Innovations in the operation of future computers

by FREDERIC G. WITHINGTON

Arthur D. Little, Inc.
Cambridge, Massachusetts

EVOLUTIONARY TRENDS

Not many years ago the standard way of operating computers was in a single stream batch mode. The operator completely controlled the computer resource, setting up each job before its initiation, mounting the appropriate tapes and I/O media, initializing the machine, putting the appropriate program deck in the cardreader and so forth. The user interacted with the operator most often in a "closed shop" mode, giving the operator a run request from a card deck and tapes as needed, and leaving it to him to take care of applying the resources to the job.

Because of the waste of computer time involved in single stream batch operation, multi-programming became general. In the multi-program environment the roles of user and operator did not change in any relative way, but the role of the operator became impossibly difficult. He was still required to specifically allocate system resources to each job in the mix, but now he had to handle as many as five to ten jobs simultaneously in the system. He was also expected to see to it that regardless of the pattern of operation of each job the system resources devoted to each partition (particularly computer time and memory space) were efficiently used. This proved humanly impossible in most cases, and the computer operator's job became a very difficult and unhappy one.

Fortunately for the operator, demand was slowly growing for systems which would project the resources of the computer directly to the user. Beginning with computational time-sharing and evolving into production areas, modern systems provided the user with a computer that would respond when needed, both to compile new programs and to execute runs of programs in the library. These interactive concepts have by now spread over the entire spectrum of data processing. Remote data entry, remote batch processing, conversational program development, transaction processing and (in general) the association of integrated data bases with the organization's employees at every level have achieved general acceptance. A diversity of developments have been needed to make this possible:

- Virtual system management techniques have proved to be desirable. First, virtual memory management evolved to allocate the storage resource in a dynamic environment. Second, to a still controversial degree multiple operating environments will be provided through the use of virtual monitor techniques. Most recently came the concept of the virtual file, first in MULTICS and now in the IBM 3850, with more to come. The virtual file may prove to be at least as important as any of the other concepts of virtual operation, because it enables the system to provide data sets, program libraries and files dynamically in response to user demand without operator intervention.
- Data base management software is becoming mature in the sense of being reliable, versatile and easy to use in an on-line environment of multiple users. Particularly important is the development of user query languages, which not only permit users to inquire about the contents of files but increasingly permit them to develop and execute new procedures involving manipulation of file data.
- Communication processing subsystems including network definition languages, versatile access methods, and economical controllers have evolved to a degree that permits economical use of communications networks by users without specialized knowledge.
- A special case of interactive processing with particular implications for local operation is the self-diagnostic and remote maintenance capability being evolved by some of the manufacturers. Designed to reduce maintenance costs while providing increased systems availability to the user, remote maintenance will further reduce the need for people at the computer's location.

All of these developments reduce—sometimes nearly eliminate—the need for operators in the computer room. Comparable developments are occurring in small systems, used both in association with large machines in networks and independently by small users. The dominant trend is toward the development of small, easy to use interactive systems typified by the new Honeywell Series 60/61 and the NCR SPIRIT. In these systems all processing takes place interactively. (Few small users were ever very interested in batch processing computers, because their small volumes of transactions had never made batch processing necessary. Using file cabinets full of
ledger cards, accounting machines and telephones, the small users already had the full advantages of interactive processing—even though semimanual—and saw no need to put up with the time delays of the batch approach.) These new small systems, taking advantage of the low transaction volumes which enable the dedication of the whole system to one transaction at the time, are also causing the user to become the operator. The order entry clerk is automatically causing the machine to execute procedures to process each order as it is entered. Outputs are delivered to the receiving dock, the storeroom, or the production line. There are few changes of programs or disk packs, since the system is dedicated most of the time. Just as with the large systems, the need for an individual in the machine room to set up and execute each application program disappears.

The intelligent terminal represents a parallel development. An intelligent terminal is usually used in a hierarchical system where a central data base exists, performing some processing of the individual transaction, sometimes using local files applying only to the location where the terminal exists. Processing performed by intelligent terminals has also obviously left the control of the central system operator. He has no idea how much processing is taking place in the intelligent terminals of the system, nor any control over who does it, when, or in what manner.

**IMPLICATIONS**

In some respects these trends should be a cause of considerable concern. No single individual or group will have knowledge at the time of what the system is doing or for whom. If any failure, fraud, confusion or loss of data occurs, it will be far more difficult than in the past to identify the cause of the problem and to rectify it. The manufacturers face a formidable responsibility of providing fool-proof software which cannot be caused to fail by any combination of permissible actions on the part of the users, and which also provides all of the necessary audit trails, journals and control reports needed for the user to detect misuse and be able to recover from failure. The software must also provide responsive service to the user and run the system reasonably efficiently; experience shows that these objectives are not easily met either.

The user also has formidable challenges facing him. He must be able to identify which users consumed what resources, where bottle-necks in the system are occurring, and where any bugs in either the applications or the system itself are cropping up. In the old batch environment these monitoring functions were not too difficult (though in multi-programmed systems considerable complexities arose). In the new world of direct interaction with the user the system must monitor itself, which is easier said than done.

Perhaps more critical than the monitoring problems are those of data quality control, protection against system misuse and provision of adequate security and privacy of data. Even though the contents of the data base are being projected to many users in many locations, the user must somehow find a way to be sure that restrictions on access to the data are enforced and that accounting controls and protections against fraud are effective in all circumstances. In the past, with an orderly flow of batched transactions through manual check-points and with all files contained in the computer room, it was not difficult to provide these satisfactory controls. No comparably effective general methods have been developed for interactive systems; the user is challenged to develop specific methods which will be adequate in his individual case.

Presuming that the manufacturer and the user can deal adequately with these problems (and most manufacturers and users alike think that they can), the roles of the people as they interact with the systems will be changed in interesting ways.

The user becomes the operator of the system, in the old sense that he causes a run to be performed at a particular time requiring a particular set of resources. Except for the remaining periodic batch runs (such as the production of payrolls and month-end accounting reports) the remainder of the system's operations will be initiated by the user without the assistance or intervention of any central operating personnel. To a considerable degree he also becomes his own programmer; such tools as query languages merge the programming and command languages indistinguishably. The user finds it easy to prepare most reports and local, small programming runs; the application programmer loses his role as a provider of fast response special reports and small runs. However, the large batch jobs of the organization continue to be implemented by specialist application programmers; a large number of them are still needed.

The computer operator is affected more than the application programmer. His primary responsibility, used to be the application of system resources to every individual job; this role has disappeared altogether. To the degree that local batch runs remain the system still needs help in the mounting of forms on printers (until such time as all output is delivered remotely), in mounting tapes (until automated tape library machines come into wider use) and in performing similar supporting functions. Also, the operator will probably have to assist the machine in instances of conflict between jobs of equal priority, in resource allocation or schedule conflicts that are beyond the machine's ability to resolve for itself. The operator becomes, in a sense, the foreman who augments the intelligence of a group of cooperating robots. The operator also remains responsible for the master schedule: informing the machine of the necessity for performing regularly scheduled batch runs, making sure that the primary operating environments are available to users during the scheduled hours, and the like.

Finally, and perhaps most important, the operator becomes the custodian of the information resource. He must see to it that appropriate analytic studies are performed
regularly, for purposes of accounting for usage and of tracking developing maintenance and bottle-neck problems. He must also (above all) see to it that the control and back-up procedures are adequately followed, that no illegitimate access to data is occurring via routes of access which he can monitor, that the appropriate journals of transactions and file images are maintained and safeguarded, and that in general all control procedures which require regular execution are followed. In this sense the operator has a more important job than he has ever had before, and the job is considerably upgraded from the applications-oriented one of previous generations. The operator becomes the overseer of the performance of the information systems resource; he becomes the conductor of the orchestra in which each user plays his preferred instrument.