CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Illinois Digital Environment for Access to Learning and Scholarship Repository

Henriette D. Avram
Assistant Coordinator of Information Systems
Information Systems Office
Library of Congress

THE EVOLVING MARC SYSTEM: THE CONCEPT OF A DATA UTILITY

Many documents have been published describing the current MARC system, the procedures and computer programs that are used at the Library of Congress (LC) for the MARC Distribution Service and other projects for divisions within LC.[1] This paper attempts to explain the rationale of the system, to summarize what exists today (the results of the efforts of 1967-1969), and to describe the emerging plans for the next generation of the MARC system (1970-1971).*

THE CONCEPT OF A DATA UTILITY

To describe clearly what exists today and the plans for the future, it is necessary to explain the thinking in 1967 that led to the present MARC system. At that time, the only requirement for an operational system was the MARC Distribution Service. However, the designers of the system assumed (with nothing more concrete to base their assumptions on than intuition of what might be just around the corner) the use of one system for many purposes. The procedures and software necessary to input, format, validate, retrieve, update, and output cataloging records for the MARC Distribution Service are basically the same as those required to process all forms of material to produce bibliographic tools for use at LC.

The decision to design a generalized MARC system, while engaged in the analysis needed to functionally describe the man-machine specifications for the Central Bibliographic System (CBS),† forced us to try to understand the relationship of the MARC system to the total automation program of LC. We accepted the premise, and

---

*System as used in the remainder of this paper is concerned primarily with the computer hardware/software configuration rather than the procedures for editing, inputting, etc.

†A program consisting of seven phases. Phase I: Survey of the Present Manual System; Phase II: System Requirement Analysis; Phase III: Functional Description of Recommended System; Phase IV: System Specifications for Equipment and Software; Phase V: System Design; Phase VI: Implementation of New System; and Phase VII: Operations of a New System.

still do to date, that work accomplished would provide the prerequisite experience and knowledge for more advanced efforts. We fully recognized that the totality of the CBS and all it implies would evolve over many years, that the hardware/software configuration and the man-machine interfaces were not fully understood. The requirement to implement the MARC system in an environment of many variables and unknowns resulted in the conceptualization of a centralized data processing service called a data utility.* The hope was that, within the constraints imposed by the complexity of a many-faceted information system, our progress would be as orderly as possible and our efforts would have meaning for the future system.

The term "data utility" is used to describe a data-oriented, computer-based centralized service, with emphasis toward generalized applications on a centrally maintained set of data files for access by a variety of users. This concept differs significantly from a computer utility. A computer utility allows many users at remote sites to use a central computer concurrently. Each user operates as though the computer were dedicated to his problem alone. In general, each user produces his own programs and maintains his own data files. He may choose his own programming languages; he may be compiling a program, debugging a program, or executing a program; he may request output on any available device, e.g., printer output, teletype, cathode ray tube, or plotter. The data utility is similar in that certain services will be available at remote sites, but all take the form of interrogating existing files with existing software. The user does not produce the application programs or maintain the data files. (Certain files will be updated on-line by qualified library personnel, e.g., MARC editors will correct and verify records on-line.)

The operations of a library are file oriented. Any approach toward the automation of library processes must be based on the improved handling of files. These files must be converted to machine-readable form and maintained. It must be possible to access these files in a variety of ways both for maintenance and for retrieval of information, and to provide a variety of products for use by the staff and/or the public.

Within the data utility framework, files are maintained in a single standardized format, i.e., the records within each file are self-defining and the data elements in each record are represented in a common structure. The LC MARC processing format[2] was specifically developed for such a role, has already been used for several files, and, by the nature of its design, can be used for many others.

---

*The author wishes to acknowledge the work of Coyle & Stewart, Computer Applications Consultants, who, under contractual support, assisted the Information Systems Office in formalizing this concept, as well as some of the advanced planning described in the remainder of this paper.

The centralization of the computer facility and the standardization of the record format across all files have allowed development of general purpose application programs. These programs operate on the data without concern for the fact that there are a variety of files. Thus, it has been possible to serve different interested library personnel, the users of the data utility.

This generalized concept, although more difficult to implement, has paid off many times. The following briefly describes some of the LC projects that are either in operation or in the check-out phases that use the MARC system for the processing of the data from input through output.

1. A variety of book catalogs have been produced for internal use of the Geography and Map Division. The off-line retrieval subsystem has been used to extract records pertaining to a particular subject matter for specialized catalogs.

2. Files are maintained for the reference collection of the Science and Technology Division reading room. This collection contains current and non-current material made up of monographs, serials, and technical reports. Book catalogs are issued for use by the reference librarians.

3. A bibliography is compiled by the Science and Technology Division for the U.S. Army Cold Regions Research and Engineering Laboratory. Quarterly and annual reports are published as well as a monthly current awareness listing.

4. The National Directory for Latin Americanists, published by the Hispanic Foundation, is processed through the MARC system. Members of the foundation staff also use the retrieval subsystem to answer specific queries from the data base.

5. A file containing citations of journal articles, government documents, congressional documents, monographs, and internal Legislative Reference Service (LRS) reports is maintained. In addition to the production of book catalogs, this file supports a selective dissemination of information (SDI) system providing "current awareness" based on profiles for the staff members of LRS in their support of Congress.

A data utility encompasses both batch and on-line processing. Its advantage is that the service is open-ended in that new applications and data files may be added with facility whenever required. Although the MARC system was designed as a batch processing, tape-oriented system, the data utility concept represented a systematic framework for the expansion of its capabilities and services.

THE CURRENT MARC SYSTEM

The current MARC system is defined as four subsystems: input, file maintenance, retrieval, and output. The programs within each

subsystem are generalized, data independent where possible, table driven where the function of the program assumes known characteristics of the data (e.g., validation of MARC tags for a given form of material such as maps), and parameterized where the user, within constraints, can specify his requirements (e.g., format control for printing of bibliographic listings).

The system is a batch processing tape system operating under DOS on a system 360/40. Figure 1 is a high-level flow diagram, each rectangle representing a complete program.

Following the numbers associated with each rectangle in the diagram, the functions of the programs are briefly described below:

1. *Preedit.* Accepts source data from a variety of input devices (punched paper tape, magnetic tape, punched cards) and converts the data from the code of the input device to EBCDIC code for 360 processing. The preedit program deletes all function codes (codes from the input device to control the input device and/or to indicate to the computer program end of field, end of record, etc.), and writes each field of a MARC record on tape as a physical record. The present input process at LC consists of a magnetic tape inscriber through a converter to computer-compatible magnetic tape.

2. *Fmtedit* (*Format Edit*). Processes each record from preedit output (each record is a field of a logical MARC record) and produces a pseudo-MARC processing record. The format edit program eliminates all redundant fields (last field in is the one retained which allows the input typist to replace an entire field while typing) and corrects all characters, words or lines that the typist corrected by the use of backspace and delete codes during the typing operation. A field containing the order of input of the MARC record is prefixed to each format edit record. Both Preedit and Format Edit are data independent. They will accept and process any variation of the LC MARC processing format following the prescribed input procedures.

3. *Conedit* (*Content Edit*). Accepts the format edit output, determines the form of material (monograph, serial, map, etc.) being processed, activates the table of tags, indicators, subfield codes (hereafter referred to as content designators), and any special program modules required for the processing of that form of material. Content designators are validated and tags converted from a mnemonic to numeric value. (The editor has the option of using either form of tagging.)

Frequently appearing indicators and subfield codes do not require explicit editing or keying. The content edit program recognizes their absence and sets them automatically. Record and field level error conditions are flagged in the record, which is written on the content edit output tape and the processing results tape. All input records to Content Edit are written on the processing results tape, but depending on the type of error, they may or may not be written on the content

edit output tape. For example, if the LC card number is incorrect (discovered through recomputing the check digit), there is no way to correct the record in error since all control is by the card number. Therefore, the record is written on the processing results tape only.

4. *Sort.* Sorts the content edit output tape in LC card number sequence for further processing.

5. *Update1.* Accepts the sorted content edit output tape which may contain new, delete, replace, or verification records. The tape is processed against the MARC working file (old) which contains records that have not been declared error free. An updated MARC working file (new) is generated containing all new and modified records (resulting from the correction of fields) in addition to all records for which there was no transaction. All verified records and correction and deletion transactions which found no match on the working file (old) are transferred to the master file transaction tape for later processing. Transaction records which have an error in the transaction record itself, as well as all corrected records, are written on the processing results tape.

6. *Print Index.* Lists the LC card number and the date of the last transaction of records on the working file (new) for the MARC editorial office.*

7. *Update2.* Accepts the master file transaction tape which contains newly verified records and correction and deletion transactions which found no match on the working file (old). The tape is processed against the master file (old), and an updated master file (new) is generated containing all verified and modified records in addition to all records for which there was no transaction. All records modified on the master file, transaction records which have an error in the transaction record itself and all transaction records for which no match was found on the master file (old), are written on the processing results tape. Deletion records are processed against the master file (old) in two ways:

a. Deletion records referring to records not yet distributed to MARC subscribers cause the deletion of the record.

b. Deletion records referring to records that have already been distributed to MARC subscribers result in a change of the status of the original record. These records are transferred and remain on the master file until the next distribution cycle when they are transmitted to the subscribers and deleted from the file.

8. *Print Index.* Lists the LC card number and the date of last transaction of records on the master file (new) for the MARC editorial office.

9. *Sort.* Sorts records contained on the processing results tape

*The office in the LC Processing Department where MARC records are edited, keyed, corrected, and verified.

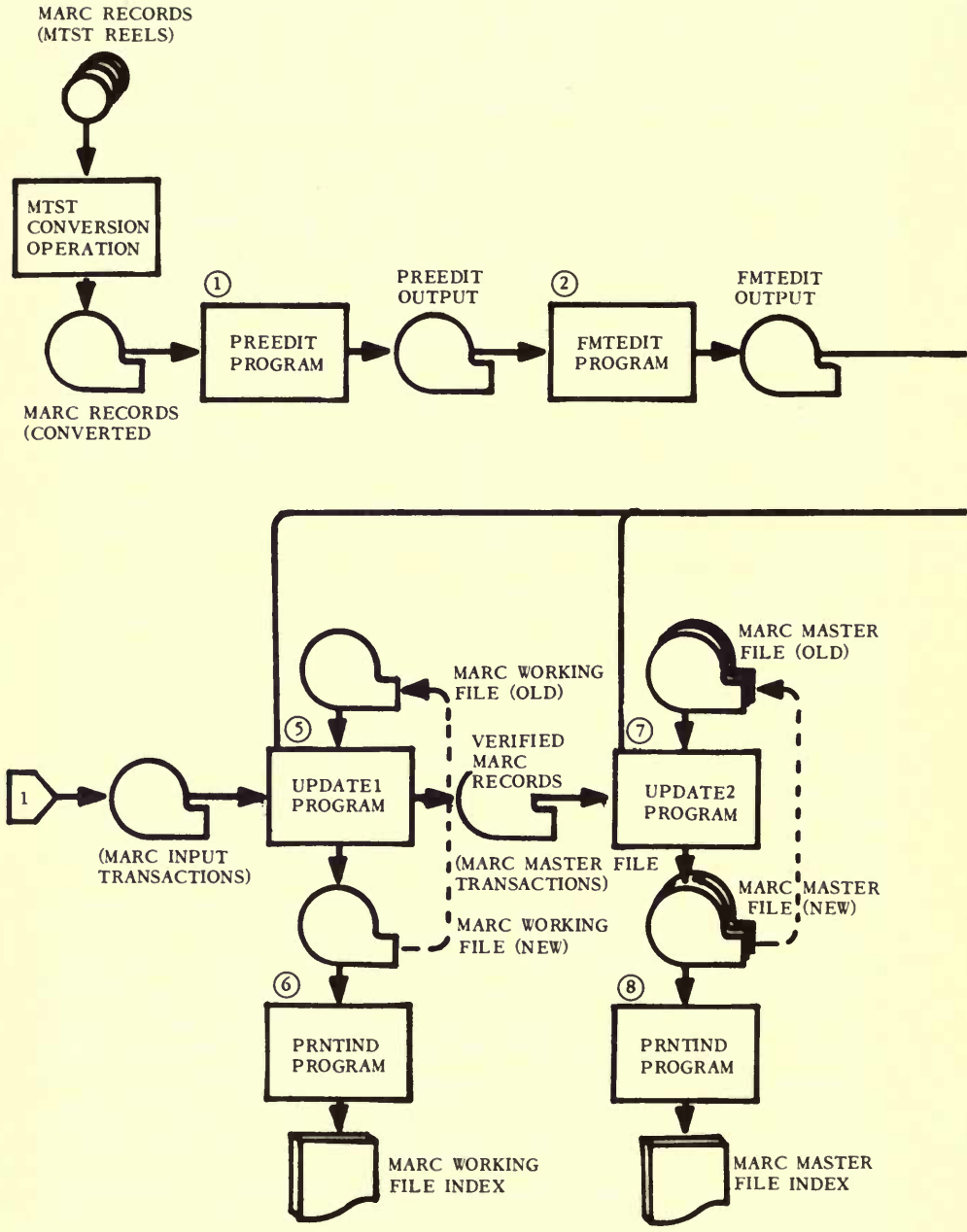
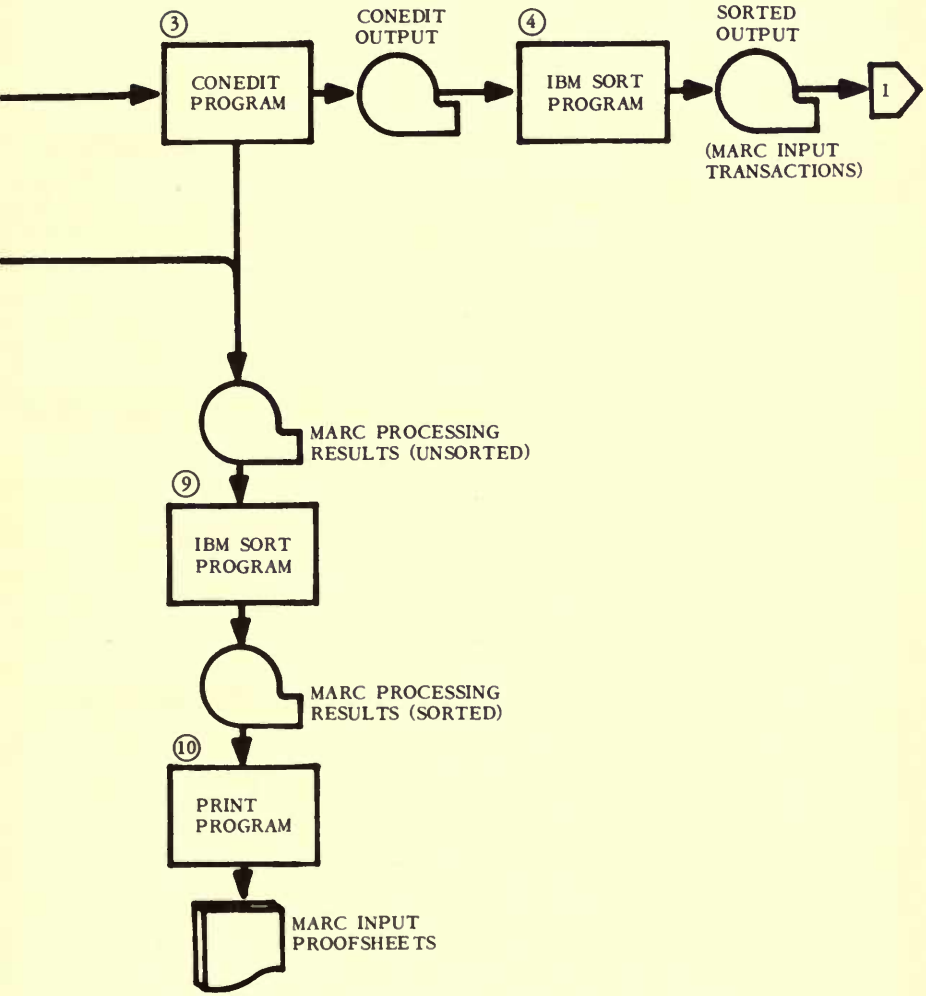


Figure 1. Current MARC II Processing System


3
CONEDIT PROGRAM
CONEDIT OUTPUT
4
IBM SORT PROGRAM
SORTED OUTPUT
(MARC INPUT TRANSACTIONS)
1
MARC PROCESSING RESULTS (UNSORTED)
9
IBM SORT PROGRAM
MARC PROCESSING RESULTS (SORTED)
10
PRINT PROGRAM
MARC INPUT PROOFSHEETS

(output of Content Edit, Update1 and Update2). The records are sorted on the field prefixed to the record during format edit processing containing the order of input. This sort serves two purposes: 1) the subsequent proofsheet run is arranged in input order, thus facilitating the matching of the proofing record with the original input worksheet; and 2) in the proofsheet run, the listing of each transaction record appears with the record it corrected. Thus, the entire transaction history of the record is displayed for analysis and for any subsequent action by editors.

10. *Print Program.* Displays all records contained on the sorted processing results tape. The contents of the record are displayed in an order determined to be the most useful for proofing against the original input worksheet. Record and field level error messages are generated to assist in the management of records within the system.

FURTHER DEVELOPMENT OF THE MARC SYSTEM

THE NEED FOR EXPANSION

The current MARC system is tape oriented in keeping with the hardware configuration available at LC at the time of implementation. LC's commitment for the operational MARC Distribution Service did not allow sufficient set-up time for the acquisition of additional equipment or the design of software required for the effective use of random access devices.

The present procedures for original input of data and correction of mistakes in editing or keying result in a cyclic process. The records are input, a proofsheet is provided, corrections are made on the proofsheet, and then they are rekeyed and run through the system producing another proofsheet. This procedure is continued until the record is declared error free. At that point, the LC card number is keyed into the system and the verified record is removed from the working tape and stored on the master tape. Control procedures are required to monitor the movement of the record from original input to the final cycle when it is considered verified.

The experience at LC indicates that on-line input of new records is not required but there is a definite need for on-line correction and verification of records on the working file and the master file. This will reduce the paper work involved, expedite the processing of records, and should result in an increase in the number of records processed. The processing time required to revise records in a tape system, the efficiency offered by on-line correction and verification, and the eventual requirement for more than one access point to a record are the factors that initiated the redesign of the MARC system. The redesign was strongly influenced by the desire to build upon what was already available and by the desire to provide a solid base for the expansion of the automation program.

FORMAT DESIGN

The MARC processing format will remain basically the same for the expanded system. Additional control codes will be added to facilitate increased demands.

Two approaches have been taken in specifying the MARC formats to fit within the concept of the data utility:

1. Care has been exercised to identify data common to different kinds of material with the same content designators. Unique content designators are assigned for data elements peculiar to any one form of material, e.g., the scale of a map for the single sheet map format.

Table 1 correlates the tags used in formats designed to date at LC. The monograph format is used in the present MARC Distribution Service; the serial format has been published to elicit comments; the map format will be published in the near future; and the motion picture and filmstrip format is still in the design stages. All tags have not been included in Table 1; only a sufficient number are listed to demonstrate how tags are assigned.

2. Specific codes have been assigned to the record for a variety of control purposes. For example, a code is used to indicate the currency of a record in relation to the record in the official catalog. This allows the initial input and processing of a large number of records which could include records requiring comparison with the official catalog. These records can subsequently be extracted from the machine-readable file for this purpose. In addition, this code provides a mechanism for controlling the distribution of records to MARC subscribers. A retrospective record converted by the RECON Pilot Project may be input and processed, but not distributed until the machine-readable record is compared with the record in the official catalog.

Other examples of the use of control codes are the source and distribution codes. The source indicates the originator of the machine-readable record (this does not in all instances mean the originator of the cataloging). For example, the source code "Main Reading Room" signals a record for a work in the Main Reading Room reference collection, although the cataloging was actually performed by the LC Processing Department. In contrast, the source code "Geography and Map" means both a record for a single sheet map in the Geography and Map Division and also that the Geography and Map Division did the cataloging. The destination code indicates the file that the record will reside on. In the case of the Main Reading Room collection, the records could reside on the MARC master file, the way records reside on a map master file. If the Main Reading Room records reside on the MARC master file, they actually become part of the future LC machine-readable official catalog; yet they can be extracted to produce a book catalog of the reference collection, recognizable by the source code.

Table 1. Tags for Representative Data Elements in Four Types of Bibliographic Records

TAG	NAME	Mono-graphs	Serials	Maps	Motion Pictures and Filmstrips
200	Title as it appears on piece		✓		
210	Abbreviated entry or title		✓		
240	Uniform title	✓	✓		
241	Romanized title	✓		✓	✓
242	Translated title	✓			
243	Collective title				
245	Title statement	✓	✓	✓	✓
246	Varying forms of title		✓		
247	Former titles		✓		
260	Imprint	✓	✓	✓	✓
261	Production and release				✓
265	Subscription address		✓		
400	Series note, personal name	✓	✓	✓	✓
410	Series note, corporate name	✓	✓	✓	✓
411	Series note, conference	✓	✓	✓	✓
440	Series note, title	✓	✓	✓	✓
490	Series, traced differently or untraced	✓	✓	✓	✓
500	General notes	✓	✓	✓	✓
501	Bound with notes	✓			✓
502	Dissertation notes	✓			
503	Bibliographic history note	✓		✓	
504	Bibliography note	✓		✓	
505	Contents note	✓		✓	✓
506	Limited use note	✓	✓		✓
507	Scale note			✓	
508	Credits note				✓
510	Indexing and abstracting coverage		✓		
511	Cast note				✓
515	Explanation of dates and volumes		✓		
520	Abstract or annotation	✓			✓
525	Supplement note		✓		
530	Additional physical forms available		✓		
555	Cumulative indexes		✓		
700	Added entry, personal name	✓	✓	✓	✓
710	Added entry, corporate name	✓	✓	✓	✓
711	Added entry, conference	✓	✓	✓	✓
730	Added entry, uniform title	✓	✓	✓	✓
740	Title traced differently	✓		✓	✓

An encoding level code has been defined and is currently in use to indicate if the machine-readable record is the result of assigning content designators with the book in hand or from a surrogate record. Current MARC is an example of assigning content designators with the book in hand, and RECON records, an example of assigning content designators from the LC printed card. Should LC proceed with pre-publication cataloging, another encoding level would be assigned to indicate this condition, i.e., the bibliographic description is not completed. In all three cases, current MARC records, RECON records, and pre-publication records, the source and destination would be "Processing Department" and "MARC master file" respectively.

Since the organization of the machine-readable records of LC is not known at this time, these control codes will provide information which may be needed for a specific choice of organization at a later date. In the interim, one system, namely the MARC system, can manipulate all the files.

The careful assignment of content designators and the use of the control codes insure compatibility for processing, take advantage of the utilization of the same program modules whenever possible, route the record to the proper file after processing through the same series of programs, and identify records in sufficient detail to facilitate their integration within a future system.

EXPANDED MARC SYSTEM

The expanded MARC system will be the composite of several subsystems. It will include what exists today (a batch tape subsystem), it will be modified to use disk instead of tapes as storage (a batch disk subsystem), and finally, it will include on-line capability for correction and verification. Any one of the subsystems may be used, depending on the file or records being manipulated.

Figure 1 remains intact as the batch tape subsystem of the composite system. Figure 2 is a schematic representation of the expanded system, and each rectangle represents a program. Section I will operate with section II as a subsystem (batch disk subsystem) and also with section III as another subsystem (on-line correction and verification subsystem). The input of new records always occurs in an off-line mode whether the file resides on tape or disk. The input of correction and verification records may occur through section I (off-line) or section III (on-line). Table 2 lists the subsystems involved and the mode of input. Access to records is by LC card number because of the immediate requirements for the MARC Distribution Service. However, it is anticipated that the on-line system will be expanded to 1) on-line input for other than the correction and

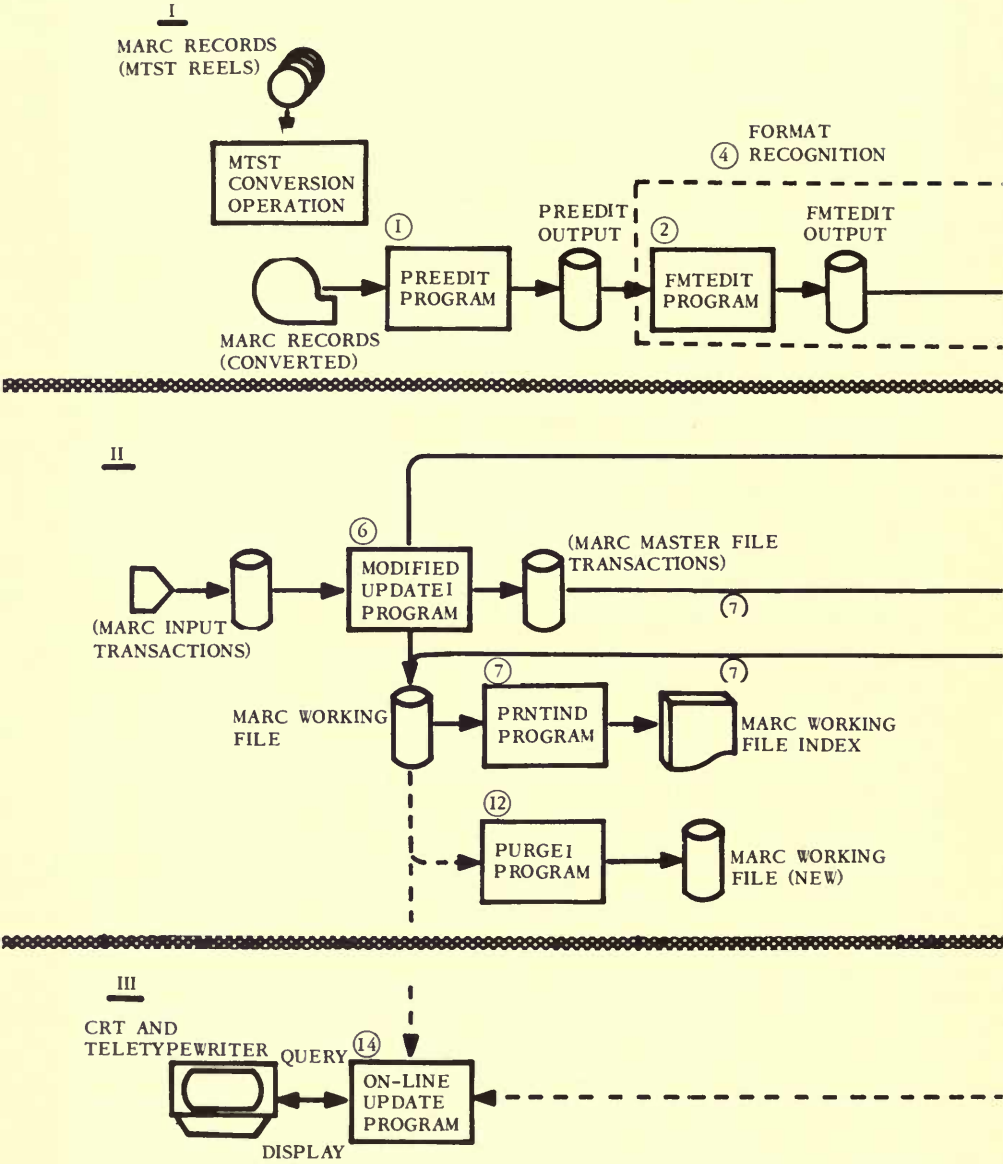


Figure 2. Expanded MARC Processing System

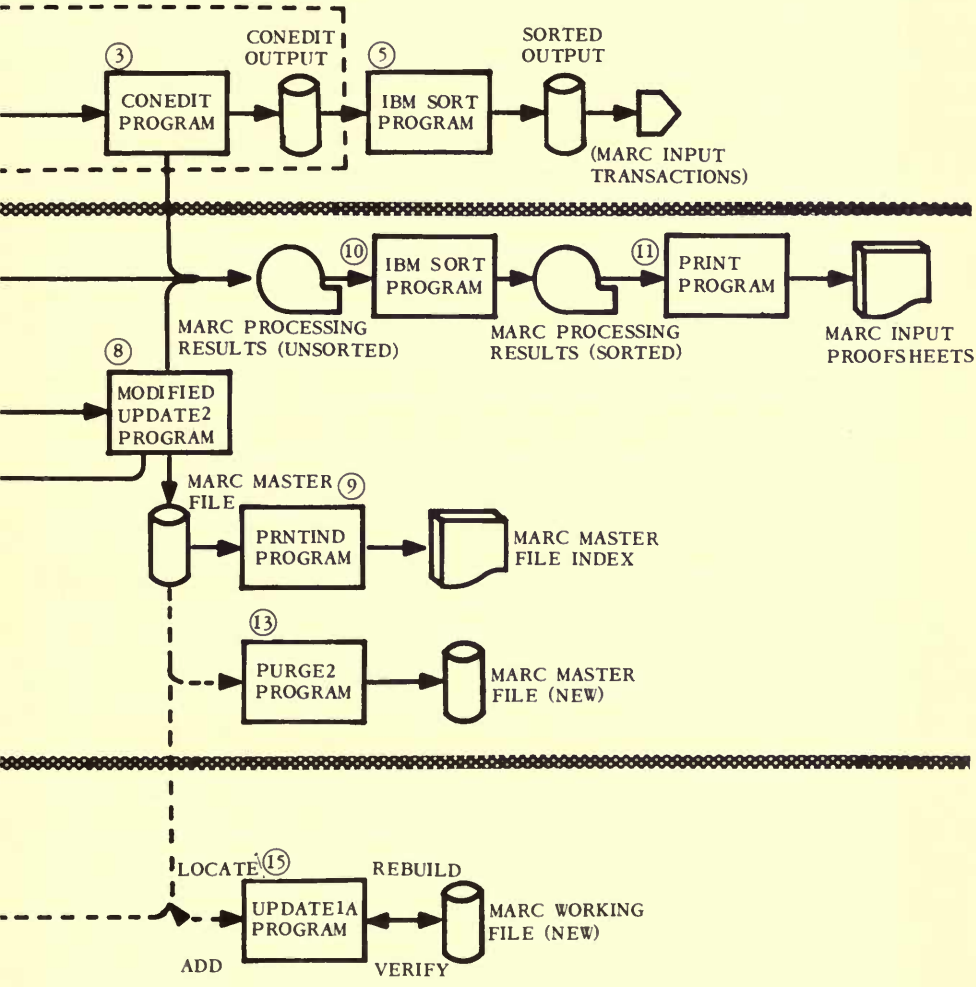

CONEDIT OUTPUT
3
CONEDIT PROGRAM
5
IBM SORT PROGRAM
SORTED OUTPUT
(MARC INPUT TRANSACTIONS)
10
IBM SORT PROGRAM
11
PRINT PROGRAM
MARC PROCESSING RESULTS (UNSORTED)
MARC PROCESSING RESULTS (SORTED)
MARC INPUT PROOFSHEETS
8
MODIFIED UPDATE2 PROGRAM
MARC MASTER FILE
9
PRNTIND PROGRAM
MARC MASTER FILE INDEX
13
PURGE2 PROGRAM
MARC MASTER FILE (NEW)
LOCATE
15
REBUILD
UPDATE1A PROGRAM
MARC WORKING FILE (NEW)
ADD
VERIFY

Table 2. Possible Input Modes

Subsystem	Input of New Records	Input of Corrections and Verifications
Batch Tape Subsystem (Figure 1)	Off-line to tape	Off-line to tape
Batch Disk Subsystem (Figure 2)	Off-line to disk (section I)	Off-line to disk (section I)
On-Line Subsystem (Figure 2)	Off-line to disk (section I)	On-line to disk (section III)

verification of MARC records and 2) access of records by key elements in addition to the LC card number.

The major change in section I (Figure 2) compared with the tape system (Figure 1) is the implementation of a format recognition program which may be used in place of the format edit and content edit programs. All three programs will reside in the system, and either format edit and content edit or format recognition will be called in, depending on the file being processed. The output, in either case, is the same.

The update programs (Figure 2) are being designed as a series of modules which handle several transactions under the conditions that control can be taken away from one transaction before completion and another transaction executed while the first is in a wait status. Such programs are referred to as reentrant programs.*

The design work of the expanded MARC system completed to date is the result of a "first cut" analysis. More detailed study is required before the specifications become final; therefore, the programs and the organization and accessing methods for the disk files are subject to change.

---

*When processing a variety of input messages, one program may be "WAITing"—for a file action, for example—and at this time another transaction wishes to use the program. This can cause problems if the program is written in such a way that it modifies itself while being executed, or stores logic information for later use in a location other than the unique message-reference block. Programs to be used by multiple transactions in this way must be carefully written so that no logic error can be caused by this. In particular, they must not modify themselves in such a way that, when control is taken away from them, another transaction can interfere with the modification. Programs which can be entered by multiple transactions without interferences are referred to as *reentrant* programs. If a program is not reentrant, it may be necessary to have more than one copy of it in core at certain times in a multi-thread environment.[4]

PROGRAM DESCRIPTIONS

The program descriptions that follow refer to the numbers assigned to the rectangles in Figure 2.

1, 2, 3, 5. *Preedit, Fmtedit, Conedit,* and *Sort.* Perform the identical functions as described in 1, 2, 3, and 4, pages 4-5, substituting disk for tape.

4. *Format Recognition.* The process which has become known as format recognition has been described in several publications.[5] Briefly, the purpose is to remove the burden of editing (assigning content designators and fixed field codes) from the human to the machine. All the algorithms have been defined to process completed unedited English-language bibliographic data and to automatically produce an edited MARC processing formated record. Flowcharting at the coding level has been completed for the implementation of required software. During the coding phase of the project, a judgment will be made for the inclusion or exclusion of each algorithm based on frequency of occurrence versus machine processing time required to assign the content designators and fixed field codes.

The result should be an optimum balance between man and machine editing. Concurrent with the definition of the algorithms, several statistical studies were made to determine the feasibility of format recognition. The result of the analysis indicates that approximately 70 percent of the records produced will be error free, and 30 percent of the records will contain one or more errors. Therefore, not only does format recognition appear to be feasible, but it should be significant in terms of the reduction of the manpower requirements to process MARC records.

The source data for format recognition is the LC manuscript card (Figure 3)* which is the record used during the cataloging process and from which the LC printed card is produced at the Library Branch of the Government Printing Office. The data from the manuscript card will be transcribed to machine-readable form by an input typist either without any prior editing or with only a minimum amount of editing.

The manuscript card is formated into paragraphs by the catalogers, and the typist follows the following organization: she uses multiple carriage returns to indicate the end of a paragraph and a single carriage return to indicate the end of a line within a paragraph. The resulting machine-readable record processed through Preedit becomes the input into the format recognition program.

---

*The characteristics of the manuscript card will be compared with the LC printed card and the procedures will be expanded, if necessary, to include the printed card, which serves as the source data for the RECON Pilot Project.

MARC 4

SD11
.A45614 Childs, Thomas White, 1908-
no.69 Elytroderma disease of ponderosa pine in the Pacific Northwest by T. W. Childs. Portland, Or., Pacific Northwest Forest and Range Experiment Station, U.S. Dept. of Agriculture, 1968.

ii, 45 p. illus.

27 cm. (U.S.D.A. Forest Service research paper PNW-69) 0.50

NC gs 11se69 sw DO NOT SET unb 76-603289

Cover title.
Bibliography: p. 38.

1. Ponderosa pine—Diseases and pests:
2. Elytroderma deformans.

I. Title.
(Series: U.S. Pacific Northwest Forest and Range Experiment Station, Portland, Or. Forest Service research paper PNW-69)

	DDC	CRD
	634.9/08 s	76-603289
Library of Congress	69 3	MARC

Figure 3. LC Manuscript Card

The format recognition process consists of five steps:

a. Establishes the framework of the MARC processing format and initializes fixed length areas.

b. Performs the preliminary identification of the input fields, breaking the data into clusters such as the title paragraph, the collation paragraph, etc. The collation is located first, and by working back through the data an attempt is made to locate and identify the call number, main entry, and title paragraph. Analysis then continues by examining the paragraphs following the collation paragraph, identifying fields when possible. The title paragraph is divided into title, edition, and imprint; the collation block into collation, series, and price. Tags are assigned to each field

identified and the tags are recorded in the preliminary record directory. The result of this step is that the variable fields in a MARC record have been located and partially identified. For many of the fields, the three-digit tag has been assigned, for others only two digits of the tag have been established. (For example, a note is recognized as a note but no attempt is made in this step to perform the required analysis for the type of note.)

c. Analyzes each variable field and extracts information from the data to assign field indicators and subfield codes and delimiters. The analysis in this step completes the assignment of the third digit to each tag. Advantage is taken of previously processed fields through a "look back approach." A "look ahead approach" of subsequent fields is used if the field was identified in step b and the tag was recorded in the directory.

d. Performs the analysis that can only be accomplished after all variable fields are processed, i.e., where identification involves relationships between fields. For example, the first indicator in the title field indicating whether a title added entry is needed, cannot be set until after the added entry fields have been processed.

e. Completes the record assembly from the various parts of the record built during the previous processing and outputs the record for further processing. The results of the format recognition program at this point are identical with the results of the content edit program.

6, 8. *Modified Update1, Modified Update2.* Performs the identical functions as described in 5 and 7, page 5, substituting disk for tape for data base storage. The only significant difference in Modified Update2 is that records requiring modification on the master file will be copied onto the working file for action. The working file record will be modified and remain on the working file until it is verified and then it is transferred back to the master file.

7, 9. *Print Index.* Performs the identical functions as described in 6 and 8, page 5.

10, 11. *Sort, Print Program.* Performs the identical functions as described in 9 and 10, pages 5 and 8.

12. *Purge Working File.* Builds working file (new) and new accessing tables.

13. *Purge Master File.* Builds master file (new) and new accessing tables. The program for purging the master file is the same for section III and is not repeated in the schematic representation.

14. *On-line Update.* Requests for on-line retrieval and/or the modification of MARC records residing on either the working file or the master file will be made via a cathode ray tube (CRT) device. The records will be displayed either on the CRT or a receive-only

teletypewriter. The teletypewriter will be used for the display of records that require more detailed examination.

All modifications will be made to records on the working file in the on-line mode, i.e., if a record is on the master file and a request is made for that record, a copy will be transferred to the working file. Three general types of commands will be implemented:

1. A display command will result in the record being displayed on one of the two output devices. No change will be made on the record.

2. A retrieve command will request the record for modification or verification.

a) If the record is currently residing on the working file, it will be located, read from disk to core, and a message sent to the user.

b) If the record is not on the working file, an attempt will be made to locate the record on the master file. If it is located, a copy will be transferred from the master file to the working file and the user notified. If the record is not located on the master file, the user will be advised of this condition. (The on-line system will have access to three disk packs, the working file, the master file index, and one disk pack of a multi-disk pack master file. Therefore, it is possible to request a record from the master file which is on a pack not mounted. The request will be placed in a queue. This queue is processed in an off-line mode and the requested records copied onto the working file for modification at some later time.)

3. A modify command will modify the retrieved working file record in one of four ways:

a) Correct: a record may be corrected by adding, replacing, or deleting data to fixed and variable fields. The modified record will be displayed on the CRT. A dialog will allow the user to further modify the record until he is satisfied, and can declare the record error free (verified) at this time.

b) Verify: a record may be declared error free. This will cause a change in the status of the record on the working file, signaling that this record may now be transferred to the master file.

c) Delete: all copies of this record should be deleted from the data base.

d) Scratch: this function is included as opposed to delete, to allow a record to be removed from the working file and to have no effect on the master file. This would allow MARC editors to train on-line.

15. *Update1.5* Builds the working file (new) and new accessing tables.

ORGANIZATION OF THE WORKING FILE

The working file will contain all those records in process, i.e., not yet verified or records recalled from the master file for updating. It is anticipated that the file will not be a voluminous file. (In the current distribution service, the working file usually contains about 3,000 records.) The working file will be maintained in LC card number sequence. When a record from the working file is modified, the new version of the record will be replaced over the old version if space permits. Otherwise, the old version will be scratched and a jumpout provided pointing to the location of the new version. Only whole records will be stored on a track. Continuation records (from one track to another) will only occur under the condition that the number of characters in a record exceeds the number of bytes in a track.

ACCESSING THE WORKING FILE

The accessing method used for the working file, called Milestone Tables, is similar in structure to the IBM Index Sequential Accessing Method.

There are two levels of Milestone Tables: the first level is the cylinder table which contains the LC card number of the last record in each cylinder, and the second level is the track table which contains the LC card number of the last record in each track of the cylinder. Each table entry in both the cylinder table and the track tables refers to the next sequential cylinder or track respectively. Therefore, no information other than the LC card number is required.

Since the working file will reside on a single disk pack, and there are 200 cylinders per pack, the cylinder track will never exceed more than 200 entries. An entry will consist of three alpha bytes for the alphabetic prefix of the LC card number and four pseudo-packed decimal bytes* for the numeric portion of the number. The cylinder table will contain 1,400 bytes maximum and can be maintained in memory during operation.

The track table, one per cylinder, will be maintained in the first track of each cylinder. Since there are twenty tracks per cylinder, the track table will never exceed more than twenty entries.

The LC card number for the requested record is searched against the cylinder table to determine which cylinder the record is in. The first track of that cylinder is read, and the card number searched against the track table to determine the track the record is in. The proper track is read, and the card number is compared with the card number of each record on the track until a match occurs. If a record has been modified and the new version has not been placed over the

*A pseudo-packed decimal byte is one that has no sign.

old version, the jumpout which points to the location of the new version may necessitate an additional read.

The number of accesses required to obtain the requested record is dependent on whether the record is in the first track of the cylinder and whether a jumpout is present in the record. If the record is in the first track of the cylinder, only one access is required; if the record is in other than the first track, two accesses are required. A jumpout situation will require the third access. Since there are twenty tracks per cylinder, the requested record will reside in the first track 5 percent of the time. Therefore, on the average, two accesses will be required to obtain a record on the working file.

ORGANIZATION OF THE MASTER FILE

The master file will be maintained in date-of-last-transaction order. This is necessary for the MARC Distribution Service. The Distribution Service is designed to supply weekly, quarterly, semi-annual and annual tapes, and must include corrections and deletions to records already distributed as well as new records. The date of the last transaction will be used and a range provided consisting of an initial date and a final date of the particular distribution cycle.

All records will be added to the end of the file, packed, and when required, the record continued onto the next track. (The length of the record exceeds the space available on the first track.) When a record is copied from the master file to the working file for modification, a "change in process" flag will be set in the record in the master file. This will guarantee that a record will not be distributed to subscribers if the record is being modified and has not been replaced on the master file prior to a distribution run. In addition, any access to the master file for this record could cause a signal to the user informing him of the status of the record. When a record is replaced on the master file, the modified record is added to the end of the file. An "obsolete" flag is set in the old record plus a jumpout inserted pointing to the location of the modified version of the record. A batch-processing purge master file program will remove obsolete records periodically as required. Since this program will build a new master file and index(es) onto new disk packs, the old packs can be retained for backup.

An index will point to a record in the master file, the pointer consisting of a pack/track number and the number of the record in the track. At the end of each track a location table of two bytes per record for each record on that track will be maintained in reverse sequence, giving the actual starting position of each record in the track.

Example:

Track i	R1	R2	R3		P3	P2	P1

R = record
P = starting position of record in track

ACCESSING THE MASTER FILE

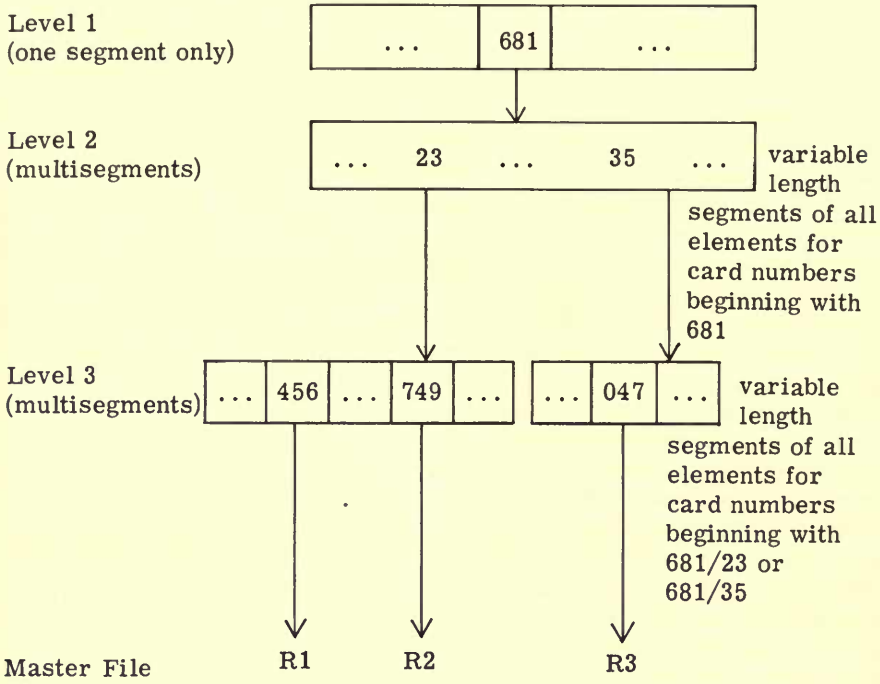
The primary access to the master file will be the LC card number. The LC card number, when expanded, is actually an eight-digit number. For example, 68-2698 expands to 68-002698. The eight digits will be divided into three parts. The first part will contain three digits, the second, two digits, and the third, three digits. The example given above would be divided as follows: 680/02/698. Since the probability is higher that all combinations of the last three digits of the card may be present, the division was made at this point. This should have the effect of shortening search time when the number is not in the index. The three parts of the number will be converted to binary to save storage space in the index; the three-digit numbers (000-999) requiring two bytes and the two-digit numbers (00-99), one byte, respectively.

The result is a three-level index and any record in the master file can be retrieved (assuming the master file pack is mounted at the time of search) in four disk accesses, including reading the track containing the record. The exception would be for those records containing alphabetic prefixes. The prefix would be considered a suffix in the index and would become the fourth level, if present. The seven-series card numbers do not include prefixes and, in the past, the number of records without prefixes far exceeded those with prefixes. Therefore, by considering the prefix the fourth level, in the majority of the cases, matching on a null prefix, i.e., a card number without a prefix, will not be necessary. The four-disk access described above and in the example below does not refer to a card number with a prefix.

The index will be a standard threaded-list tree structure in which an element in one level will point to a segment containing elements in the next level. Consider the following LC card numbers assigned to records:

Record Number	Card Number	Division of Card Number
R1	68-123456	681 23 456
R2	68-123749	681 23 749
R3	68-135047	681 35 047

The resulting index structure would be:



Each element within a segment contains information data as well as the data itself (the actual binary number portion of the divided LC card number). The information data will consist of 5 bytes assigned as follows:

Byte	Information Conveyed
1	Count of all records which can be reached from this element.*
2	Length of data portion of element (refers to byte 6 or bytes 6 and 7).
3-5	Link to segment in next lower level of index or to the master file record.

*This binary counter (1-250 and 251 meaning over 250) would equal the number of records reached if this branch of the index were traced through to completion. If this method of indexing is used when the system is expanded to allow

(Bytes 6 or 6 and 7 will contain the binary number portion of the LC card number. Whether one or two bytes will be needed depends on the level involved, i.e., level 1 and 3 require two bytes for the three-digit portion of the number and level 2, one byte for the two-digit portion of the number.)

The link portion (bytes 3-5) is constructed differently depending on whether the link is to the next lower level index or the master file record. The three bytes are considered to be twenty-four bits in length.

	Number of Bits	
Link Part	Lower Level Index	Master File Record
Flag	1	1
Pack number	1	6*
Track number in pack	12†	12†
Segment (for lower level index or record number) (for master file record in track)	10‡	5**
Total	24	24

searching by author, title, etc., this counter could serve as a guide to the user. For example, before requesting a list of books written by John Smith, the counter could be displayed and the search continued or not, depending on the resultant answer. The counter does not have meaning in the index of LC card numbers, but was included in consideration of using simpler techniques for all indexes in the system when it is expanded.

*The system is being designed projecting three million records in machine-readable form. This volume of cataloging data would require sixty-four packs. Thus, 6 bits ($63_{(10)}$) is adequate for the pack number. The one bit in the "lower level index" column is a "filler" bit.

†There are 4,000 tracks per disk pack. Thus, twelve bits ($4{,}095_{(10)}$) is adequate for the track number.

‡The minimum segment length on level one or three is six bytes plus a two byte location or nine bytes total. Therefore, the maximum number of segments per track is 7,294 bytes/9 or 810 segments per track. Thus, ten bits ($1{,}023_{(10)}$) is adequate for the segment number.

**The average MARC LC processing format record length is 633 bytes. Therefore, the number of records per track is 7,294 bytes/633 or approximately twelve records per track. Thus five bits ($31_{(10)}$) is adequate for the record number. (Allowance has been made for smaller records which will cause an increase in the number of records per track.)

Segments within a level will be variable in length. The end of a segment will be flagged with a ϕ in byte 1 (count of all records which can be reached from this element) since this byte by definition cannot contain a ϕ. If it is necessary to continue a segment onto another track, byte 6 will be set to ϕ and the bytes 3-5 (link) will point to the track where the segment continues. Segments will be packed into a track from left to right, with the same two bytes per record location table at the right-hand end of the track as described on page 21 for the master file organization.

S_1	S_2	S_3	---	L_3	L_2	L_1

S = Segment
L = Starting position of segment in track

Present thinking concerning additional indexes (author, title, etc.) includes these indexes referencing the LC card number which, in turn, will reference the master file. Thus, the system is open ended.

In the time period 1970-1971, the operation of the LC data utility will continue to be predominantly batch processing. However, we are proceeding as rapidly as possible to make on-line processing available to certain applications. The unified approach of the data utility as defined in this paper best employs the knowledge gained in developmental work and operational experience across all projects.

"Getting the show on the road" at the Library of Congress involves an effort far beyond the year 1971. Just the sheer logistics of reducing pertinent existing files to machine-readable form, as well as gearing up to include all current cataloging (approximately 200,000 titles per year) in the MARC system, will be a staggering undertaking.

If we were dealing with a simple system, we would start at the beginning and progress through each operation within LC. We would have minimal problems setting our sights and accomplishing our objectives. However, the fact is that given all the resources required, it still would not be possible to design and implement the total system in the near future. Therefore, we are attempting to place all ongoing projects and planned projects, operational or research and development, within a frame of reference, so that we have a common understanding of how we are developing and have the coordination required to do the job.

I cannot improve on Oettinger addressing this point:

> There is today a widely held point of view from which most anything, and education in particular, can be described as a collection or system of interdependent parts belonging to a

hierarchy in which a system may have subsystems of its own while acting as a mere part of a suprasystem. The process of analyzing or synthesizing such systems, called "systems analysis" for short, is touted as one of the shiniest of new technologies.

To some extent, speaking of systems is little more than appealing to a fashionable metaphor for the sake of snowing someone. Even then, this fad is not without merit as an antidote to that other pseudo-scientific fad, the precise and exhaustive analysis of an insignificant isolated effect under artificial conditions. Thinking "systems" at least reminds one that everything is related to everything else. Although always necessary in practice, ignoring any of these relations can be perilous; thinking "systems" alerts us to this peril.[6]

From the original input through the production of output (cards, MARC record, book catalogs), i.e., from the moment the book is in hand through the generation and use of the bibliographic record, each function has a high degree of dependence on every other function. In addition to this and many other technical difficulties (large character sets, complex voluminous files, etc.), we are concerned with the design and implementation of an information system over a span of time. We are planning and designing today, in terms of the potential for the future, and tied to all the work of the past. These are some of the factors that make the automation of libraries a task so difficult to accomplish.

REFERENCES

1. Leach, Theodore E. "A Compendium of the MARC System," *Library Resources & Technical Services*, 1:250-75, Summer 1968; U.S. Library of Congress. Information Systems Office. *The MARC Pilot Project: Final Report.* Prepared by Henriette D. Avram. Washington, D.C., Library of Congress, 1968, pp. 19-45; Rather, John C., and Pennington, Jerry G. "The MARC Sort Program," *Journal of Library Automation*, 2:125-38, Sept. 1969; Avram, Henriette D., *et al.* "MARC Program Research and Development: A Progress Report," *Journal of Library Automation*, 2:242-65, Dec. 1969; McCabe, Charles E. "Computer Applications in the Library of Congress Science and Technology Division." *In* American Society for Information Science. *Proceedings of the Annual Meeting.* Vol. 6. Washington, D.C., 1969, pp. 63-67; and Reimers, Paul R., and Avram, Henriette D. "Automation and the Library of Congress: 1970." (To be published in *Datamation.*)

2. Avram, *et al., op. cit.*, pp. 244-49.

3. U.S. Library of Congress. Information Systems Office. *The MARC II Format; A Communications Format for Bibliographic Data.* Prepared by Henriette D. Avram and Lucia J. Rather. Washington, D.C., Library of Congress, 1968, pp. 40-44.

4. Martin, James. *Design of Real-Time Computer Systems* (Prentice-Hall Series in Automatic Computation). Englewood Cliffs, N.J., Prentice-Hall, 1967, p. 148.

5. RECON Working Task Force. *Conversion of Retrospective Catalog Records to Machine-Readable Form; A Study of the Feasibility of a National Bibliographic Service*. Washington, D.C., Library of Congress, 1969, pp. 169-82; Avram, *et al., op. cit.*, pp. 250-53; and Avram, Henriette D. "The RECON Pilot Project: A Progress Report." (To be published in *Journal of Library Automation.*)

6. Oettinger, Anthony G. *Run, Computer, Run; The Mythology of Educational Innovation* (Harvard Studies in Technology and Society). Cambridge, Mass., Harvard University Press, 1969, p. 53.