THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 5, May–June 1982
Printed in U.S.A.

Human Performance Engineering Considerations for Very Large Computer-Based Systems: The End User

By I. S. YAVELBERG

(Manuscript received May 15, 1981)

Effective Human Performance Engineering for a large-scale, computer-based system involves many complex strategic and tactical decisions regarding the computer system design, the target user's behavior, and the organization/environment. Descriptions of the more important performance considerations are presented. These are based primarily on the experience accrued during the last several years in the building of Bell Laboratories centrally developed computer systems for use by telephone company loop operations personnel in assisting them to do their job. The target population can be broken down into these distinct classes: End Users, Database Maintainers, and the Data System Support Staff. This paper focuses on the End User, and specifically the Bell System Service Representative. Important points include: (i) early emphasis of human performance considerations in the computer system design process can reap valuable benefits; (ii) care must be taken to specify input/output design features which have gone through the human/system engineering step of identifying a favorable payoff versus penalty ratio; and (iii) based on measurement data and user interrogation, computer system availability and transaction failures and response times can seriously damage user performance and system acceptance.

I. INTRODUCTION

Effective Human Performance Engineering (HPE) for a large-scale computer-based system involves many complex strategic and tactical decisions with regard to the computer system design, the target user's behavior, and the organization/environment. Descriptions of the more important performance considerations are presented. These are based primarily on the experience accrued during the last several years in the building of Bell Laboratories centrally developed computer systems for use by telephone company loop operations personnel in assisting them to perform their job. Wherever possible, observations are compared to recent literature associated with other systems or experiments.

The major in-place elements that affect human performance are the organization/environment and the behavioral considerations of the workers themselves. The introduction of a computerized support system adds another element. Presumably, the computer improves the effectiveness of the worker in performing the job. It also provides some design flexibility to counter some of the negative aspects of the other three existing elements. However, the computer system itself introduces serious behavioral effects on its users that may negate its benefits.

The target population can be broken down into these distinct classes: end users, database maintainers, and data system support staff. This paper focuses on the end user with special emphasis on the Bell System Service Representative. Principal points made are as follows:

- An understanding of the target population's attitude and behavior toward computers is necessary.
- Early emphasis of human performance considerations in the computer system design process (workflows) can reap valuable benefits. Unworkable or unreasonable performance requirements on the user can be substantially avoided.
- Care must be taken to specify input/output (I/O) design features which have gone through the human/system engineering step of identifying a favorable payoff versus penalty ratio.
- The introduction of a computer to the existing organization/environment should stimulate some adjustments (job assignments) to optimize the mechanized features. Every attempt should be made to do this as early as possible. However, if workers in one department are asked to increase (or even change) their workload to make it easy to program a computer helping another department, there is extensive resistance.
- Operational—more so than functional—characteristics of the computer system (availability, transaction response time, transaction failures) can have a serious, and often underestimated, negative effect on user performance. For example, many sporadic short outages can have a significantly greater effect on user performance than a single, extended outage.

II. MAJOR CATEGORIES OF HUMAN PERFORMANCE CONSIDERATIONS

The model that represents the relevant physical and behavioral elements and interrelationships has, as the central element, the collection of people whose performance is what this paper is all about: the target population or the end user of the computer system. Surrounding the user are three major categories, each of which introduces human performance considerations. The computer system is related to the user by a two-way interface which represents physical, as well as behavioral, interactions. The two in-place categories represent organizational/environmental considerations and the behavioral considerations of the user. These two categories have unidirectional interfaces representing behavioral influences only.

In contrast to other models in the field, I chose to group organization and environment together and add the design and constraints of the mechanized elements (main frame, communications, terminals) under the umbrella category, computer system. I do not advocate waiting for the performance deficiencies to occur before addressing solutions. We can predict potential performance problems and avoid them. This important philosophy is emphasized throughout this paper. Most of the material concerns the human/computer interface: in a positive sense, where interface features relieve user performance problems associated with the precomputer job and, in a negative sense, where operational characteristics of the interface itself introduce new performance difficulties.

III. TARGET POPULATION

Without question, knowing the target population is fundamental in solving any human performance problems. In particular, it cannot be overstressed how necessary an understanding of the target population's behavior towards computers is with respect to effective performance.

3.1 Population classes

For many large computer-based systems, the target population really changes with delivery. In the predelivery (usually manual) environment there are generally two classes of users: customer service end users and back-room record maintainers. With the introduction of the computer system, three classes, generally distinguished by their level of expertise with computers,[1] emerge: the unskilled, the semiskilled, and the skilled. In a Bell operating company (BOC) environment, the jobs associated with these levels are often end users and their supervision at the unskilled level; database maintenance clerks and their supervision at the semiskilled level; and data systems support staff at the skilled level. It is important to remember that skill here applies specifically to computer use and not to the whole job. I prefer to identify these classes as end users, maintenance clerks, and operating staff.

Within this target population framework, human/computer interface design should be multifaceted, and groomed specifically for each class.

3.2 End user: the Service Representative

This paper deals with the end user. Most of the field experience, obtained between 1978 and the present, is associated with the Bell System Service Representative (Service Rep). The computer system for which the considerations are based is Premises Information System (PREMIS), which is a Bell Laboratories system designed to provide BOCs with customer-related information to help determine service order information.

PREMIS supports the Residence Service Center Rep on-line during new customer negotiation and also provides some important data that must go on the service order. When a customer calls the BOC to place an order for new residential service, there are a number of tasks that must be performed by the BOC Service Rep taking the order:

(*i*) Get the new address correctly.
(*ii*) Sell telephone service at the new address.
(*iii*) Establish the credit class for the customer to determine if a deposit is required.
(*iv*) Quote charges and rates.
(*v*) Give the date service will begin.
(*vi*) Assign a telephone number.

The Service Rep records all of this information on a service order, which will be used by other departments in the BOC to provide service. PREMIS aids in all of these tasks and replaces the paper records, microfiche, telephone calls, or guesswork used previously. The physical architecture consists of a very large UNIVAC computer, a communications network built around BANCS, another Bell Laboratories product, and 40/4 terminals located at each Service Rep's position. The computer is administered by a data systems group, and the database is maintained by several dispersed maintenance groups. Primarily, PREMIS is an address-keyable system providing the needed address-related data. The Service Rep simply keys the new address into a preformatted mask on the terminal. Usually all that is needed is the house number and street name.

PREMIS responds with information about the geographic area that is needed on the service order. This includes wire center, exchange, rate zone, tax area, directory group, and the service features available for the address. This same display will also include the existing customer's name, telephone number, and presence of an in-place connected circuit or loop from the address back to the central office.

As of December 1980, PREMIS served almost two million residences, accounting for almost half of South Central Bell Telephone Company's residence service orders. About 1000 Service Reps are now accessing the system.

3.3 Behavioral considerations

Human behavioral factors that are particularly relevant to the introduction of computer support have been widely recognized: short-term memory limits, the need for closure, the desire for control, motivational characteristics, and the fear of computers themselves. Computer system designers—especially those responsible for the human/machine interface—can effectively incorporate design features which protect against the negative behavioral effects of these factors. This paper will cover many of these features. Perhaps, the anxiety or fear of computers is the most underrated and is discussed next.

Extensive research, experimentation, and observations have been noted by Shneiderman showing that user attitudes—fear of computers—can have a major effect on learning and performance.[2] His data survey suggests that "Novices with negative attitudes towards computers learned editing tasks more slowly and made more errors. Anxiety, generated by fear of failure, may reduce short-term memory capacity and inhibit performance." In a later paper,[3] Shneiderman notes that part of the training requirements must be to overcome a distorted role of a computer as perceived by the potential user. The media and computer manufacturers have unfortunately characterized computer capabilities by life-like behavior which results in potential trainee resistance in the form of increased apprehension, resistance to technology, and anxiety interfering with learning potential. Shneiderman cautions against predictions such as "people and machines are so similar that with a few years effort, they should be able to produce machines that are superior to people." He states that "this naive view is useless as a goal and harmful in destroying people's expectations of themselves and how they will use computers." Experience with new users of PREMIS confirms this apprehensive attitude generally among the most experienced, entrenched, settled-in population. However, the younger, newer employees often showed enthusiasm and genuine wonder at how the computer can help them do their job. In fact, they tended to press for an education into "how the computer works inside," rather than limit their involvement to the minimal training on transaction I/O.

The negative attitude towards mechanization has sometimes extended in serious directions. While I have never encountered sabotage, evidence of disgruntled employees damaging a delivered computer system exists,[4] apparently out of frustration at not being properly trained. One case resulted in a prison term for consciously sabotaging a system dozens of times during a period of 18 months.

Jones emphasizes the need for building up the confidence of a potential user by incorporating into the design of the computer system features which give the user the feeling that: "his or her commands will be obeyed; the data are in safe hands; a good and thorough job is being done; and the machine is going to help the user.[5]

The desire for control can be best satisfied if the user is the initiator of all human/computer interface sessions. This principle is easy to support if the end user is getting "information retrieval only" support from the computer, as in the case of the Service Rep. For data system support positions where the computer initiates the need for work, this is, of course, not possible.

The psychological needs associated with short-term memory limits and the need for closure (the completion of a task leading to relief), are interrelated. Excesses can lead to delays, forgotten items, and increased error rate. The key computer-related factors affecting these needs are the careful functional design of human/computer interface sessions into discrete subtasks and the transaction response time delays and variations for a variety of operational conditions. Both of these factors will be examined in this paper. The important point, made by Miller, is that "a psychological closure permits at least a partial purging of short-term memory; waiting time invokes stress." It is believed that more extended delays (as in computer response time) "can be tolerated just after closure rather than in the process of obtaining closure."[6] The idea is to have a system design which does not lead to loss of user concentration, that is, maintain continuity of human thought processes.

Finally, certain characteristics associated with the design and delivery of a computer system can seriously damage the motivation of the user. Three such characteristics which have proven important in PREMIS were generally identified in an earlier paper by Hackman:[7] experienced responsibility for work outcomes; experienced meaningfulness of the work; and knowledge of results. These motivational needs, as well as the other psychological considerations discussed above, will be related to PREMIS field experience discussed later in this paper.

IV. COMPUTER SYSTEM CONSIDERATIONS

This category is a catch-all for mechanized system elements, hardware, and software. It also includes the physical attributes associated with the human/machine interface design. Discussed below are examples of design, development, and field experience in this category.

Major elements of the physical architecture of PREMIS are as follows:

- Computer—UNIVAC 1100 series.

- Communications Network—BTL/South Central Bell BANCS THP System resident on Control Data Corporation (CDC) Cyber 1000; Connectivity via 50-kb high-speed lines.
- Terminals—Dataspeed Mod 40/4 series; Input device—QWERTY keyboard with function keys; English language; output device—CRT display.

At a lower level, many operational and functional system considerations remain which can fundamentally affect user performance. Before developing some of these specific issues, I offer some recommendations on the staffing and timing of the system design activity.

4.1 Early HPE influence in system design

I believe human/machine interface issues can be best understood from a user perspective by professionals in the broad HPE field. Early analysis by HPE professionals can help provide a sound basis for system design decisions. By working with hardware and software professionals, they can substantially avoid unworkable or unreasonable performance requirements on the user. Bennett agrees, noting that human/machine interface design "will be most expeditiously advanced if those already trained in human engineering join the design team rather than if software people attempt to learn engineering."[8] This position is not universally accepted, but certainly trending in this direction. Bennett goes on to paraphrase McCarn: "... experts in computers see well-bounded problems with great precision" but have difficulty in accepting the idea "that human communication is much more complex, and its context more extensive, than was initially conceived by computer programming staffs." I believe this may have been generally true several years ago, but lately, with extensive teamwork among HPES and computer science types under our belt, it is not unusual for a genuine interest and insight of the human side to evolve in software professionals.

Regarding the detailed human/machine interface design, one key to ultimate high user performance is simplicity in learning. Shneiderman stresses this by noting that simplicity of design can best meet the bottom-line economic objectives of the system (e.g., computer processing efficiency, storage capacity, communication network load) and still generate the desired user performance in terms of reliability, reduced error frequency, and enhanced satisfaction. Closure, which is strongly influenced by short-term memory considerations, must also be considered. The user receives "great relief when information is no longer needed to be retained. There is . . . a powerful desire to complete a task, reduce memory load, gain relief." In terms of operational design, transaction response time requirements obviously are tightly coupled with this behavioral consideration. From a functional design view, a direct design guideline is to formulate transactions which allow the same user to complete each task in sequence. These issues are all part of a fundamental principle: A key to influencing ultimate positive acceptability of a large computer system is the initial approach taken in system definition. I am a believer in formally going through these steps:

1. Functional decomposition, analysis, and allocation—This, in effect, divides up the tasks between humans and machines, making the best use of the strengths of each.
2. Work flows—This shows how the new job will get done in a logical sequence.
3. Human/machine interface design—Finally, from the work flows, the interface points can be supported by detailed I/O design.

Martin refers to these three steps as "functional," "procedural," and "syntactical." If Step 1 is not done, the ball game may be lost before it starts. Jordan recognized this many years ago: "Men are flexible but cannot be depended upon to perform in a constant manner, whereas machines can be depended upon to perform consistently but have no flexibility whatsoever." Martin, more recently, says it best: "The difference in "thinking" talent—the computer being good for ultrafast sequential logic and the human being capable of slow but highly associative thinking—is the basis for cooperation between man (human) and machine. It is because the capabilities of man and machine are so different that the computer has such potential ... It is important that system designers ... do not try to make the computer compete with man in areas in which man is superior."[9]

A very important system design principle which is being used to great advantage in PREMIS is the concept of modular structure. Each separate system function is architecturally built independent of the others. This applies to internals (processing logic and database structure) and externals (transaction groupings). In addition to the obvious benefits of understandability and simplicity, this modular structure allows an upward compatibility in later releases. It also allows the trialing or "soaking" of a new feature in a limited area before general deployment. In a similar way, by working with the target organization staff, the existing target organization framework (e.g., methods, work flows, job, measurements, etc.) can be examined. User performance deficiencies can then be avoided by influencing the change of the organizational structure when mechanized support is introduced. This is covered later in this paper.

4.2 Input/output design considerations

4.2.1 Interactive transaction characteristics

PREMIS has incorporated a variety of Service Rep I/O design features which help smooth the interaction from a human point of view. Many features were part of the original design; others were designed and delivered via field reviews and feedback from the trial users—South Central Bell. A few are in the process of being delivered. Listed below are those believed to influence user performance and satisfaction. Some specific examples can be found in other papers by Hicks[10] and Ferrer.[11]

(*i*) Input Mode—Inputs can be classified as either "coded" or "prompted."[12] With coded input, the user supplies the labels and the data values; with prompted input, only the data values need be entered because the labels are supplied by the computer via form-filling (a mask). The latter is recommended for the end user because of its inherent advantages in reducing input errors and work time, while increasing satisfaction. Prompted input can be further divided into "interactive" or "batch." With interactive prompted, the computer waits for the user to enter a data value after sending each prompt; with batch prompted, the user fills out the entire mask and the computer gets only one transmission. For a computer system without local intelligence at the point of entry, the batch-prompted mode is much more efficient from a communication system overhead and computer usage capacity point of view and was selected for PREMIS.

(*ii*) Command Structure—Short, simple, consistent structure emphasizing clarity.

(*iii*) Single Display Frame (Input)—Transaction inputs always limited to single screen (mask); no input paging required. The particularly awkward paging design associated with the MOD 40/4-system software combination made this very important.

(*iv*) Single Display Frames (Output)—Transaction outputs usually limited to single screen (mask); however, in some cases, in "prompting" mode, multiple output pages can be returned to the user. Because paging commands require full transmission back to the computer, the transaction response and human search time delays are proving impractical for more than a very few pages. A revised design includes providing more limited output via either more selective data returned or asking the user for more input data to narrow the scope of the stored database information. The concept of putting the most likely choices on the first output page has obvious benefit, but as of yet, such an algorithm has not yet been implemented for PREMIS. This issue is covered in more detail later.

(*v*) Discrete Transaction Per Function Design—Several independent subfunctions requiring computer assist each has their own independent transactions. This is in support of the user's need for closure. In addition, this allows optimization of transaction queue control to maximize the priority of the end-user transactions.

(*vi*) Transaction Sequencing—While transactions are discrete, each output screen is designed so that over-typing a portion of the input control field, leaving the desired data, adding new data and hitting the SEND key can request the next transaction. This achieves a high level of "Reusability"[13] of former inputs and outputs.

(*vii*) Cursor Movement—Nondestructive forward and backward position-by-position movement and programmable tabbing for automatic movement to the beginning of next and previous input fields.

(*viii*) Prompting—Software logic which detects some shortage of input information and requests the user to provide additional input.

(*ix*) Menu Selection—A type of prompting where a limited set of valid responses is presented on the screen and the desired response can be chosen by keying in the choice, the number of the choice, or by positioning the cursor next to the choice.

(*x*) Parameter Defaults—For each community of interest (e.g., city or state), a set of table-driven parameter values which represent an agreement by the user on what are normal, or the most often used, values that can be assumed if the field is left blank.

(*ci*) Minimal Input—The user need enter only a "shorthand" input, and the computer employs some pattern recognition logic to fully interpret the input. (This feature has sensitive system performance implications and is discussed later.)

(*xii*) User Control—The user initiates and controls all human/computer interactions.

(*xiii*) User Messages for Transaction Failures—Currently, the screen goes blank for certain transaction system failures and uninformative messages are returned for others. This is a serious cause of user discomfort. Meaningful messages should be returned in every case, and efforts are underway to make this possible.

4.2.2 Multisystem considerations

(*i*) System-to-System Switching—Currently, the Service Rep has access to another mechanized support system in addition to PREMIS via the same terminal. During the same work session, one switch (or more) between systems is often necessary. The current log-off-log-on procedure involves several keystrokes and response waits. Serious user dissatisfaction with the procedure has led to the recommendation of a quick-switch procedure of perhaps just one function key. In addition, the entry mask for the appropriate system being logged on to should appear automatically on the screen.

(*ii*) System-to-System Consistency—There exists system-specific function keys and dialogue mneumonics for each of the systems accessed by the same user. Lack of system-to-system consistency is a training problem and a potential source of error. Multisystem agreement on design is currently being negotiated.

4.2.3 Screen display physical characteristics

Most terminals in the field today generally provide acceptable physical characteristics from a human performance point of view. In the case of the Dataspeed MOD 40/4 used for PREMIS, the users reacted favorably when asked about screen brightness and contrast and character size, sharpness, and spacing. The only negative reaction to the screen was the eye strain associated with users whose terminal screen faced a nearby window where the sun glare was strong. This was easily remedied by drapes or blinds.

4.2.4 Information display characteristics

In terms of information display, the PREMIS design was open to human performance considerations. A variety of human-engineered screen information display principles were followed in the PREMIS design.[10] These design features were incorporated to improve the cognitive behavioral response of the user, mostly by minimizing data searching, increasing awareness, and by increasing recognition and distinction. PREMIS users gave very high values on the ease of use of this information display (8 to 10 on a scale of 1 to 10, 10 being the highest). However, no experimentation has yet been done on PREMIS to determine sensitivities to varying display techniques.

(*i*) Positioning of Input Data on Screen—As close as possible, input data should be entered in a left-to-right and up-to-down sequence consistent with the work operations.

(*ii*) Input Data Preservation—All input data appears on the output screen completely and precisely in the same position.

(*iii*) Positioning of Output Data on Screen—All data displayed on an output screen (mask) always appears in the same spatial position in multiple operations to minimize user search time, "reduce disruptive movement and help highlight the impact of the last operation."[14]

(*iv*) Highlighting—For key output parameters, where particularly high detectability is desired, highlighting is used.

(*v*) Data Labeling—All variables on I/O screens appear as pairwise identifiers and values.

(*vi*) Number Displays—Long number sequences in the same field are broken down into subsets.

(*vii*) Screen Partitioning—Separate, dedicated areas for input, normal output, and user messages.

(*viii*) User Message Semantics—All user error control and/or instructional messages are constructive and supportive, not condemning and confrontive.[3]

(*ix*) User Message Syntax—All user error control and/or instructional messages are in English (no cryptics/codes).

4.2.5 System engineering considerations

While ensuring understandable, simple human/computer interface features, the designer must also consider the effect of these features on the hardware/software capacity of the system. Gilb notes, "Extensive use of humanized input designs can easily result in substantially greater consumption of central processing cycles and secondary storage search time over programs which effectively place the equivalent work processes on human beings."[4] Care must be taken to specify I/O design features which have gone through the human/system engineering step of identifying a favorable payoff versus penalty ratio. This important design activity is not always done in computer systems and, for some features, in the PREMIS project also, subsequent field analysis has sometimes suggested the wrong emphasis.

For example, the misuse of prompting and menu-select features associated with interactive transactions can be a potential source of unexpected (and undesired) induced load. The additional transactions generated from these features can significantly degrade the system load capability. That is, the payoff from reduced training costs, reduced keystrokes, and reduced entry errors may not offset the economic penalty in increased transaction load. A recent example of this is the minimal input feature in PREMIS.[15,16]

This feature allows the Service Rep to key into the computer as few as the first four characters of a street name. The computer searches the database for a match and, if found, returns the desired information. If more than one internal match is found, a menu is returned to the terminal screen and the user, after perusing the menu, selects his or her choice and resends the transaction. Training was provided on how to use this feature but not when to use it. Analysis of audit tapes showed that, for some locations, 50 percent of the time the menu-select response was happening. Not only was this causing an unacceptable additional load on the system, but, the user was paying a penalty in increased time per event. This is evident from Fig. 1. Figure 1a shows that if a second transaction per address is required more than 30 percent of the time, it takes more time on the average to use minimal input than to use full spelling. Figure 1b shows the rate at which transaction volume increases for additional transactions per address beyond the 15 percent error rate assumed for full spelling.

The potential user and system costs of what was intended as a humanized input design was uncovered in an operational review of the system. However, this same review found considerable variability between locations on the percent of time that minimal input required additional transactions. These findings suggested that rather than remove the feature altogether, the objective should be its efficient utilization. Work was undertaken to develop the software analysis and

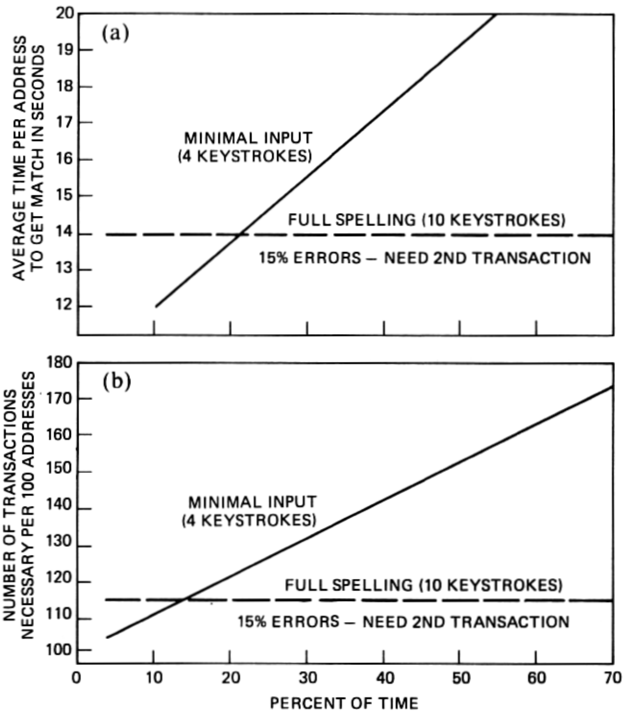

Fig. 1—Impact of minimal input. (a) Service Rep time. (b) Transaction volume. (More than one transaction is required to get a match.)

monitoring tools which could be applied per location, so that usage guidelines per location could be developed. Until these software tools are available, however, the potential cost was deemed too great and, as a short-term remedy, the recommendation was to stop using minimal input.

Another issue which was not properly engineered in the initial delivery of PREMIS was establishing and enforcing limits on the amount of computer processing or output pages which any individual Service Rep transaction could consume. Assumptions were made about real world data which proved not to be true. As a result, information retrieval transactions were allowed which searched vast areas of the database for a "hit." After minutes of chewing up processing time, having failed to find a certain hit, a very large number of possible menu-select choices are output back to the Rep's terminal requiring many pages to go through. In an environment where the Rep is negotiating with a customer on the telephone, these occurrences resulted in Rep frustration leading to a bypass of PREMIS altogether. A careful review of these cases resulted in changes in the software logic and controls to include firm processing and output upper bounds for certain transactions, while providing user messages and enough information to proceed through the customer contact. For example, it is often possible for the user to change the input in such a way that the search is narrowed and the output limited. (To illustrate, if a Rep is dealing with a customer moving into an apartment complex with many units and the units are identified by a combination of letter and number, say Apartment A-162, the customer would often state their address as 162-A. Rather than output all apartment numbers as possible matches, a quick perusal of a single output page can quickly identify this inconsistency which can be easily handled by overtyping the reversed characters and resending the transaction.

Unfortunately, the software processing safeguards were not applied to all potential overload situations. Providing all possible database matches on the first four characters of the keyed-in street name caused the entire system to go down several times in the first few weeks of 1981. These occurrences were all associated with a geographic area which happened to have many highways. By matching HIGH, all highways were attempted to be sent back to the user. However, the software output buffer could only handle 200 of these in one transaction, and this number was exceeded. The software had no programmed safeguards against this overload, and it caused the whole system to lock up. To make matters worse, no output message identifying the specific transaction or the condition was sent to either the user or the computer center. To relieve this condition, the software processing was changed to limit potential matches to 50 and, if more than 50 occur, send none back to the user; instead, provide a new user output message stating "NO EXACT MATCH ON ADDRESS ENTRY—TOO MANY SIMILAR ADDRESSES TO DISPLAY."

In the same vein, other obvious user-oriented I/O features can cause substantial induced load on the system. Two examples which should not be incorporated into the design without careful analysis are (*i*) allowing the end user access to a status transaction which provides what state a previously entered transaction is in, and (*ii*) allowing the end user to enter later transactions without first receiving a response from the first.

Both of these capabilities tend to further degrade an already overloaded condition.

4.2.6 Novice versus experienced variations

Several papers in the field today suggest that there be two varieties of I/O design: one for the brand new end user, as part of a training strategy, and another for the expert user. The novice version would have very simplistic inputs and expanded outputs associated with prompting, menus, and full-error messages; the expert version would be optimized for minimal load on the system, that is, flexible inputs and outputs, but designed to require only a single human/computer interaction per task. The idea is to do a "switch" per terminal or per end-user identification when a certain proficiency level is reached. It has even been suggested that the computer itself track the progress of each user, and via an adaptive switch, change the I/O design from novice to experience accordingly. In PREMIS, only one I/O design exists, but for the new user a Training Subsystem is provided which allows training transactions to be tracked and processed independently from live transactions using a training database. Further field analysis is necessary to determine the cost effectiveness of these various approaches.

4.2.7 User errors

No matter how hard system designers try to make transaction inputs simple, user errors will always be somewhat of a disappointing surprise when the system is in operation. As part of the system design, there should be easy monitoring (mechanized) of error patterns so that some action can be taken in response to frequent, costly errors. The system modification might be in the semantic or syntactic elements of the input or might be solely associated with training.

A mechanized means of sorting user errors by the Service Rep has recently been designed. The statistical information is conveniently displayed in report format and is available on request by management personnel. This report capability is scheduled to become operational in 1981. It is anticipated that feedback to the Service Reps on their error rates will be a positive influence on their performance and attitude. A user error rate objective standard will be established as soon as a substantial amount of data is collected and analyzed. This standard, which may vary depending on local conditions, should also be made known to the Reps. Experimental results[17] have shown that in a repetitive human/machine interface task, the combination of feedback on both individual performance and an accepted (high) standard can have significant positive effect on reducing user errors, while uplifting individual morale and uplifting the organizational climate. A 15 percent decrease in the average error rate was ultimately achieved. In addition, the results of observations and interviews suggests that the increased pressure on performance did not cause any undue physiological or psychological stress.

4.3 Operational considerations

In my experience, three computer operational considerations are by far the most significant: availability, transaction failures, and transaction response time.

The users should be (but usually are not) prepared for the effect of these on their job performance. Our experience has shown that these factors have a potentially drastic and traumatic effect on the user, especially when they first use the system. They must be prepared for these effects.

4.3.1 Availability

The availability of interest to the user is the end-to-end variety; that is, the probability that all components of the system are functioning at the time the user desires access.

Repeated availability problems caused by equipment outages can result in long-term attitude problems. If a terminal, for example, goes down often (low mean-time-to-failure), it not only interferes with work performance during the outages, but it is also viewed with distrust when it becomes workable. This results in the user checking and double-checking results constantly, thus, increasing task time.

Often, however, there are socio-technological obstacles in the way of providing realistic availability information to potential users. For example, an on-site review associated with FACS* uncovered the following:[18]

Overall Application User Opinion: "The system is down on the average of 3 hours per day!"

Operations Staff: "Not so! Availability is in the high 90's!"

How is this widely different view possible? The system administrator is generally measured on how often and for how long the system is down. Therefore, it is highly unlikely that availability—as seen at the user's terminal—is either known or, if known, publicized. The traditional fallacy is the official reporting of "computer main frame downtime during business hours." There are two serious problems with this:

(*i*) As mentioned earlier, care must be taken to deal with end-to-end availability—the product of on-line hardware/software/communications links reliabilities. That is, the terminal, the transmission lines, the message switcher(s), etc., must all be considered to get a true picture of user availability.

(*ii*) Often, just as much work is done during off hours to keep the system up to date. Availability can be just as important here.

A classic case of a difference between perceived availability (by the user) and measured availability (by the operations staff) occurred during an on-site operational review of FACS.[18] At the user's terminals, on many occasions during the day, transactions could not be entered

* FACS: Business Information Systems Customer Service/Facility Assignment System, FACS is a Bell Laboratories-developed mechanized support system, an early version of which was trialed in the late 1970s and is now being redesigned.

into the computer. On the other hand, the official availability of the computer for the same time frame was in the high 90s. After intensive investigation, it was found that a major cause of the confusion was tape handling. As part of the recovery procedures, audit tapes are generated saving a transaction history. These tapes would each handle about 40 to 60 minutes worth of activity. After each tape was full, it had to be manually removed from the tape drive and a blank tape mounted. This takes up to 3 minutes per occurrence. During these time intervals, no user transactions could be accepted by the computer; however, they were not considered outages in terms of reporting availability.

A very interesting user behavioral effect results from these multiple, very short outages. Say, a user attempts to enter a transaction during one of these outages. Normally, a "message received" appears on the screen; in this case, the screen goes blank. The user mentally records that the system is not available, but does not know exactly how long this condition has existed. Suppose now, another try is made. Again, a blank screen. The user gives up, walks away, and does something else. Thirty minutes later, the user walks over and tries again. Nothing. Another mental note: "The system's been down anywhere from a half hour to an hour." Two outages of maybe 3 minutes each resulted in a magnified user performance degradation. If nothing else, this phenomenon suggests that many sporadic, short outages can have a significantly greater effect on user performance than a single, extended outage. For extended outages, telephone calls were generally made from the computer center to each user work center informing them of the situation thereby limiting the perceived availability problem to the actual. (The audit tape problem was subsequently relieved by the introduction of a mechanized transfer process.)

During a PREMIS field visit, the negative impact on the Service Rep of many short outages was further magnified. Over a period of 4 hours there were at least 6 system outages of 5 to 20 seconds each. The total actual time the system was unavailable was less than 1 min. (None of these occurrences was officially included in availability statistics because the minimum outage reportable was 5 minutes.) However, every Rep accessing PREMIS when an outage occurs is also on the phone to the customer. There is, of course, no way of knowing when the system will again be available. The Rep can either choose to bypass PREMIS and resort to back-up procedures (extra time, motion, and chance of error) or skip the portion of the customer contact associated with PREMIS and hope the system comes back quickly. The latter was generally done. However, the Rep must formally log back on to PREMIS each time and either call up or clear out of the system any message or partial messages that were in process at the time of the outage. This might take a minute or two to accomplish. The cognitive load associated with these required actions in concert with verbal continuity of customer negotiations created obvious stress on the Reps. They were visibly disturbed and often made keystroke errors, as well as wrong decisions. Worse, the fact that these outages were not counted (because of their short duration) as official availability measurements put no pressure on the data systems organization to investigate the problem and attempt to resolve it.

Another important issue is the fallacy of partial availability. It has been my experience that the hardware/software portion of the system does not "fail softly" as Martin puts it. No matter what mechanized fall-back processes are built in, the user is usually dramatically affected. What is needed mostly, I believe, are not gradual, stepwise degraded user procedures, but a simple alternate design that bypasses the computer altogether. The user must be able to conduct business for periods of time when the computer is totally down or not available anyway.

Simply waiting for the system to come back up is not acceptable in dealing in real-time tasks with a third party (the customer) such as the Service Rep's job. In effect, the fall-back position closely resembles the procedures and aids which were in the pre-PREMIS environment. After these procedures are designed and tested, they should be included as part of the formal training. In PREMIS, the use of periodic printouts (albeit somewhat out of date) keeps the business going reasonably well.

4.3.2 Transaction failures

Transaction failures are defined to be cases of user-initiated transactions which do not process to successful conclusion. The causes of these failures can be hardware-related or software-related. A high rate of these failures, especially if combined with poor user notification, can seriously affect user performance. An example of such a situation was observed with FACS. Table I shows representative failure statistics.[18]

The causes of these failures were spread among management soft-

Table I—Failure data

Month	Total Transactions	Total Failures	Failure Rate (Percent)
August, 1978	140,770	2873	1.8
September, 1978	135,720	2992	2.2
October, 1978	158,175	1987	1.3

(For comparison purposes, several other large-scale systems manage to maintain failure rates of less than 0.1 percent.)

ware bugs, application software bugs, and database logical and physical inconsistencies. The effect on all three classes of users was serious: the operations staff received all the failure messages independent of the transaction entry source. They were inundated with paper each day and generally were unable to analyze, pattern, or react in any effective way. The paper continued to stack up each day so there was virtually no real-time support provided. This left them frustrated at their obvious ineffectiveness. Each aborted transaction was taking up to eight days to evaluate. The database maintenance clerks were continuing to do their database changes generally unaware which of their transactions "bombed out," because they were not notified by output messages or calls from the support staff. Often, they would discover these events by later transactions not processing properly, that is, the database change may have tried to load some inventory but since it never worked, later accesses against this inventory would fail. This caused frustration on their part because they would receive calls from the application users when their transactions did not work. The application users were negatively affected because their ability to serve the customers' needs in a timely way was degraded. All of the users who initiated transactions that failed were further frustrated because failure messages were not returned to the entry terminal! The lack of job continuity and efficiency, because of these transaction failures, was having a measurable effect on the acceptability of the system from a user point of view. A variety of fixes were designed—the prime effort, of course, focused on improving the quality control of the transaction processing. In addition, the transaction failure messages (in English!) were returned to the initiator's terminal, and extensive training and documentation was developed to help people cope with these events, including straightforward error correction procedures.

4.3.3 Transaction response time

Transaction response time is defined to be the interval between sending the transaction until the first element of the response appears. It has many of the same characteristics as availability and transaction failures in trying to get a true handle. The software designers are sensitive to poor transaction response time perceptions because these perceptions imply less than optimal programming in many cases, or the system administrator may have installed a queueing structure that is inefficient from a user point of view or the communications manager may have installed a poorly balanced communication layout.

It should be noted at this point that fast response time is not cheap. Neither computer processing power or telecommunications network capacity can be indiscriminately increased for fear of negating the economics of the system itself. Therefore, extreme care must be taken

in specifying response time objectives. In PREMIS, the various transactions are grouped according to specific users and discrete response time requirements are established for each grouping. A table-driven queueing structure built into the computer management system supports this structure. In addition, communication lines vary from dedicated to a single terminal to heavy time sharing with others. Martin points out—and I agree—that, in practice, justifying response time based on pure time economic considerations is virtually impossible. It is the psychological factors which can affect user performance. These factors include "expectance," "chunking," "short-term memory," "closure," etc. For example, experiments have shown short-term memory limits are severely stressed when the response time increases. This stress results in user discomfort, as well as increased error rate. Experiments have also shown that user annoyance is very low if the response time is under 8 seconds; some annoyance appears from 8 to 13 seconds, and the user becomes very annoyed if the response time is greater than 13 seconds.[19] This is very consistent with Service Rep reaction on PREMIS. Williams reports on experiments which show that for fairly routine data entry tasks "User performance deteriorates with system delays of greater than 15 seconds."[20] Other experiments have shown a discontinuity at 15 seconds. It becomes more than an annoyance and disrupter—it becomes a demoralizer and reduces motivation.[6] I do not have statistical data on user error rates in PREMIS as a function of transaction response time. However, a recent study by Barber,[21] showed that for a response time of, on the average, 8 seconds, the subsequent transaction had an error rate of about 0.10; when the response time was increased to 20 seconds, the error rate increased to 0.25. For the case of the Service Rep position, Table II gives recommendations based on qualitative behavioral observations (mostly my own judgmental estimates of disturbance levels). These values are also based, unfortunately, on systems without local terminal intelligence so that every user action requires full transmission to the computer:
The recommendations given have the built-in presumption that the

Table II—Response time objectives

Action	During Busy Hour (90 Percent Confidence) (Seconds)
Bring-up screen	4
Paging	4
Input error (syntactic only)	5
Input error (all others)	7
Simple transactions (direct access key to data)	7
Simple transactions (limited search in data base)	10
Complex transactions (extensive search in data base)	15
Transactions allowing parallel tasks	25

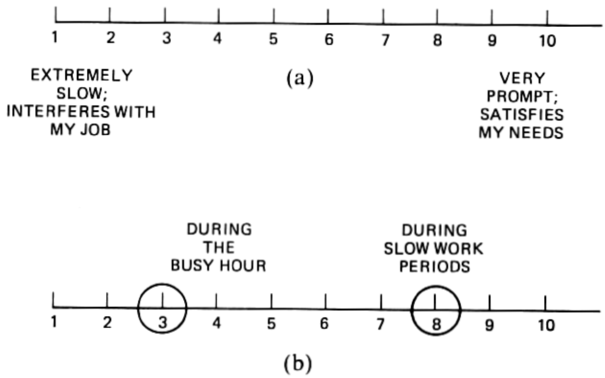

Fig. 2—(a) Original scale for Service Reps opinion of response time. (b) Revised scale.

Rep is willing to wait somewhat longer if they perceive the computer task to be more difficult (simple versus complex transactions) or the system is in a high load condition (busy hour). In other words, the user discomfort level is related to the extent to which the real delay exceeds expectation.

Dealing with response time as seen by the user can have some very interesting and important quantitative and behavioral effects. Recently, the PREMIS users in Memphis reported the response time for simple transactions to be about 30 seconds, while users in Birmingham and Jackson reported a 7-second average. An examination of the terminal and communications network layout revealed that there were excessive terminals in Memphis competing for the same line.

A 1979 PREMIS field review in Jackson, from the Service Rep's perspective, yielded the following:

The Service Rep's attitude towards response time was obtained via formal questionnaires. However, a real-time adjustment to the questions had to be made. Originally, the Service Reps were asked to indicate their opinions of response time. They were asked to choose an answer based on the scale shown in Fig. 2a.

It quickly became apparent that the Reps were having great difficulty with this request. Informal probing led to a decision to pose the request in two parts (See Fig. 2b.), which ultimately provided a true picture when the stop-watch measurements were analyzed (See Fig. 2b.)

Stopwatch measurements were extensively taken at three different user positions over a one-week period. In addition, the users' behavior was observed, and they were requested to complete questionnaires.

Tables IIIa, IIIb, IIIc, and IIId provide some key statistical summary data on response time for the most important transactions. Referring to the table, a success, or "hit" condition refers to a user transaction

Table III—Response time characteristics

(a)

Variations across locations (means)

	Terminal Locations	Computer Time (Mean)	Stopwatch Time (Mean)
Hits (second)	A	4.39	8.87
	B	3.90	12.09
	C	4.80	10.38
	Average	4.27	10.20
No-hits (seconds)	A	5.93	8.15
	B	6.62	13.63
	C	8.45	11.35
	Average	6.56	10.65

(b)

Variations across locations (details)

	Terminal Locations	Stopwatch Timings		
		Mean	Mode	Ranges
Hits (second)	A	8.87	6	5–29
	B	12.09	11	5–60
	C	10.38	6	5–39
No-hits (second)	A	8.15	7	5–39
	B	13.63	11	6–54
	C	11.35	12	5–35

(c)

Variations across transaction outcomes

	Stopwatch Times		
	Mean	Mode	Ranges
Hits (second)	10.68	6.0	5–60
No-hits (second)			
Outcomes			
1	9.61	9.3	5–19
2	9.33	6.0	6–19
3	9.38	8.0	5–29
4	11.68	8.0	5–21
5	11.86	7.0	6–27
6	9.65	7.0	6–18
7	11.43	11.0	7–26
8	10.00	6.5	5–23
9	29.64	26.0	22–54
10	11.23	7.0	6–30

(d)

Variations across business day (means)

	9–10 a.m.	10–11 a.m.	10–12 a.m.	12–1 p.m.	1–2 p.m.	2–3 p.m.	3–4 p.m.	4–5 p.m.
Hits (second)	6.3	10.2	12.5	7.0	8.7	11.5	7.5	8.1
No-hits (second)	10.6	17.5	12.2	8.8	9.3	9.8	9.9	9.1

which returned the requested information to the user; a "no-hit" condition implies an undesirable result usually accompanied by an error message. The response time objective was 7 seconds. Table IIIa shows the variation in mean response time across users' locations for both user response time (measured by stop watch) and computer response time (automatically measured by the computer, representing only the main frame residence time). Note that the use of the computer response time as the only official reportable number had provided a false sense of security. The variations between user sites is significant; therefore, a more detailed statistical breakdown is shown in Table IIIb. (It turned out that the computer polling algorithms were out of balance causing these variations). The response time variability between transaction outcomes is shown in Table III (one specific result was the redesign of the software processing for the ninth no-hit condition). Finally, Table IIId shows the variation in response time across the business day. It showed the longest response times before and after lunch time which is consistent with the known customer activity level variations (this information resulted in recommending to the maintenance and computer center staff that offline transaction activity be avoided during these time periods).

Based on my observations and later interviews, I believe the wide variability in response time for different outcomes was apparently at least as troublesome to the users as the mean response time. This is no longer surprising based on several recent experiences and experiments appearing in the literature. First of all, what variability is detectable by the users? A recent experiment by Butler[22] showed that for a response time in the 8-second range, a 2-second increase was consistently detectable by 75 percent of the subjects. Now, in terms of discomfort, Miller notes "Increasing the variability of the output display rate produces both significantly decreased human performance and a poorer attitude toward the system and the interactive environment." The results of this experimentation showed that user satisfaction, measured on a relative scale, decreased by 25 percent as the response time variability went from low to high. In addition, the amount of time required by the user to analyze the output and take action increased by 15 percent under the same conditions.[6] In addition, not preparing the user for this compounds the negative effect. Shneiderman, in reviewing several experiments, states "If the variance of response time is small, users incorporate the waiting into their work patterns by pre-planning future queries or attending to other functions, but if the variance is large, users must maintain a continued high level of awareness and become anxious if response time grows."[2] Quantitatively, Martin recommends as a rule of thumb that the standard deviation of response time should not be more than half the mean response time. Therefore, for an 8-second mean, two-thirds of the real response time should be less than 12 seconds.

Finally, a condition with both flaky transaction failures and slow and erratic response time is more damaging to user performance than the sum of each effect. A recent review of PREMIS, which had such a combined condition, disclosed high-user frustration at not knowing if a blank screen meant something wasn't working or the system was just slow. On-site observations showed that they would often send the same transaction several times in succession, thereby compounding an already overloaded input queue. This phenomena of an induced load was supported by data on audit tapes. An analysis of the data[16] revealed that as the system approached the busy-hour, users repeatedly hit the send key in their frustration at the slow response time. The Service Reps openly admitted that they were doing this. Figure 3 shows the measured response time characteristic versus offered load for the system as it existed in September, 1980. The effect of the induced load on response time is shown for a particular Monday morning busy hour, where a serious computer problem caused repetitive entry of very long running transactions. In these cases, each transaction was, in effect, worth many normal transactions. It is estimated, based on audit tape data analysis, that this user-initiated phenomena combined with transaction processing difficulties increased the average response time from a normal 10- to 20-second range to a 60- to 70-second range.

V. TARGET ORGANIZATION/ENVIRONMENT CONSIDERATIONS

It should not be surprising that the performance of a very large computer-based application can be greatly affected by (and can greatly affect) the target organization/environment. Every attempt should be made to analyze this impact up front and force appropriate adjust-

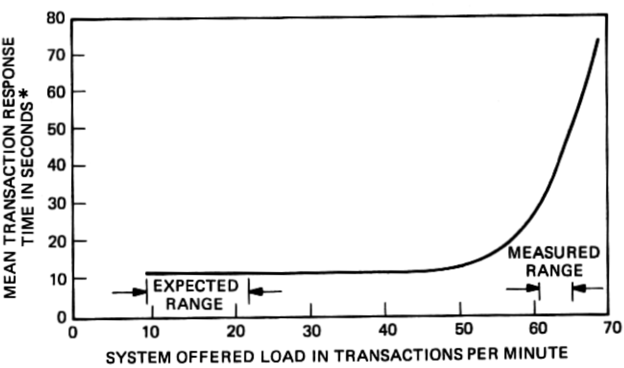

Fig. 3—Load characteristics.

ments in the organization or environment to allow a graceful introduction of the computer system. The first consideration—the relationship of the computer tasks to their appraisal through existing job performance measurements—is extremely important and often underestimated, especially for the application user. This can be the focal point of serious user performance problems. The Service Rep job is one such case.

5.1 Relationship of existing user job performance measurements to use of the computer

There are three major possibilities here:

(*i*) Does the Rep perceive the computer system to have a positive effect on their performance measurements?

(*ii*) Are the measurements themselves tied to bottom-line system performance objectives?

(*iii*) Are the Reps provided feedback on how well they are doing?

Assume a Service Rep is negatively measured, at least in part, if the customer contact is broken off for any reason and the customer is put on hold. In a pre-PREMIS environment, paper records are accessed during the contact; with PREMIS, the computer is accessed via the terminal. The computer-based records may be more accurate than the paper records. This guarantees better service to the customer which presumably has positive satisfying value to the Rep. But the more significant behavioral influence on the Reps is the effect of the computer on their job performance measurement. So, for example, if the system response time is so slow that Reps believe that customer contact breaks will increase, they are going to be unhappy.

An on-site review of the Service Rep's use of PREMIS was held in Jackson, Mississippi, in 1979. Under the existing measurement structure at that time, breaks in customer contact played an important role. That is, Reps with frequent and lengthy contact-time breaks would be negatively effected. Personal interviews with the Reps included the question, "How do you perceive the effect of PREMIS on your average contact time? Their verbal responses generally showed a keen sensitivity to the issue and a uniform distress over the transaction response time degradation in the busy hours (the observation of response time as a sensitive behavioral characteristic will be further amplified in the next section). There is no easy way out of this without cooperation of the target organization. A slight increase in contact-time breaks and break durations may, of course, be a small price to pay if a substantial reduction in key information errors occurs which ultimately translates into reduced total work or some other bottom-line measure. One such measure is input error frequency.

A data collection summary[23] of the effect of PREMIS on the frequency of errors in an important parameter makes this point. One of the items that a Service Rep must determine is the address of a customer who wants service so that the electrical connections can be assigned and hooked up to the right house, as well as provide the downstream accounting people with the right place to mail bills. The Service Rep places the address on a contact memo which later is used to generate the formal service order. If the address is incorrect, a variety of difficulties can develop, depending on how long the error goes without being detected. If the error is caught early, the order is sent back to the Rep to change and re-submit; if the error succeeds in surviving the entire service provisioning process, billing discrepancies may occur. In any case, there is substantial cost associated with each of these errors. One objective of PREMIS is to provide the Rep with on-line address verification support during the customer contact, thereby avoiding address errors up front. Table IV summarizes the history of address error statistics since PREMIS was cut live in Jackson, Mississippi. The precut-live error rates were on the order of 10 percent in 1978.

It is generally acknowledged that PREMIS—both from an ease-of-use and a database-accuracy point of view—allowed this dramatic improvement.

The key is to revise the job performance measurement closer to the bottom line instead of to a secondary parameter. This is easier said than done, especially if this condition is not recognized before the computer system is installed.

Recently, the Service Rep job performance measurements were changed. The emphasis is now on customer service rather than on the contact time. That is, questionnaires on how gracefully and efficiently the job is being done, from both a behavioral and efficiency point of view, are now being sent directly to customers on a per-Rep basis. If an address error was not caught early and proliferated all the way to the installation step, the customer would be negatively affected (by the missed appointment) and would so report. This change of emphasis on individual performance measurements was implemented totally

Table IV—Address error trend

Quarter	Total Service Orders	Orders With Address Errors	Percent
1Q79*	58,707	3,919	6.5
2Q79	77,746	4,570	5.9
3Q79	80,800	4,154	5.1
4Q79	69,217	3,305	4.8
1Q80	68,717	2,730	4.0
April–May	47,167	1,388	2.9

* Premise begins live operation.

independent of PREMIS; however, the beneficial effects to PREMIS are obvious.

Of utmost importance is the direct feedback to the user of both the performance objective and the user's personal performance. In the famous Hawthorne investigation on worker productivity, the investigators found that the experimental subjects demonstrated improved performance regardless of the variables. Many years later, H. M. Parsons concluded that the two major influences on the workers were (*i*) they were given frequent feedback on their performance, and (*ii*) they were aware that their performance was directly tied to their salary increases.[24]

In addition, the Service Rep is now handling new functions on a PREMIS environment. The associated new tasks, of course, take additional time; but from a total organization point-of-view, the functions are being performed more efficiently. This very important consideration is discussed next.

5.2 Relationship of the existing organization-to-function mapping to an optimal strategy

In an organization/environment where the work is broken down into several discrete functions performed by discrete work groups, the flexibility to re-distribute certain tasks across traditional boundaries can often improve the effective use of a computer system. A case in point are the traditional organizational entities in the BOC, where several vertical management structures each are involved in the processing of a horizontal customer service order. To illustrate the example, consider the roles of three of these organizations: The establishment of the service order (Service Reps); the assignment of facilities (assigners) and; the installation of the facilities (installers). PREMIS provides on line, up-to-date information up front to the Service Rep associated with the service history at a particular address. Now, one specific issue for which PREMIS can provide benefits is the interfering station condition. Simply put, it is a request for new telephone service at an address where previous service has not been terminated. This most often occurs when the new customer's request for service is processed before the processing of the disconnect service order of the existing customer at the same address. Before PREMIS, this condition was often not detected until the installer goes to the address, thereby causing significant extra work by the Rep, assigner, and installer. With PREMIS, the information about the existing customer is displayed on the screen while the Rep is negotiating with the new customer. Now, if the Rep's job can be expanded to resolve this condition right up front, significant wasted work can be avoided downstream; that is, a modest increase in the Rep's work load is certainly justified from a

Table V—Interfering station condition (ISC) workload analysis. Assumptions: Percent of inward service orders for which an ISC exists is 7.3%; percent ISC identified by Service Rep with PREMIS is 92.0%.

Workload Characteristics	Service Rep	Assigner	Installer	Other
Average wasted work effort per ISC occurrence *not* detected by Service Rep	12 min	20 min	45 min	10 min
			ISC Identified by:	
		Assigner	Installer	Other
Annual benefit per line with PREMIS		4.7¢	4.4¢	0.56¢
Annual expenses per line with PREMIS because of increased Service Rep work efforts	(1.6¢)			
Annual benefit* (cost) per year with PREMIS for 4 million line entity	($64,000)	$188,000	$176,000	$22,400

* Annual benefits per line are linear.

corporate point-of-view if a much larger savings is realizable by the assigner and installer. An economic study by Hafelfinger,[25] based on on-site observations, interviews, and several months' worth of data, convincingly demonstrated this. South Central Bell initiated new work flows in a PREMIS environment across the relevant organizations, thereby allowing pre and post data to be gathered. Table V summarizes the key results.

Thus, per PREMIS system, almost a quarter of a million dollars can be saved yearly by optimizing the task requirements across organizational boundaries. (It is believed that these figures are conservative and much more savings is likely.)

It is important that the system design and development plan include an activity associated with the optimal re-association of functions across organizational boundaries. The best vehicle and methodology to accomplish this depends on the relationships of the system design and customer organization and the complexity and breadth of the mechanized functions. A vehicle that has proved effective for PREMIS is the formal establishment of a working committee of several BOC representatives working under the auspices of the AT&T Residence Service Center, with consulting support from the Bell Laboratories PREMIS HPE group. The committee recommended a generic approach to the changes in existing job functions in the Residence Service Center to best use PREMIS. Detailed before and after work flows formed the basis of the method used in this activity and proved to be an effective device.[26] An adjunct method—to be used in conjunction with work flows—has also shown some advantage for certain environments, i.e., the formal groupings of like tasks into a series of work modules.[27] These modules have characteristics as follows:

(*i*) Not splittable between people.
(*ii*) One (or more) make up a person's job.
(*iii*) Each can be independently assigned to the appropriate group.

The strategy is to define the work flows (and work modules) as part of the system design process early enough to influence the adjustments in the target organization's job structure. The work modules can also be used as the basis for organizing procedural user documentation and training, independent of job definition.

Make no mistake, there is a real threat to the success of a computer system if the organizational dynamics are not well understood and dealt with. The worst possible condition exists when the computer system is almost totally supporting one department but dependent on another to increase its workload to make the system work effectively in a mechanized environment. To ensure an effective working system, an early corporate, multidepartmental commitment is vital. It has been my experience that when it is shown that additional human effort in one department can save much more human effort in another department and improve the corporate bottom line, with the organization doing the extra work getting the credit and a good public relations job is done, then the chances of acceptance will be improved. This was the case of the ISC discussed earlier. On the other hand, if workers in one department are asked to increase (or even change) their jobs to make it easy to program a computer helping another department there is extensive resistance.

5.3 Customer interface

Another potentially important environmental influence on user performance is the customer interface. This is appropriate when the performance of the user tasks includes interfacing with the customer of the service being provided. This condition exists for the case of the Service Rep where the customer is initiating action with respect to obtaining (or changing) telephone service.

An earlier example discussed the potential problem of the target organization using customer contact time as a measure of a Service Rep's job performance. This presumes direct involvement with the customer over the telephone.

In the telephone work mode, the Service Rep is visually interpreting screen responses from the computer, while simultaneously listening to the customer provide additional data all under contact-time pressure. We know from experimentation[28] that there are load stress bounds beyond which there can be some degradation of human performance if, during a person's job, he or she is time-sharing between auditory and visual inputs.

PREMIS is also being used in an environment where the Rep is face-to-face with the customer. Now the behavioral damage to the Rep of a slow computer response time is compounded. The visibility of the interaction to the customer has been dramatically extended, thereby introducing a "fish-bowl" effect on the user. Over the phone, perhaps the Rep can disguise the delay by gathering additional information for a time, and then, out of desperation, discuss the weather or last Saturday's football game. In a face-to-face contact, I have seen the Rep and the customer staring nervously at a blank screen for a time that seemed like forever, but which was only for about 20 to 25 seconds. This was very disturbing to the Rep, especially if the response time increased during the busy period, with other customers waiting in line (also watching the blank screen).

Corrective strategies to minimize the pain in this environment had to do with adjusting the work station layout and changing the work-flow steps. When possible, screens were not made visible to the customer. This was relatively easy to do when the customers and Rep were separated by a counter or barrier. For layouts where there is free flow of customers throughout the work area (as in many Phone Center stores), the terminals were put off in a corner and accessed privately by the Rep after he or she had collected the pertinent information from the customer on a scratch pad.

VI. SUMMARY

The introduction of a very large computer-based application system to an existing work environment provides the opportunity for enhancing the application user's job performance. However, an awareness is necessary of the special computer considerations associated with the organization/environment pressures, the interface with the customer, and the user's own behavior. The features of the mechanized support should be carefully tailored to make the human/machine interface as effective as possible. Principles used in the functional and operational design of PREMIS—with the Service Rep as the user—emphasize simplicity, consistency, and clarity.

The field experience gathered over the last two years has shown that certain operational aspects of PREMIS can have a cumulative, very potent negative effect on user performance. As reported by the Reps, these include "flaky" availability, slow response time, and a cumbersome log-on procedure. Also, the Reps report that the time to complete the functions and the level of task difficulty were not decreased by the introduction of PREMIS. Balancing this, the Reps feel that the system has allowed them, as well as the rest of the workers, to serve telephone customers more effectively and efficiently with higher confidence and less errors. The general positive influence of PREMIS—as seen by the

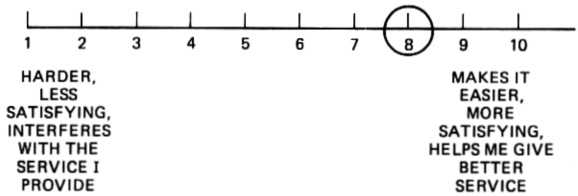

Fig. 4—Service Reps response to "How has PREMIS affected your job?"

Service Reps—was apparent in the final question of their most recent field review questionnaire, "How has PREMIS affected your job?" with the average score circled (see Fig. 4).

In my judgment, this positive reaction is in large part because of a systematic influence in the system design decisions of conscious human performance engineering principles. In addition, a total system engineering view must be taken of all design features so that the user benefits are not outweighed by hardware/software costs.

The time is past where large-scale computer systems can be initially delivered, found to be unacceptable from a user point of view, and then virtually re-done. The applied technology of human performance engineering has reached the stage in maturity and know-how to influence the a priori system design and help achieve a successful initial product.

VII. CONCLUSION: A SYSTEM DESIGN PERFORMANCE AID

The following techniques work when designing computer systems that are easy for people to use:

(*i*) Have HPE people perform an early task analysis. This can substantially avoid unworkable or unreasonable user performance requirements. Divide tasks between people and machines, making the best use of the strengths of each; for example, allocate simple, repetitive jobs to the computer and complex, judgmental jobs to people.

(*ii*) Develop work flows early to ensure that the work will get done in a logical, workable sequence.

(*iii*) Design each system in an incremental fashion—both from a software-architecture (internals) and user-transaction (externals) point of view.

(*iv*) Involve the existing target organization in the system design process to help determine appropriate work shifts to achieve optimum use of the mechanized assist features. If certain work groups had added tasks, make sure they will get credit.

(*v*) Understand the impact of the computer system on the existing work performance measurements of individuals and groups to avoid behavioral resistance.

(*vi*) Design the human/computer interface and transaction command structure to be consistent and simple—simple to use and simple to learn.

(*vii*) Allow the user to initiate and control all human/computer interactions.

(*viii*) For a computer system without local intelligence, use a batch-prompted (with mask) input mode.

(*ix*) Limit inputs and outputs to a single display frame (page).

(*x*) Design output screens to allow over-typing of input commands to request the next transaction.

(*xi*) Return meaningful messages in English to the user for all possible transaction outcomes.

(*xii*) Incorporate computer-assist features (menus, prompts, defaults, minimal input) but only after task analysis is done to ensure that these features do not induce unnecessary transactions causing computer overload, as well as forcing users to take longer to complete their tasks. Perform iterative analysis of alternate designs to establish a quantitative computer/human performance balance.

(*xiii*) Make system-to-system switching from the same terminal an easy operation.

(*xiv*) Make terminal function keys and dialogue pnenomics consistent among systems.

(*xv*) Incorporate ease of use principles in information display design (data displayed in actual work sequence, data preservation across outputs, labeling, etc.).

(*xvi*) Make all user messages constructive and supportive, not condemning and confrontive.

(*xvii*) Establish quantitative system performance objectives and computer costs for

a. Transaction Response Time by transaction class, processing type, and outcome
b. Total System Availability number of occurrences, as well as total time
c. Transaction Failure Rate by transaction class.

(*xviii*) Establish quantitative human performance objectives and costs for

a. Human error rate by transaction class
b. Training time
c. Work time.

(*xix*) SELL THE SYSTEM. Use any means—user group meetings, presentations, deliverable documents, informal visits—to convince the target population, top down and bottom up, that the addition of the computer to the work environment is positive in terms of job effectiveness and user satisfaction.

VIII. ACKNOWLEDGMENTS

Several colleagues supplied many useful suggestions. In particular, D. F. Lee, L. Bernstein, S. R. Fagan, and M. W. Soth, by their reviews, provided valuable guidance in the organization of this paper; P. R. Hawkins and M. P. Tarka helped with data analysis; and H. Dravis was involved in most of the editing and revision work. I thank them all.

REFERENCES

1. M. J. Heffler, private communication.
2. B. Shneiderman, "Human Factors Experiments in Designing Interactive Systems," Computer, December, 1979, pp. 9–18.
3. B. Shneiderman, "Only Non-Human" Datamation, Forum (May, 1980) p. 295.
4. T. Gilb and Gerald M. Weinberg, *Humanized Input: Techniques for Reliable Keyed Input*, Englewood Cliffs, N.J.: Winthrop Publishing, 1977, pp. 629–33.
5. R. F. Jones, "Four Principles of Mass-Computer Dialogue," IEEE Trans. Professional Commun. *PC-21*, No. 4 (December, 1978), pp. 154–9.
6. L. H. Miller, "A Study in Man-Machine Interaction," National Computer Conference, *46,* AFIPS Press, 1977 p. 409–21.
7. J. R. Hackman and G. R. Oldham, "Development of the Job Diagnostic Survey," J. Appl. Psych. (April, 1975), pp. 159–80.
8. J. L. Bennett, "The User Interface in Interactive Systems," *Annual Review of Information Science and Technology*, *7,* ed., C. Cuadra Washington, D. C.: Amer. Soc. Inform. Sci., 1972, pp. 159–96.
9. J. Martin, *Design of Man-Computer Dialogues*, Englewood Cliffs, N.J.: Prentice Hall, 1973.
10. B. L. Hicks, private communication.
11. N. L. Ferrer, T. G. Hicks, and N. E. Lyness, private communication.
12. M. J. Rothkind, private communication.
13. F. Gebhardt and I. Stullmacher, "Design Criteria for Documentation Retrieval Language," J. Amer. Soc. Inform. Sci. *29,* No. 4 (July, 1978), pp. 191–9.
14. W. J. Hansen, "User Engineering Principles For Interactive Systems," Proc. Fall Joint Computer Conf., *39,* AFIPS Press, 1971, p. 528.
15. P. R. Hawkins, unpublished work.
16. M. Tarka, private communication.
17. B. Das, "Consequences of Production Feedback and Standards in a Human-Machine Task," Proc. Human Factors Soc., 1980, 24th Annual Meeting, pp. 629–33.
18. AT&T, BISCUS/FACS Review, 1978. Dallas, Texas, Section on System Administration.
19. M. S. Schoeffler, private communication.
20. J. D. Williams, private communication.
21. R. E. Barber, Experimental Results from 1979 Ph.D. Dissertation, N.Y.U.
22. K. Butler, private communication.
23. I. S. Yavelberg, private communication.
24. H. M. Parsons, "What Happened at Hawthorne?" Service Magazine, March, 1974.
25. J. Hafelfinger, private communication.
26. HOMAC Committee, "PREMIS Work Flows," AT&T Residence Department, Basking Ridge, New Jersey, 1980.
27. M. W. Soth, private communication.
28. Mowbray, private communication.