Foster M. Palmer
Associate University Librarian
Harvard University Library
Cambridge, Massachusetts

A LIBRARIAN'S VIEW OF DATA PROCESSING

Choosing a title for this paper five months before it was to be given was something of a problem. Knowing I had the broad field of library applications of data processing to discuss added to the number of possibilities. "Testament," however, seemed too final; "confessions" seemed too lurid—finally I settled on "view." However, one's view may change, depending on the viewpoint, and, in fact, I will offer not a single view but three. First I will discuss the rather expansive experimental days of the MARC Project, then the problems of getting an actual daily production job on the road, and finally, in a different vein, I will present some ideas on possible future systems.

Since all three are really personal views, I should explain that the speaker is an old reference librarian, with considerable background in circulation. He knows the catalog fairly well, but as a consumer rather than a producer. He has no personal experience in library acquisitions work, although he has managed to acquire a rather large personal collection. Like most librarians, he is more a practitioner than a theoretician. His machine background came late, via circulation and punched cards; he got his start with computers at the University of Illinois not quite five years ago from Kern Dickman and Hillis Griffin. His first actual computer job was to prepare the initial systems design and programs for the Widener Library shelfist series (the early volumes, all in upper case).[1] In mid-1966 he was freed from departmental duties so that he could work full time in the field of library automation, starting with the Library of Congress MARC Pilot Project, in which Harvard was one of the sixteen selected participants.

It would be hard to overestimate the stimulus which the MARC Project provided to the library community in general and the Harvard University Library in particular. Conditions at Harvard were propitious. An 8K IBM 1401 with four tape drives and upper and lower case printing capability had

References

1. De Gennaro, Richard. "A Computer Produced Shelf List," *College & Research Libraries,* 26:311-315+, July 1965; Palmer, Foster M. "Conversion of Existing Records in Large Libraries, with Special Reference to the Widener Library Shelf List." *In* John Harrison and Peter Laslett, eds., *The Brasenose Conference on the Automation of Libraries. . . .* Chicago, Mansell, 1967, pp. 57-83.

2. Avram, Henriette D. *The MARC Pilot Project: Final Report on a Project Sponsored by the Council on Library Resources, Inc.* Washington, D.C., Library of Congress, 1968, pp. 103-111.

3. Palmer, Foster M. *Punched Card Circulation System for Widener Library, Harvard University.* Cambridge, Mass., Harvard University Library, 1965.

4. Reynolds, Carl H. "Notes on Estimating and Other Science Fiction," *Data Processing Magazine,* 9:44-45, June 1967.

5. Morehouse, Harold. "The Future of Telefacsimile in Libraries: Problems and Prospects," *Library Resources & Technical Services,* 13:42, Winter 1969.

recently been placed in the Library as a substation of the Harvard Computing Center. If in 1969 this seems like a small and old-fashioned piece of equipment, remember that the year was 1966. At that time the dominant theme in the news concerning third generation computers was slippage of software support. Even now I am not willing to concede that the 1401 was or is a poor thing, and it was indeed our own–not fiscally, for we paid by the hour, or, to be more exact, by the hundredth of an hour, but freely available for hands-on use with only mild competition even in prime time. The budget for machine usage was of course not unlimited, but a generous allowance was made for experimental use because the administration of the Library placed a high value on the long term potential worth of the MARC Project. I do not intend to repeat here the description of Harvard's participation which appears in the Library of Congress's published report.[2] Rather, I will try to convey what it means to operate in an experimental, developmental hands-on environment with a computer with which one feels at ease.

If one had an idea, it was easy to try it out on the spot. No sending of jobs to the computing center by messenger, waiting to get them back the next day, telephoning if they did not come, sometimes getting them back with cryptic messages indicating that they had not run but giving only the faintest of clues why, with the prospect of going through the whole cycle again and again as problems were solved one by one. With the computer right in the Library, when something went wrong (and of course it often did), the author of the program was right there to evaluate the problem. When the dread red light "process" appeared on the computer, sometimes a similar, although figurative light would be the response by the operator when the address at which the program hung up was compared with the program listing. If not, at least there had been an opportunity to observe the sequence of events leading to the crash. Had this tape moved? What had been written on that one? Would it be worthwhile to print out the contents of the computer memory? (I do not know how many of you have seen one of these arcane documents–a printout showing exactly what is in each position of the computer memory at a given time. Baffling to the novice, it soon becomes an extremely helpful aid in finding program "bugs.") Such a printout for the 1401 would cost only a matter of cents anyway. Contrast the situation with a closed shop and a large computer whose storage printout is inherently more expensive (because the much larger memory takes longer to print and the cost per minute of the larger machine is much higher). One has the option of specifying in advance that the printout be made whenever there is a problem, and perhaps over a period of time accumulating several of the expensive documents, only one of which may turn out actually to be useful, or skipping the option and failing to get the record of the memory the one time it would have saved hours of looking for a needle in a haystack. (Of course storage printouts can also be haystacks, but in a hands-on situation one at least has a better idea of when the printout is likely to be helpful or essential.)

With a machine in house, it was often possible to be back on it minutes after the problem was solved, rather than the next day. All this is not said to make a virtue of necessity and extol small machines as such, but to call attention to the advantages of close contact with the computer. For economic and administrative reasons such contact is not likely to be feasible with a monolithic large machine; whether time sharing with remote consoles really makes it possible to have the best of both worlds you will have to learn from someone else who has actual experience with it.

We anticipated, rightly, a great deal of nonce or ad hoc programming in conjunction with the MARC tapes, and to facilitate this prepared a whole family of macro instructions–another term that may require some explanation. Each computer has a built-in set of instructions. Using as we did a low level assembly language, normally each instruction written by the programmer was the equivalent of one machine instruction. However, it is possible to gather together a series of instructions that may be useful on some other occasion, give them a name, put them on the system tape used in assembling programs, and thereafter call them forth at will by the name given. If desired, parameters–values appropriate to the particular use of the moment–may be inserted. Such macros are an intermediate step in the direction of a higher level language such as COBOL or PL/I.

The term "family" was used advisedly in speaking of our group of thirty or so, since many of the macros referred to other macros, especially to a sort of master one called BEGIN. The description of macro capability in the handbook on the Autocoder system which we used for assembling programs specified that labels within macros in this language should begin with a right parenthesis. However, we discovered by experimentation that ordinary alphanumeric labels could also be used. This was an essential factor in the transition from a series of individual routines to an interlinked family or system. For example, by providing suitable areas named LNCTR (line counter) and PGNO (page number) in the master macro BEGIN, any of the several line printing or page formatting macros could increment these counters at will. They could test LNCTR to see if the page was full (page length in lines, if other than the standard fifty-eight provided by the macro, being specified by a parameter). They could test PGNO to see whether an upper or lower page (the fanfold equivalents of verso and recto) was being printed and adjust top and bottom spacing accordingly, or arrange to start a new section on the equivalent of a recto.

Getting the paper started right in the first place was the job of another very simple macro which printed a message to the operator. It would seem that a technology which can send men around the moon could devise some sort of a sensing finger for fanfold forms which could detect whether an external or internal fold had passed more recently and set a program-accessible indicator accordingly, but demand for this humble but useful device has so far been insufficient. In fact, operators from the scientific world of the computing center usually seem surprised that we care which page we start on; administrative and business data processing types do understand and that is one subsidiary reason they typically send people called data controllers along with their jobs. In addition to more weighty matters such as keeping straight which tape is the new master and just what data it incorporates, they see that the forms are set up correctly.

As you may have gathered, I am rather fond of the 1401, the DC-3 of computers. Much of its tremendous success in my opinion stemmed from the fact that it is an accessible machine, easy to learn without a doctorate in computer science. One of its less lovable features, however, is the fact that whenever the 120-character print chain is mounted, the distinctive character associated with the termination of a reading or writing of magnetic tape changes from a group mark to a tape mark. People far more experienced in the computer world than our group was have come to grief over this little fact, and coping with it was an important feature of our system of macros. A position labelled GORTM contained either a group mark word mark or a tape mark word mark, depending on which print chain was mounted. This could then be moved to wherever a tape stopper or a test character was desired.

No doubt this is more than you want or anyone wants to know about an obsolescent machine. The point is that we had our own environment, limited though it might have been, and we learned to live comfortably in it for that time by developing our own aids. Even if you have the latest, largest computer, do not underestimate the effect the peculiarities of the exact configuration and the operating system will have on your work, both for good and ill—mostly the latter when it comes to exchanging programs with other institutions.

Any macros mentioned so far are quite general, with no particular orientation to libraries. A somewhat more specialized one divided streams of continuous text into lines at word breaks or, optionally, at hyphens. It did *not* supply hyphenation to words, a tremendously more sophisticated task calling for a large computer and great expertise in computer science as well as linguistics. A macro that was never completed started with centering a line on a page, not too difficult a problem. However, it got bogged down in the process of expansion to a generalized title page layout routine incorporating judgments involving aesthetics and psychology. One tentative lesson from this effort I will record. If some lines are an even number of characters and others are odd it is impossible to center them all perfectly in relation to each other, therefore the longest line should have the half space excess to the right.

There were other macros to move characters and provide for the overprinting of diacritics whenever necessary (but not slowing down the printer to do so when there were none). Table macros to convert MARC or our own shelflist character codes to those giving the best available output from each of three print chains were also more library related. The longest and also most specific to libraries was NAMES, which derived sort keys from personal names; this work is discussed more fully in the MARC report already mentioned.[2]

Work with selection of MARC records as well as name and title indexing is recounted in that report and will not be further detailed here, except to say that toward the end, when the data filled two and a fraction tapes, not lightly to be run through, most searches were multiple. That is, several things were being looked for during the same run, and appropriate indicators were being put in different locations in the local use field, recording which particular set(s) of criteria the record met. Another development too late to be mentioned in the printed report was a limited error correction program. This did not reach the level of changing the length of fields or records, but could change lower case to capitals or vice versa (the latter actually the commonest correction required), or change any character in a specified location to any other character. A good number of observed or program-detectable errors were corrected in Harvard's final MARC master by this program, but we learned that batch processing correction of this type is tedious and expensive, even though we were prudent enough to select and transfer the records to be corrected onto a short special tape which was then passed several times until letter perfect. Any thought that all corrections could have been done in one pass through the whole master would have been wildly illusory. Even if the program had had the capacity to make as many changes in one pass as a few of the records required, human error in specifying some of the changes would have spoiled the job. As happens with conventional proofreading and typesetting, new errors are easy to introduce while changes are being made. For example, the correct character is moved to the spot next to the one to be corrected, leaving two wrongs instead of one. On-line capability seems especially attractive for editing and error correction. It would certainly save human (and elapsed) time, though higher machine costs might prove to eat up much or all of the saving in salaries.

Tedious as it may have been making corrections with this rather primitive program, inspecting a printout, and making further corrections, it was far simpler than it would have been if each pass had had to be submitted to a closed-door computer in another building. Had this been the case, it is certain that our final MARC master would not have been as relatively clean as it was.

These remarks so far have been a sort of informal parallel or supplement to the official published report of Harvard's participation in the MARC Project, slanted to give a view of one type of situation–hands-on use of a small, relatively easy to understand computer whose inherent limitations were to a large degree compensated for by its exceptional accessibility and by a set of aids that were at the same time homemade and tailor-made.

Now let us pass on to a situation that in many ways is the same but is also very different, although it was some time before the full extent of the difference became apparent. The time is the present academic year. The configuration is the same. There is now more competition for the machine, both from library colleagues and from refugees from another 1401 that was turned in. Local work on MARC II is in a state of suspended animation because a change to a third generation machine is on the horizon. These, however, are not the differences of which I speak. Let me explain.

The Widener punched card circulation system was instituted in the summer of 1963. As early as 1964, computer processing was applied to the cards for two periodic tasks, fine billings and semiannual overdue reminder lists for faculty members, as described in the 1965 document on the system.[3] In 1968 it was decided to replace one of the two card files–the information file in call number order, which had to pass through the collator every day–with a magnetic tape and a daily printout. Since I had designed the punched card system five years earlier, it seemed only logical that I develop the new printout system for circulation. This has been a most instructive experience and has provided the second of my three differential views.

Do not interpret this view as one of disillusionment or disappointment. The printout system has worked very well. A month's parallel operation of the old system and the new was planned; well before that time was over, the circulation division unilaterally dropped maintenance of the old system. No disaster ensued; this is not the account of a fiasco. Still, as I shall show, there was much to learn.

First draft programming, given the system designer's familiarity with the punched card format of his own invention, went quite rapidly—a matter of a few weeks. Debugging and polishing time was not excessive. While the macros were used in the original assembly of the programs, they were expanded to one card per line in a new source deck, and some of the luxury options, or safety features such as the provision for warning of illegal parameter combinations (by now known not to be present here) were weeded out in a hunt for core positions. One of the programs ended up in the 7990's of our 8K machine even after these excisions.

Revisions in the format of the printout based on experience with its actual use took a little more time, as did slight shortcuts in the actual running of the program. For instance, at first the opening message to the operator always included a line, "Use 48-character print chain." The revision was to print this only when the 120-character chain was mounted, as determined by a test tape write. Furthermore, it was arranged that if the wrong chain were on, the computer would do nothing but repeat the message until it was changed. In this case a machine feature already spoken of as potentially troublesome was turned to positive advantage.

This last example, although perhaps trivial, begins to get close to the heart of the matter. There was far more than anticipated to making the transition from the old hands-on experimental way to a package that could simply be given to an operator to run, backed up by the documentation required for both the operator and the circulation division. For a highly relevant example, what happens if the sort program prints out an unreadable tape block and the system designer is away?

Please note that the departure from hands-on operation in this different situation was voluntary. An alternative would have been for circulation to have its own data controller who accompanied the job and held its hand every night. Systems work would have been greatly simplified but we would be back to the question of a backup for the data controller, not to mention the question of his salary.

Let me cite a small detail on what proved to be the nub of the problem in achieving our goal. It was decided to substitute a computer sort for a card sort. This meant that there would be one program to edit the cards and put them on tape, next the sort program supplied by the manufacturer, then a tape updating program, and finally a printing program. It was convenient to carry the last two programs on successive master tapes, but the sort program was available only on cards with our particular configuration, and it worked out best to use cards also for the card editing and taping program.

It was nothing new for us to load a following program automatically at the end of a preceding one—in fact, the standard closing macro XYZ99 had a parameter which provided for this. There was just one thing—how did you direct the operator to set up the tapes for the tape update when you did not know in advance which unit the sort output was going to come out on? This varied with the size of the file and could also vary with its degree of randomness. One early thought was to use two job cards, clearly marked 1 and 2. On the second card, instead of calling for a specific reel number on tape unit 4, there would have had to be the statement (or a reference to a statement on the other side of the card, since there was not sufficient room on the front), "Use sort output from previous program." While a regular operator would no doubt get used to this quickly enough, the chance that sometime a substitute would find it too confusing and there would be a disaster seemed too great for us to take.

We finally elected to make the job a continuous one and to print out instructions (including a table with actual reel numbers) which varied according to the unit on which the output appeared. This routine had to be grafted onto the end of the manufacturer's sort (while the output tape unit number was still available), which led into some study of that 8700 line program. Adding our own remounting instructions (once they were written) was simple enough, but weeding out several hundred cards for unused options to save half a dollar's worth of card reading every night was more of a task. Then there were such details as making changes so that the entire deck including parameter cards could be given one straight through numbering, important when there was card reading trouble.

Our own addition at the end needed to read a card which was changed daily, giving reel numbers of tapes to be used and of the previous day's master. The print program also required one card to be read, containing the date. When we began, these were separate cards; after all, the entire tape update program (which read no cards) came in between. One evening there was a substitute operator, who was given a brief verbal rundown on what to expect (documentation was not yet ready at that time). The fact that one card remained in the reader for a considerable time after all others were read was mentioned but apparently not heard, for he called me at home shortly and said, "It wants to read a card." He had run out the reader after the rest of the cards were read, thinking he was a good operator to do so. The solution? Try to change the ingrained habits of an operator? (The regular operator when told of the incident confessed that he had done the same thing on occasion, but he had had the wit to retrieve the card when it was called for.) The solution was instead to combine the daily data required by the second and fourth programs onto one card, read by the second, and to hold the twelve characters of information required for the fourth over in the highest core positions by clearing storage for the last two programs not from the usual 7999 down, but from 7987.

Murphy's Law ("If anything can go wrong, it will") tells us that we should provide an error or exception routine for every eventuality; one of the programs has thirty possible messages for the operator. Our earlier programs of course had messages, but they were messages to ourselves, reminders to do something we understood or notices that something anticipated had happened. What might be called life or death messages are quite another matter. At this point, the problem is no longer computer science but communications. How will the operator understand the message? In what ways could he misunderstand it?

All in all, it is necessary to build into the system proper, if possible, and into the documentation if not, a substitute for the knowing presence of the systems designer. This is no mean task, and the revision of the circulation system has taken months rather than weeks. It has given us a fuller appreciation of why an article on software costs by Carl H. Reynolds was entitled "Notes on Estimating and Other Science Fiction."[4]

What is the combined lesson of my first two views? If one is developing something new, experimenting, or making corrections (whether in program or data), a hands-on environment is priceless. When a job becomes routine and must be done whether or not the individual is present, things are very different and the ideal is to wrap up a package that can be given to someone who has never seen it before and still run successfully.

Let us now consider the future. The future of computer applications in libraries is wide open. If I can make a generalized prediction, perhaps it is that economics will be a more severe limitation than technology.

It has already been clearly demonstrated that computer processing of bibliographical records (whether for whole books or for journal articles) is possible not only technically but in many cases economically. Most work so far has used batch processing; on-line systems are very attractive for many purposes but also tend to be very expensive. Even in batch processing of bibliographical records, which is the most widespread form of activity, there are limitations largely based on cost. Machine-readable records may be limited to current books processed since the new system was introduced; or if the entire collection is covered, the records are probably brief, or the total size of the collection small. No large library has yet completed the task of converting its entire catalog, not to mention that of upgrading the level of its subject cataloging to the standards desirable for new machine systems.

Still dealing with bibliographic records, but at a level beyond batch processing, we may consider on-line random access, whether by means of a dedicated computer or by time-sharing. This is a far less well-explored field. The possibilities are exciting but the price tags are high. One library which actually uses this type of facility, a disk on a time-sharing basis, for a few weeks a year while editing its periodical list, reports data storage costs which work out to be on the order of a dollar a year per title. This is clearly prohibitive for general use; for many large libraries it would tend to double total expenditures (recent observation of the statistics issued by the Association of Research Libraries indicates that annual budgets in dollars and total holdings in volumes are often in the same range).

Of course the dollar per title per year was a service bureau type charge (though non-commercial) and no doubt included a great deal of overhead. It may have allowed for much more activity in writing data on and off the disk than actually took place. A far more favorable cost estimate comes from the NELINET (New England Library Information Network) project where the storage cost per title per year on a large owned disk tied to a dedicated computer might be as low as five cents. At this cost level one could consider it, however painful financing even this may be where books are counted in the millions. One could take the line that for future accessions a certain average sum should be set aside for each title (hopefully out of savings realized through having machine-readable catalog data available from a central source) as what might be called endowment for that book's bibliographical record being kept on-line. At five cents and 5 percent interest, this works out to $1.00, not too discouraging an amount. These figures are not absolutely solid, but they do provide food for thought.

When we see how much remains to be accomplished even in this relatively developed area of machine-readable bibliographic records, is it worthwhile even speaking of the tremendously more ambitious possibility of putting the entire text of books and journals in our libraries into machine-readable form? The answer is yes, if only to indicate how remote it is as a practical economic possibility, and to consider possible alternatives.

One reason that the question is inevitably going to come up is the problem of indefinite expansion of library buildings. The hopes of miniaturization first raised by microphotography and now by digital storage are worthy of fairly extended discussion. First, let us consider and compare the quite different approaches of ordinary microfilm (whether roll or fiche) and computerized storage.

Microfilm has these advantages: data conversion is relatively cheap, perhaps on the order of two cents a page in quantity production. Reading machines are low in cost and reader-printers are available. Typographic style, manuscript annotations, and illustrations are reproduced. Separate color films of colored illustrations are not beyond the realm of serious consideration (the main film could include both a black and white copy of such an illustration and a target referring to placement in a collected color roll or sheet). Microfilm can readily be copied. Despite recent flurries, it is relatively permanent. Microfilm technology is familiar and proved, although library filming typically uses equipment and materials developed for other purposes and this is not entirely satisfactory for use in the library. Harold Morehouse has a good phrase in an issue of *Library Research & Technical Services*; he is speaking of telefacsimile but might as well be referring to microfilm or to computers: "It sometimes seems as though we are doing something like using an electric dishwasher to wash our clothes."[5] In the roll microfilm field, the absence of standards for cartridges seems particularly unfortunate.

In comparison with film, digital storage has one potentially tremendous advantage—it is machine-readable, with all that that implies. The cost of copying is typically even lower than microfilm for equivalent volumes of text, but machine-readability is the one outstanding advantage.

Just what does having full text in this form make possible? For one thing, there is an advantage in transmissibility. Optical images can be transmitted by using scanning devices, but reducing each letter to the typical eight bits rather than an area to be scanned has inherent efficiencies. Even so, large scale over-the-wire transmission of texts of books or even articles is a particularly clear cut example of a situation where the economic factors are more limiting than the technical. Instant display via a character-generating cathode ray tube is another somewhat similar area. Again, there is no question but that it can be done once the information is encoded, but the serious question is how economic is it even in the absence of the long line tolls which are such a large factor in data transmission.

The situation with regard to automatic indexing and abstracting, and information retrieval based on this capability, is quite different. These are primarily software rather than hardware problems. While some aspects—concordances, KWIC indexes, vocabulary studies—are subject to what might be called definitely attainable solutions, others—automatic indexing and abstracting, machine translation—offer what amounts to an infinitely extensible challenge.

Even if one assumes a scarcity of indexers, an ample supply of tape typists, and free machine time, it is implausible to expect that articles may be indexed equally well with less total effort (conceding a much heavier weighting to the indexer's time) by re-keying the entire text and then submitting it to machine analysis than by more conventional ocular and cerebral processes. Automatic indexing is a very fascinating intellectual problem that draws gifted scholars because, like Everest, it is there; but with all due respect to the intelligent, ingenious, and hard-working people in this field, it will become a practical economic method only if machine-readable text is already available.

How does one obtain this? Let us first consider the optical scanner or page reader. Such devices are already in fairly wide use, most successfully with material typed in a font specially designed for the purpose but still quite legible to the human eye—much more so than the magnetic numerals imprinted on bank checks. Ordinary typewriting is also handled fairly successfully if paper and ribbon quality as well as type face are uniform. Books offer problems of quite another order, including varying type styles, varying width of different characters within the same font—an *m* is wider than a *t*—and a much larger number of different graphic units than in typewriting, for example the ligature for *fl*. This is not to mention such problems as page turners if original non-expendable books are to be used as source copy.

If scanners sophisticated enough to read books rather than typed documents are developed, it is reasonable to assume that they will be more expensive than the machines of more limited capability, though to be sure there is both hope and some reasonable expectation that machine costs generally may decline if considered on a constant dollar basis. However, even assuming both that the problems of scanning books can be resolved and that a secular decline in machine costs cancels the premium for greater sophistication, the fact that the current service bureau rate for scanning the easier typewritten material is on the order of one cent per *line* helps put this avenue in proper perspective.

There is one other possibly royal road to machine-readable full text, although it is possible that if I had cost data on it I would have another reason for a lapse into pessimism. It is of no help at all with books of the past, unless they are reprinted. However, I believe it may have great worth in the future. Many type composition processes right now and all photocomposition devices that I know of use a machine-readable tape as a regular step in the process. If these compositors' tapes could be translated into regular computer tapes, we might at last be on the track of machine-readable full text at a reasonable price. However, of course, there are problems. The tape (commonly punched paper at the present time) is machine-readable, but usually only by a very special machine. It may be wider than standard, with more channels; special hardware would be required for translation and hopes for economy would fade. If the tape is physically compatible with computer tape, there will probably still be problems. Simple code differences—the use of different punch or blip combinations for the same letter—are readily susceptible to a software solution. The presence of special typographic codes may be more troublesome, but can still be solved by software. A larger problem, and really the crux, is the matter of errors. Many otherwise advanced systems still use manual methods of correction, so that the final corrections never get on the tape. Compositors' tapes will become a really useful source of full text data only when computer-aided editing systems are used and final tapes are clean. Fortunately, such editing systems also offer great possibilities for improved efficiency to publishers, who cannot be expected to go out of their way merely to produce a by-product useful to someone else.

The preceding discussion has assumed that full text in machine-readable form would be desirable if we could get it. Certainly there are many purposes for which this is so, even if we defer the concept of remote display. Full text analysis has already revolutionized the making of concordances. One wonders how many uses of words earlier than the first recorded in such works as the *New English Dictionary* would be turned up by a computer search of a whole library. An algorithm that would detect proper names, or even potential proper names, by their typographical characteristics, could make possible prodigies of indexing. It will be a long time before such capabilities can be applied to large general collections, but it is worth beginning to think about them.

I have not meant to create the impression that bibliographic records only and full text are the alternatives and that there is nothing in between. Among intermediate steps are machine-readable (but humanly composed) abstracts and citation indexing, the latter a particularly active and promising field.

Where does all this leave us? Through a glass darkly I see something like this: while I agree that we should be accumulating data for the future insofar as it is available from compositors' tapes, and I hope such availability will be encouraged by a standards effort and by development of central repositories, I think that full text processing will long be limited to special situations and will not be in the realm of financial possibility for universal application in the foreseeable future (whatever the foreseeable future may be).

I do think conversion of bibliographical records for books, including full retrospective conversion (but not immediate upgrading), is within the realm of the feasible, given a major effort by the national library and full cooperation from other major libraries. Our old friend the analytic, seen less and less often among new additions to card catalogs, may make a comeback in machine systems. Machine-readable indexing of current journal articles is already a reality in a number of fields and it seems reasonable to hope that it may be extended to all, but retrospective coverage of all articles is a much more formidable problem.

I do not think whirling magnetic disks or comparable larger devices, inherently high in cost and subject to hazards of data loss which require elaborate and expensive backup, are the last word in storage for library records. Rather, I see a more stable, permanent, read-only tape memory (read only after original entry, of course), hopefully of vast size and low unit cost, which will contain the body of the description of each title, with numbers referring to the various headings—author, added, subject—which refer to it. The description could have a permanent number and additions of new batches could be made in accession order at the end of the file. A revised entry would be a new entry, leaving the old one unused until such time as a general rewriting of the entire file was required, when it could be dropped.

While the various names, subject headings, etc., would also appear in their own read-only authority file, they would also have to exist in a more active medium which would refer by number to all the full records to which the heading applied, and to which new accession numbers would constantly be added. Title and classification number files with provision for interfiling would also be needed. Ordinary access to the main permanent descriptive file would be indirect, through the active heading, title, or class files. There would be provision, however, for periodic seriatim search of the entire file for combinations of characteristics not findable through the usual keys.

Until a full on-line catalog of this sort can be developed and afforded, it might be worth considering having the basic file on tape, used via a book catalog, with supplementary titles (including information on those on order and in process) in an on-line system.

These remarks on the possible future of the computerized catalog are speculative; even more speculative will be anything I can say about automated text access. There would seem to be considerable promise in mixed systems, storing text (and illustrations) in micro-image form and retrieving, transmitting, and displaying it under computer control. Project INTREX at the Massachusetts Institute of Technology is doing some interesting work in this field, but the fact that a device mentioned favorably in their reports has a capacity of 45,000 pages is some suggestion of what a long road there is to travel before large general libraries are covered by systems of this type. In the meantime, perhaps it is worth thinking about wider use of ordinary hand-retrievable microfilm as an interim backup facility. For really is there not something incongruous about investing very large sums in advanced computerized catalogs when they refer solely to original copies of books that are subject to all the hazards of being out, not on the shelf, etc., that are so familiar in present-day university library collections? This would also be a step toward the desirable goal of taking some of the wear from our aging stocks of original books.

This paper has raised many questions to which I do not know the answers. However, my general conclusions are that libraries and computers have a great future together, but that economic factors and the sheer size of large libraries will slow full application of many exciting possibilities that will first be explored and developed in more specialized situations. Finally, microfilm, both conventional and automated, must not be overlooked as a complement to computerized systems.