10. C. A. R. Hoare, *Commun. Assoc. Comput. Mach.* **12** (No. 10), 576 (1970).
11. B. Wegbreit, *IEEE Trans. Software Eng.*, in press.
12. E. W. Dijkstra, *A Discipline of Programming* (Prentice-Hall, Englewood Cliffs, N.J., 1976).
13. D. Gries, *IEEE Trans. Software Eng.* **SE-2** (No. 4), 238 (1976).
14. H. D. Mills, *Commun. Assoc. Comput. Mach.* **18** (No. 1), 43 (1975).
15. C. Reynolds and R. T. Yeh, *IEEE Trans. Software Eng.* **SE-2** (No. 4), 244 (1976).
16. E. W. Dijkstra, *Commun. Assoc. Comput. Mach.* **11** (No. 5), 341 (1968).
17. ——, in *Programming Languages*, F. Genuys, Ed. (Academic Press, New York, 1968).
18. A. N. Habermann, *Commun. Assoc. Comput. Mach.* **15** (No. 3), 171 (1972).
19. C. A. R. Hoare, in *Operating Systems Techniques*, C. A. R. Hoare and R. H. Perott, Eds. (Academic Press, New York, 1973).
20. P. Brinch Hansen, *Assoc. Comput. Mach. Comput. Surv.* **4** (No. 4), 223 (1973).
21. C. A. R. Hoare, *Commun. Assoc. Comput. Mach.* **17** (No. 10), 548 (1974); *ibid.* **18** (No. 2), 95 (1975).
22. J. H. Howard, *ibid.* **19** (No. 5), 273 (1976).
23. S. Owicki and D. Gries, *ibid.*, p. 279.
24. D. L. Parnas, *ibid.* **15** (No. 5), 330 (1972).
25. B. H. Liskov and S. Zilles, *IEEE Trans. Software Eng.* **SE-1** (No. 1), 7 (1975).
26. W. A. Wulf, R. L. London, M. Shaw, *ibid.* **SE-2** (No. 4), 253 (1976).
27. A. N. Habermann, L. Flon, L. Cooprider, *Commun. Assoc. Comput. Mach.* **19** (No. 5), 266 (1976).

Human Performance Considerations in Complex Systems

H. O. Holt and F. L. Stevenson

Acceptable human performance in complex systems depends upon precise human-machine interaction. Such interaction is the focus of attention of design engineers and computer programmers, for example, on the one hand, and of human performance psychologists on the other. The meaning and extent of that interaction has evolved and expanded over the years. We now commonly find computers of various sizes on the machine side of the human-machine interface, and their presence has changed human performance considerations markedly. On the human side we have seen an accelerating emphasis upon man as an information processor, thus adding many considerations to the older—but persistent—anthropomorphic concerns.

In this article we review the "human factors engineering" field briefly, and then discuss in some detail the requirements that computers have put on people and human performance technology in computer-based systems. For examples, and in the citation of solutions, we draw heavily upon our own experience in the Bell system.

There have been human-machine interaction concerns of a sort ever since primitive man first extended his own abilities with simple weapons and tools. In more recent history the industrial revolution accelerated greatly the transfer of work functions from people to machines and complicated the human-machine interface problems considerably.

Dr. Holt is director of the Human Performance and Support Center, and Mr. Stevenson is head of the Systems Training Department at Bell Laboratories, Piscataway, New Jersey 08854.

Human Factors Engineering

It is generally agreed that World War II marked the beginning of a professional approach to what came to be called human factors engineering, that is, a systematic approach to studying problems of human-machine interaction and to arriving at practical solutions on a scientific basis. Before the war, going back into the late 19th century, systematic work had been done by psychologists, but it tended to focus upon selecting or training people to interact with machines. But the tremendous industrial and military expansion brought on by World War II, and the greatly increased complexity of the weapons systems being produced, complicated human-machine interaction considerably, so that the selection and training approach no longer was sufficient. For example, it was a simple matter to get a relatively small number of men to fly the slow uncomplicated fighter aircraft of World War I as compared to getting thousands of men to perform satisfactorily in the high performance P-38's and P-51's of World War II. Thus, a recognized professional specialty, usually known as human factors psychology or human engineering, was spawned.

Since human factors psychology came into being during World War II, it is to be expected that it would continue to thrive in a military environment after the end of that war. The Army, Navy, and Air Force all established substantial centers for the study and application of this discipline, and many of them still exist today. Human factors practitioners spread into the industries that supplied military equipment and systems. There was a particularly large concentration in the aerospace industry. The movement also took hold in many nonmilitary fields such as transportation, telephony, and occupational safety.

In 1953, Paul Fitts and others typified the work being done in the 1940's and 1950's as follows: "In the design of equipment, human engineering places major emphasis upon efficiency as measured by speed and accuracy of human performance in the use of the equipment. Allied with efficiency are the safety and comfort of the operator. The successful design of equipment for human use requires consideration of the man's basic characteristics, among them his sensory capacities, his muscular strength and coordination, his body dimensions, his perception and judgment, his native skills, his capacity for learning new skills, his optimum work load, and his basic requirements for comfort, safety, and freedom from environment stress" (*1*).

Thus traditional human factors engineering concerns itself with data gathering and experimentation meant to yield precise information about human capabilities. With such information, machines can be built to fit humans. For example, studies revealed design requirements for visual displays of understandable information so that correct decisions or control actions can be made without delay. Comfortable physical fit between man and machine can be established with the use of such information as the average human's physical size, strength, and reach. This information has been stored in handbooks for the use of equipment designers and for human factors personnel who work with equipment designers.

Personnel Subsystem Development

By 1960 there was a growing recognition that profound changes needed to be made in the organization and execution of complex system designs. In brief, an opinion was growing that there should be a simultaneous, integrated design and development of the hardware and software, and of the human elements of the system. In 1960 Van Cott wrote the following: "Traditionally, the goal of human engineering has been to engineer machines for human use—to make equipment compatible with man's characteristics and limitations. In this book, however, it is assumed that the primary goal of human factors engineering is to help design an optimal system. The human factors program helps design not only the equipment, but also the jobs, and to some extent the characteristics of the personnel in the system through appropriate selection and training programs. This approach changes the typical role of the human engineer from that of a design critic to that of a component specialist and a member of the system design team" (*2*).

By 1960, too, not only was there a healthy appreciation of complex weapons systems and the important role of human factors engineering in their designs, but the importance of the computer was being realized and there was a rising concern about the complexity of the interaction between humans and computers. In that year, an article by Licklider appeared in the first issue of the *IRE Transactions* on "Human Factors in Electronics," in which he raised three issues of "man-computer symbiosis": the language mismatch between computers and people, the physical interface (computer console and terminals), and the speed-cost mismatch between human and computer (*3*).

In 1977 we no longer hear much said about Licklider's three problem areas, at least not expressed in the same terms. We have been able to compensate for the speed-cost mismatch between man and computer, by such approaches as timesharing. In information systems, our concern often is the opposite: the response time of a computer to an operator inquiry sometimes is not fast enough for optimum system performance.

With regard to the physical interface, the variety of available well-designed terminals is much better than it was a few years ago. This improvement may be due at least partly to the negative reactions to many of the early terminals which were so poorly designed.

The language mismatch between people and computers—at least one aspect of it—still is a major problem. The matter of preparing and debugging programs, especially for large software projects, still is, as one authority puts it, "in the category of a cottage craft" (*4*).

But today there is another major problem of "human-computer symbiosis." It concerns the fit between the computer and its software (computer subsystem) and the people (personnel subsystem) in the same system. The term "personnel subsystem" refers to the manual procedures, the human-computer interface, training, documentation, and performance aids.

Today's problem and its solution are familiar to every human factors psychologist who has worked in a large human-machine development. The software is designed, written, and debugged with meticulous care, while the personnel subsystem is left to the last minute and is given little or no consideration. In the worst cases the software simply is turned over to the clients who attempt to fit in the people functions some way—often with disastrous results. (Have you ever attempted to get an error corrected in a computer-based charge account?) In less traumatic cases, the software designers, with or without the help of human factors people, attempt to adapt the human functions to the computer functions, which may allow the system to run, but often in a suboptimum manner. The correct way to proceed, according to Van Cott and others, is with a simultaneous, integrated design of both subsystems.

We will now consider the issues raised by Van Cott and Licklider, and give illustrations from our own experiences of how these issues have been addressed in the development of computer-based systems, particularly in the past 10 years. To repeat the issues: Van Cott said ". . . it is assumed that the primary goal of human factors engineering is to help design an optimal system. . . . This approach changes the . . . role of the human engineer from that of a design critic to that of a component specialist and a member of the system design team" (*2*). Licklider raised the issue of human-computer symbiosis, which we interpret to include the difficult task of coordinating human information processing and computer data processing into an integrated system.

System developers have addressed the issue of human-computer symbiosis in varying ways. One approach has been to pay only casual attention to human factors and personnel subsystem considerations. A second approach has been to devote extensive resources to the development of training courses for the personnel. A third approach has been to concentrate on the human engineering of the human-computer interface. The fourth approach has been to design the complete personnel subsystem concurrently with the computer subsystem. We will discuss each of these approaches separately.

Develop Hardware and Software, but Very Little Personnel Subsystem

Many new systems involve the application of a new technology to existing business or scientific functions. Making full use of a new technology—for example, computer technology—is a challenging undertaking. Development organizations sometimes become preoccupied with applying the new technology; all their attention and resources are devoted to it, and little or no attention is given to the personnel subsystem.

One consequence of such an approach is that human errors often become intolerable. This can be illustrated by an early instance of computerization. The plan was to mechanize part of a customer records and billing process. Hardware was selected and software developed. Soon after installation, the system became swamped with errors. Eventually, and with great difficulty, improvisations were made in the personnel subsystem which brought the error rate within tolerable limits.

The worst problems are traditionally with input errors. Computer edits identify errors and reject the error-containing transactions. Error messages are then fed back to personnel in the error-correction feedback loop. There is a snowballing effect: "temporary" personnel are added to correct errors; system performance keeps dropping; the rate of system throughput drops because most of the data processing effort is expended in the error-correction process.

Abnormally high error rates have three consequences. First, if outputs are delayed, customer schedules (and customers) are upset. Second, the cost of extra processing required to correct errors can be high; for example, a study of errors on customer service orders showed the total cost to correct each error to be $7.25. Such costs can wipe out the savings which the new system was to effect. Third, high error rates demoralize system users. We once asked a clerk whose sole job all day long was to prepare a complex input form, how many forms were returned for correction of errors. The answer: "All of them, and some more than once."

Another consequence of the single-technology approach is that system users often are unable or unwilling to use system features which have been devised at great expense. Licklider, describing the SAGE (semi-automatic ground environment) system, reported: "According to credible reports, there were makeshift plastic overlays on the cathode-ray displays, and the scope watchers were bypassing the elaborate electronics—operating more or less in the same 'manual mode' used in World War II" (5).

Produce Extensive Training Course Materials

System designers, when confronted with widespread human performance problems in their systems, search for a broader design approach. They sometimes believe that training equates to satisfactory human performance. There are at least two flaws in such an assumption. First, human performance in systems is the complex interaction of many variables, only one of which is training. Fox, citing experience with developmental testing, described eight major variables affecting human performance in systems; only one was training (6). (The other variables are: design, human-machine interface, information transfer, environment, personal factors, supervision, and documentation.)

Second, simplistic as it may sound, system personnel should be trained to do the manual procedures of the system. However, the most carefully developed training courses cannot compensate for a poorly designed or suboptimized personnel subsystem. The procedures themselves may be incorrect; for example, some information should be stored in performance aids instead of memorized. No matter how much training they receive, system personnel probably will be unable to overcome many of the error-producing procedural and interface problems inadvertently incorporated in the personnel subsystem design.

During the 1970's, instructional technology has become widely used. This method of training course development features thorough analysis of tasks to be taught. We have found that a task analysis, done for training purposes, often reveals personnel subsystem design deficiencies. It also often reveals the need for new performance aids and procedural documents. The net effect usually is reduced training time.

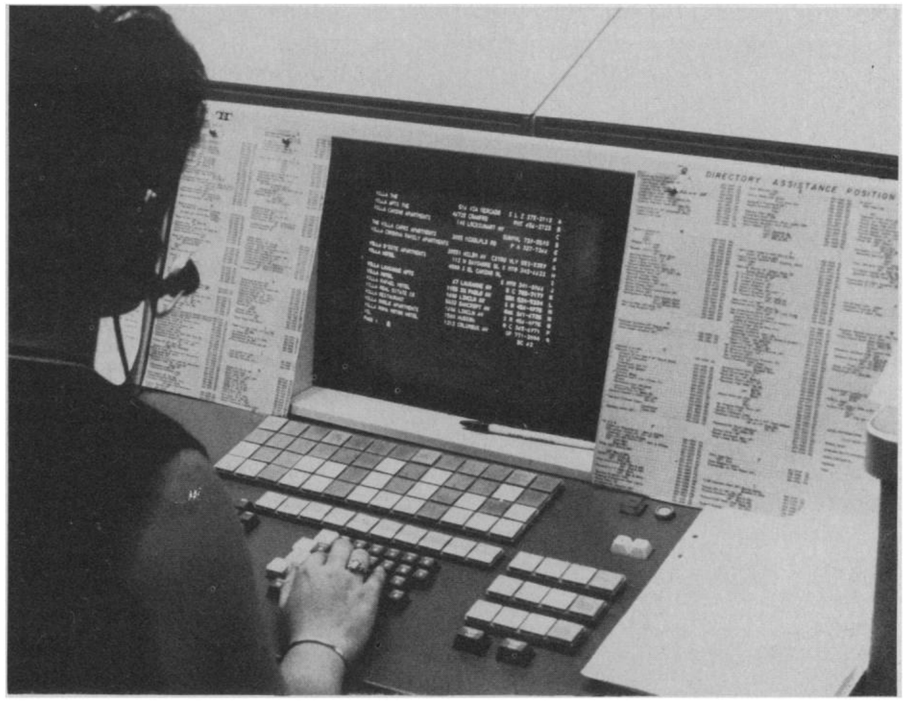

Fig. 1. Directory assistance operator working at specially designed terminal of experimental computer-based system.

For example, we were asked to help with training course development for a new system which provides standardized circuit designs. A task analysis revealed the need to add performance aids and restructure user documentation. The result was a 30 percent reduction in training costs, improved human performance, and a reduction in the volume of system documentation that was required.

Design the Human-Machine Interface

Systems designers are often familiar with some of the terminology and concepts of human factors engineering; therefore, the "human factoring" of interfaces has traditionally received support in both large and smaller systems.

One such project was MECHSIM (mechanical simulation of a computer-based directory assistance system). This project had as its ultimate goal the reduction of the amount of time an operator spends responding to a request for directory assistance. The MECHSIM study followed some 10 years of low budget research and experimentation directed to reducing operator work time.

Control or reduction of operator work time is important to telephone companies. Directory assistance is expensive to provide and, until recently has been provided universally at no charge. In 1970, for instance, the Bell companies had more than 44,000 people working in directory assistance offices at an annual cost of \$300 million. The average duration of a directory assistance call is about 33 seconds; therefore, the annual value, so to speak, of each second is in excess of \$9 million. So, if Bell laboratories could show the telephone companies a way to cut a few seconds off average call duration, the savings would be considerable.

Operators on the job quickly get quite proficient at using the traditional "paper data base," that is, telephone directories, thus early attempts to improve the process were unsuccessful. For example, in 1956, a trial was run in which directory pages were put in microfilm "sticks," as they were called, and projected for the operator on a ground glass screen. This method did not save look-up time, largely because the state of the arts of optics and microfilm were not sufficiently advanced.

In the early 1960's, analytical studies were undertaken with the view to utilizing the computer and its related cathode-ray tube (CRT) terminal. A major question was, how would the operator gain access to the file if it were in the computer instead of on paper? When operators used paper telephone directories, access was by full name of the customer; however, the studies showed that if the operator were to key in trigrams—that is, the first three letters of last names instead of the entire name, input time and spelling and key stroke errors would be reduced.

In 1965, a human factors trial was instituted for the purpose of measuring op-

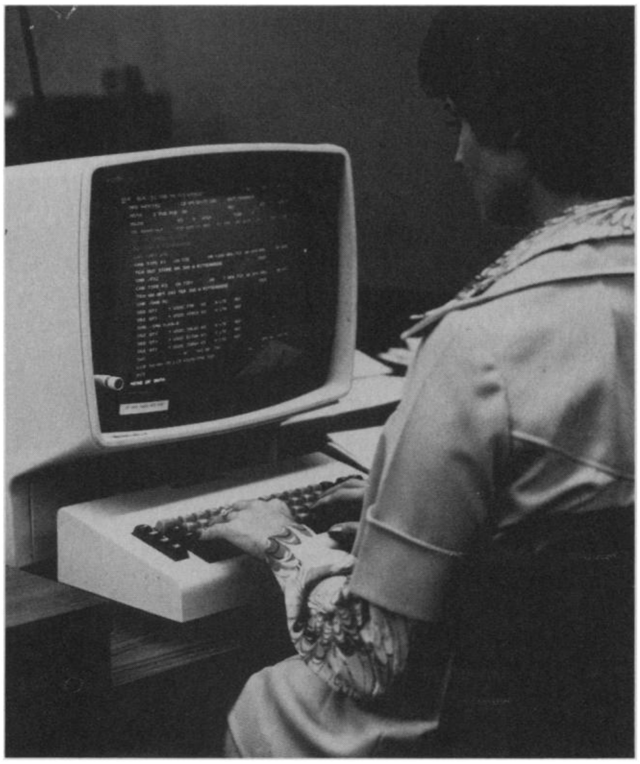

Fig. 2. Customer records being checked for accuracy during conversion from the existing manual system to a new computer-based system.

erator performance with such a system. This is the mechanical simulation (MECHSIM) referred to by Lindgren in a 1966 article, in which he said: "The Bell project also presents an interesting sidelight. Human factors engineers always stress the importance of being in on design studies as early as possible so that their investigations can properly influence the final system design. In the Bell case, the engineering department is not even going to begin to implement the computer look-up system until they have seen the complete results of the human factors study . . . which amounts to the realization of a human factors dream" (7).

The dream, if that is what it is, has continued. The mechanical simulation of computer-assisted directory assistance, when it was completed in 1966, indicated that there probably would be sufficient savings in operator look-up time to justify the development of an actual computer-based system. As a result, a computer-based human factors experiment was begun in late 1970. It was referred to as the "live traffic experiment" because it involved ten operators who worked at specially designed CRT terminals and who received actual "live" customer requests for directory assistance. They used the terminals to locate customer telephone numbers which were stored in a computer memory.

Operators talked directly with customers and, using keys on the CRT console, keyed into the computer the necessary trigrams indicating name and address information (Fig. 1). The computer then activated a display on the CRT from which the operator selected the desired telephone number and relayed it to the customer.

The study was in two parts. The first was an experiment or, more correctly, the last stage of an experiment that had several preceding steps in other parts of the laboratories going back over some 12 years. Operators systematically tried eight search (keying) strategies to retrieve telephone numbers from computer storage. The best strategy was then selected for the second part of the experiment, a pilot study of the computer-based directory assistance operation. Its purpose was to see if sufficient savings in work time could be realized to justify the use of the computer subsystem for directory assistance.

The outcome of the pilot study indicated that the computer-based directory assistance would indeed save sufficient operator work time to justify a change to that type of system when development resources became available.

More important, perhaps, is the fact that the live traffic experiment represents in several ways the coming of age of human factors psychology or "human performance technology." For example, the human factors experiment and pilot study were undertaken *before* huge sums were committed to system development. Next, there was the specific commitment of designer resources to the attainment of an integrated operator-computer interaction. This required an explicit understanding that software designers were dependent upon human performance requirements developed by human factors psychologists.

Finally, the live traffic experiment served to call management attention to the principle that the personnel subsystem of a computer-based system requires careful design attention just as does the computer subsystem.

In information systems, the human-computer design is important for error prevention. Human-caused errors are an anathema to system designers. A large facilities engineering and assignment system that we studied illustrates the effect of human error on system performance; it also illustrates why, when systems have human-error problems, designers tend to focus their attention on interface and data display designs.

Almost immediately after the change to the new system, human errors snowballed. Computer-processed transactions bogged down to the point where the computer was bypassed entirely; later, the errors were located and corrected, and the computer data base and manual protective records updated. The system's error-correction process quickly became overburdened. System performance then fell below acceptable or even tolerable levels.

Most of the human errors were detected by computer edit of manual inputs. System designers proposed a two-part solution: Replace the input clerks with more skilled people; and substitute CRT terminals for input forms. Before this solution was implemented, a special task force worked with the system users. As a result, CRT's were not installed, nor were more skilled people hired. Instead, many of the paper forms were redesigned, performance aids developed, and system personnel trained. Error rates immediately dropped, and system performance was soon above acceptable levels, where it remained.

A key point, however, is this: The most visible changes made were to the forms used by humans to transmit information to the computer; however, task force members believed the other personnel subsystem work (procedures, performance aids, and training courses) to have been much more important.

This final example points out a critical issue. On many systems, the design of the man-computer interface itself is not enough. After all, many large computer-based systems use off-the-shelf terminal equipment which satisfactorily meets human capabilities and stereotypes. As DeGreene says, in regard to design of data displays: "Many of the baffling human factors problems, then, relate not so much to vision as to cognition. The display can be considered a direct extension not only of the inner structure and workings on the computer, but also of what is going on within the user's [operator's] problem-solving head" (*8*).

For most system designers, human-machine interface design should focus on integrated, complementary human-machine procedures. Human procedures and machine procedures come together at the interface, whether console, CRT display, or paper form. The interface design should take into account these interactions, and should be consistent with both human and hardware capabilities.

Design the Subsystems Concurrently

Human performance technology integrates elements of human engineering, engineering psychology, instructional technology, and industrial and organizational psychology. The application of human performance technology is called personnel subsystem development (PSD), which means the integrated design of human procedures, human-machine interfaces, training, performance aids, and documentation as part of the total system. It also includes the rigorous testing of the personnel subsystem together with the computer and hardware subsystem counterparts. The products of PSD are the controlled utilization of human resources, and the means (procedures, training, performance aids, interfaces, for example) for obtaining required human performance in systems.

A recent Bell Laboratories example of designing the personnel subsystem simultaneously with the computer subsystem was the enhancement of a computer-based message switching system. During design there was continual interaction between personnel subsystem and software designers. The personnel subsystem and software were fully tested before installation. During a 3-month period following installation of the enhanced systems, only 6 percent of the problems encountered required design changes, and none of these involved human procedures or a human-machine interface. Previous experience suggested that, if integrated, concurrent design of the computer and personnel subsystem had not been done, 50 percent of the encountered problems would have required changes to the personnel subsystem (*9*).

Another example illustrates that even as a retrofit, the thorough design of the personnel subsystem of a computer-based information system can accrue system benefits. When the PSD design team began work, the system design included hardware and software. More than 50 documents already had been developed for system personnel by persons who did not follow the PSD process.

The PSD designers first analyzed system functions, then analyzed tasks to be done by system personnel. From the data collected, work procedures were designed. Common human procedures were identified. Performance aids were developed. As a result, the existing 50 procedural documents were reduced to 17. This in turn simplified document maintenance and reduced training time.

A key design activity was to allocate the human information load among system personnel and system products, without duplication. For example, attendants (operators) were taught to recall from memory certain procedures and data. Other data were contained in performance aids. Still other data were incorporated in masks displayed on the CRT. Attendants were trained to use the performance aids and CRT masks, not to remember all the information contained therein; there was no need to duplicate such information in training courses, therefore the volume of system documentation was reduced markedly.

This system also illustrates how manual procedures and interface design are interrelated. A service attendant at a CRT terminal talks directly with a customer who is reporting telephone malfunction. A mask is displayed on the CRT. As the customer gives information, the service attendant uses the CRT keyboard to fill in the mask. The original mask design had been controlled by software requirements, and assumed the typical customer would provide information in a certain order. Actually, the information from the customer came in a different order; therefore, attendants had to go through several extra operations to record the data, thus cutting productivity and increasing the chance of error (Fig. 2). Redesigning the mask to match the operator-customer dialogue reduced the number of operations required of the operator and reduced the chance of error.

One final example deals with the conversion of a system which maintains an inventory of available facilities, and assigns these facilities to customer orders for telephone service. The conversion requires the transcription of the records of 120 million telephones from paper to computer data base, at an estimated cost of $305 million.

The computer and personnel subsystems of the conversion system were designed and implemented concurrently and systematically. Training and documentation were developed, and the personnel subsystem was tested as part of the design process. When the testing program was completed, the transcription error rate had been reduced from a projected 10 percent to an overall 5 percent. Thus, when the necessary records have been converted to the computer data base, there will have been an estimated cost avoidance of $46 million.

Conclusion

Clearly, the years since World War II have seen increasing complexity of human-machine interaction. Machines have become much more complicated and the responses required of humans may be said to have moved from "brawn to brains," that is, from the physical interaction between person and machine to an interaction which depends more on the information processing abilities of human beings.

In our experience with computer-based systems, we observe an increasing emphasis upon personnel subsystem development. This is largely because today's complicated systems, particularly those involving computers, put increasing demands upon people in the system for fast and nearly error-free performance. Careful design attention must be given to the human element if such performance is to be attained.

References and Notes

1. "Human Engineering in the National Defense," Panel on Human Engineering and Psychophysiology, Research and Development Board, HPS 205/1, (29 June 1953), cited by N. Lindgren, *IEEE Spectrum* 3, (No. 3), 133 (1966).
2. H. P. Van Cott, *Human Factors Methods for System Design* (American Institutes for Research, Washington, D.C., 1960), foreword.
3. J. C. R. Licklider, *IRE Trans. Hum. Factors Electron*. **HFE-1**, PA-11 (March 1960).
4. V. A. Vyssotsky, private communication.
5. J. C. R. Licklider, *Comput. Autom*. **18**, 48 (August 1969).
6. F. W. Fox, paper presented at the Seventh International Symposium on Human Factors in Telecommunications, Montreal, Canada (1974). The eight variables are: training, procedural documentation, design of human procedures, interface design, information transfer, environment, supervision, and personal factors.
7. N. Lindgren, *IEEE Spectrum* 3 (No. 4), 70 (April 1966).
8. K. B. DeGreene, *Systems Psychology* (McGraw-Hill, New York, 1970); p. 319.
9. M. W. Soth, "The Human Work Module: A Structural Entity for Personnel Subsystem Design" (Bell Telephone Laboratories, Murray Hill, N.J., 1976).