Techniques for Special Processing of Data within Bibliographic Text

Paula GOOSSENS: Royal Library Albert I, Brussels, Belgium.

An analysis of the codification practices of bibliographic descriptions reveals a multiplicity of ways to solve the problem of the special processing of certain characters within a bibliographic element.

To obtain a clear insight into this subject, a review of the techniques used in different systems is given. The basic principles of each technique are stated, examples are given, and advantages and disadvantages are weighed. Simple local applications as well as more ambitious shared cataloging projects are considered.

INTRODUCTION

Effective library automation should be based on a one-time manual input of the bibliographic descriptions, with multiple output functions. These objectives may be met by introducing a logical coding technique. The higher the requirements of the output, the more sophisticated the storage coding has to be. In most cases a simple identification of the bibliographic elements is not sufficient. The requirement of a minimum of flexibility in filing and printing operations necessitates the ability to locate certain groups of characters within these elements. It is our aim, in this article, to give a review of the techniques solving this last problem.

As an introduction, the basic bibliographic element coding methods are roughly schematized in the first section. According to the precision in the element identification, a distinction is made between two groups, called respectively field level and subfield level systems. The second section contains discussions on the techniques for special processing of data within bibliographic text. Three basic groups are treated: the duplication method, the internal coding techniques, and the automatic handling techniques. The different studies are illustrated with examples of existing systems. For the field level projects we confined ourselves to some important German and Belgian applications. In the choice of the subfield level systems, which are MARC II based, we tried to be more complete. Most of the cited applications, for practical reasons, only concern the treatment of monographs. This cannot be seen as a limitation because the methods discussed are very

general by nature and may be used for other material. Each system which has recourse to different special processing techniques is discussed in terms of each of these techniques, enabling one to get a realistic overview of the problem. In the last section, a table of the systems versus the techniques used is given. The material studied in this paper provided us with the necessary background for building an internal coding technique in our internal processing format.

BIBLIOGRAPHIC ELEMENT CODIFICATION METHODS

Field Level Systems

The most rudimentary projects of catalog automation are limited to a coarse division of the bibliographic description into broad fields. These are marked by special supplied codes and cover the basic elements of author, title, imprint, collation, etc. In some of the field level systems, a bibliographic element may be further differentiated according to a more specific content designation, or according to a function identification. For instance, the author element can be split up into personal name and corporate name, or a distinction can be made between a main entry, an added entry, a reference, etc.

This approach supports only the treatment of each identified bibliographic element as a whole for all necessary processing operations, filing and printing included. This explains why, in certain applications, some of the bibliographic elements are duplicated, under a variant form, according to the subsequent treatments reflected in the output functions. Details on this will be discussed later. Here we only mention as an example the Deutsche Bibliographie and the project developed at the University of Bochum.[1–4]

It is evident that these procedures are limited in their possibilities and are not economical if applied to very voluminous bibliographic files. For this reason, at the same time, more sophisticated systems, using internal coding techniques, came into existence. These allow one to perform separate operations within a bibliographic element, based on a special indication of certain character strings within the text. As there is an overlap in the types of internal coding techniques used in the field level systems and in the subfield level systems, this problem will later be studied as a whole. We limit ourselves to citing some projects falling under this heading. As German applications we have the Deutsche Bibliographie and the BIKAS system.[5] In Belgium the programs of the Quetelet Fonds may be mentioned.[6, 7]

Subfield Level Systems

In a subfield level system the basic bibliographic elements, separated into fields, are further subdivided into smaller logical units called subfields. For instance, a personal name is broken into a surname, a forename, a numeration, a title, etc. Such a working method provides access to smaller logical units and will greatly facilitate the functions of extraction, sup-

pression, and transposition. Thus, more flexibility in the processing of the bibliographic records is obtained.

As is well known, the Library of Congress accomplished the pioneering work in developing the MARC II format: the communications format and the internal processing format.[8–11] These will be called MARC LC and a distinction between the two will only be made if necessary. The MARC LC project originated in the context of a shared cataloging program and immediately served as a model in different national bibliographies and in public and university libraries. In this paper we will discuss BNB MARC of the British National Bibliography, the NYPL automated bibliographic system of the New York Public Library, MONOCLE of the library of the University of Grenoble, Canadian MARC, and FBR (Forma Bibliothecae Regiae), the internal processing format of the Royal Library of Belgium.[12–21]

In order to further optimize the coding of a bibliographic description, the Library of Congress also provided for each field two special codes, called indicators. The function of these indicators differs from field to field. For example, in a personal name one of the indicators describes the type of name, to wit: forename, single surname, multiple surname, and name of family. Some of the indicators may act as an internal code.

In spite of the well-considered structuring of the bibliographic data in the subfield level systems, not all library objectives may yet be satisfied. To reduce the remaining limitations, some approaches similar to those elaborated in field level systems are supplied. Some (NYPL, MARC LC internal format, and Canadian MARC) have, or will have, in a very limited way, recourse to a procedure of duplication of subfields or fields. All cited systems, except NYPL, use to a greater or lesser degree internal coding techniques. Finally some subfield level systems automatically solve certain filing problems by computer algorithms. This option was taken by NYPL, MARC LC, and BNB MARC. Each of these methods will be discussed in detail in the next section.

TECHNIQUES FOR SPECIAL PROCESSING OF DATA

Methods for special treatment of words or characters within bibliographic text were for the most part introduced to support exact file arrangement procedures and printing operations. In order to give concrete form to the following explanation, we will illustrate some complex cases. Each example contains the printing form and the filing form according to specific cataloging practices for some bibliographic elements.

Consider the titles in examples 1, 2, and 3, and the surnames in examples 4, 5, and 6.

Example 1: L'Automation des bibliothèques
AUTOMATION BIBLIOTHEQUES

Example 2: Bulletino della R. Accademia Medica di Roma
BOLLETINO ACCADEMIA MEDICA ROMA

Example 3: IBM 360 Assembler language
I B M THREE HUNDRED SIXTY ASSEMBLER LANGUAGE

Example 4: Mc Kelvy
MACKELVY

Example 5: Van de Castele
VANDECASTELE

Example 6: Martin du Gard
MARTIN DUGARD

We do not intend, in this paper, to review the well-known basic rules for building a sort key (the translation of lowercase characters to uppercase, the completion of numerics, etc.). Our attention is directed to the character strings that file differently than they are spelled in the printing form. The methods developed to meet these problems are of a very different nature. For reasons of space, not all the examples will be reconsidered in every case; only those most meaningful for the specific application will be chosen.

Duplication Methods

We briefly repeat that this method consists of the duplication of certain bibliographic elements in variant forms, each of them exactly corresponding to a certain type of treatment. In Bochum, the title data are handled in this way. One field, called "Sachtitel," contains the filing form of the title followed by the year of edition. Another field, named "Titelbeschreibung," includes the printing form of the title and the other elements necessary for the identification of a work (statements of authorship, edition statement, imprint, series statement, etc.). To apply this procedure to examples 1, 2, and 3, the different forms of each title respectively have to be stored in a printing field and in a sorting field. Analogous procedures are, in a more limited way, employed in the Deutsche Bibliographie. For instance, in addition to the imprint, the name of the publisher is stored in a separate field to facilitate the creation of publisher indexes. The technique of the duplication of bibliographic elements has also been considered in subfield level systems. The NYPL format furnishes a filing subfield in those fields needed for the creation of the sort key. This special subfield is generally created by program, although in exceptional cases manual input may be necessary. In the filing subfield the text is preceded by a special character indicating whether or not the subfield has been introduced manually. MARC LC (internal format) and Canadian MARC opt for a more flexible approach in which the filing information is specified with the same precision as the other information. The sorting data are stored in complete fields containing, among others, the same subfields as the corresponding original field.

Because in most subfield level systems the number of different fields is much higher than in field level systems, the duplication method becomes more intricate. Provision of a separately coded field for each normal field

which may need filing information is excluded. Only one filing field is supplied, which is repeatable and stored after the other fields. In order to link the sorting fields with the original fields, specific procedures have been devised. MARC LC, for instance, reserves one byte per field, the sorting field code, to announce the presence or the absence of a related sorting field. The link between the fields themselves is placed in a special subfield of the filing field.[22] In the supposition that examples 3 and 4 originate from the same bibliographical description, this method may be illustrated schematically as follows:

tag	sorting field code	sequence number	data
100	x	1	aMc Kelvy
245	x	1	aIBM 360 Assembler Language
880		1	ja1001$MacKelvy
880		2	ja2451$I B M Three hundred sixty Assembler Language

As is well known, the personal author and title fields are coded respectively as tag 100 and tag 245. Tag 880 defines a filing field. In the second column, the letter x identifies the presence of a related sorting field. The third column contains a tag sequence number needed for the unequivocal identification of a field. In the last column the sign $ is a delimiter. The first $ is followed by the different subfield codes. The other delimiters initiate the subsequent subfields. In tag 100 and 245, the first subfields contain the surname and the short title respectively. In tag 880 the first subfield gives the identification number of the related original field. The further subfield subdivision is exactly the same as in the original fields. In Canadian MARC a slightly different approach has been worked out. Note that in neither of the last two projects has this technique been implemented yet.

For an evaluation of the duplication method different means of application must be considered. If not systematically used for several bibliographic elements, the method is very easy at input. The cataloger can fill in the data exactly as they are; no special codes must be imbedded in the text. But it is easy to understand that a more frequent need of duplicated data renders the cataloging work very cumbersome. In regard to information processing, this method consumes much storage space. First, a certain percentage of the data is repeated; second, in the most complete approach of the subfield level systems, space is needed for identifying and linking information. For instance, in MARC LC, one byte per field is provided containing the sorting field code, even if no filing information at all is present. Finally, programming efforts are also burdened by the need for special linking procedures.

In order to minimize the use of the duplication technique, the cited systems reduce their application in different ways. Bochum simplified its cataloging rules in order to limit its use to title information. As will be explained further, the Deutsche Bibliographie also has recourse to internal

coding techniques. NYPL, MARC LC, and Canadian MARC only call on it if other more efficient methods (see later) fail. They also make an attempt to adapt existing cataloging practices to an unmodified machine handling of nonduplicated and minimally coded data.

Internal Coding Techniques

Separators

Separators are special codes introduced within the text, identifying the characters to be treated in a special way. A distinction can be made among four procedures.

1. Simple separators. With this method, each special action to be performed on a limited character string is indicated by a group of two identical separators, each represented as a single special sign. Illustration on examples 2, 3, 4, and 6 gives:

   Example 2: £Bolletino £¢Bulletino della R. ¢Accademia Medica ¢di ¢Roma

   Example 3: £I B M three hundred sixty £¢IBM 360 ¢Assembler Language

   Example 4: M£a£c¢ ¢Kelvy

   Example 6: Martin du¢ ¢Gard

   The characters enclosed between each group of two corresponding codes £ must be omitted for printing operations. In the same way the characters enclosed between two corresponding codes ¢ are to be ignored in the process of filing. In the case that only the starting position of a special action has to be indicated, one separator is sufficient. For instance, if in example 1 we limit ourselves to coding the first character to be taken into account for filing operations, we have:

   Example 1: L'/Automation des bibliothèques

   where a slash is used as sorting instruction code.

   The simple separator method has tempting positive aspects. Occupying a minimum of storage space (maximum two bytes for each instruction), the technique gives a large range of processing possibilities. Indeed, excluding the limitation on the number of special signs available as separators, no other restrictions are imposed. This argument will be rated at its true worth only after evaluation of the multiple function separators method and of the indicator techniques. The major disadvantage of the simple separator method lies in its slowness of exploitation. In fact, for every treatment to be performed, each data element which may contain special codes has to be scanned, character by character, to localize the separators within the text and to enable the execution of the appropriate instructions. For example, in the case of a printing operation, the program has to identify the parts of the text to be considered and to remove all separators. The sluggishness of

execution was for some, as for Canadian MARC, a reason to disapprove this method.[23] As already mentioned, another handicap with cataloging applications is the loss of a number of characters caused by their use as special codes. It is self-evident that each character needed as a separator cannot be used as an ordinary character in the text. For Bochum this was a motive to reject this method.

Many of the field level systems with internal codes have recourse to simple separators. We mention the Deutsche Bibliographie, in which some separators indicate the keywords serving for automatic creation of indexes and others give the necessary commands for font changes in photocomposition applications. In order to reduce the number of special signs, the Deutsche Bibliographie also duplicates certain bibliographic data. BIKAS uses simple separators for filing purposes. The technique is also employed in subfield level systems. In MONOCLE each title field contains a slash, indicating the first character to be taken into account for filing.

2. Multiple function separators. Designed by the British, the technique of the multiple function separators was adopted in MONOCLE. The basic idea consists of the use of one separator characteristic for instructing multiple actions. In the case of MONOCLE these actions are printing only, filing only, and both printing and filing. In order to give concrete form to this method we apply it to examples 3, 4, and 6, using a vertical bar as special code.

Example 3: |IBM 360 |I B M three hundred sixty |Assembler Language
Example 4: M|c |ac|Kelvy
Example 6: Martin du| ||Gard

The so-called three-bar filing system divides a data element into the following parts:

data to be filed and printed	data to be printed only	data to be filed only	data to be filed and printed

In comparison with the simple separator technique, this method has the advantage of needing fewer special characters. A gain of storage space cannot be assumed directly. As is the case in example 6, if only one special instruction is needed, the set of three separators must still be used. On the other hand, one must note that a repetition of identical groups of multiple function separators within one data element must be avoided. Subsequent use of these codes leads to very unclear representations of the text and may cause faulty data storage. This can well be proved if the necessary groups of three bars are inserted in examples 1 and 2. Of the studied systems, MONOCLE is the only one to use this method.

3. Separators with indicators. As mentioned in the description of subfield level systems, two indicators are added for each field present. In

order to speed up the processing time in separator applications, indicators may be exploited. In MONOCLE the presence or the absence of three bars in a subfield is signalled by an indicator at the beginning of the corresponding field. This avoids the systematic search for separators within all the subfields that may contain special codes.

The number of indicators being limited, it is self-evident that in certain fields they may already be used for other purposes. As a result, some of the separators will be identified at the beginning of the field and others not. This leads to a certain heterogeneity in the general system concept which complicates the programming efforts.

Under this heading, we have mentioned the use of indicators only in connection with multiple function separators. Note that this procedure could be applied as well in simple separator methods. Nevertheless, none of the subfield level systems performs in this fashion because it is not necessary for the particular applications. This method is not followed in the field level systems as no indicators are provided.

4. Compound separators. A means of avoiding the second disadvantage of the simple separator technique is to represent each separator by a two-character code: the first one, a delimiter, identifies the presence of the separator and is common to each of them; the second one, a normal character, identifies the separator's characteristic. Taking the sign £ as delimiter and indicating the functions of nonprinting and nonfiling respectively by the characters a and b, examples 2 and 4 give in this case:

Example 2: £aBolletino £a£bBulletino della R. £bAccademia Medica £bdi £bRoma
Example 4: M£aa£ac£b £bKelvy

Thus the number of reserved special characters is reduced to one, independent of the number of different types of separators needed. In none of the considered projects is this technique used, probably because of the amount of storage space wasted.

Indicators

As the concept of adding indicators in a bibliographic record format is an innovation of MARC LC, the methods described under this heading concern only subfield level systems. Although at the moment of the creation of MARC LC one did not anticipate the systematic use of indicators for filing, its adherents made good use of them for this purpose.

1. Personal name type indicator. As mentioned earlier, in MARC LC one of the indicators, in the field of a personal name, provides information on the name type. This enables one to realize special file arrangements. For example, in the case of homonyms, the names consisting only of a forename can be filed before identical surnames. Using the same indicator, an exact sort sequence can be obtained for

single surnames, including prefixes. Knowing that the printing form of example 5 is a single surname, the program for building the sort key can ignore the two spaces. The systems derived from MARC LC developed analog indicator codifications adapted to their own requirements.

This seems to be an elegant method for solving particular filing problems in personal names. Nevertheless, its possibilities are not large enough to give full satisfaction. For instance, example 6 gives a multiple surname with prefix in the second part of the name. The statement of multiple surname in the indicator does not give enough information to create the exact sort form. Because of this shortcoming, MONOCLE had recourse to the technique called "separators with indicators."

2. Indicators identifying the beginning of filing text. BNB MARC reserves one indicator in the title field for identification of the first character of the title to be considered for filing. This indicator is a digit between zero and nine, giving the number of characters to be skipped at the beginning of the text. Applying this technique to example 1, the corresponding filing indicator must have the value three. Without having recourse to other working methods, this title sorts as:

   Example 1: AUTOMATION DES BIBLIOTHEQUES

   Notice that the article *des* still remains in the filing form.

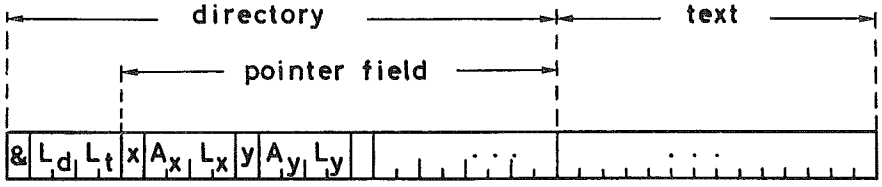
   This procedure has the advantage of being very economical in storage space and in processing time. Moreover the text is not cluttered with extraneous characters. On the other hand we must disapprove of the limitation of this technique to the indication of nonfiling words at the beginning of a field. The possibility of identifying certain character strings within the text is not provided for. Taking examples 2 and 3 we observe that the stated conditions cannot be fulfilled. Another negative side is the number of characters to be ignored, which may not exceed nine. Also one indicator must be available for this filing indication. After BNB MARC, MARC LC and Canadian MARC also introduced this technique.

3. Separators with indicators. The use of indicators in combination with separators has been treated above.

Pointers

A final internal coding technique which seems worth studying is the one developed at the Royal Library of Belgium for the creation of the catalogs of the library of the Quetelet Fonds, a field level system. The pointer technique is rather intricate at input but has many advantages at output. Because there is inadequate documentation of this working method, we will try to give an insight into it by schematizing the procedures to be followed to create the final storage structure. At input, the cataloger in-

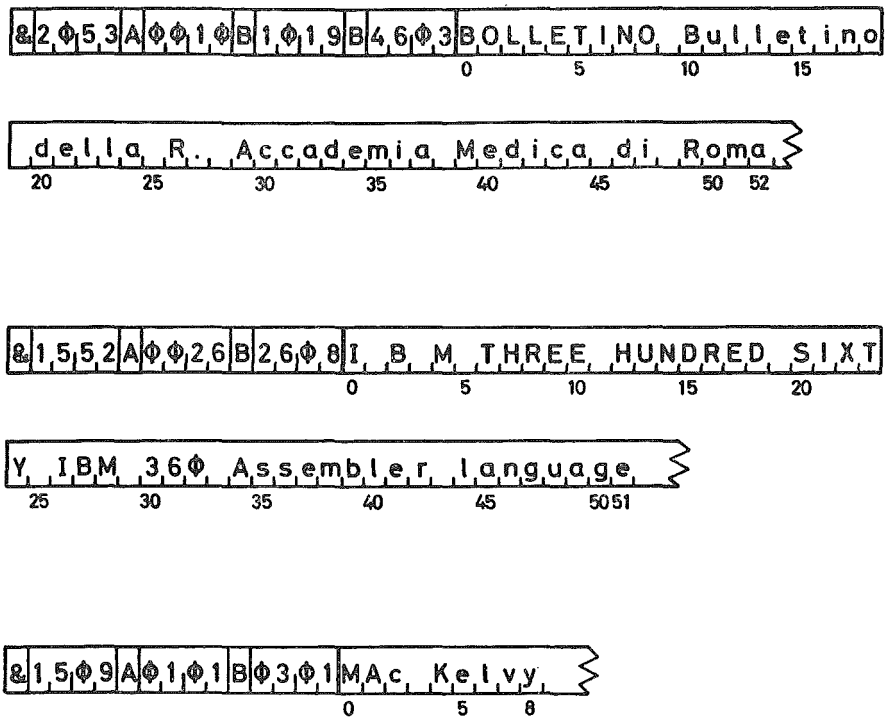
serts the necessary internal codes as simple separators within the text. These codes are extracted by program from the text and placed before it, at the beginning of each field. Each separator, now called *pointer characteristic,* is supplemented with the absolute beginning address and the length of its action area within the text. In the Quetelet Fonds the pointer characteristic is represented by one character, the address and length occupy two bytes each. The complete set of pointers (pointer characteristics, lengths, and addresses) is named pointer field. This field is incorporated in a sort of directory, starting with the sign "&" identifying the beginning of the field, followed by the length of the directory, the length of the text, and the pointer field itself. This is illustrated in Figure 1. Note that each field contains the five first bytes, even if no pointers are present. In the Quetelet Fonds, pointers are used for the following purposes: nonfiling, nonprinting, KWIC index, indication of a corporate name in the title of a periodical, etc. Examples 2, 3, and 4 should be stored in this system as represented in Figure 2.



Representation of the structure of a field in the internal processing format of the Quetelet Fonds system. The codes respectively represent: &: field delimiter; Ld: length of directory; Lt: length of text; x, y, . . . : pointer characteristics; Ax, Ay, . . . : addresses of the beginning of the related action area inside the text; Lx, Ly, . . . : length of these action areas.

Fig. 1. Structure of Directory with Pointer Technique.

The advantages of the pointer technique are numerous. First, we must mention the relative rapidity of the processing of the records. In fact, in order to detect a specific pointer, only the directory has to be consulted. All subsequent instructions can be executed immediately. In contrast with most of the other methods discussed, there is no objection to using pointers for all internal coding purposes needed. This enables one to pursue homogeneity in the storage format, facilitating the development of programs. Further, the physical separation of the internal codes and the text allow, in most cases, a direct clean text representation without any reformatting. Finally, unpredictable expansions of internal coding processes can easily be added without adaptation of the existing software. A great disadvantage of the pointer technique lies in the creation of the directory. The storage space occupied by the pointers is also great in comparison with the place occupied by internal codes in other methods. A further handicap is the limitation imposed at input due to the use of simple separators.



Representation of examples 2, 3, and 4 in the Quetelet Fonds format. A represents the pointer characteristic for nonprinting data; B is the pointer characteristic for nonfiling data.

Fig. 2. Pointer Technique as Applied to Bibliographic Data.

In spite of these negative arguments, we see a great interest in this method, and wish to give some suggestions in order to relieve or to eliminate some of them. Initially we must realize that the creation of a record takes place only once, while the applications are innumerable. The possibility of automatically adding some of the codes may also be considered. Data needing special treatment expressed in a consistent set of logical rules can be coded by program. Only exceptions have to be treated manually. In considering the space occupied by the directory, some profit could be imagined by trying to reduce the storage space occupied by the addresses and the lengths. There is also a solution to be found by not having systematically to provide pointer field information. One must realize that only a small percentage of the fields may contain such codes. Finally, the restrictions at input may be removed by using complex separators. Such a change does not have any repercussion on the directory.

As far as we know, the pointer technique has not been used in a subfield level system. At our library an internal processing format of the subfield level type, called FBR, is under development, in which a pointer technique based on the foregoing is incorporated.

Automatic Handling Techniques

In order to give a complete review of the methods of handling data within bibliographic text, we must also treat the methods in which both the identification and the special treatment of these data are done during the execution of the output programs. The working method can easily be demonstrated with example 1. Only the printing form must be recorded. The program for building the sort key processes a look-up table of nonfiling words including the articles *L'* and *des*. The program checks every word of the printing form for a match with one of the words of the nonfiling list. The sort key is built up with all the words which are not present in this table. To treat example 4, an analogous procedure can be worked through. An equivalence list of words for which the filing form differs from the printing form is needed. If, during the construction of the sort key, a match is found with a word in the equivalence list, the correct filing form, stored in this list, is placed in the sort key. The other words are taken in their printing form. In our case, using the equivalence list, Mc should be replaced by MAC. In order to speed up the look-up procedures, different methods of organization of the look-up tables can be devised. Other types of automatic processing techniques can be illustrated by the special filing algorithms constructed for a correct sort of dates. For instance, in order to be able to sort B.C. and A.D. dates in a chronological order, the year 0 is replaced by the year 5000. B.C. and A.D. dates are respectively subtracted from or added to this number. Thus dates back to 5000 B.C. can be correctly treated. This technique, introduced by NYPL, is also used at LC.

The advantages of automatic handling techniques are many. No special arrangements must be made at input. Only the bibliographic elements must be introduced under the printing form and no special codes have to be added. There is no storage space wasted for storing internal codes. As negative aspects we ascertain that not all cataloging rules may be expressed in rigid systematic process steps. Examples 2 and 3 illustrate this point. One must also recognize that the special automatic handling programs must be executed repeatedly when a sort key is built up, increasing the processing time. This procedure may give some help for filing purposes, but we can hardly imagine that it really may solve all internal coding problems. Think of the instructions to be given for the choice of character type while working with a type setting machine. The automatic handling technique is very extensively applied in the NYPL programs, MARC LC has recourse to it for treating dates, and BNB MARC for personal names.[24] None of the field level systems considered here uses this method.

SUMMARY AND CONCLUSIONS

Table 1 presents, for the discussed systems, a summary of the methods used for treating data in a bibliographic text. The duplication and indicator techniques have the most adherents. However, we must keep in mind

Table 1. Review of the Techniques for Special Processing of Data within Bibliographic Text Used or Planned in the Discussed Systems

Systems	*Techniques*							
	Duplication	*Internal Codes*						*Automatic Handling*
		Separators		*Separators with Indicators*	*Indicators*		*Pointers*	
		Simple	*Multiple Function*		*Personal Name Type*	*Beginning of Filing Text*		
Deutsche Bibliographie	X	X						
Bochum	X							
BIKAS								
Quetelet Fonds							X	
MARC LC	X				X	X		X
BNB MARC					X	X		X
NYPL	X							X
MONOCLE		X	X	X	X			
Canadian MARC	X				X	X		
FBR							X	

that in most of the systems the duplication of data only represents an extreme solution. On the other hand, indicators are very limited in their possibilities. As far as the flexibility and application possibilities are concerned, the simple separators and the pointers present the most interesting prospects. Automatic handling techniques may produce good results for use in well-defined fields or subfields.

From the evaluations given for the different methods, we conclude that for a special application the choice of a method depends greatly on the objectives, namely the sort of special processing facilities needed, the volume of data to be treated, and the frequency of execution.

REFERENCES

1. Rudolf Blum, "Die maschinelle Herstellung der Deutschen Bibliographie in bibliothekarischer Sicht," *Zeitschrift für Bibliothekswesen und Bibliographie* 13:303–21 (1966).
2. *Die ZMD in Frankfurt am Main;* Herausgegeben von Klaus Schneider (Berlin: Beuth-Vertrieb GmbH, 1969), p.133–37, 162–67.
3. Magnetbanddienst Deutsche Bibliographie, *Beschreibung für 7-Spur-Magnetbänder* (Frankfurt on the Main: Zentralstelle für maschinelle Documentation, 1972).
4. Ingeborg Sobottke, "Rationalisierung der alphabetischen Katalogisierung," in *Electronische Datenverarbeitung in der Universitätsbibliothek Bochum;* Herausgegeben in Verbindung mit der Pressestelle der Ruhr-Universität Bochum von Günther Pflug und Bernhard Adams (Bochum: Druck- und Verlagshaus Schürmann & Klagges, 1968), p.24–32.
5. *Datenerfassung und Datenverarbeitung in der Universitätsbibliothek Bielefeld: Eine Materialsammlung;* Hrsg. von Elke Bonness und Harro Heim (Munich: Pullach, 1972).
6. Michel Bartholomeus, *L'aspect informatique de la catalographie automatique* (Brussels: Bibliothèque royale Albert Ier, 1970).
7. M. Bartholomeus and M. Hansart, *Lecture des entrées bibliographiques sous format 80 colonnes et création de l'enregistrement standard;* publication interne: Mecono BO15A (Brussels: Bibliothèque royale Albert Ier, 1969).
8. Henriette D. Avram, John F. Knapp, and Lucia J. Rather, *The MARC II Format: A Communications Format for Bibliographic Data* (Washington, D.C.: Library of Congress, 1968).
9. *Books, a MARC Format: Specifications for Magnetic Tapes Containing Catalog Records for Books* (5th ed.; Washington, D.C.: Library of Congress, 1972).
10. "Automation Activities in the Processing Department of the Library of Congress," *Library Resources & Technical Services* 16:195–239 (Spring 1972).
11. L. E. Leonard and L. J. Rather, *Internal MARC Format Specifications for Books* (3d ed.; Washington, D.C.: Library of Congress, 1972).
12. *MARC Record Service Proposals* (BNB Documentation Service Publications no.1 [London: Council of the British National Bibliography, Ltd., 1968]).
13. *MARC II Specifications* (BNB Documentation Service Publications no.2 [London: Council of the British National Bibliography, Ltd., 1969]).
14. Michael Gorman and John E. Linford, *Description of the BNB MARC Record—A Manual of Practice* (London: Council of the British National Bibliography, Ltd., 1971).

15. Edward Duncan, "Computer Filing at the New York Public Library," in *Larc Reports* vol.3, no.3 (1970), p.66–72.
16. *NYPL Automated Bibliographic System Overview. Internal Report.* (New York: New York Public Library, 1972).
17. Marc Chauveinc, *Monocle: Projet de mise en ordinateur d'une notice catalographique de livre.* Deuxième édition (Grenoble: Bibliothèque universitaire, 1972).
18. Marc Chauveinc, "Monocle," *Journal of Library Automation* 4:113–28 (Sept. 1971).
19. *Canadian MARC* (Ottawa: National Library of Canada, 1972).
20. *Format de communication du MARC Canadien: Monographies* (Ottawa: Bibliothèque nationale du Canada, 1973).
21. To be published.
22. Private communications (1973).
23. Private communications (1972).
24. Private communications (1973).