Input/Output Considerations for Large Data Bases†

HERMAN SKOLNIK* and JOHN C. SNYDER

Hercules Incorporated, Research Center, Wilmington, Delaware 19899

Received May 23, 1974

Input/output devices, format designs, and programming for rejecting defective input, updating and correcting input, and for producing a variety of outputs from the input are discussed, especially from the viewpoint of a large data base and that of the information needs of the users.

This Conference has defined a large data base as one which involves the processing of 100 million characters or more within a year. Quite a few data bases now process this much information and, when combined with the input of previous years, some may approach or even exceed a billion characters.

The magnitude of 100 million characters is difficult to comprehend unless converted to more familiar terms. Many books, such as a novel or text, average 300 pages of about 400 words each, for a total of 120,000 words or 840,000 characters. It takes 119 such books to approach the lower level of our definition of a large data base.

If we read at a rate of 400 words per minute, it would take us 595 hours to read the 119 books which comprise the lower limit of our defined large data base. Put another way, it would take us 15 40-hour weeks, about a third of a man-year. But the purpose of a large data base is not to give the user 100 million characters to read. On the contrary, the purpose is to eliminate the user's need to read anything except that which is essential to his purposes. The name of the game vis-à-vis the user is *information for use.* Consequently, the computerized information system must be designed to select information relevant to the user and present it in a highly readable format which is compatible with the user's thought processes.

OBJECTIVES OF THE DATA BASE

Output is the ultimate objective of any data base. Most importantly the output must be defined in terms of the needs of the users. These needs tend to multiply as the size and scope of a data base increases.

There is a large variety of types of output. Until the recent past, most data bases were conceived as massive input of lines to be sorted and merged for output with the high-speed printer operating essentially as a Linotype machine. An example of this type of output is the telephone book in which updating is frequent and extensive.

More recently, data bases have been conceived for multiple and selective outputs from a single input, both with and without updating.[3] In these more sophisticated data bases, programming is concerned with display compositions from stored information, so that a variety of outputs are produced which differ in the kind, amount, and arrangement of information in accordance with how it is to be used.

Another output mode of large data bases consists of direct interaction, usually but not necessarily from a terminal, using a screening and selection process provided for the user from a need profile, or developed by him as he interacts with the data base. At present most of the output by this mode consists of reproduction of selected records or selected parts of selected records from the data base.

Input records of many data bases contain a certain amount of similar identifying information, *viz.*, the title, author, reference, and an abstract. Many data bases make use of computer processing of titles, abstracts, or text to provide keyword access to the documents in the data base. These machine-derived keywords are stored as an index to the master file itself. This type of data base eliminates the activities of an information scientist to interact with the original body of information.

Quite at variance is the mode which uses information specialists to interact with the original body of information to provide a subject index by which selection from the data base will be made. The advantage of this mode is the extended value judgment of the direct and potential usefulness to a wider scope of users than can be provided by the words an author used in his title or abstract, or even in the text itself.

INPUT DEVICES

Until relatively recently, keypunching was the primary method for delivering input. Keypunching is rather slow and fallible, particularly in comparison with other computer components. As a result of considerable activity, the keypunch has been replaced extensively by the key-to-tape or key-to-disk terminal, and in some data base operations by optical (or magnetic) character recognition devices. Tab cards generally have been limited to 80 characters, are difficult to locate for correction, and are subject to somewhat lower productivity than typing for most input with predominantly alphabetic and nontabular data.[3]

Considering that our large data base requires the input of 100 million characters, and that this operation is one of the more expensive steps in the process, it is apparent why typing has largely displaced keypunching. Typing at the rate of 50 words per minute, typed input would require 119 40-hour weeks for 100 million characters, or over two man-years. In our experience, keypunching would require about twice as much time (due inherently to the slower functioning of the punch), or well over four manyears. Furthermore, such input is confined to all caps and is restricted to the limited personnel with superior keypunching skills, as compared to the wider availability of skilled typists.[4]

We do not mean to imply, however, that there is no place for keypunching. Some data bases, particularly those which are heavily numerical and highly formatted, and require considerable machine duplication from record to record, are still best input by keypunching of cards.

We are particularly puzzled by the absence of direct key-to-tape or -disk devices which incorporate the keypunch numeric input design. We would also like to have two other

† Hercules Research Center Contribution Number 1641. Presented in the "Conference on Large Data Bases," sponsored by the NAS/NRC Committee on Chemical Information, National Academy of Sciences, Washington, D.C., May 22–23, 1974.

convenient features of the keypunch: ease of programmed tabulation at several input levels and ease of duplicating portions of input.

Initially, when typewriter terminals were introduced they were restricted to on-line input. Computer downtime frequently impeded operations, and failures sometimes required extensive reentry of input. Furthermore, on-line input requires constant use of communications lines and computer time. On-line input at a typist's speed of about six characters per second (about 50 words per minute) is a rather inefficient use of such facilities. These disadvantages were eliminated with the advent of off-line typewriters with on-site storage capacity, such as magnetic cards (for about one page of input) or magnetic tape (for many pages of input), which could then be placed on-line later for transmission at rates of two or three or more times faster than by straight typing.

Ease of correcting during input or after it has been completed, whether a character, word, or line, as well as rearrangement or deletion of text, is a major advantage of typewriter input. Upper- and lower-case input yields an output of markedly higher readability than of all caps and generally makes proofreading easier for the technical people responsible for the data base. These two factors alone have appreciably reduced input errors for our data base operations.

Whereas all input typewriters allow character, word, and line corrections as each line is input, extensive correction procedures are available only in conjunction with some sophisticated software system, such as that associated with the IBM Administrative Terminal System (ATS).[4]

Numerous typewriter terminals are on the market today, so many that it is difficult to evaluate the advantages and disadvantages of each relative to costs and options. Most are designed, however, for only on-line operation, and only a few are designed for off-line operation with storage capacity for an appreciable amount of input. Direct coupling with and control of on-site minicomputers with editing and correcting capabilities for eventual transmission from a remote site to a master computer are destined to be developed.

INPUT DESIGN

The larger the data base the more the investment that can and should be made in input software to maximize the operations that are to be done by the computer, rather than by human effort. In the production of any information system, even those based on the simplest types of documents, a certain amount of editing must take place during the input process. Especially with large data bases, where we must minimize costs and time, the flow of information from the input document should be natural for and require a minimum amount of editing and tabulating by input personnel. The operator should be required to spend essentially no time scanning the input document for information, and no time rearranging or right- or left-justifying items. These operations tend to introduce errors. The computer should be used for these functions.

While some consideration needs to be given, when designing input systems, to updating requirements, error corrections, and the like, the primary thrust should be to transfer the information with a minimum of effort and strain.

For some data bases the primary output required is of tabular design, and the input itself, particularly for subsequent updating, may have the same format. In this case, the input system might well use the output format, but input in the output format tends to increase input typing or keypunching time and complexity, and to increase the complexity of the programs that have to be written to extract the information needed for a variety of outputs with different formats.

In most cases, particularly when a wide variety of outputs will utilize the information, it is best to design input, especially typed input, with an informational item per line format.[3] This method is the least complex for input typists and can be typed the fastest with a minimum probability of error. It is the easiest for proofreading and correcting input. A unique advantage is the ease with which the items can be arranged in conformity with the flow of information in the documents being processed. Information processed in an harmonious flow utilizes most productively the time of the information scientist extracting information from documents, the input typist, and the programmer.

Each item of this type of input record is preferably associated with an identifier, such as

a = author item
t = title item
r = reference item
x = abstract item
s = subject item

(In this connection, the use of alphabetic identifiers has proven to be superior to numeric ones.)

These item identifiers permit the input programmer to examine, by computer conducted census, the overall nature of the information being input and to establish the nature of the record he will use to contain it and the bounds he must place upon information scientists and input personnel for their information to be processed without error or program failure. When input was designed for tab cards, there were compelling reasons to minimize the number of cards to be handled: reduced card costs, computer input reads, card punching time, and the shear volume to be handled. This tendency, as well as the 80 columns available, frequently limited input operations unduly by restricting and structuring the input format well before the characteristics of the data base were fully realized. Frequent costly reformatting of input design resulted.

With item-by-item input, especially with typed input, these characteristics can be examined by computer, and not only can the record be designed and scaled, but acceptance standards can be more easily established for processing of the record.

An absolute necessity for input to large data bases is computer screening of the record being input for processability by whatever program it will be processed eventually. Missing items, over-sized items, data out of range, and the like can be computer checked at input time and the record rejected; a computer-generated diagnostic provides guidance for correction and resubmission of the item. During the early stages of development of a new information system, the programmer needs to be keenly aware of errors that might be introduced, and he should be encouraged to be resourceful in developing checks which will eventually catch any unprocessable input as early as possible.

The programmer also should provide for right- and left-justifying of input items, extraction of information that must be relegated to fixed fields for identification and sorting purposes, the creation of compound fields containing a number of informational items to be output by several programs, the coding of information for selection purposes, and the compression of information to save storage and computer space.

Records are generally input into a data base in random order. A variety of selection and sorting processes eventually will be used to bring the records into desired order or to compartmentalize the records. We code records at input time according to areas of interest, such as research programs, disciplines of science, or fields of technology relevant to our user groups. The coding mechanism not only

reduces computer costs, but allows us to package our various outputs from each data base in terms of maximum benefit to the users.

Coding is a mechanism that is operable most effectively for data bases in which documents are processed for input by information scientists. It takes a human brain to react with the contents of a document to establish its true relevance to a user group. As a document is examined for deciding whether or not it belongs in a data base, it is then indexed and abstracted from the point of view of the information needs of the users. It is a simple matter for a knowledgeable information scientist to code the record for output in as many ways as the users may require. These may include many different groups, such as analytical, organic, physical, and polymer chemists. Even within one group, for example, polymer chemists, interests vary over a spectrum, such as preparations of specific and generic polymers, specific and related properties, and uses and applications. It is a disservice to give a user undesired along with relevant information. When codes are properly assigned, output for each group of user can be tailored to be most highly relevant.

We relegate one line in the item-per-line format for coding. Codes may consist of one or more characters and are assigned specific locations in the record for processing purposes. They increase record size insignificantly and provide a net savings in computer operations.

Selective dissemination of information (SDI) has enjoyed a high level of acceptance in recent years. In our experience, SDI by means of keywords in titles, even when expanded with keywords in abstracts, yields products of varying quality. Rarely does the user have assurance that he has the best information, and rarely is he happy with the massive doses of trivial and nonrelevant information that leads to needless reading. Consequently, we introduced the Multiterm concept,[1,2] an indexing system that relates and associates things and actions by a minimum number of subject words. In its simplest form, a Multiterm is a combination of correlatable subject terms, such as

C/R/P/A//

in which C may be a chemical prepared from reactant R by process P using catalyst A. The virgule or oblique stroke following each term is easily programmed to wrap-around so that each term in turn is first in the rearranged Multiterm, as follows:

R/P/A//C/
P/A//C/R/
A//C/R/P/

The double virgule is used in the Multiterm to indicate the end of the initial Multiterm. It informs the reader of the order or precedence chosen by the indexer.

The Multiterm concept is economical of the indexer's time and gives the user of the index as much information as is given in many abstracts. It is considerably more informative than titles of papers, and furthermore the subject terms are chosen from the user's point of view, whereas titles of papers reflect only the author's point of view. Multiterms, in general, are shorter than titles, thus reducing the area to be searched by either the computer or user. Even a printout display is smaller with Multiterms, as the number of subject terms is less than the number of keywords in the average title.

OUTPUT DEVICES

Many sophisticated large data bases are still wedded to output devices that yield products of poor readability. The whole field of computer output devices is in a state of flux with many new systems being developed and marketed for the first time only recently. Nevertheless it is still generally true that the dependability and readability of output decreases as the speed of printing increases.

The best products are those associated with photo-offset printing. A rather good product is produced with on-line typewriter terminals, but the output rate is rather low, between 100 and 150 words per minute, restricting output to a relatively few pages. High-speed printers, producing well over 1000 and even 4000 lines per minute, leave much to be desired. For optimum output they require considerably more attention than is generally provided routinely. This is particularly so when masters for Multilith or photo-offset reproduction are desired. Then centering, inking, and printer chain or drum adjustment require special attention. The expanded character set of upper/lower case printing devices provides a more naturally readable output. However, the more characters available of necessity slow down the printing process from a third to a half of the speed of all-cap devices.

Now that COM output is available with upper/lower case characters, and assuming that users will accept microfilm, this may be the medium of preference for massive outputs in the future. High readability coupled with high speed of output, however, is a goal yet to be achieved.

The cathode ray tube, as such, or in conjunction with moderate speed printing devices, has made quite an impact recently for delivery of output from large data bases to remote terminals. This trend will certainly continue for on-line conversational mode operation with large data bases for which the essential operation is screening and selection of whole records or selected parts of records in relatively fixed output format.

OUTPUT DESIGN

We have pointed out that output is the objective of an information system. It is the interface with the user and should enable him to receive the information as nearly as possible in the order in which he needs it, with clearly expressed ideas, and in a layout that allows him to receive or reject information as he proceeds through it.

To this end there are a number of generalized requirements for good output, which essentially involves the composition of the output (in most cases a page, or a part of a page).

First the document needs proper identification, usually with a suitable title page. Individual pages similarly should be identified, preferably with a running page title, and with pages numbered. In a great many cases, subtitles within a page are highly desirable from the reader's standpoint. Consideration should be given to providing content information at the bottom of the pages as well, particularly if the document is to be bound with the typical flip-over binding customary with computer printouts. If the output is voluminous, such added information is a necessity.

Computer-produced compendia are much more usable to the reader if they contain adequate subheadings so that information can be found without strain. Such titling and subtitling are readily provided to the output composition programs by (a) sorting, (b) providing the necessary classification contained with each record, (c) retaining information from record to record for comparison, and (d) taking appropriate action (printing heading or subheading) when changes are encountered. These procedures involve the retention of information from page to page and displaying proper continuations at the bottom of a page, and even at the top of the next page for certain types of displays.

It is particularly desirable not to split certain parts of a record between pages, nor is it desirable to begin an item too near the end of a page. These requirements are charac-

teristic of the particular output being processed and must be established for the advantage of the user. In any event, the layout of each item in the page being produced is greatly aided by passing to the composition program whatever description of the record is necessary for proper layout. Such information is best obtained and added to the input record by the input processing. For example, to lay out the item in the space remaining on a page, the number of lines of each information type which is subject to variation is required, such as lines involved in the title, number of author lines, reference lines, abstract lines, etc. Not only can the total length be computed quickly at output time without testing the content of each line, but also discrete directions can be provided to the output program on how to proceed to output the information from each record area.

It generally is more efficient to program a line count of the space remaining on a page than to call system routines for each line printed to obtain the line number.

In many instances it is advantageous to provide, at input time, certain prefabricated lines produced from individual items which must be kept separate for sorting and record selection purposes. When upper/lower case records are involved, it is also necessary to provide both all upper case fields for sorting and upper/lower case fields for printing. The translation procedure for converting to all upper case is provided preferably at input time, and is another example of output processing that begins at input time.

On reviewing some of the requirements for good output processing, it is apparent that the purpose of good input is to provide fully processable records for output. Thus input ends where rearrangement of records for output begins. Input provides the bridge between the input document and the output products. The outputs provide the bridge between the computerized information system and its user.

LITERATURE CITED

(1) Skolnik, H., "The Multiterm Index: A New Concept in Information Storage and Retrieval," *J. Chem. Doc.*, **10**, 81–84 (1970).

(2) Skolnik, H., and B. E. Clouser, "Designing an Information Awareness and Retrieval System for Chemical Propulsion Literature," *J. Chem. Doc.*, **11**, 39–43 (1971).

(3) Skolnik, H., "The What and How of Computers for Chemical Information Systems," *J. Chem. Doc.*, **11**, 185–189 (1971).

(4) Skolnik, H., and W. L. Jenkins, "Evaluation of the IBM Administrative Terminal System and Magnetic Tape Selectric Typewriter for Text Processing," *J. Chem. Doc.*, **11**, 170–173 (1971).

Character Sets†

DONALD F. RULE

Chemical Abstracts Service, Columbus, Ohio 43210

Received November 13, 1974

Historically, character sets have grown and developed with the growth of the "frame" or byte size of data processing equipment. Starting with the 5 level (bit) Baudot codes, the popular industry codes have progressed through the 6-bit BCD, the 7-bit ASCII, the 8-bit EBCDIC. In keeping with the *convenient* code size limit of 2^n where n is the byte size of the computer, I/O hardware offerings and programming languages have generally constrained themselves to available byte-size environments. Fortran uses a character set of 49 symbols; Cobol uses a set of 51 symbols; PL/1 uses 60 symbols; and IBM 360/370 Assembler uses 51. Impact line printers have typically progressed through offerings of 48, 60, 120, and 240 characters. CRT's have progressed through a series of offerings similar to impact printers.

The 8-bit version of ASCII plus the proposed code extension techniques for ASCII represent the most significant industrywide thrust toward information interchange that is not so dominated by the byte-size environment of the current generation of computers. These extension techniques provide a mechanism for defining "pages" of character sets and for declaring pages to be active. In a 7-bit environment, one page of 128 characters can be active at a given moment. In an 8-bit environment, two 128 character pages may be active at a given moment.

The character-set needs of many information systems involving large data bases exceed the byte size of today's industry codes (excluding extended ASCII). Character usage studies of the published literature in the disciplines of chemistry, biology, and math show that the character set size needed for (English-based) information processing is in the neighborhood of 1000 characters and growing slowly. It is unlikely that the needed set size for other disciplines such as medicine, law, and physics is much larger although the 1000 characters for each discipline might be somewhat different. These character-set sizes do not include typographical variations such as type style, type face, or point size. Such variations are useful in printed matter, but are not (should not be) part of the intended information content of machine-readable information data bases.

Input of large character sets on devices with a limited number of keys and discrete device generated output codes need not be a major problem if the system designer keeps two points in mind. First, most special symbols occur very infrequently. Second, many characters are simply shape or placement variations of some basic symbol. For example, A, a, $_a$, ^{a}, and *A* are all variations of a single basic symbol. With these points in mind, the system designer can develop input conventions which are simple and yet efficient in terms of average number of keystrokes per character. Software routines translate the input code of a particular device into the internal character code which should be constrained only by the byte environment and not by a particular input or output device.

Output of large character sets is getting easier. Photocomposers meet the printing need for large character sets. Some non-impact printers and a few CRT devices provide

†Summary of paper presented in the NAS/NRC "Conference on Large Data Bases," sponsored by the NAS/NRC Committee on Chemical Information, National Academy of Sciences, May 22–23, 1974.