For analytical purposes, a teleprocessing system can be characterized as a digital computer with unscheduled inputs from a number of remote points. In the design of such a system, various queuing problems arise as a consequence of the unscheduled inputs, and the necessity of linking remote points to the central computer leads to a problem in combinatorial mathematics.

To show the origin of these problems, a functional classification of teleprocessing applications is given, a schematic of a basic teleprocessing system is introduced, and the relative merits of mathematical analysis and digital simulation are discussed.

On teleprocessing system design

Part I Characteristic problems

by W. P. Margopoulos and R. J. Williams

In the typical teleprocessing system, communication facilities link remotely located input/output devices with a centrally located computational facility. But while communication links and remote devices are the most conspicuous marks of a teleprocessing system, they are not the most significant. From an analytical standpoint, the characteristic that best differentiates teleprocessing from conventional processing lies in the nature of the job inputs. In the teleprocessing application, the primary inputs are unconditioned; in the batched-input application, the primary inputs are under the control of installation managers and operators.

The amount and kind of work to be performed by a teleprocessing system may be undeterministic in various ways. For example, a remote device may communicate haphazardly with respect to time, or it may routinely submit observations that vary widely in their processing implications. In either case, the system must be prepared to deal with a work load that is influenced by chance. To say that teleprocessing job inputs are *unscheduled* is to encompass all of the uncontrolled factors that thrust a varying work load on the system.

The existence of unscheduled inputs not only leads to shifts of emphasis among long-accepted operating principles, but leads to a number of new system-design problems as well. Our purpose is to identify these problems and show their origins. To this end, it seems best to start by characterizing teleprocessing applications and discussing a typical teleprocessing configuration.

discussed more and more often today, usually includes all four functions in one system.

response time

Although the term "response time" is relatively easy to define in the context of actual application, operating mode, and device configuration, it eludes a general definition. For the sake of discussion, let *response time* mean the time that elapses between the completion signal of a terminal entry and a "response" signal indicating that posting is completed or that a desired answer has been received. Thus defined, response time is analogous to turnaround time in a batch system.

The desired response time in a teleprocessing application may vary from milliseconds to hours. In the operating mode known as remote job entry, the user employs a terminal and a relatively high-speed line to transmit the data (and often the programs as well) for an entire job as one continuous entry. The computer processes the job, as permitted by priorities, and returns the desired output as one transmission. The response time may be hours in such a system; for example, the user may submit a job at closing time and have no need for the results until morning. On the other hand, he may submit a small job and desire the answer in seconds; moreover, the economics of skilled labor and project deadlines may well justify immediate answers.

The advantages of rapid response times have generated a good deal of interest in the "conversational" mode of operation for job-shop computers. In this mode, the terminal user is given the ability to control, interrogate, observe, and modify a task during the course of computation. This mode of operation is most feasible where the central computer is endowed by design with features and programs that facilitate so-called "time-sharing."

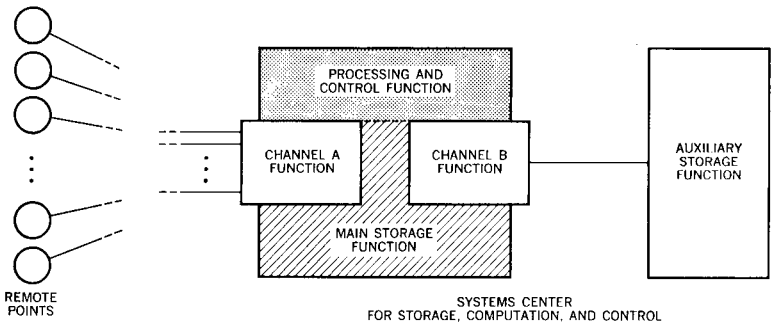
The remote job-entry and conversational-entry modes are particularly exacting in their requirements because they permit a whole spectrum of jobs, including program compilation and debugging. The teleprocessing type of inventory or accounting application normally dedicates all or part of a system to one continuing job. Examples are airline reservation systems, on-line banking systems, and data-gathering systems for production scheduling.

The response-time requirements get tighter and tighter as we get down to process-control applications, particularly to those that monitor and control missiles. Although most teleprocessing applications have stricter response-time requirements than most batched-input applications, it should be noted that response time is not entirely satisfactory as a criterion for distinguishing between the two types of applications. To reiterate, the distinguishing characteristic of teleprocessing is unscheduled input.

A functional view of the teleprocessing system

A schematic of a simple teleprocessing system is shown in Figure 1 (where miscellaneous input/output units are ignored). Terminals

Figure 1 Schematic of a teleprocessing system



at remote points are connected to a systems center via communication lines. The systems center consists of the following primary functional units:

- Auxiliary storage
- Main storage
- Channel for connecting lines to main storage
- Channel for connecting main storage to auxiliary storage
- Processing and control

The typical message is characterized by a gross flow that starts with a terminal, goes through Channel A, main storage, Channel B, and finally ends in auxiliary storage; the return message or result takes the same path but in reverse order.

The distance between the remote input device and the processing unit usually makes the cost of multiconductor cables prohibitive. Engineering advances in the ability of the common carriers to transmit digital information over conventional communication facilities have, in part, made teleprocessing systems feasible. The transmission rates of available communication lines vary all the way from the keying rate of a typewriter to over five thousand characters per second. Compared to the rate at which a human can enter data on a terminal, the rate at which most lines transmit data is relatively fast. Yet, in most cases, the line rate is slow compared to the rate at which a processing unit can accept data. This allows the system designer to time-multiplex terminals on communication lines and to time-multiplex communication lines on a channel at the systems center.

lines

The control of lines and terminals may be accomplished in a number of ways. Multiterminal lines can operate in at least three different modes. The first is the *contention* mode, wherein the terminals and processing unit request a line on an as-needed basis. If the line is busy, the terminal must persist in its request until it succeeds. Where it is desirable for the processing unit to maintain positive control over the order in which lines are recognized, the processing unit can accumulate input messages from the terminals by a periodic poll of each terminal. This is

referred to as the *roll call* mode of line control. In rapid-response systems, the *go-ahead* mode of line control can eliminate a substantial amount of the control communication between the processing unit and the terminal. In this mode, the processing unit initiates polling with the first terminal on the communication line; subsequently, the poll of each remaining terminal is initiated by its predecessor terminal in a fixed polling order.

The terminals in a teleprocessing network may consist of devices with many different configurations and performance characteristics. A terminal is fashioned from functional devices such as keyboards, card readers, card punches, paper tape readers, paper tape punches, printers, visual displays, and analog-to-digital converters. A terminal may consist of several different devices, in which case each device is distinguished as a *component* of the terminal. The range of available devices is expanding as user needs become more clearly defined through operational experience.

central computer

The devices needed for processing and control, main storage, and channel functions may be integrated into one general purpose computer. In the IBM SYSTEM/360, for example, Channel A functions are met by a *multiplexor channel* and Channel B functions by one or more *selector channels*. To a large extent, the computer requirements are those needed for batch processing with overlapped operation of a number of conventional input/output devices. Because it will be expected to handle multiple unscheduled messages on an interleaved basis, the central computer should provide efficient means for program interruption.

The systems center may include extra buffering devices that collect bit-by-bit information from transmission lines and supply character-by-character information to Channel A.

auxiliary storage

The allocation and scheduling problems inherent in all computer operation are aggravated by the occurrence of unscheduled inputs. As a result, the teleprocessing application tends to require more direct-access storage than a batch application. This need is usually met by supplementing main storage with magnetic drums, disks, or similar devices with direct-access capabilities.

control program

The computer programs for a teleprocessing system are necessarily complicated; many detailed steps are required to handle the diverse aspects of the allocation, scheduling, and control functions. The control-program requirements of conventional batch-processing operations can be taken as the base requirements for a teleprocessing application.[1] Additional requirements are generated by the peculiarities of telecommunications equipment and the unscheduled mode of operation; these magnify the amount of detail that the programmer has been accustomed to by ordinary punched card and magnetic tape operations. Moreover, the presence of unscheduled inputs complicates the debugging phase of program preparation.

To help minimize the training, coding, debugging, and program maintenance phases of program preparation, the programmer needs

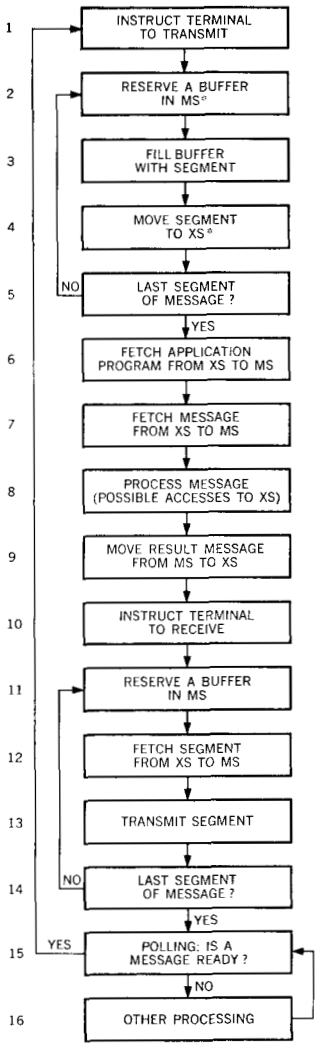
some of the advantages of a high-level language system. One possibility now being used is to bolster the control program with a communications control program that can be generated for a given application by the use of system macroinstructions.[2]

To give an idea of the kinds of interactions that occur among the units of a teleprocessing configuration, the flow chart in Figure 2 indicates a serial pattern of events for receiving one inquiry message and subsequently sending out one response message. Initially, we assume the system is in a processing and polling loop (Statements 16 and 15). Then the polling procedure recognizes the inquiry message and branches to Statement 1.

The incoming message is assumed to be chopped off into segments, and each segment to be placed in a main-storage area set aside as a buffer. A buffer is emptied into auxiliary storage as soon as it is filled, and buffers are filled and emptied as necessary (Statements 2–5). When the message transmission is completed, an appropriate application program is fetched from auxiliary storage (Statement 6), as is the message itself (Statement 7). The message is processed (Statement 8) to produce a result message, which is moved to auxiliary storage (Statement 9). The result is then buffered and transmitted to a terminal (Statements 10–14). One of the points to note in Figure 2 is the major role played by auxiliary storage.

Figure 2 not only subordinates a good deal of detail, but it is idealized in assuming that terminals, lines, channels, processing unit, and auxiliary storage are always readily available. In reality, because such a mode of operation would be far from economical, many other messages will be flowing through the system during the same period of time. Various messages will share a pool of buffers, and programs will have to queue their requests for auxiliary storage devices and channels, as well as for the processing unit. As a result, the design of a teleprocessing system is sometimes described as a solution to several interacting queuing problems.

Figure 2 Flow of inquiry and answer messages



* NOTE: MS – MAIN STORAGE
XS – AUXILIARY STORAGE

Subsystem analysis

The analysis of a teleprocessing system is usually divided into the analysis of three subsystems: communication, central computer, and auxiliary storage. Because each of these subsystems is partially dependent upon the others, the overall design is an iterative process taking into account trade-offs in all three areas. For example, given a system that is not input/output limited, less main storage will be required with a faster processing unit, yet overall response time may be decreased by adding more lines and/or terminals without increasing the processing unit capability.

communication subsystem

The design of the communication subsystem involves selection of terminals, lines, and line-control procedures. In the analytical sense, two basic problems arise. The first involves appropriate line-loading factors, which in turn affect the response-time performance of the communication subsystem. This problem is difficult to generalize about because it is heavily dependent upon

the application characteristics, such as the manner in which peak loads are to be handled. The problem is best attacked by simulation.

The second problem, which concerns the line configuration of the communication network, is more easily generalized. Given information on acceptable line loadings, the problem is to connect terminals to lines in a manner that minimizes the line operating charges. Although no speedy method of finding a minimum-cost solution for non-trivial networks has come to our attention, Esau and Williams discuss (in Part II of this paper) a simple and useful method of constructing a near-optimal network.

computer subsystem

Given a suitable processing unit and operating system as a starting point, the main estimation problems in the computer subsystem concern main storage. The required amount of main storage is a function of the size of control programs, application programs, and input/output buffer areas. Because the traffic of a teleprocessing application is unscheduled, a probability analysis is needed to estimate the appropriate number of buffers. For a case of dynamic buffering (akin to that of Figure 2), Bricault and Delgalvis discuss (in Part III) a method for estimating the number of buffers.

auxiliary storage subsystem

As suggested by Figure 2, message handling can lead to a considerable number of accesses to auxiliary storage. A reasonable number of asynchronous messages, in various stages of completion, lead to a haphazard stream of access requests that can be analyzed by probability theory. In Part IV, Seaman, Lind, and Wilson discuss a probability model that can be used to estimate file response time, queue lengths, and device utilization factors.

Channel units, such as those represented by Channels A and B in Figure 1, use main storage in the course of their operations. For good reasons, these channels normally have a higher priority for storage cycles than the processing unit. As a result, the processing unit is reduced in its potential by what is termed *channel interference*. In the case of some channels, the SYSTEM/360 multiplexor channel being an example, interference is not entirely a linear function of the message rates. Gay provides a simple method for estimating channel interference (see Part V).

System analysis

Today, simulation programs represent the most general and powerful tools available to a systems designer. With them, a system designer can predict the performance of an entire system, or investigate any portion of the system that requires analysis. The results, of course, can be no more valid than the input data and the assumptions used in constructing the simulation model.

The modeling of a system using a simulation program usually consists of two phases. First, the model is written, debugged, and verified; second, a simulation run is made. This entire two-step process may have to be repeated if the simulation results lead to structural changes in the system being modeled. Early

runs will normally suggest additional runs as parameter ranges converge toward a final solution. The amount of time required to write a model depends heavily upon the desired level of detail and the complexity of the system. Moreover, a detailed model will require far more computer running time than a simple model.

In the early stages of system design, there is usually insufficient data to construct a meaningful simulation model. At this point, mathematical techniques based on queuing theory can prove indispensable in determining the general directions that should be taken in outlining the system. As more data become available, a simulation model becomes feasible. As the system is developed, and the application programs and other parameters become better defined, the model can be used to "tune up" the system and evaluate new alternatives. By maintaining the model throughout the development cycle, and even during the operational phase of the system, the impact of increased activity or additional applications can be readily evaluated. In Part VI, Seaman discusses the nature and use of digital simulators at greater length.

Summary

Because a teleprocessing system is characterized by a multiplicity of unscheduled inputs, performance predictions are somewhat more difficult to make than in the case of the batch-processing system. Nevertheless, probability and queuing theory can be used to analyze various subsystems within the total system. The nature of a teleprocessing system is outlined, and the subsystems most deserving of mathematical analysis are isolated.

The advantage of direct mathematical solutions for systems design, where applicable, is that results can be obtained rapidly and inexpensively, often without even running a computer program. The advantage of using digital simulation is that the peculiarities of a proposed teleprocessing system can be modeled more accurately and therefore evaluated more completely. The advantages of using the two methods together are twofold. Formulas can be useful when it is still premature to write a simulation model. Later, they can be helpful in targeting the parameter spaces of most interest, thereby cutting down on simulator running time.

CITED REFERENCE AND FOOTNOTE

1. For an introduction to the functions of an operating system and control program, see G. H Mealy, B. I. Witt, and W. A. Clark, "The functional structure of OS/360, *IBM Systems Journal* 5, No. 1, 2–51 (1966).
2. For the reader who is interested in pursuing the concepts and use of a communications control program, the following manuals published by the IBM Data Processing Division, White Plains, New York, should be helpful: IBM Operating System/360 Telecommunications (C28-6553); IBM System/360 Disk Operating System Extended: Basic Telecommunications Access Method (C30-5001); IBM System/360 Operating System: Basic Telecommunications Access Method (C30–2001); and IBM Operating System/360 QTAM User's Guide: Message Control Task Specification (C20-1640).