experimentation to be done and all accumulated data had been transferred to disk storage. When the last laboratory signed off, all of the data which had been collected from each laboratory and stored on disk was converted to a decimal format and delivered in the form of punched cards. This latter operation is a nonprocess application as contrasted to the data acquisition and control function. Our original system did not allow for time sharing between process control functions and nonprocess functions because of the lack of core. Thus the first laboratory to sign off could not get its data until the machine had been dedicated to nonprocess activities. Moreover, the lack of core severely restricted the number of laboratories that could be serviced by the system. It will be recalled that the operating system required 5,500 words of core. With an 8K system, this left only 2,500 words to be distributed among user programs. Thus, when a core load was developed, it could not exceed 2,500 words. Any number of laboratories could be serviced, providing the servicing programs for these laboratories could all be contained in a 2,500-word core load. By servicing, we are referring only to the data acquisition and control function. No core was available for data retrieval, new program assembly, or other nonprocess applications. By advance scheduling and building of core loads with different laboratory routines, we were able to function, although somewhat inefficiently.

However, the recent expansion of our system from an 8K to a 16K core will enable us to use another method of operating. Core is divided into basically two regions: one area is devoted to the system skeleton which includes the operating system; the second is called variable core. The skeleton area will contain the basic system routines—such as skeleton I/O, MIC, ITC, etc. User subroutines will also be added to the system skeleton by a program, similar to the Core Load Builder, called the Skeleton Builder. By including all laboratory control programs in the skeleton rather than in the variable core, as previously done, the variable core may be used exclusively for nonprocess core loads—the assembler, core load builder, data retrieval routines, statistical analysis programs, etc. The nonprocess activities will then be carried on without disturbing the on-line data acquisition programs that service the various laboratories. Since the data acquisition programs to service the various laboratories vary between 400 and 1,000 words, and 5,000 words of core are available in the skeleton section of core for user programs, it can be seen that a fairly large number of laboratories, each containing a number of devices can be included in the skeleton, leaving a considerable amount of core in the variable core section for nonprocessing applications. Still additional laboratories can be added to the variable core section, provided their servicing requirements can be kept within the time it takes to switch the nonprocess programs out of core and replace them with process programs.

In short, with a 16K-word centralized data-acquisition system, most, if not all, of the departmental laboratories can be serviced with the various users functionally unaware of the fact that they are, in fact, sharing the resources of the system. Finally, since 16K is only one-half of the 1800 machine's modular growth capacity (32K is maximum capacity), there is still room for additional expansion. This can be achieved at nominal cost, since additional core can be installed in 8K blocks.

REFERENCES

IBM 1800 Data Acquisition and Control System, functional characteristics. Order No. GA26-5918-6. IBM Systems Development Division, San Jose, California, 1966.

IBM 1800 Time Sharing Executive System, concepts and techniques. Order No. GC26-3703-1. IBM Systems Development Division, San Jose, California, 1970.

UTTAL, W. R. *Real time computers: Technique and applications in the psychological sciences.* New York: Harper & Row, 1967.

WEISS, B., & LATIES, V. G. Reinforcement schedule generated by an on-line digital computer. Science, 1965, 148, 658-661.

Computer-managed instruction: IBM System 360

JOHN A. CONNOLLY and THOMAS F. LAMBERT
American Institutes for Research
8555 Sixteenth Street, Silver Spring, Maryland 20910

Managing the educational process by means of an on-line computer system is a relatively recent application of computers to education. It is only within the last few years that large-scale instructional systems have been developed and managed from a computer base (Baker, 1971). The potential for on-line computer management of these instructional systems has not yet been fully explored.

The Instructional Management Program (IMP) is an on-line computer-management system. Its educational approach is based on principles derived from the psychology of individual differences, organized by system design techniques and implemented by means of computer technology. The educational and psychological aspects of the system have been described elsewhere (Connolly, 1970). The present report focuses on the computer-management features of the model.

IMP was developed by the American Institutes for Research for the School District of Philadelphia under a Title III grant. It is not related to any other learning program, such as Project PLAN or IPI. The system has been implemented on an IBM 360 computer, which is presently available at the School District headquarters. IBM was not responsible for the development of the program, and other general-purpose computer systems may be equally appropriate.

AN OVERVIEW OF THE SYSTEM

IMP is a computer-based system of individualized instruction designed to assure that all children master all of the basics (e.g., reading, writing, arithmetic). The prototype was developed and tested in a poverty-area school in Philadelphia, where it is now operating. The system is currently being disseminated to five additional schools in Philadelphia.

The model consists of three basic components. First is the evaluation system, which involves four different kinds of tests. Terminal measures indicate mastery or deficiency with respect to 70 different educational objectives. Diagnostic tests assess prior knowledge of the instructional subtopics available in the learning sequence for each objective. Aptitude tests measure learning strengths and weaknesses in terms of reading, verbal aptitude, learning style (i.e., visual, auditory, or kinesthetic), and cognitive style (i.e., abstract or concrete). Progress tests indicate knowledge of the material just presented in each learning unit. Hundreds of different scores on each student are derived from these tests and entered into a pupil data bank.

The second major system component is the curriculum bank. The bank consists of coded data on hundreds of curriculum packets. The material in each packet covers one topic in the path leading to mastery of the related objective, and the packets are sequenced accordingly. Packet variations have been developed in order to teach the same instructional topic in alternative ways, e.g., at different reading levels or using different modes of presentation. All packets are coded according to objective and topic, as well as the reading level, difficulty level, teaching mode, and cognitive method most appropriate to it.

The computer-management system is the third major component. The computer's primary purpose is to prescribe a learning path for each student corresponding to his individual needs by matching the student's assessment file to the curriculum catalog. This kind of computer-managed instruction (CMI) involves an on-line prescription for an off-line learning activity. It should be distinguished from the continuous on-line learning which is the essence of computer-assisted instruction (CAI).

THE ROLE OF THE COMPUTER

One way of describing the system is from the standpoint of the computer's functions. The computer system was relatively easy to develop once the educational process was organized in terms of its objectives, measures, and curriculum units. The original version of this program was operated manually, and even today it can function without a computer in a simplified form. Yet, the computer does add a significant dimension to the approach. In particular, real-time processing of data concerning students and curriculum materials allows truly immediate and highly appropriate scheduling of student learning activities.

Computer System Configuration

Computer system overview. The instructional program presently operates on an IBM System/360, Model 40 computer system with CPU storage capacity of 196K. The program was originally designed for a Model 30 and used 14K of the available 64K storage capacity. The computer system was upgraded to a Model 40 partly to allow expansion of the system to additional schools.

Remote terminal. A remote terminal (IBM 2740-2) was installed in the school where the program is currently operating. The terminal is in direct communication during school hours with the central processing unit located approximately 5 miles away. A teletypewriter device was chosen since it provides adequate flexibility at the lowest cost. Typed messages are stored in a buffer prior to transmission, thereby freeing the communication system during the typing operation.

Communication and control. Information is transmitted to and from the terminal over common-carrier leased telephone lines at the rate of 14.8 characters per second. A Bell modem is located at each end of the line for data conversion. Communication with the central processing unit is through an IBM 2701 transmission control unit which operates in a half-duplex mode.

Central processing unit. The CPU's core storage is partitioned so that a number of processes can run at one time. A partition dedicated to the IMP system handles all incoming and outgoing transmissions and does some of the actual processing of the data. At present, approximately 8K is used for line-handling routines and 6K for an application programs supervisor. These programs reside in the first foreground partition, which has top priority for the use of the CPU. Consequently, response time on transmissions is at a real-time level, with delay at the terminal of less than 2 sec.

Data storage. All files supporting the system are stored on a disk pack of an IBM 2314 direct-access storage facility. This unit has four drives operating in one unit, and data can be recorded on 18 surfaces on each of the disk packs. An average access time of 75 msec and a high-speed data-transmission rate allow operation with relatively little dedicated core storage and make possible a large variety of real-time functions in the system.

Other devices. A number of peripheral devices are used in the batch-processing mode in support of the real-time applications.

Software Support

Operating-system programs. The computer system is currently under control of Version 25 of the IBM Disk Operating System. This system is contained in a 16K partition in core and provides supervisory functions for the entire system. In addition to the operating system, the actual teleprocessing routines (written in 360/Assembler language) provide access to data at the BTAM (Basic Teleprocessing Access Method) level. These line-handling routines continuously poll the terminal to determine if a transaction is ready to be transmitted. When a transaction is received, the program edits and reformats it and passes the message on to the core resident application program for processing.

Core resident processing program. This program involves a set of modules which supervise the real-time processing of data. Only the basic executive program and buffers for data and logic modules are permanently located in core storage. The modules that act on the data are called into main storage as they are required, along with the appropriate data record. When the transmission has been processed, the updated record is returned to secondary storage on the disk pack and the next required module is read in. The core requirements for the real-time processing are limited to 6K by means of this overlaid use of main storage. All of the program logic modules are catalogued on disk for fast retrieval, and all of the files of student and curriculum information are organized in an indexed sequential fashion for direct access to pertinent records.

Stand-alone programs. There are a number of supportive programs designed to operate in a batched mode in the background partition. These programs, written in COBOL, are designed to accomplish tasks that are either impossible or unnecessary to perform on a real-time basis. All of these programs are catalogued as object modules on disk and are called into use with a minimal amount of operator intervention.

Computer Operations

The computer's functions in managing the instructional program are divided into three major categories—remote batch processing, real-time processing, and stand-alone batch processing in two phases. The following discussion shows how the various batch and real-time processing operations come into play as students move through the program.

Stand-alone batch processing—Phase I. The instructional management process starts with the optical scanning and machine scoring of terminal measures and aptitude tests. A student record is created from this test information and stored in a direct-access file. The coded data on curriculum units is used to create a curriculum bank, which is also stored on disk.

Summary reports of these data provide information that is important to the instructional management staff and to the teaching staff as a whole. The teaching staff receives a list of all student deficiencies on the terminal measures as well as an indication of the learning aptitudes of the children. This information focuses the teacher's attention on problems in the student's learning progress. The instructional management staff receives a report showing areas where new curriculum materials are needed. These processes are all run as stand-alone programs on a batch basis, either cyclically or on demand.

The final batch run in this phase starts the actual instructional process. The student's deficiencies are arranged by the system according to a predetermined hierarchy of importance. A list of the top 10 deficiencies in priority order is prepared for each student to permit selection of two objectives for initial concentration. This decision is made in a conference between the student and the instructional staff.

Real-time interactive processing. The real-time characteristics of the system now come into play as the students interact directly with the computer. The student goes to the terminal and enters the diagnostic test results for the selected objective and indicates that he would like the first assignment. The computer adds the diagnostic data to the student record and accesses the curriculum file for the chosen objective. The actual assignment is then made by means of a complex matching algorithm which selects a learning unit representing the best fit between the student's learning aptitudes and the available curriculum treatments. The matching process is described in detail elsewhere (Connolly, 1971). The total updating and assignment procedure takes less than 2 min.

When the student completes his current assignment, he takes a brief progress test and returns to the terminal to indicate his success or failure. In the fail case, the computer implements a decision made by the instructional staff concerning the appropriate remedial action. The system could, for example, assign another unit on the same instructional topic with a different learning style, move to the next topic in the instructional sequence, or begin assignments for another objective. If the student has successfully passed the progress test, the computer will search the next learning topic in the instructional sequence for the best assignment. This cycle is repeated throughout the student's learning path, with the system tracking the student's progress, branching to remedial paths when needed, and adapting instructional assignments to the student's learning abilities. An interactive language leads the student through the process so that he is not required to memorize any transaction formats or complex data codes.

Other real-time applications are less dramatic but equally important. New information about the student can be posted in his records and used immediately in determining his next learning assignment. Also, the instructional staff can request data on student performance for immediate display at the terminal.

Daily remote batch processing. Remote batch processing is used for certain updating and display functions that cannot be accomplished on a real-time basis. The system takes information from the terminal on a real-time basis and stores it on a direct-access device for overnight processing. Every morning, before the opening of school, the data stored from the previous day are sorted and sequentially run against a series of programs in the background partition.

A variety of applications are processed in this way. New students or curriculum units are added to the file, curriculum sequences are rearranged or deleted, student and curriculum records are updated, and new curriculum listings or student histories are obtained. These listings are printed on a high-speed 1403 printer and returned to the school by messenger. It is in this same operating environment that the usual housekeeping, security, and file back-up functions occur.

Batch processing—Phase II. This batch-processing phase essentially involves the preparation of reports from the mass of data generated in the system. All information transmitted from the terminal is stored cumulatively on a log file. Summary reports are prepared by combining these data with the information contained in the student and curriculum banks. Student reports indicate number of units passed, number failed, number of objectives completed, number of units still active, etc. Curriculum reports show number of times the unit was failed, passed, skipped because of a diagnostic test, etc.

The most significant use for these data is in the area of research and evaluation. Statistical treatment of these data can provide insight into many important aspects of the program, such as the effectiveness of various aptitude treatments and learning sequences, and the relationship between the degree of successful curriculum matching and learning performance. The possibilities along these lines are exciting and not yet fully implemented.

Future Development Possibilities

Expansion to additional schools. The system described above was installed as a prototype in one school. Steps are now being taken to expand the communication network to include five additional schools at various locations throughout the Philadelphia School District. Except for some additional disk storage space, no further hardware expansion is required at the central computer facility.

A new IBM software package will be installed to handle the teleprocessing demands of a number of additional schools on a real-time basis. The newly released CICS (Customer Information Control System) will replace the BTAM routines. The outstanding features of this new teleprocessing system will be its relative independence with regard to the type of remote terminals and its efficient queuing and processing of messages. Each of the schools involved in the system will have one or more remote terminals, all connected to one communications-line network. All terminals will be in direct contact with the central processing unit and will operate independently. No delay in computer reaction time is anticipated with the addition of these terminals.

Upgrading of hardware. As more and more schools are added to the network, increasing the computing power becomes an important consideration. More efficient remote terminals, with a variety of input and output devices, might be appropriate. A more sophisticated time-sharing system or a totally dedicated computer may be required.

Cybernetic features. A base now exists for the development of a much more sophisticated computer-management system. The system now operates on relatively fixed parameters. Self-adjusting rules could be developed which would permit a dynamic system of cybernetic feedback. Information about student performance might be systematically examined by the computer and the system made self-correcting.

Research and evaluation. The research possibilities inherent in the system should be fully explored. As more students use the process for longer periods of time, an immense amount of empirical information will be collected. Thse data may provide useful clues to the nature of the learning process.

PROBLEMS AND PROSPECTS

There are many serious problems in designing and implementing a computer-managed instructional system. Computer costs are a formidable problem; structuring the educational process is a mammoth undertaking; accommodating to the educational bureaucracy is an endless frustration. Perhaps the most fundamental problem, however, is our limited knowledge of learning psychology.

Many basic questions about the learner, the instructional process, and the learning environment are still under study. Which student aptitudes, if any, interact significantly with learning treatments? Is there an optimal instructional sequence which allows maximum transfer of training from one instructional unit to another? Which factors in the learning environment enhance student learning? The design of instructional systems is often based on assumptions about the learning process rather than on definitive research.

On-line computer systems cannot operate at full potential without more definitive research on the learning process. Batch computer processing is the most economical way of performing most of the management functions in an instructional system. Real-time processing becomes crucial when continuous progress data are used in determining the next learning assignment. These cybernetic adjustments require a thorough understanding of developmental stages in the learning process. In the end, psychological research may dictate the future of on-line computer systems.

REFERENCES

BAKER, F. B. Computer based instructional management systems: A first look. Review of Educational Research, February 1971.

CONNOLLY, J. A. A computer-based instructional management system: The Conwell approach. Interim report. Silver Spring, Maryland: American Institutes for Research, 1970.

CONNOLLY, J. A. Using computers to individualize instruction: Another approach. Computers & Automation, March 1971.