restrictions, dictate to humans, structuring how functions will be accomplished and therefore limiting what will be accomplished.

On the one hand, humans create marvelous, intricate programs designed to accomplish thousands of simultaneous banking transactions nationwide. On the other hand, the bank clerk notifies you your address is restricted to 12 characters because "the computer won't allow more."

Theodor H. Nelson candidly discusses the "leetle (sic) white lies about how such-and-such is the computer's fault and not your decision . . . 'The computer won't let us' (means) WE DO NOT CHOOSE TO PROVIDE, IN OUR PROGRAMS AND EQUIPMENT, ANY ALTERNATIVES . . . Now, it is often the case that good and sufficient reason exists for the way things are done. But it is also often the case that companies and the public are inconvenienced, or worse, by decisions the computer people make and then hide with their claim of technical necessity."

Individuals responsible for initiating data management systems must mediate between the needs of the information user and the requirements of the computer system. The goal is to use the powerful capabilities of the computer system (remembering it does have legitimate limitations) to efficiently accomplish a task defined by the user. The following suggestions and examples are intended to be helpful in this process.

Try to determine how immutable a software package is, especially if you are assessing a commercially available, pre-programmed software package. The greater the flexibility offered by the software, the greater the likelihood of meeting the information user's needs — accompanied perhaps by a greater risk of offering too many possibilities and being excessively cumbersome to use.

- can large, system-wide changes be made?
Example: can the program be initialized on the

computer system to accept dates in the format you desire (01/24/84 instead of January 24, 1984), or is that format predetermined and unchangeable?

- as the program is run on the system, does it provide alternatives?
Example: after entering expenditures in a computerized accounting system can you exit the program or *must* you proceed to a balance sub-program?
- if the program is purchased, is it "locked up" or can your programmer customize the commercially available program to your needs?
Example: can you insert a brief description of invoiced items as a customized part of a standard accounting package?
- if the program is purchased, can or will the software development firm from whom you purchased the programs make alterations for you? For what fee? Who retains ownership of the new elements?
Example: will the company develop a linking program from your "spread sheet" program (where you can do math calculations) to your word processing program (where you can edit)?

Pay special attention to the planning phases of designing how the entire system will function, how components will interact, and what the accomplishments of the system will be and could be in the future.

- has the information user thoroughly planned the functions of the program so mid-stream changes will be less likely?
Example: will a list of the parks need to be alphabetized, in which case the "key" to each park unit must be the park name, rather than an alphabetic or numeric code?
- what system-wide, basic restrictions apply to this type of programming?
Example: a new program designs graphic displays of research results. Is the printer capable of producing graphics?
- at what point does making the computer system and the programmers "jump through a hoop," although possible, become inefficient?
Example: a program will sort park unit information to three levels (region, then unit type, then alphabetized park unit), but a user would like the program altered to go to five levels.

Individuals who mediate between information users and data manipulators (computers) often find themselves "interpreting" the requirements for each involved party. On the one hand, the user may not understand the necessity of abiding by computer system requirements. Perhaps because of an over-estimation of the omnipotence of computers, the user may feel responsible for only the broadest scope of a data management system: the ends, rather than the means. However, the critical argument remains "garbage in, garbage out."

Planning, to a great degree, determines the quality of the results. It behooves the user to protect his or her own interests by carefully defining the information according to the data requirements of the computer system.

On the other hand, the computer and its programmer may specify overly restrictive requirements or not appreciate the informational needs of the user. Computers do have legitimate hardware and software requirements which are either unavoidable or circumvented only with great difficulty. People control the inanimate machine, but there still exists a point beyond which it is unreasonable to expect the system to flex. That point needs to be recognized and appreciated.

The two outlooks comprise a power struggle:

- humans, according to their needs, command computers vs.
- computers, because of technical necessities, overrule user needs.

The two outlooks greatly affect computerized management design and implementation, but they remain continually in conflict. It is not too utopian to propose that the tension produced also can be viewed as a constructive force. The phenomenon of tension in physical structures and even in humans serves useful purposes, and this tension is no different. It prevents the user or the computer from dominating data processing.

The over-all efficiency of a given program is better served by striking an equitable and efficient compromise between user needs and computer requirements. Users may find that the computer capabilities more than compensate for imposed restrictions. The structure the computer insists upon prevents the user's needs from creating inefficient and unreasonable programs. The computer system, on the other hand, may be challenged to increased capabilities by the demands of the user. Without the impetus of the other, neither aspect would be goaded to the quality of which it is capable.

Awareness of the user-computer power struggle should alert data management system designers to potential "tension headaches" in store for them. However, awareness of the problem also affords opportunities to develop creative, efficient, and very useful applications which challenge even our current advanced stage of technological development.

Walker is with the Air and Water Quality Division, NPS, Fort Collins, CO.

¹ Nelson, Theodor H., *Computer Lib*, the distributors, South Bend, Indiana, 1974, p.8. This unusual book is based on the premise that everyone should know about computers and deserves explanations in layman's terms. It deserves status as "required reading."