CRITERIA FOR USEFULNESS OF COMPUTERS IN OFFICES

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305
jmc@cs.stanford.edu
http://www-formal.stanford.edu/jmc/

1999 Jul 8, 5:02 p.m.

Abstract

The thesis of this lecture will be that there is no difficulty in getting people to use computers in offices provided the computer and its applications are genuinely useful. However, the criteria for usefulness are often not what one would imagine, and some further research is required before the **real computer revolution** happens.

I became interested in office use of computers in 1957, and this was one of the motivations for my research on time-sharing - the main one being use in artificial intelligence research. The first time-sharing system at Stanford was a PDP-1 in 1964. For that we provided no off-line program preparation equipment, and when we specified a display system, we insisted on both upper and lower case with a view to using the system for preparing documents as well as other applications.

A. Office Computing at the Stanford Artificial Intelligence Laboratory

The Stanford Artificial Intelligence Laboratory received its PDP-6 computer in 1966, and it was planned to use the computer for office applications from the beginning. All displays and the printer permitted upper and lower case and a reasonable set of mathematical symbols, and we began improving on-line editors. The first PhD thesis written and printed on the computer was in 1971.

Our progress in office use of the computer was mainly paced by hardware acquisition. While we could print documents from the beginning, there was no motivation to prepare them on-line as long as we were using uppercase-only teletypes for terminals. When we acquired our first display system, on-line preparation of documents began, but until we installed our 60-terminal Datadisc display system in 1971, the terminals were in a terminal room. Putting the terminals in offices, which included supplying the secretaries with terminals, was a major step. Gradually more and more Laboratory administrative files were kept on line, and the secretaries could help prepare papers. However, since it also became easier for researchers to enter and edit their own papers, there was less typing per secretary. People differ in the extent to which they work through typists and secretaries, and an office system should provide for these differences.

Getting on the ARPA net gave a big stimulus to message sending inside the Lab as well as over the net. The E editor permitted more effective use of the displays. The POX and PUB document compilers automated many editorial aspects of document preparation. When Xerox gave us a Xerox Graphics Printer, this made possible preparing multifont documents with arbitrary character sets.

Many people work both in the Lab and at home, and their easy use of office computing requires home terminals, of which we now have a fair number.

Donald Knuth's TEX and the associated acquisition of high quality printing equipment have substantially increased the documentation use. Knuth's vigorous publicizing of TEX including the book has been at least as important as the program itself. Even mathematicians are beginning to use our computers for producing theses and papers. They have always been among the slowest to make use of computer facilities. This is because mathematics is mostly done at a high level of abstraction, and we are only beginning to develop computer programs that communicate this abstractly.

Besides the main programs associated with office use, many auxiliary programs for looking up data in files and even computing have been developed.

B. Conclusions from Our Experience

1. Up time of the computer and safety of its file system determine whether

it will be accepted for office use.

2. Secretaries and other clerical people can use computers even without much training. Their motivation to do so requires that they have terminals on their desks and that the computer be reasonably reliable. Some amateur human engineers imagined that they might have a problem with a keyboard in which the top row of keys was displaced from standard. Like almost all of the researchers, they never noticed. For workload reasons, we have had to use many temporary secretaries from manpower agencies. This has proved unexpectedly easy. I saw one temporary typing at a terminal twenty minutes after her arrival under the supervision of a regular secretary working at another terminal in the same office.

3. Display terminals are much better and cheaper than hard copy terminals. The latter are noisy, and waste paper gets spread around. Of course, there needs to be a good accessible printer, but many people print only daily.

4. Having many display terminals, provided they have at least minimal facilities, is more important than having a few super terminals. We will shortly have a really good experimental test of this proposition, because the Computer Science Department now has perhaps 75 terminals in offices and about 15 Xerox Alto systems in terminal rooms. I am betting that people who have Datadiscs in their offices will use them rather than take their papers to a terminal room down the hall.

5. Reducing the noise level is important. A big improvement comes from eliminating typewriters, and it will probably prove worthwhile to develop less noisy keyboards.

6. If they have a proper interactive style, many programs can be used without formal documentation - whether a hard copy manual or full interactive documentation. If the program is written in a style familiar to our users and its general capabilities are known, most people will try to use it without reading the manual. For example, punctuation of the arguments of commands must be standard: Don't require a comma in one place and a semicolon in another. "?" should always get information about the options available at the present point in the interaction.

7. Programs should interact at a single level as much as possible. Even an experienced user often gets lost in a hierarchy of modes and submodes. Menus are bad, because as soon as a user gains the slightest experience, he hates having the screen cluttered up with changing menus. The information provided by menus and question-and-answer formats can be provided by letting the user say "?" whenever he needs to know what his options are. Worst of all are interactive programs that clear the input buffer before accepting a user's command in a new situation. The even slightly experienced user will want to type ahead, often totally ignoring what is on the screen, in order to get the program into a desired state.

8. Keeping up with new programs and improvements in programs has proved impossible so far. People use only a part of the facilities of our interactive programs. This isn't a tragedy; everyone has his own appropriate balance of effort between learning about new features and using the old ones.

9. Many jobs do not involve continuous use of the computer, and it is more important to meet the needs of the casual user than those of the beginner. A person will invest considerable effort in first learning how to use a computer, but if he has to learn all over again after a two-month layoff, he won't put in the effort a second time. Computer use should not be like instrument flying - requiring lessons if you haven't done it for six hours in the last ninety days.

10. The utility of many proposed applications of computers is limited by the work required to put the information in the computer. The prize example is the proposal to use home computers to keep track of items in the pantry and warn the householder when to buy more. Even if the terminal were in the pantry, it would be too much trouble and people would forget. A bar code reader in the pantry might make it reasonable, but the geneticists may have to breed hens that lay eggs with bar codes.

C. Some More Controversial Contentions

I cannot claim that these contentions have been verified by experience, because I haven't enough.

1. Executives will use computer systems provided they are genuinely useful, but when they are not useful, the complaints will often be misleading about the real reasons. An executive will always be a casual user. Therefore, the terminal must be unobtrusive and quiet; his secretary must have one too, and if he works at home, there must be a terminal at home too. In fact, the message use of a computer is most helpful out of normal working hours. I got out of bed last night, because I remembered a message I had been intending to send for a week, sent it and forgot the matter till I received the reply this morning.

   Usefulness for executives will depend on how many of the people with whom they must communicate also have terminals.

   Anyone who does much work at home should have a terminal at home.

2. Idiot-proofing programs is often a bad idea. It is easy for the designer of an on-line system to get into a state of mind where he regards the user as an idiot who must be prevented from making all kinds of mistakes. Indeed the books and papers on interactive programming take this attitude. In fact, the people who write about supervising programmers take that attitude towards their charges. However, it has several disadvantages.

   First it must concentrate on the kinds of mistakes that can be detected and prevented by bureaucracy - whether it be the programmed bureaucracy of a field that allows only numeric input in a certain range or the administrative bureaucracy that requires a comment for every statement in a program. There are many situation in which the bureaucracy spends its time preventing trivial errors, while major substantive errors are ignored, because the input embodying them is "grammatical" according to the lights of the system.

   Second, idiot-proofing takes time, and it often happens that the idiot-proof programs are insufficiently debugged. There is nothing as annoying as trying to get a program to accept input that it is rejecting for trivial reasons.

   Third, idiot-proof programs are usually extremely inflexible and are difficult to modify to take new data into account.

   Let me describe an experiment that unfortunately was never carried out. A certain university found its on-line registration system terribly late, full of bugs, and expensive of computer resources. The experiment was to have the clerks prepare the registration material using an ordinary editor - labelling the items in the text of the record. The files prepared by the clerks would then be processed by programs to get it in the desired form. Unlike on-line input-receiving programs, the processing program could be written while the input was taking place, and if bugs showed up, they could be corrected after the fact. Even last-minute changes in the information to be included could be accommodated. The results of the data-entry could be printed and checked by supervisors or the supervisors could examine them on line. However, the university took the "safer" path of buying another computer.

Unfortunately, the task of writing a computer program for others to use seems to bring out the latent tyrant in many people.

D. A Step Further Out

There are many opportunities for expanding the usefulness of computers in offices, but many of them require the development of standardized facilities.

1. The Dialnet project. Many rival networks for interconnecting computers have been developed, but in my opinion, the possibilities of the ordinary dial telephone network have not yet been fully exploited. That network has the advantage that it already connects all the offices in the world.

   The Stanford Artificial Intelligence Laboratory is developing the Dialnet system. This consists of a telephone dialer and suitable modems connected to our computer and software implementing the Dialnet protocols. Anyone else in the world can similarly equip his computer and users of any computer equipped with Dialnet can communicate with users of any other.

   Sitting at my terminal I will be able to type "MAIL MIKESMITH@202-666-6666 Mike are you free for lunch on Thursday?" Once I have done this, I can use my terminal for other purposes. My computer calls a computer at that number and tells it that it has a message for a user called MIKESMITH. He gets the message immediately if he is logged in - later otherwise. We can do this now for computers on the ARPAnet, but why go through all that politics, when the telephone system is available? Dialnet can also be used for transferring files between computers.

The 1200 baud limitation of present Dialnet is important for some applications but not for messages and transfer of medium-size files like reports. If one specificies NIGHTMAIL the telephone cost for a 9000-byte message will be only a little more than the price of a stamp.

2. National file-naming system. A major application of Dialnet or other inter-computer communication systems will be to transfer files from one computer to another. This is done now but it almost always involves specific technical arrangements between the managers of the computers. In order to transfer files freely (except as restricted by password fences), a national file-naming system is required.

3. Describing other people's files. Many programming languages contain features for describing data structures so that the compiler will generate them and compile programs that use them. However, no one such system will conquer the world and indeed, if progress is to continue, it is not even desirable that a single system be adopted. Therefore people will always want to refer to other people's data structures.

   This can be made possible by a universal system for describing existing files which can be developed using the techniques for describing grammars and data structures.

4. A standardized style of interactive programming will help people use each other's programs.