International Journal on Document Analysis and Recognition manuscript No.
(will be inserted by the editor)

Retrospective Document Conversion

Application to the Library Domain

A. Belaïd

LORIA UMR 7503 Campus scientifique B.P. 239 54506 Vandœuvre-lès-Nancy Cedex France
email: Abdel.Belaid@loria.fr

Received March 10, 1998 / Revised August 12, 1998

Abstract. This paper describes a framework for retrospective document conversion in the library domain. Drawing on the experience and insight gained from projects launched over the present decade by the European Commission, it outlines the requirements for solving the problem of retroconversion and traces the main phases of associated processing. To highlight the main problems encountered in this area, the paper also outlines studies conducted by our group in the MORE project for the retroconversion of old catalogues belonging to two different Libraries : National French Library and Royal Belgian Library. For the French Library, the idea was to study the feasibility of a recognition approach avoiding the use of OCR and basing the strategy mainly on visual features. The challenge was to recognize a logical structure from its physical aspects. The modest results obtained from experiments for this first study led us, in the second study, to base the structural recognition methodology more on the logical aspects by focussing the analysis on the content. Furthermore, for the Belgian references, the aim was to convert reference catalogues into a more conventional UNIMARC format while respecting the industrial constraints.Without manual intervention, 75% rate of correct recognition was obtained on 11 catalogues containing about 4548 references.

Key words: Retrospective Conversion, Library Catalogue, Reference Recognition, Structure Analysis, OCR, UNIMARC

1 Introduction

The success of library automation, resulting in user-friendly on-line catalogues[1] integrated with the WEB and other circulation-systems facilities, has created an urgent need for retroconversion of the older parts of catalogues [2,14,28,31]. As users get familiar with the new catalogue medium, the documents not registered in machine-readable form become "invisible" and unreadable. This has meant for many libraries the relegation of an important part of their rich stock of documents to a state of inaccessibility.

Such obvious waste of library collections in addition to the cost difference between manual handling and an equivalent set of automatic routines has made a strong case for the need to convert a library's entire collection of works to machine-readable records, in the interest of ensuring an efficient use of the investment in the new technology.

This has led to the search for cost-effective tools for the conversion of old catalogues into machine-readable forms. This search has not been limited to the sole problem of conversion but has been extended to embracing other objectives such as ensuring very high rates of distribution and sharing of documents between several libraries.

In this framework, the European commission launched a big library program in order to help libraries in the use of modern data processing [10,11,13]. Its specific objective was to promote:

- the availability and the accessibility of modern library services throughout the community, taking into account existing geographic discrepancies in library provision;
- a more rapid penetration of information and communications technology in cost-effective way;
- the standardisation required for resource sharing among libraries;
- the harmonisation and convergence of national policies for libraries.

The plan involves four lines of action within which a range of shared-cost cooperative projects could be launched. The first line is more specifically connected with the retroconversion problem. It proposes projects to create, enhance and harmonize machine-readable bibliographies (principally national bibliographies used for international bibliographic services) and union catalogues as well as the development of the necessary tools and methods for retrospective conversion of catalogues of internationally important collections.

[1] A catalogue is a list of bibliographic descriptions of works.

Three projects, based on OCR/ICR approaches, were launched in order to cover the main problems encountered in the automatic conversion of catalogues such as:

- the search for a common format to harmonize the representation of different kinds of references and for a system of indexing and classifying, in BIBLIOTHECA project [11];
- the search for OCR packages adapted to the problem of retroconversion involving a large set of characters and tools for fast and cheap mass conversion of card catalogue[2], in FACIT project [22,35,37];
- and the study of the role and use of dictionaries in the structure modelling and recognition of catalogues by OCR techniques in MORE project [5,26].

Drawing heavily on the experience and insight gained from MORE[3] [4,26], and particularly on the various reports and recommendations published on the other two projects, this paper outlines the main phases of retroconversion for a real production chain and states the relevant requirements of the retroconversion operation on such a chain.

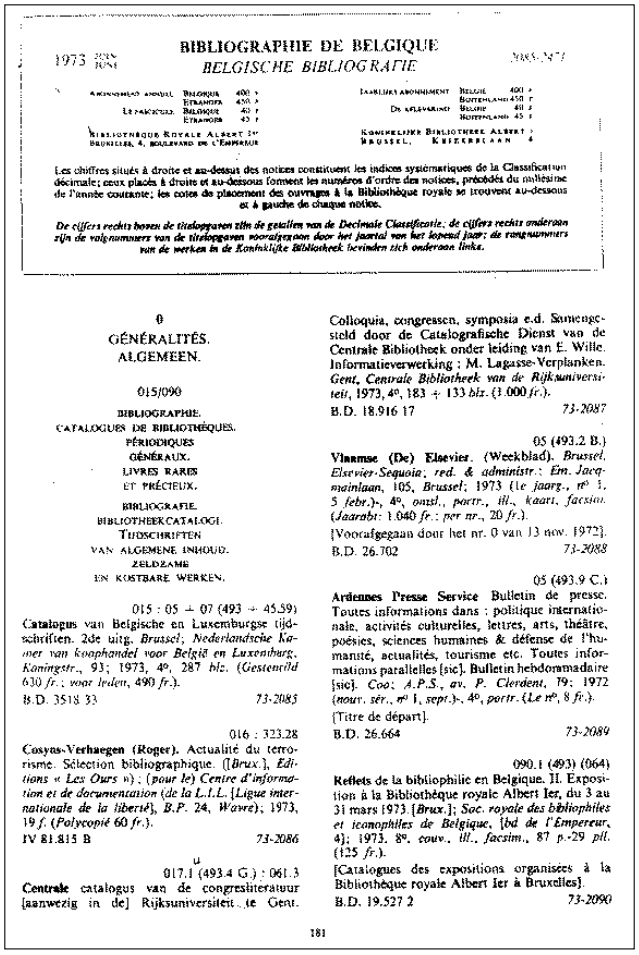
The paper is divided into four main sections.

The first section briefly outlines the cataloguing specification and describes in general terms the bibliography organization and gives some basic concepts used in this area.

The second section tries to sketch what could be a recognition process for the bibliography problem and describes its main phases.

The last two sections are devoted to the description of the experiments conducted on the recognition process for the two libraries studied. Each section reviews the library specification, the methodology used and the results obtained from some selected references.

In the conclusion, we will give a synthesis on the retroconversion problem and propose some perspective in a larger domain such as digital library.



BIBLIOGRAPHIE DE BELGIQUE
BELGISCHE BIBLIOGRAFIE

0
GÉNÉRALITÉS.
ALGEMEEN.

015/090
BIBLIOGRAPHIE.
CATALOGUES DE BIBLIOTHÈQUES.
PÉRIODIQUES
GÉNÉRAUX.
LIVRES RARES
ET PRÉCIEUX.
BIBLIOGRAFIE.
BIBLIOTHEEKCATALOGI.
TIJDSCHRIFTEN
VAN ALGEMENE INHOUD.
ZELDZAME
EN KOSTBARE WERKEN.

015 : 05 + 07 (493 + 45.59)
Catalogus van Belgische en Luxemburgse tijdschriften. 2de uitg. *Brussel; Nederlandsche Kamer van koophandel voor België en Luxemburg, Koningstr.*, 93; 1973, 4°, 287 blz. (*Gestencild 630 fr.; voor leden, 490 fr.*).
B.D. 3518 33 73-2085

016 : 323.28
Cosyns-Verhaegen (Roger). Actualité du terrorisme. Sélection bibliographique. ([*Brux.*], *Editions « Les Ours »*) ; (*pour le*) *Centre d'information et de documentation* (*de la L.I.L.* [*Ligue internationale de la liberté*], *B.P.* 24, *Wavre*); 1973, 19 *f.* (*Polycopié 60 fr.*).
IV 81.815 B 73-2086

017.1 (493.4 G.) : 061.3
Centrale catalogus van de congresliteratuur [aanwezig in de] Rijksuniversiteit te Gent. Colloquia, congressen, symposia e.d. Samengesteld door de Catalografische Dienst van de Centrale Bibliotheek onder leiding van E. Wille. Informatieverwerking : M. Lagasse-Verplanken. *Gent, Centrale Bibliotheek van de Rijksuniversiteit*, 1973, 4°, 183 + 133 blz. (1.000 *fr.*).
B.D. 18.916 17 73-2087

05 (493.2 B.)
Vlaamse (De) Elsevier. (Weekblad). *Brussel, Elsevier-Sequoia; red. & administr.: Em. Jacqmainlaan*, 105, *Brussel*; 1973 (*1e jaarg., nr 1, 5 febr.*)-, 4°, *ongepag., portr., ill., kaart, facsim.* (*Jaarabt: 1.040 fr.; per nr., 20 fr.*).
[Voorafgegaan door het nr. 0 van 13 nov. 1972].
B.D. 26.702 73-2088

05 (493.9 C.)
Ardennes Presse Service Bulletin de presse. Toutes informations dans : politique internationale, activités culturelles, lettres, arts, théâtre, poésies, sciences humaines & défense de l'humanité, actualités, tourisme etc. Toutes informations parallelles [sic]. Bulletin hebdommadaire [sic]. *Coo; A.P.S., av. P. Clerdent*, 79; 1972 (*nouv. sér., nº 1, sept.*)-, 4°, *portr.* (*Le nº, 8 fr.*).
[Titre de départ].
B.D. 26.664 73-2089

090.1 (493) (064)
Reflets de la bibliophilie en Belgique. II. Exposition à la Bibliothèque royale Albert Ier, du 3 au 31 mars 1973. [*Brux.*]; *Soc. royale des bibliophiles et iconophiles de Belgique*, [*bd de l'Empereur, 4*]; 1973, 8°, *couv., ill., facsim.*, 87 *p.*-29 *pl.* (125 *fr.*).
[Catalogues des expositions organisées à la Bibliothèque royale Albert Ier à Bruxelles].
B.D. 19.527 2 73-2090

181

Fig. 1. Example of the first page of the Belgian Library Catalogue in bookform.

2 Cataloguing Specification

2.1 Bibliography Organization

A bibliography is organized into catalogues as a list of bibliographic descriptions of items according to a set of cataloguing rules set up by the library to give access to the items. Items can be stored physically on paper cards in alphabetical order with one item per card. They can also be stored on microfiches, in bookform or as electronic records.

The content of an item depends on the type of catalogue : alphabetic, dictionary, systematic, topographical, etc. and on the type of materials described : books (monographs), periodicals (serials), maps, prints, etc. An alphabetic catalogue is organized by headings (author's names and titles), including main entries[4], added entries[5] and abbreviated entries[6]. A dictionary catalogue is a catalogue with subject headings, as well as authors' names, titles and reference headings are organized in an alphabetic sequence. A systematic catalogue is a catalogue organized according to a classification system such as Universal Decimal Classification (UDC). Finally, a topographical catalogue is organized according to the geographical or topographical areas described in the works.

Figure 1 gives an example of the first page of the Belgian catalogue in bookform. Its organization is of the UDC type. This catalogue is divided into monthly catalogues. Each one contains references led to a specific theme and stored on two columns as separated line blocks.

[2] A catalogue whose elements are cards

[3] Marc Optical REcognition, supported by the same organism between 1992 and 1994.

[4] The entry in the catalogue containing the fullest bibliographic description and the prefered heading (main author or title according to the cataloguing rules used)

[5] An entry supplementing the main entry to give extra access points

[6] Usually an added entry (title, secondary author, translator or subject)

2.2 Reference Specification

The macrostructure organization of all bibliographical catalogues can be divided into references. Each reference may itself be divided into elements which can be coded into a readable format on the target machine.

The bibliographical reference contents of the catalogues share a number of similar features just as do the bibliographical reference contents of catalogues belonging to the same library but covering different periods. Normalized in form as they are today, all bibliographical references invariably contain information which either guides the reader in his research or advertises published documents available elsewhere, *e.g.* in national bibliographies.

The Structure

The reference structure obeys some rules relating to the *physical*, the *logical* and the *semantical* level. Figure 2 gives a real example of a reference from the Belgian catalogue and shows its logical description.

- The physical level describes the reference in terms of consecutive areas (i.e. format). It gives for each area the number of characters or digits, the descriptor and the interest of the information contained.
- The logical level gives the nature of the information put in each area such as "*title*" in the first area, "*author*" in the second, "*edition*" in the third, etc.
- The semantical level corresponds to the specific cataloguing rules of each library. It gives the formal and informal rules in producing a specific catalogue (electronic). These rules determine what bibliographic elements to select, how to formulate the entries and how to represent them in the electronic catalogue.

The Standards

The middle of the 19th century saw the birth of an international movement towards the unification of catalogue construction [34]. This marked the beginning of a progressive development of research tools culminating in the ISBD[7] of the nineteen-seventies, representing an international effort to harmonize cataloguing rules [17–19].

ISBD is a standard which specifies for each type of document:

- the full list of information elements that a complete bibliographical reference can contain along with the associated hierarchical structure,
- the sequential conventional notation for the normalized presentation of these elements (initially on paper support). ISBD rules aim among other things at a presentation of bibliographical information which favours easy user understanding without the need to be familiar with the language of publication, specific names, etc.

This normalization not only led to the emergence of machine-readable bibliographical format, but was also

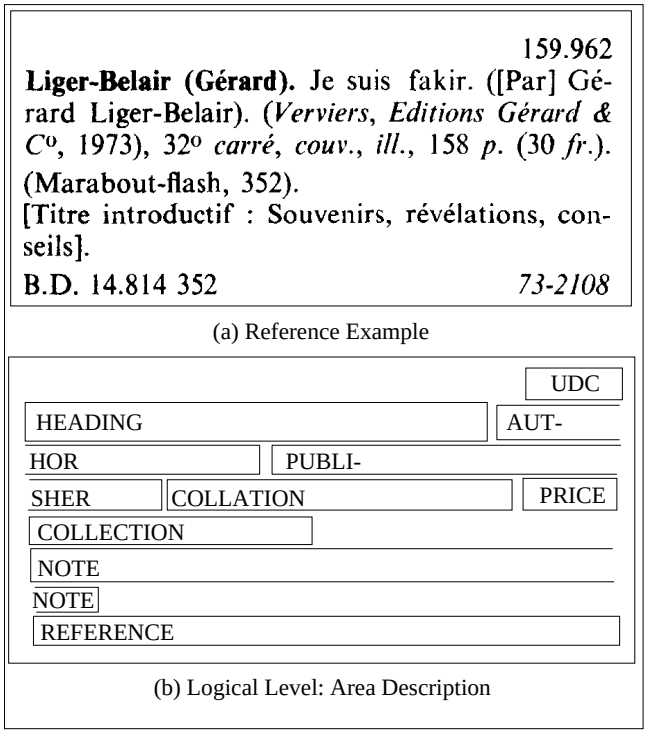


Fig. 2. Reference Structure.

instrumental in making it compatible with the 1976 international UNIMARC[8] format. UNIMARC is an attempt to standardize the different varieties of the MARC[9] format in use in the U.S.A. and several European countries [9,15]. As seen in the example of the figure 3 for the same reference given in 2.a, each field in UNIMARC has a three-digit code. Fields correspond to UDC, heading, author, etc. Each code is followed by an information giving the importance and the origin of the field coded on two or three digits. Subfields, when they exist, are given by means of a specific letter preceded by the dollar symbol.

The ISBD punctuation facilitates the labelling of bibliographical information during catalogue conversion into machine-readable records. The segmentation of pre-ISBD written references into bibliographical units is still possible, as the realization of catalogues almost invariably follows the same rules which have been preserved and have evolved into national and international harmonization.

2.3 The Specific Problems in Recognition

The use of generic tools to manipulate bibliographical information almost invariably poses the same problems. These are related to the following facts:

- *Heterogeneous Content.* The reference catalogue is usually produced over a long period of time during which the cataloguing rules change. It contains references produced by different cataloguing agencies

[7] International Standard Bibliographic Description.

[8] UNIversal MAchine Readable Catalogues.

[9] MAchineReadable Cataloguing format.

FIELDS	UNIMARC
UDC Vedette	<675 I=bb <$a 159.962</$a></675> <200 I=0b <$f Gérard Liger-Belair</$f> <$a Je suis fakir </$a>,/200>
Author	<700 I=b0 <$a Liger Belair</$a> <$b Gérard</$b></700>
Publisher	<210 I=bb <$a Verviers</$a> <$c Editions Gérard & CO]</$c> <$d [1973]</$d> </210>
Collation	<215 I=bb <$d 320 carré</$d> <$c couv.,i11.</$c> <$a 158 p.</$a></215>
Price Collection	<010 I=bb <$d 30 BEF</$d></010> <225 I=2b <$a Marabout-flash</4a> <$v 352</$v></225>
Note	<517 I=0i1<$a Souvenirs, révélations, conseils</$a></517>
Ref.	<900 I=bb <$a B.D.14.814352</$a> <$b 73-2108</$b></900>

Fig. 3. UNIMARC Coding.

each applying their own rules. Many catalogues to be converted contain many different types of references: main entry references with headings representing authors or titles. Added entries by secondary authors, title, subjects, etc. Entries covering more than one reference. The system must be able to differentiate between these types and handle the information according to the type.

– *Typographic imperfections*: Bibliographical information is made up of text containing a large number of abbreviated words, not only in the document language but in the cataloguing language as well. It also contains numerical information, sometimes in Roman numerals, and an important quantity of names. To these must be added the multiplicity of languages and the use of a wide range of stressed characters in contrast with Latin writing styles. There is higher frequency of punctuation marks than in ordinary text. In addition to their natural role, punctuation marks are used as separators to delimit logical elements of information. The presence of several similar character sets such as hyphens and long dashes, parentheses and square brackets, further increases their frequency. Printed catalogues make use of typography to differentiate between sets of elements belonging to the same logical category. Unlike card catalogues, the layout is more elaborate, including systematic justification of text, variable spacing, and at times word cutting at the end of line. Some of the word cuts belong to the publication language of the catalogue to convert.
– *Linguistic variabilities*: The recognition of some fields depends on the recognition of some key words in specific lexicons. In these lexicons we can find all the cataloguing vocabulary and all the words that exist in bibliographical work titles and insertions concerning the "authorship responsibility". Punctuation is usually less reliable than that of ISBD. Some words are related to the publication language (title fields, edition, address, collection) and others are related to the cataloguing language (collation and notes). Finally, all the words have to be taken into account in a complete form and also in an abbreviated form, knowing that they were not normalized at the time of the tests.
– *Higher Density of Structure*: The main problem posed by the bibliographical references resides in the density of their logical structure and the multiplicity of the choice of information sequences. In fact, several cataloguing entities are optional and repetitive. These information elements are required only for the cataloguer if the information exists in the catalogued document. Furthermore, these elements can depend on the kind of the document and of course on the kind of references, such as "monograph" or periodical publications, or as in certain catalogues, on "principal" or "secondary" reference. Finally, a practice inherited from printed catalogues is at the root of the current use of punctuation marks as a means of condensed representation of information. The ISBD normalization on the international level further reinforces this.

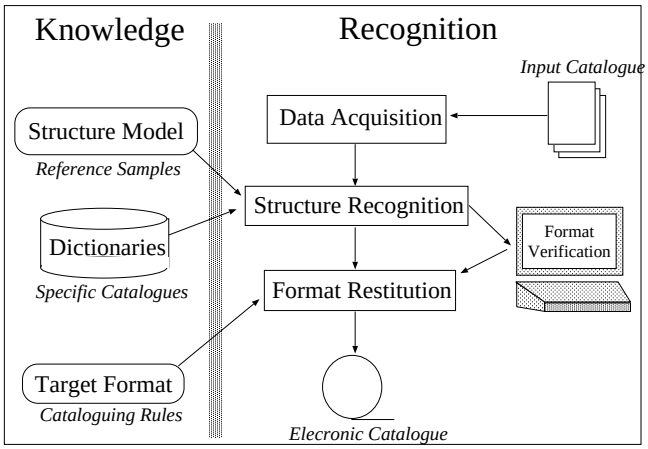


Fig. 4. Retroconversion System Overview.

3 Retroconversion System

Figure 4 shows the main phases of the recognition processing of references. In the following sections, we briefly describe the different components and outline what is general for the two applications performed.

3.1 Data Acquisition

The main problems with handling catalogues are related to the automatic feeding of the pages or cards, the existence of cards printed on both sides, and the variable quality of machine written cards.

For catalogues in bookform as in the MORE project, pages are separated and fed separately. The project has

identified scanners capable of handling a great quantity of pages at acceptable speed. In fact, the speed of the scanning process does not depend entirely on the scanner itself, but also on the controller page as well as the speed of the controlling system. For the resolution and because of the variations in printing quality, many tests were operated in order to determine the acceptable resolution (in our case 300 dpi) which can be used for all the pages of the catalogue.

Data acquisition also includes data formatting. Individually pasting into pages alters the reference images (skew angle, font changing, cut or connected characters, etc.). In order to take into account such particularities, specific algorithms had to be developed [12] to handle:

- *Skew Correction*: As references are pasted manually within columns and have their own typography, a skew detection technique is applied separately on each block after the segmentation procedure. The skew detection technique makes use of connectivity analysis as well as of grouping stage to partition the connected components (*cc's*) into homogeneous group blocks containing the references. We used the Baird technique [7] for the determination of the skew angle by maximizing the histogram projection of the midpoint of the bottom side of the cc bounding box. The objective function is computed as the sum of the squares of the profile bins.
- *Block Segmentation*. It is based on connected components analysis and studies for each set of *cc's*, the classes of different lengths of spaces between the *cc's*, as well as their size and regularity. The analysis is done in two steps.
  In the first step, *cc's* are merged into sets of approximately aligned *cc's*. For example, a text line can be partitioned into three sets of *cc's*, the first for accents and apostrophes, the second for letters and the third for punctuation (cf. fig. 5). In this manner, two successive text lines are never merged, and large connected components are easily isolated. The *cc's* in each set are analyzed individually if they are few, or globally otherwise.
  In the global analysis, the width of the *cc's* as well as the space between them are studied. If there are more than three types of different lengths of spaces, the analysis is recursively applied to the two sets of *cc's* around the largest space (this allows the separation of two columns, for example). If there is a *cc* whose width is much larger than those of the rest, it is separated and analyzed apart.
  In the second step, the sets obtained in the previous step are globally analyzed with respect to their neighbors in order to either correct the errors of the previous classification or to merge similar sets into lines. At last lines are merged into blocks by considering line spaces parameters. More details on the segmentation algorithm can be found in [3].
- *Text Conversion*. Text is seen in this application as a series of visual features each one composed of text (connected characters or words) and a number of typographic parameters such as font style, mode, punctuation, etc. Different extraction procedures for these visual features are applied in the two versions of the system.
  In the one concerning the French National Library, a pattern matching approach is performed for the extraction of all visual features using specific templates trained from samples. This did not involve OCR.
  In the study concerning the Belgian Library, different commercial OCR systems were used. They produced better results for the feature extraction [16, 27,32,33,36]. Because of the large number of characters used in catalogues, including letters with diacritics and special symbols (such as Greek characters), a retroconversion OCR package must be able to handle the 16-bit UNICODE or a near subset [21,23].

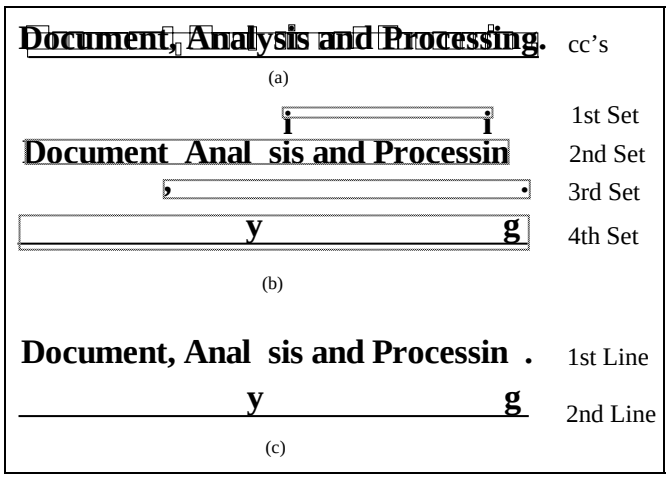


Fig. 5. Connected Component Sets.

3.2 Dictionaries

The number of languages found in the the same catalogue prohibits the use of large standard dictionaries (with 200. - 300.000 wordforms). It is normal to find 15 different languages in the same catalogue. The prototype for retroconversion must include some specific procedure for identifying the language of the current field (in general more apparent in the title).

For verification of words and strings, a set of general and specific dictionaries are provided by the user. Specific dictionaries include: marks, list of publication places (with country codes), list of typical names (family names, special names, like name of kings, princes, popes, list of publishers' names, list of typical words and phrases used in cataloguing such as "*edited by*", "S.L.", "S.A." (with indication of languages where that words occur). The prototype must include routines for dictionary look up in order to validate the word spelling.

3.3 Structure Analysis

The main difficulty is to segment the reference text into a hierarchy of fields and sub-fields (called logical structure) according to the standard. This requires the use

of a structure model for the reference class and an analysis approach that is able to extract from the image a valid instance for this model. For the structure analysis, the idea is to make use of a generic model which informs about the general appearance of the fields in the data strings. Because of the poverty of the physical structure, the emphasis is rather put on the logical structure which is more informative. The hierarchy is exhibited by generic constructors and qualifiers highlighting the different structure occurrences. Because of the presence of "repetitive" and "optional" cases complicating the problem of field separation, the structure analysis is based on segmentation hypotheses management .

We applied this schema on the two different libraries. Since the needs are not similar, we used two different approaches for the analysis. In the first case, the method is influenced by the image which orients the analysis to a merging process analysis into sub-fields and fields using visual features. In the second case, the strategy is more influenced by the model which conducts the text segmentation process in a top-down manner.

3.4 Structure Model

The model is derived from the examination of different reference samples. It describes the generic structure in terms of a hierarchy of fields and sub-fields.

Knowing that the problem is to find the sub-fields within reference areas, the model specification concentrated on the description of sub-field properties, by the distinction of their typographic styles, the existence of particular words or group of words and their appearance in certain lexicons, and especially their limits (type of initials and finals such as capital letters, particular words or type of punctuation separating the sub-fields).

3.5 Error Correction

As the objective is to produce a valuable electronic catalogue capable of providing useful help to the user in his bibliography search and consultation, the final catalogue must contain as few error as possible. Routines for error detection and correction will have to be adapted to the error conditions of the actual catalogue, scanner, OCR, character representation in the plain text, and essentially data formatting and logical structure of the reference highlighting main and secondary entries of the references. This means that the application will have to be presented with the result of the error analysis in a suitable form [22].

3.6 Target Format

After the formatting and correction of errors, the resulting records will have to be converted from the internal format into the target bibliographic format and a character set acceptable to the cataloguing system where reference will be used. This always involves the use of two specific tables:

- a specific table showing the system how to tag the current bibliographic element in conformity with the target format. When the conversion is not straightforward, one will then have to resort to some specific procedures to realize the conversion;
- a conversion table for character representation. This table tells the system how the characters in the internal 16-bit character set are to be represented in the output file. UNIMARC is expected to support UNICODE characters in the foreseeable future, in which case conversion will no longer be needed.

4 The French Library

In this study, the idea was to explore the feasibility of a recognition approach avoiding the use of OCR and basing the strategy mainly on visual features. Visual features are first computed from the original image (particular characters, style of words, numbers, etc.). Then, labels corresponding to these features are searched in the generic model and a list of labels is attached to each word of the analyzed reference. On this chain of words (with their possible labels), a neighbourhood constraint propagation method is applied to prune the list of possible labels. The recognition process terminates with a mixed analysis which gives the hierarchical organization of the structure recognized [6].

4.1 Structural aspects

The French Library catalogue is composed of 23 volumes containing about 1000 pages printed in recto-verso. This corresponds to the total of 550 000 references for the year 1973.

Reference Classes

There are two reference classes:

- **Bibliographic references**: *principal* (P), *secondary* (S), *analytic* (A) and *collection* (C). These represent 97% of the references.
- **Link references**: *link reference of title* (T), of *congress editing* (T), and the others (R). They represent 3% of references.

The bibliographic references P, S or A always comprise a general reference, eventually followed by subreferences in volume (cf. figure 6) for 5% of references with an average of 3 volumes. In this last case, it is a question of references with *deprivation* (D). The bibliographic references C comprise only one general reference.

Reference Typography

There are two different typographic sets for the references with equal percentage. This particularity may complicate the visual features extraction in images and principally the search of possible separators between the different fields of a reference. In fact:

- the *printed references*. They concern French books. The style (italic, underlined, bold) and the mode

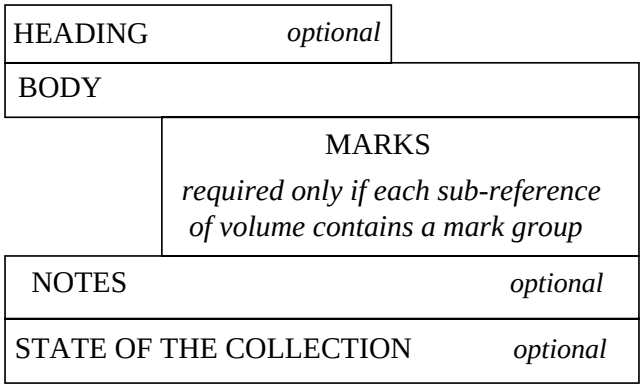


Fig. 6. The Layout of the French National Library References.

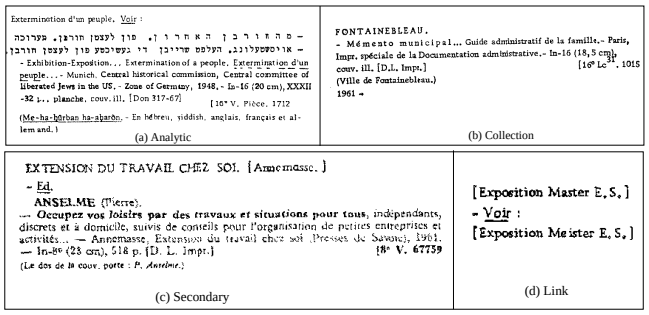


Fig. 7. Examples of French National Library References.

(upper-case, lower-case, small lower-case) are pertinent indications for the field location;
- the *typed references*. They concern foreign books and link references. Here the "pertinent indices" are richer than those of printed references.

Logical Structure

The layout structure is very poor, essentially limited to some typographic differences. The logical structure on the other hand is much denser providing a different content architecture for each kind of reference.

Figure 7 gives examples of important references. Principal references represent more than 60% of the references to convert. A principal reference is very complete containing in the *heading*, an *author* name or the beginning of a *title* without such mentions as `Ed.` or `Collab`, and without such links as `Voir`, `In`, `Class`, `Devenu`, etc. A secondary reference is similar to a principal one except that it contains function mentions in the author entry. An analytic reference is characterized by an author-title entry followed by the mention `Voir` or `In`.

4.2 Structure Modelling

The problem is to find the sub-fields within reference areas. The model specification therefore concentrated on the description of sub-field properties based on the distinction of typographic styles, the existence of particular words or group of words and their appearance in certain lexicons, and especially their limits (type of initials and finals such as capital letters, particular words or type of punctuation separating the sub-fields).

The model is given in the form of a context-free grammar written in the EBNF formalism. The syntax of a production rule is as follows :

Term	::=	Constructor Sub_Objects[Qualifier] | Constant | Terminal
Constructor	::=	seq_td | seq_lr | seq | aggr | cho | import
Separator		Name Sub_Objects
Attributes		[Name Weight]$^+$

Constructors and qualifiers

A term, the left hand of a rule, can either be simple (constant or terminal) or composed of subordinate objects. In the latter case, a constructor describes the relationship between objects. The constructor specifies the order of the appearance of subordinate objects such as SEQUENCE: top-down (*seq_td*), left-right (*seq_lr*) or logical (*seq*), AGGREGATE (*aggr*) or CHOICE (*cho*). A special constructor, "*import*", is used to inherit for the term some or the total description of another term. Furthermore, to express the object occurrence in the term, each object may be accompanied by a qualifier such as OPTIONAL (*opt*), REPETITIVE (*rep*) and OPTIONAL-CONDITIONAL (*optc*) giving the condition under which an object may appear.

Separators

As the structure is not sufficiently complete for characterizing the fields and separating them, the limits between consecutive fields are introduced to reinforce the field description. Separators can be *specific punctuation marks* such as point, comma, bracket, parenthesis, etc. *mode changing* (Capital letter in the beginning of the field), numeric area, *font style changing*, etc.

Attributes

Because of the poverty of the physical structure and the multitude of choices represented in the model, some attributes given in the library specification were added to the previous description to enhance the description of the reference components.

Several kinds of attributes have been defined: *Type* (string, line, word, char, etc.), *Mode* (capital, numeric, alphabetic, punctuation, etc.), *Style* (bold, italic, standard, etc.), *Position* (beginning of line, inside, end), *Lexicon* affiliation (author index, countries, towns, abbreviations, articles, etc.), *Weight* which specifies the degree of importance of subordinate objects, etc.

The model contains about 150 objects, mostly content fragments representing both characters and words. Figure 8 describes some of the model rules implemented in `Common-Lisp`.

The first rule (cf. figure 8.a) describes the frame **ReferenceB** as a choice (**cho**) between 7 references plus **ReferenceCV5** (for Reference Collection in Volume) and **Sub_Reference_Volume**. The latter is optional-conditional in the sense that it cannot participate in the choice if its condition is not verified. This condition is provided by the Library cataloguing rules indicating that

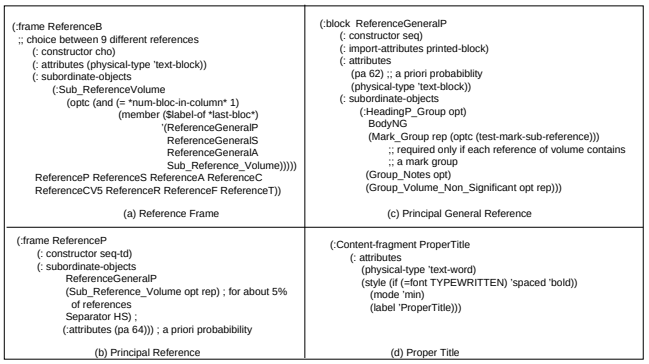


Fig. 8. Examples of Model Rules.

a page or a column always begins by a new reference, except for references like P, S or A. So, a sub-reference in a volume cannot appear as a separate block except when located at the beginning of a column and follows another sub-reference in a volume when it is the general reference of a reference P, S or A.

A reference P (cf. figure 8.b) is described as a top-down sequence (`seq-td`) of a principal general reference. It may be followed by a list of references in volume. The latter, if they exist, are separated from the general reference and from the others sub-references by a horizonal space (`HS`), described in the model as a particular fragment (cf. figure 8.b.).

Going through the micro-structure within the reference content, we can observe in figure 8.c the internal structure of the principal general reference. It is composed of a sequence of logical entities, without information on the orientation (top-down or left-right). A Mark group can for example appear on the same line as the last line of the reference body. This sequence deals with:

- a heading group which is optional (author entry, collectivity, etc);
- a reference body which is required (qualifier by default), this body is composed with fragments: "title", "address-date" and "collation" fragment;
- a mark fragment which can be repetitive and required, except if the reference contains sub-references in volume and if each one of them contains a mark group. The condition "`test-mark-sub-reference`" gives a constraint for the validation of a principal reference;
- a note group (notes between parentheses);
- an optional list of non-significant volumes (without mark and where body does not completely contain "title", "address-date", "collation").

The most refined level of the description gives more information on the content. For example, in figure 8.d, we describe a terminal fragment representing a word belonging to the "proper title". The example shows how we represent a title with a bold style in the case of printed reference, or "spaced" if the reference is typewritten. The variable `font` controls the change of the font style during the analysis.

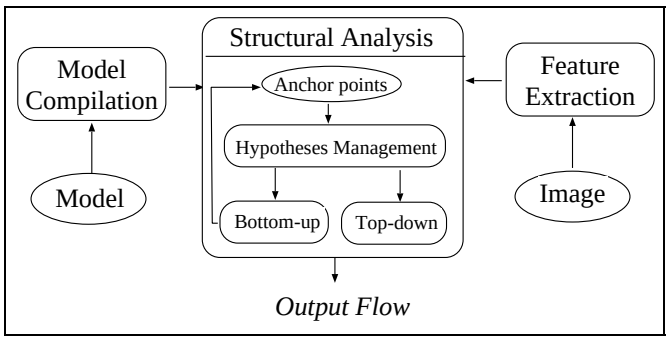


Fig. 9. System Overview.

4.3 System Overview

Figure 9 shows the principal components of the recognition system. As presented in the schema, the extraction of features plays a central role in the running of the process. The next section briefly describes the different components.

4.4 Structural Analysis

The strategy is driven at the same time by the model and by anchor points extracted from the visual features of the current reference. It operates in a bottom-up / top-down scheme. For each anchor point, the system proposes in a bottom-up manner the most probable model hypothesis and tries to verify in a top-down manner its left and right contexts. This strategy is adapted to an item where the beginning is noisy and does not favour a top-down scheme. This strategy is efficient only when the number of anchor points and the number of the grammar hypotheses are limited which is not the case of the references processed. In fact, visual features are numerous and not sufficient for generating the relevant anchor points. This is so for different reasons: finer degree of some features such as the punctuation generating ambiguity, irrelevant answers stemming from the style, and paucity of the physical representation where some content fragments have no physical characteristics. Consequently, we use as anchor points tokens that minimize the number of hypotheses.

4.5 Model Compilation

This step allows the transformation of the model into a more directly usable structure. The model is first parsed in order to extract the visual indices to be searched in the references (separators, styles, etc.). Then, for each object o of the model, three sets are built: the set of *initials* (I_o^*), the set of finals (F_o^*) and the set of compatible neighborhood. These sets will be used during the constraint propagation and mixed analysis stages.

Target Features

Each generic object of the model (content fragment) is observed so that a list of specific attributes is summarized. These attributes correspond to pertinent visual

features to be searched for in the image, such as the style (italic, bold, underlined, spaced), the mode (capitals, numbers) or the size of the words. Furthermore, special characters correspond to separators between fields and subfields. The list of these separators is generated during the compilation stage and is added to the list of pertinent visual features to search for.

Initials and Finals

The model can be viewed from a formal point of view as a grammar $G = (V_n, V_t, P, S)$ where P is the set of production rules of the model, V_n is the set of non terminal objects, V_t is the set of terminal objects (which cannot be decomposed any more) and S is the starting axiom (the reference).

Let S_a be the set of subordinate objects of a (the right part of a rule where a forms the left part). The set of initials of a (I_a) is defined as the subset of S_a where each element can appear in first position according to the construction (choice, sequence, aggregate). By extension, I_a^*, the transitive closure of I_a, can be recursively defined as follows:

if $a \in V_t$ then $I_a^* = \{a\}$ else $I_a^* = I_a \cup (\cup_{i \in I_a} I_i^*)$.

The set of finals of a (F_a) and its transitive closure F_a^* are defined in a same manner but correspond to elements that can appear in last position. I_a^* and F_a^* are extracted for each object a during the compilation stage.

Compatible Neighborhood

Let $N_{l_{a,p}}$ be the set of the possible neighbors at the left of a in the rule:
$p \rightarrow \lambda a \delta$, with $\lambda \in (V_t \cup V_n)^*$, and $\delta \in ((V_t \cup V_n) - \{a\})^*$.
A fictitious rule $\Lambda \rightarrow \lambda$ is imagined so that $N_{l_{a,p}}$ can easily be recursively defined as follow:

if $a \notin S_\Lambda$ then $N_{l_{a,p}} = F_\Lambda$ else $N_{l_{a,p}} = F_\Lambda \cup N_{l_{a,\Lambda}}$.

$N_{r_{a,p}}$, the set of the possible neighbors at the right of a in the rule:
$p \rightarrow \lambda a \delta$, with $\lambda \in ((V_t \cup V_n) - \{a\})^*$, and $\delta \in (V_t \cup V_n)^*$.
is defined in a same manner. A particular case concerns repetitive objects. In this case the result is $N_{l_{a,p}} \cup \{a\}$ and $N_{r_{a,p}} \cup \{a\}$ because repetitive objects are their own neighbor.

Let $nl_{a,p}^*$ be the transitive closure of the left neighborhood finals and $nr_{a,p}^*$ the transitive closure of the right neighborhood initials of a in p.

$$nl_{a,p}^* = \cup_{l \in N_{l_{a,p}}} F_l^* \qquad nr_{a,p}^* = \cup_{r \in N_{r_{a,p}}} I_r^*.$$

These neighborhoods have to be generalized to each rule where a appears in the right. NL_a^* and NR_a^* are the left and right neighborhood of a in the model and are defined as follow:

$$NL_a^* = \cup_{p \in E(a)} nl_{a,p}^* \qquad NR_a^* = \cup_{p \in E(a)} nr_{a,p}^*.$$

where $E(a)$ represent the set of possible father for a, that is the set of the left part of the rules where a appears in the right. Finally, the neighborhood compatibilities of two objects A and B (see figure 10) are defined as follows:

- A is right compatible with B if $B \in NR_A^*$ (case (a) and (c)) or $A \in NL_B^*$ (case (a) and (d)) or $\exists P_A \in E(A)$ and $\exists P_B \in E(B)$ / P_A is right compatible with P_B (case (b));

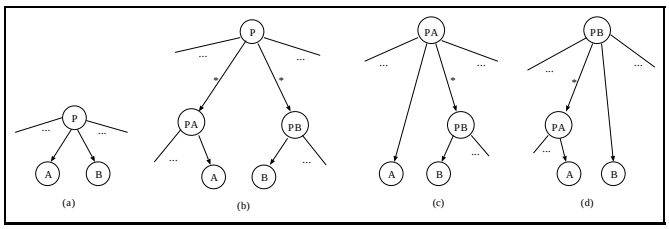


Fig. 10. Neighborhood compatibility between two content fragment A and B.

- B is left compatible with A if $B \in NL_A^*$ or $A \in NR_B^*$ or $\exists P_A \in E(A)$ and $\exists P_B \in E(B)$ / P_A is left compatible with P_B;
- furthermore, A is right compatible with B $\Longleftrightarrow$ B is left compatible with A.

During the compilation stage, the set of couples $((i,p),(j,p'))$ are extracted from the model. They represent for each couple of objects i and j a father (non-terminal) p (i.e. $p \rightarrow \lambda i \lambda'$) of i compatible with a father p' of j. This last set contains about 3700 pairs for the model of French references.

4.6 Extraction the Visual Features

Visual features extracted during the compilation of the model are observed on each data component (token). As this method does not involve OCR, specific tools had to be developed to recognize typographic styles (italic, bold, underline, spaced letters), font family (printed or typed), mode (number, capital, etc.), separator characters (punctuation, parentheses, arrows, particular symbols, etc.) and particular words such as cross-references responsibility mentions (`Adapt.`, `Collab.`, `Comment.`, etc.).

Style Identification

- *Bold words.* Two measurements are associated with each connected component: *npn*, the number of black pixels and *nbseg*, the number of black runs. The analysis of the histogram of values of *npn*/*nbseg* shows a characteristic peak of the average of the line thickness. This measurement allows locating the bold words if they are in minority in the space area.
- *Italic words.* The profile of the vertical projection of the word is compared to the projection profile at the skew α (α varying around the skew standard of italic). The skew retained is the one which maximises the criterion: $\sum_i N[i]^2$ where $N[i]$ is the number of points accumulated at i.
- *Underlined words.* The horizontal projection profile of the word is analyzed in order to precisely locate the top and bottom edges of the line. A partial edge detection allows the deletion of the line preserving the possible descenders.
- *Word Mode.* The analysis of the size and the space of connected components allows the determination of the word mode (upper-case, lower-case, spaced words, etc).

masks / Patterns	[	[	(	(
Ma	4.7%	5.3%	16.1%	18%
[	76.7%	60.8%	43.3%	19%
(	37.5%	28.2%	91%	80%
la	55.5%	50.3%	31.5%	10%
Ca	37.5%	35.4%	61%	55%

(a) Correlation Results

	Total Words	Correctly Classified	Wrongly Classified
Bold Words	210	203	7
Underlined Words	133	131	2
Upper-Cases / Numeric	238	132	6
Lower-cases	190	175	5
Spaced Words	190	188	2

(b) Mode and Style Results

Fig. 11. Result of the Visual Feature Extraction.

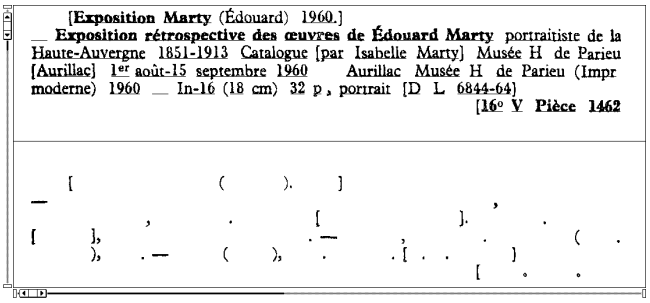


Fig. 12. Result of Separator Extraction.

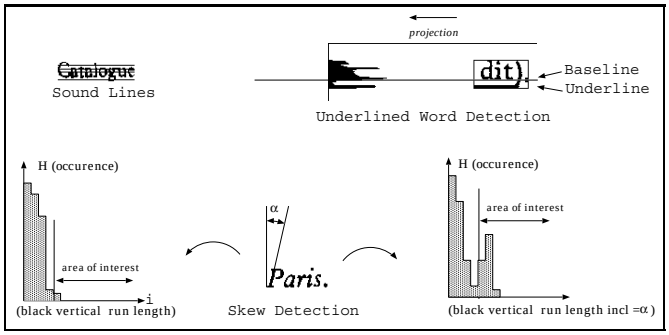


Fig. 13. Visual Features Extraction: Segmentation techniques.

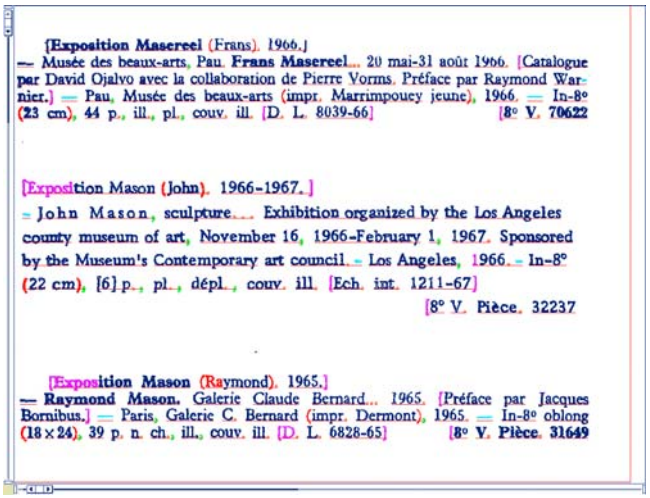


Fig. 14. Visual Features Extraction: Separator, Style and Mode Extraction.

Key-words

They correspond to certain function mentions such as `Ed.`, `Collab`, etc. We have chosen a simple, fast and global method which takes into account the high and low profiles of the word, its length and the ratio N_i/E where N_i is the number of pixels along the sounding line i, and E, the average thickness of the word. Five horizontal line sounds are judiciously used.

The recognition method was tested on a sample set of 300 words from four pages not used for the learning. For words whose fields are known, the recognition rate is 86.27% for top 1 and 99.01% for top 3. For words of unknown fields, the recognition rate is 83.33% for top 1 and 95.09% for top 3.

Knowledge about the field to which a word belongs is very important for its recognition. In fact, each field has a local lexicon context, compiled during the learning step. This context helps to limit the number of possible choices and to improve the recognition rate.

Separator Recognition

We used the template matching technique for the recognition of separators. Templates corresponding to each kind of separator (comma, parenthesis, bracket, etc.) are learned during a previous step from different samples extracted from the references. During the recognition, the template is moved on the line by small translations in the four directions and the maximum of the correlation is retained.

Figure 11.a gives a table of correlation between some visual features and masks learned for brackets and parentheses. Figure 11.b relates in terms of number of words the mode and style results. These tables show the limit of the method.

Figure 12 shows an example of separator extraction from a principal reference.

Figure 13 gives examples of visual feature extraction. Figure 14 shows the result of the separator extraction. Different colours are used to distinguish the different kinds visual features.

4.7 Anchor Point Extraction

For each content fragment in the analyzed reference, the sets of their possible fathers are searched. If the symbol value of fragment A is known (for example a particular recognized character), the set of its possible fathers is initialized with P_A as defined in the model compilation. The problem is that in most cases, the symbol value of the fragment is not known in advance. However, some of its physical characteristics are given by the visual feature extraction. These characteristics will allow the initialization of the possible fathers with the set of objects that verifies them. For each fragment, the set of its possible fathers is pruned in order to keep only those which are compatible with at least one of the possible fathers of each of its neighbors. Each suppression of a possible father constitute an information to be processed by the neighbors (propagation). The compatibility constraints are extracted during the model compilation step.

```
% Initialization step
L ← φ
for each (i, j) ∈ A do
  for each p ∈ E(i) do
    for each p' ∈ E(j) do
      if P_ij(p, p') then C_ijp ← C_ijp + 1
                          S(j, p') ← S(j, p') ∪ {(i, p)}
      end if
    end for
    if C_ijp = 0 then L ← L ∪ {(i, p)}
    end if
  end for
end for
% Discrete propagation
consistency ← true
while L ≠ φ and consistency do
  choose (i, p) in L; L ← L − {(i, p)}; E(i) ← E(i) − {x}
  if E(i) = φ then consistency ← false
  else
    for each (j, p') ∈ S(i, p) do
      if p' ∈ E(j) then
        C_jip' ← C_ijp' − 1
        if C_ijp' = φ then E(j) ← E(j) − {p'}
                           L ← L ∪ {(j, p')}
        end if
      end if
    end for
  end if
end while
```

Fig. 15. AC4 algorithms.

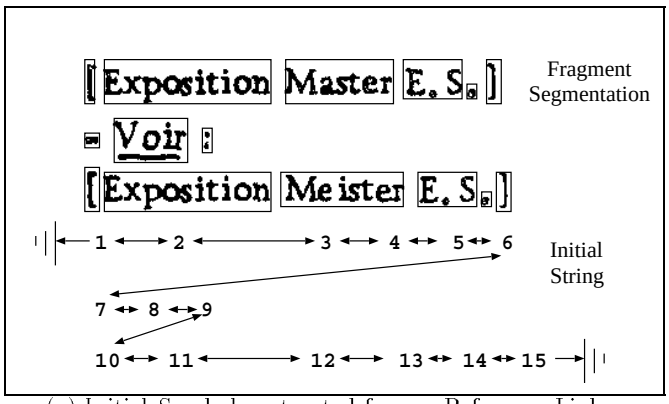
This method is based on the arc consistency algorithm AC4 [24].

The basic idea is to associate with each arc-label couple a counter C_{ijp}. This counter contains the number of possible fathers of j ($E(j)$) compatible with the possible father p of i. Furthermore, the set $S(i,p)$ is memorized. It represent all the node-label couples (k,p') compatible with the father p of i. This set helps to determine the counters $C_{kip'}$ to be decreased when p is suppressed from $E(i)$. The cost is $O(ae^2)$ where a is the number of fragments is the chain and e is the maximal number of fathers per fragment. The algorithm involves two steps. The first consists of initializing the counters C_{ijp} and building the sets $S(i,p)$. The node-label couples to be suppressed will be added to the set L initialized as empty set. The second step prunes the sets of labels associated with the fragments. When a counter C_{ijp} becomes zero, it means that no element in $E(j)$ is compatible with the father p of i. p has to be suppressed from $E(i)$ and the information propagated to all the nodes that have a compatible label with p, i.e.:

$$\forall (j,p') \in S(i,p), C_{jip'} \leftarrow C_{jip'} - 1$$

This operation is repeated each time a counter becomes zero. The fact that the set $E(i)$ becomes empty is a proof of the inconsistency of the chain (see figure 15).

The example in figure 16 shows the incidence of the constraint propagation for anchor points extraction.



(a) Initial Symbols extracted from a Reference Link.

	Before Propagation		*After Propagation*	
Fr.	S	Father	Symbol	Father
1	ϕ	{ "(", "[" }	"["	ZTitreF
2	ϕ	23 labels	TypeF	TitreF
3	ϕ	23 labels	ϕ	{ TypeF, NomF }
4	ϕ	23 labels	ϕ	{ Pren1, TypeF, NomF }
5	ϕ	{ ".", "," }	"."	TitreF
6	"]"	7 labels	"]"	ZTitreF
7	"-"	10 labels	"-"	NoticeT
8	ϕ	7 labels	Renvoi-T	NoticeT
9	":"	3 labels	":"	NoticeT
10	ϕ	{ "(", "[" }	"["	ZTitreF
11	ϕ	23 labels	TypeF	TitreF
12	ϕ	23 labels	ϕ	{ TypeF, NomF }
13	ϕ	23 labels	ϕ	{ Pren1, TypeF, NomF }
14	"."	24 labels	"."	TitreF
15	ϕ	{ ")", "]" }	"]"	ZTitreF

(b) Extraction of Anchor points: those having only one father after propagation.

Fig. 16. Incidence of the constraint propagation.

4.8 Bottom-up / Top-down Analysis

The analysis algorithm proposed by [25] starts from a well chosen anchor point o_k. It searches for the rule $(A \rightarrow \lambda o_k \lambda')$ in the model. The left and right contexts (λ and λ') are then verified in a top-down manner, from right to left for λ and from left to right for λ'. This procedure is then repeated on A by searching for another reducing rule of the form $(B \rightarrow \mu A \mu')$ and so on until reaching the grammar axiom (cf. figure 17). In order to optimize the bottom-up step, we choose the objects that minimize a criterion which depends on the number of hypotheses and on an *a priori* probability for each of them. This probability is given during the model construction step as an additional attribute to help the analysis.

The method for the left and right context verification is illustrated in figure 18. Considering an already verified object o_k (which can be an anchor point at the beginning), a rule $A \rightarrow \lambda o_k \lambda'$ is proposed from the model where λ and λ' are the left and right context to be verified. We next describe the left verification which is processed from right to left. The right verification is

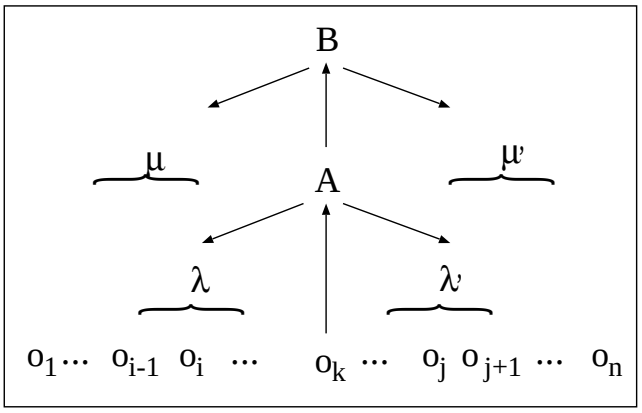


Fig. 17. Mixed Strategy.

performed in the opposite direction in identical manner. The left context λ can be rewritten as: $\lambda = \lambda_1 \ \ldots \ \lambda_i$. λ_i is first verified in a top-down manner, then λ_{i-1} and so on. λ_i is first checked to avoid an unnecessary top-down analysis.

Considering $(a_1 \ a_2 \ \ldots \ a_n)$ as the input chain to be recognized, different cases can be encountred:

- λ_i is repetitive. It is transformed into the following sequence:
  $\lambda_i \rightarrow$ sequence(λ_i (optional repetitive), λ_i).
- λ_i is a terminal. In this case, we observe a_f (see fig. 18.(a)), the left neighbor of o_k. λ_i is validated if it is a possible label for a_f (obtained during the propagation step).
- λ_i is a non-terminal. Here, we check if there is an intersection between the possible labels of a_f and $F^*_{\lambda_i}$, the set of finals for λ_i (see fig. 18.(b)). The syntactic analysis of λ_i is performed only in this case.
- If λ_i is verified or if it is optional, the verification is continued on λ_{i-1}. The problem is recurrent and a possible final for λ_{i-1} (a_x) has to be found in the input chain, as a left neighbor for a possible initial of λ_i (a_y) (see fig. 18.(c))
- If λ_i is required but not verified then the current rule is abandoned. The failure is propagated to the upper level (A) so that other alternatives can be tested. If all the alternatives end in failure for the object A, then another possible father is chosen from object O_k.
- If λ and λ' could be verified for the rule $A \rightarrow \lambda o_k \lambda'$, the set of possible fathers for A is pruned according to the neighborhood compatibilities between A and its left and right context in the chain.

4.9 Results and Discussion

A test was performed on 10 catalogue pages corresponding to about thirty references per page (a total of 300 references). Results show that the structure is always correctly recognized when the initial data is consistent. The few errors encountered arising from erroneous choices (for Human) were in the end verified with the help of

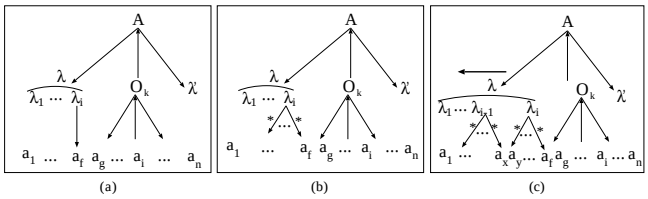


Fig. 18. Different Left-to-Right Top-Down Context Verification.

the model. We can estimate at about half, the references that gave a consistent chain after syntactic propagation. For the other half, by redefining the model in a less strict manner, or by enhancing the low-level tools, 50% of the errors coming from the extraction of visual features can be recovered. This constitutes the major weakness of this global method.

The drawback of the method

The neighborhood constraint propagation needs robust tools to extract the visual features from the image. At this stage, if the structure recognition could be achieved without OCR, we have to admit that the use of one or a combination of several commercial OCR's would have improved the recognition rate. We have had to develop specific tools to recognize particular characters and words, and to recognize the styles and the modes of the tokens. This task proved difficult because of the noisy multifont environment of the catalogues. For example, the punctuation is often attached to the preceding word and is not identified. This drawback is propagated along the chain which often leads to an inconsistency.

The strength of the method

A bottom-up/top-down analysis allows the building of the reference hierarchy from the fragment called *anchor points*. The other fragments are just verified in a top down manner on their left and right context. This mixed analysis is much faster than a full top-down analysis because the input chain is filtered according to the extracted features and the neiborhood constraint propagation. Fancy references are rejected at the propagation stage. In the top-down method, such references would have to wait for the analysis stage before being rejected.

5 The Belgian Library

In order to improve the recognition rate of the reference structures, we based the second system on an extensive use of OCR and on the context [4,6,8]. Documentalist experts have contributed to the determination of a basic logical structure description. Furthermore, different knowledge sources were added such as general and specialized lexicons (indexes of subjects, names, etc.). These indexes, present in the catalogues, were also modelled and recognized by the same system.

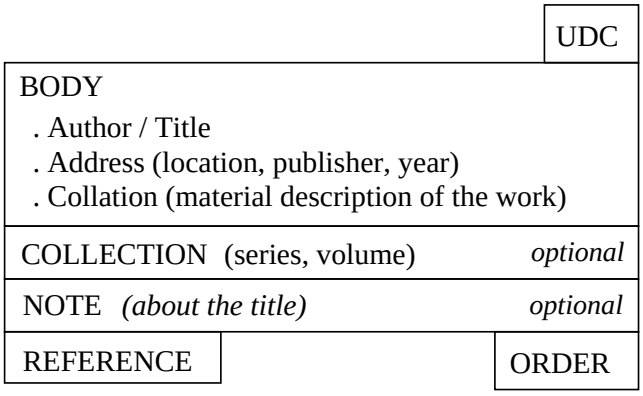


Fig. 19. The Structure Layout of the Belgian References.

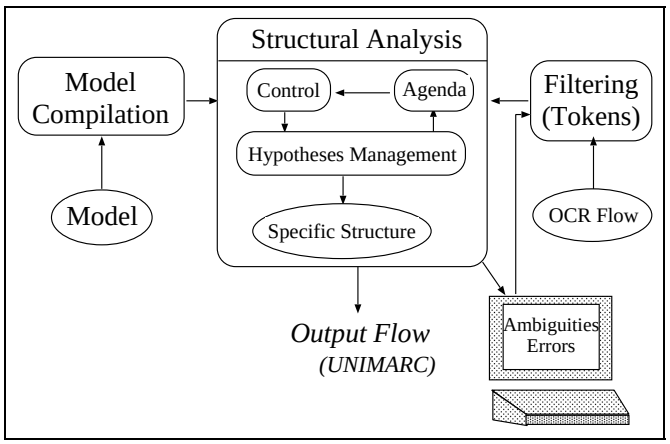


Fig. 20. System Overview.

5.1 Structural Aspects

The Belgian Library is presented as a series of monthly catalogues on paper. Each catalogue is divided into two parts. The first part contains the bibliography body while the second is filled with pointers to authors, subjects treated (titles, collections, rubrics in French and in Dutch, etc.).

Layout Reference Structure

The layout structure is very poor; it is partitioned into five areas (cf. figure 19). The first area, composed of the first line, contains on its right hand side, the "CDU" code (Classification Décimale Universelle) which gives some information about the library classification of the reference. The second area contains the reference body. It is composed of a series of fields describing the work referred in the reference such as "*heading*" (author name or beginning of title), "*title*", "*address*", "*collation*" (material description of the work : location, editor, year, format, etc.). The body is often typed in many lines. The third area contains the "*Collection*" field (description of the series, volume, etc.). The fourth area contains the "*Note*" field which for example gives information about the title (abbreviated, complete, original, etc.). These last two areas are optional and so are not always present in some references. The last area, located on the last line of the reference, contains the "*reference*", on the left hand, and the "*order number*", on the right hand.

Logical Reference Structure

The logical structure on the other hand is more dense. A "heading area", representing the first author or the beginning of a title is always located at the beginning of the "body". As with the rest, there is a great variety of possibilities. We can find, for example, depending on the references, "principal authors" or "secondary" (introduced by some characteristic expressions) which can be physical persons or institutions, some "main titles", "parallel" (printed in different languages), or "partially", "sub-titles", "publishers" with their "addresses" and the "date" of publication, an area "collation" describing the characteristics of the work (number of pages, format, supporting documents, etc.).

5.2 System Overview

Figure 20 shows the major components of the system developed. The *input* is given by tagged OCR output references. The *Domain knowledge* contains the generic structure of the references (model) and also the data flow to recognize. The flow is given in SGML formalism [20]. The model is compiled and the flow is filtered to extract interesting information for the analysis.

The *Databases* contain all lexicons used for term verification (index of author names and titles, towns, countries, expressions, etc.). These indexes were recognized in a previous step in the same manner, that is, with rules and tagged OCR output. What follows describes only the reference analysis.

The *Structural analysis* is the main module of the system. The system control acts on the hypotheses management process to produce and verify hypotheses on the specific structure. The specific structure is represented by a tree of terms. Hypotheses are produced for each node of the tree and construction is first pursued for the node which looks the most likely (having the highest score).

The *Output*, tagged in UNIMARC, contains the specific structure identified by the system.

Data Flow

Each reference is passed through a series of commercial OCRs. The results of these OCRs are combined to obtain the best response. The reason for this is that references contain a lot of different symbols (such as punctuation, indices, exponents, and multilingual words typed in different sizes and styles) which are very difficult to recognize using only one OCR. We thought that combining the results of different specialized OCRs will give a maximum of information on the text, its style, its language, and its separators.

The result of these tasks is a data flow containing the reference text coded in SGML. The tags separate the lines and different information such as style or lexical class corresponding to each word (token). Figure 21 shows the flow corresponding to the reference of figure 19.

```
<DOC  % image source 1st reference last reference
Directory TY=N PROV=ENRLEX EG=OK NPN=2085 NDN=2114
IMA=users/brb/juin73/images>
<PAG % number bounding box NP=1 NOM=0008.ima>
<COL XHG=63 YHG=1900 XBD=1027 YBD=2912>
<NOT % number coordinates NON=2108 EN=OK>
<LIG XHG=870 YHG=2215 XBD=1000YBD=2266 YBSL=2256
ST=t> <REDF=85.69>159.962</LIG> <LIG XHG=149 YHG=
2260XBD=1001 YBD=2313 YBSL=2298 ST=p><B>Liger-
Belair</B><I>(Grard).</I><LEX L=GFR,GNL><REDF=
50.00>Je <LEX L=GFR>suis <LEX L=GGB,GFR>fakir.<LEX
L=GGB,GFR,GNL><RED F=99.99>([Par] <REDF=99.97>
G-</LIG><LIG XHG=148YHG=2304 XBD=1002 YBD=2356
YBSL=2342 ST=p>rard Liger-Belair).<RED F=89.99>
<I>(Verviers, <LEX L=GGB> <RED F=100.00>Editions
<LEX L=GNL><REDF=99.99>Grard</I> <I>\&amp; </I>
</LIG><LIG XHG=151 YHG=2350 XBD=1000 YBD=2403
YBSL=2388 ST=p>C0,<RED F=83.33>1973),320<LEX L=GFR,
GNL><I>carr, <RED F=66.66>couv.,ill.,</I><RED
F=99.97>158 <RED F=25.00>p.30 <I>fr.</I>).</LIG>
<LIG XHG=149 YHG=2406 XBD=549 YBD=2457 YBSL=2443
ST=p><LEX L=GGB,GFR,GNL> Marabout-flash,352).</LIG>
<LIG XHG=148 YHG=2447 XBD=1001 YBD=2499 YBSL=2485
ST=p> <LEXL= GFR><RED F=99.99>[Titre <LEXL=GFR>
introductif:<LEX L=GGB,GFR> <RED GFR>F=89.99>
Souvenirs,<LEX L=GFR>rivilations, <LEX GFR> L=GGB,
GFR,GNL>con-</LIG> <LIG XHG=149 YHG=2494 XBD=245
YBD= 2545 YBSL=2530 ST=p>seils].</LIG> <LIG XHG=148
YHG=2546 XBD=1002 YBD=2599 YBSL=2584 ST=t> <RED
F=43.75>B.D. 14.814 <REDF=99.97>352<SN=15><I>
73-2108</I> </LIG> </NOT> </PAG></DOC>
```

Fig. 21. Flow of the reference given in figure 2.a.

The reference is located in this flow between two successive tags "<NOT" and "</NOT>". Useful tags for the document analysis are "LEX" which gives the lexicon affiliations of words, "I" for italic style, "B" for bold style, and "S" for the number of spaces. The defaults style is standard and as such not tagged. It is possible to have some errors during this first conversion (especially in recognition of style and punctuation). For example, the exponent "o" in c^o is replaced by the character "o". Another initial recognition error concerns the style of the end of the secondary title which is identified as "standard" instead of "italic".

OCR Filtering

The content fragment extraction is easier than in the French Library case because the OCR flow directly gives the list of separated words. Only the hyphenation poses the problem of word extraction and needs a particular interpretation of dashes at the end of lines. Furthermore, a filtering stage is used before the extraction either to eliminate some useless terms given by the OCR flow (global information on the image and on the location of its lines) or to convert some peculiar symbols coded in a particular manner by SGML such as the "&" (\&amp in SGML) or the accents such as the "" (e\acute in SGML), etc. Once the data flow is purged and words are extracted, a table is filled containing tokens (words or word fragments) accompanied by their attributes (given by the OCR such as language, style, mode, etc.). This table is used during the analysis step for the validation of syntactic associations of terms.

5.3 Structure Modelling

The model is given by the Library experts. We also used a context-free grammar where the terms are described by a combination of constructors, qualifiers, subordinate objects and attributes.

Attributes

Because of the weakness of the physical structure and the multitude of choices represented in the model, we added to the previous description, some attributes given in the Library specification to give a more precise description of the reference components.

Several kinds of attributes have been defined, among them, *Type* (string, line, word, char, etc.), *Mode* (capital, numeric, alphabetic, punctuation, etc.), *Style* (bold, italic, standard, etc.), *Position* (beginning of line, inside, end), *Lexicon* affiliation (author index, countries, towns, abbreviations, articles, etc.), *Separator* between subordinate objects (space, comma, hyphen, etc.), *Weight* which specifies the degree of importance of subordinate objects, etc.

The following example describes the term "TITLE" as a logical sequence of two objects: "PROPER-TITLE" and "REST-TITLE" where the style is not italic (may be bold or standard). It is located at the beginning of the line with a comma as a separator. The minus sign indicates the negation of an attribute value.

TITLE	::=	*seq* PROPER-TITLE REST-TITLE
Style		-Italic
Position		Begline
Sep		Comma

Weights

In the case of an uncertain OCR flow, weights are used to obtain an evaluation of the solution retained. These weights are specified in symbolic form, for example from A (very important) to Z (not important). The corresponding numerical values are determined from a base value specified by the user. In this manner, the user can specify the importance he attaches to each subordinate object. In the following example, the optional object "PARTICLE" (A) is more important than "LCAP" (G). This specification is logical since an optional object normally helps in reinforcing the possible presence of a term more than an object that is always present.

ZPB	::=	*seq* LCAP RP PARTICLE?
Weight		PARTICLE A LCAP G

Actions

The processing of a rule sometimes needs a check procedure before or after its processing, either to prepare the analysis of the rule, to recover from its failure,

or to post-process the result. For example, if the analyzer is expecting only numerical values in a region being analyzed but fails to find them, it is called. Usually this function should check for OCR substitution errors, ex : "l" for "1", "O" for "0"). The production rule given in the sequel describes a choice between two terms neither of which should contain any of the strings in the lexicon "Abn" (expressed by the attribute Clex). There are two actions. The first one indicates the need to verify before the rule analysis that the search zone does not contain the string "fr.". In the event this hypothesis is verified, the second function is executed to create a UNIMARC tag before the restitution of the result in the required format.

FIP	::=	*cho* FIP1 IPA
Clex		-Abn
Style		Italic
Action		+VerifyStringInField(fr.,false)
		Restitute(215,bb)

Inheritance

We distinguish essentially two kinds of inheritance: the succession and the external reference. In the first case, the inheritance of attributes between an object and its subordinate objects is made in direct line (direct filiation), according to the constructor type. In the case of "choice", all the attributes are sent to subordinate objects. For *sequences*, only the *typographic* attributes (style, mode, etc.) are inherited by the subordinate objects. Attributes describing the geometrical structure (position, width, height, etc.) are filtered as a function of the kind of the physical structure. For example, a top-down sequence transmits only the width to subordinate objects ; the location attributes in a page, region, or line are inherited only by the first object of a sequence.

The inheritance by external reference is obtained by the constructor `Import`. In this case, the characteristics of the subordinate object (imported object) are inherited by the current object. There is no restriction on the imported attributes. Furthermore, the constructor and the subordinate objects of the imported object are also transmitted to the current object. The following example describes the object `Notes` as a repetition of the object `OtherNotes` the characteristics of which are imported. The attributes `Sep, Action` and `Style` of the object `Notes` take the place of the eventual attributes of the same name inherited from `OtherNotes`.

Notes	::=	`Import` OtherNotes+
Sep		LongDash B
Style		Standard
Action		+VerifyStartWith(–,FALSE)

5.4 Structure Analysis

The structure analysis is based on the model and the entry data flow. For the model, the grammar rules are converted by a compilation procedure into a working structure. The input data flow is also reorganized into a working table by a filtering task. This table contains useful tokens extracted from the flow such as style, token, size, etc. and a pointer to a buffer containing the corresponding content. Figure 22 summarizes the principal functioning mode of the structural analysis.

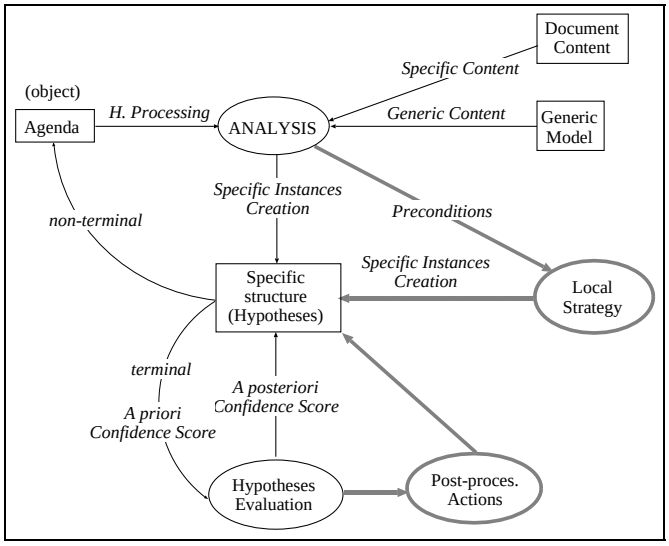


Fig. 22. Functioning Scheme of the Structural Analysis.

Model Compilation

This step helps to adapt the analysis process to the application model. It generates working files containing the specific terms, actions and attributes for the application. References as well as indexes (containing authors and subjects) are modelled as three different applications. During the analysis, these files are converted into dynamic tables of terms where the entries correspond to term codes. Each term is given by a list of characteristics gathered in a characteristic table. This allows the system to read rapidly the characteristics of each term analyzed.

Hypotheses Management

At each step of the analysis, the system proposes for the current object different choices for its decomposition (analysis). Those choices which are not already verified are called *hypotheses*. We use a structural tree to store these hypotheses. A confidence score (*a priori* score) is computed for each generated hypothesis. This score allows to choose, among all the current hypotheses in an *agenda*, the one to process first. The score computing is initialized by the weights given in the model for the current object (for its attributes and subordinate objects). This score is successively updated as the hypotheses are verified and becomes a recognition score. At the end of the analysis, each tree path corresponds to a possible structure (for the input reference) weighted by a recognition score. This qualitative reasoning helps to reduce errors and isolate possible ambiguous areas.

The hypotheses are chosen from the agenda according to the importance of their *a priori* scores (apr). Thus, the analyzer is said to function in an opportunistic mode.

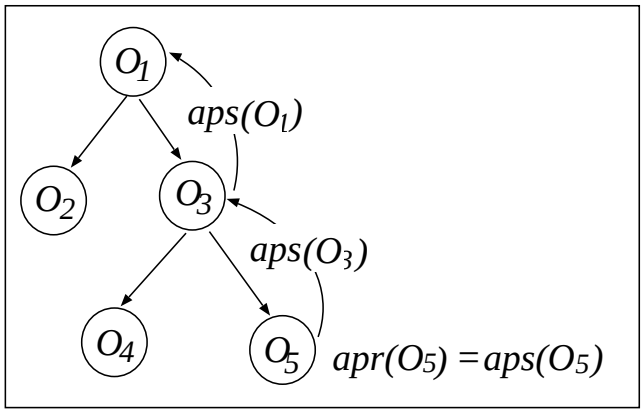


Fig. 23. Score Propagation.

Terminal terms (tree leaves) are directly verified. On failure or success, the *a priori* score is up-dated and becomes an *a posteriori* score (aps) which is propagated from bottom to top in the corresponding path (see figure 23).

The *a priori* score of a current object o depends on the result of the observation of its attributes (a_t) for each token t_k of o ($C(a_t, t_k)$). It is also function of the token length (L) and of the weight W of each attribute.

$$apr(o) = \frac{\sum_{a_t}\sum_{t_k} C(a_t,t_k).W(a_t).L(t_k)}{\sum_{a_t} W(a_t).L(o)}$$

The *a posteriori* score of o is updated from the *a posteriori* scores of its subordinate objects (o_i) by taking into account their corresponding weights (p).

$$aps(o) = \frac{\sum_{i} p(o_i).aps(o_i).L(o_i)}{\sum_{i} p(o_i).L(o_i)}$$

With this method, the different objects and attributes influence the final score according to their importance in the model (weight) and in the input data string (length).

Local Strategies

We now present some examples of actions executed before the general analysis. Depending on the status returned by these actions, they can either play the role of pre-conditions: in which case, the analysis continues normally, or of local strategies, or stopping general strategy. When an action plays the role of a local strategy, it has the control of new hypotheses (possible decomposition of the current object) to submit.

Author searching

In the library references studied, it was found fitting to identify the secondary authors of the publication. Contrary to principal authors, secondary authors are introduced by a particular expression ("par", "introduit par", "illustration de", etc.). It suffice to recognize this expression and to verify that what follows corresponds to an author. The problem here comes from the fact that authors are not necessarily presented in the same format in indexes and references. Furthermore, the list of expressions is not exhaustive. It is fitting to refine the syntactical analysis to recognize these secondary authors as shown by this example:

ZATZME	::=	Seq ZAT ZME?
Sep		Ponct1
Action		+InitAuteurs(Expressions, IndexAuteurs,...)

Parameters `Expressions`, `IndexAuteurs`, etc. correspond to a list of lexicons used by the local strategy `InitAuteurs`

Searching for style

In order to minimize the number of hypotheses submitted during the analysis, we developed some heuristics using typographic characteristics to delimit an area. The following example shows an action which cuts the current object at the first punctuation preceding the beginning of the italic area. This pre-cutting in fact allows part by part analysis and avoids hypotheses which are doomed to fail

ZATX	::=	Seq ZATZME ZIC
Sep		Ponct
Action		+SplitField(italic,Ponct)

Suppression of irrelevant hypotheses

Some objects to be recognized are easily identifiable (for example, a town found in town dictionary). In this case, it is interesting to delete all hypotheses in the queue which contain the same search area in an another context. The action `KillAmbiguities` in the example below is activated if the object `MotEd1` is perfectly recognized. It goes through the specific structure tree and suppresses all waiting hypotheses that contain the same content as `MotEd1` and that do not belong to other instances of `MotEd1`. This action must be used carefully because every new hypothesis on this area, which is not an instance of `MotEd1`, will be forbidden.

MotEd1	::=	Terminal
Alex		Edition //opl. tir. uitg. éd, etc.
Nature		mot
Action		KillAmbiguities() RestituteField()

Output Flow Restitution

When the analysis is finished, it is necessary to go through the structure tree to produce a structured flow corresponding to the result. This is realized in depth first. The structure is represented by a mark up format like SGML. Each tagged field is given by a confidence score. If, for the same specific object, many hypotheses are successful, we have an ambiguity in the structure.

Some types of documents are intrinsically ambiguous. For example, in the sequence:

A	::=	Import B+
Sep		Virg
B	::=	Terminal

A is a repetition of objects B, separated by a comma. But nothing indicates that a comma can appear within a B. Thus, there is ambiguity. Furthermore, there is error if the examination of the tree reveals an object which verifies no decomposition hypothesis. Errors can either come from the OCR (the separator characters are not recognized, the style and mode are incorrectly identified , etc.), or from a document which does not correspond to the generic model (required field is absent, styles are not exact, etc.). A special tagging is used to restitute every ambiguity in order to correct it manually later.

5.5 Results and Discussion

This section shows the statistical results obtained from tests effected on the real production chain of the company (JOUVE).

Problems Encountered

The main problems encountered in the technical realization of this project concern the OCR treatment of the bibliographic information, the modelling of the Library catalogues structure and the adaptation to an industrial production.

OCR and Bibliographic Information

The variability of the typography seriously handicapped the straightforward conversion of the bibliography using OCR techniques. Several reasons were signalled in section 2.3. The main deficiencies of the Belgian catalogues reside in:

- *its typographic aspects*: connected characters for bold data and the use of standard numeric characters within textual areas in italics,
- the fact that diacritics are added by hand,
- the use of long dash line for the parallel areas and for some of the collection sub-areas;
- the intensive use of square brackets.

3.6% of references were returned to the Library for the transilteration of non Latin characters found in the references.

Bibliographic Catalogue Modelling

This problem is already encountered in a traditional retrospective conversion process in which the Library writes specifications for the conversion of its catalogue. These specifications must be validated on several references and modified in a continuous manner in order to adjust the model so as to take into account exceptions and newly encountered problems. In the MORE project, these specifications were greatly refined with the emphasis more on cataloguing than on computer automatic conversion. In this regard, librarians and system designers must come to a common understanding on the specification of the model.

The three main characteristics listed below account for the difficulty in processing the structure of bibliographical information:

- Catalogues are written before the ISBD standard, based on different structures with particular rules for layout and punctuation;
- The correspondence between the pr-ISBD cataloguing rules is sometimes difficult to establish with the UNIMARC format for the transcription of the title areas and responsibility mentions. This difficulty is less pronounced in USMARC where the main cataloguing elements are grouped into three sub-areas in only one possible non-repetitive sequence. In UNIMARC, the same information can be shared in six sub-areas all of which are repetitive and have a high number of possible sequential combinations. The same difficulty is encountered in the modelling of the edition and collection areas.
- Some catalogues involve hierarchical levels as in the case of a monography in severals volumes, with significant titles for each volume. The model has to take into account some specific considerations for the treatment of these volumes. In the Belgian Library, the cataloguing of volumes belonging to a monography in many volumes, as well as the treatment of collective titles was very difficult to model and had created some anomalies which were returned to the client (the Belgian Library). The presence of many official languages in this bibliography (French and Dutch) have also led to a great number of parallel mentions in titles and notes, teading to a very complex structure for titles and responsibility mentions. 78.4% of the 246 references were returned for manual control because of bad structure, 46.53% for the title structure and 12.25% for the validation of collective authorship.

Specific Results

The industrial prototype

Figure 24 shows the industrial prototype developed by JOUVE. The functional architecture is meant for production and it is very modular facilitating adaptation to different situations and cases. Among the important modules we find:

- *the module for final verification and formatting.* It receives the SGML flow as input and produces references in UNIMARC format, after effecting the exhaustive controls and generating the missing information in the reference. It comprises modules for exhaustivity control, country and language codes for publication, quality control of the structure and coded information.
- *a module for anomaly management.* It handles all questions and errors which can appear in the production chain and which may need help from the Library. It comprises sub-modules for formatting note edition for the Library. In return, and depending on the Library comment, the reference is taken into account at different level in the production chain.

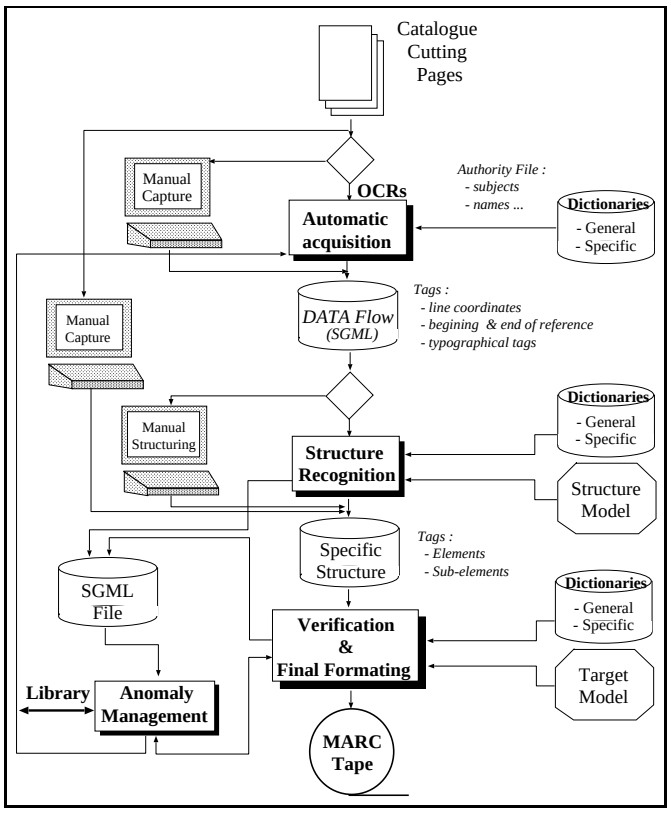


Fig. 24. Industrial Prototype Overview.

Quality Control Evaluation

The quality control was performed on a random set of 5% of references which are not in anomaly, namely 187 references of the volume of June of the year 1973, and 61710 characters. A professional librarian effected a re-reading of the controlled sample and indicated the different errors found.

OCR Quality

The project has investigated the following commercially available OCR packages: Calera, Wordscan, Caere Omnipage. The accuracy of the packages when used without postprocessing modules seems to be at the same level for the investigated OCR programs, that is about 99,8% correct (about 0.2% of the characters of the source misrecognized or not recognized), when used with trained typefaces. With catalogue references this means 2 to 3 errors per reference on the average. With much worn out references or very uneven printing as well as multi-font references the accuracy is obviously much less. For the 17 errors obtained on the 187 references, 10 are for alphabetic, 4 for the digit 1 converted to: l,[or I, 3 for an indice wrongly positioned in a numerical area, 3 for dash lines confused with an underscore, 3 for diacritics and 2 for double points.

Structure Quality

The set of 187 references comprises about 4000 UNIMARC samples. There was no fault, no detection of limits between fields, but only errors within fields and errors on the type or the category of fields.

Table 1 gives the nature of structure errors and their percentage.

References	Percentage
Recognized without error	75.5
Recognized with ambiguities to resolve manually	3
Recognized but with "risk" to be re-examined	8
Recognized with structure error	1
Unrecognized : anomalies	7.5
Unrecognized : unknown cause	3
Unrecognized : model	2

Table 1. Structure Results.

An ambiguity is announced when there are two structure hypotheses of equal importance for the same portion of the content.

References with "risk" are references for which errors are not already taken into account in the modelling phase such as the presence of multiple sub-areas in one field not allowed to contain more than one area, detection of the presence of names (belonging to dictionaries) in the title, detection of acronyms in title and collection areas, etc.

Structure errors detected are due to the deficiency of the model and to ill functioning of the dictionary author-titles during the structure phase.

References in anomaly correspond to a catalogue which is inconsistent with the cataloguing rules.

Language and Country Codes

The generation of country codes for publication is derived from the publication place identified in the structure by comparison with an atlas. The generation of the language codes for publication is made by analyzing the title language, for the fixed field and parallel titles for codes associated with each of the titles. The analysis is performed only on the common names using extended language dictionaries. All the errors, about 15, have to do with language codes.

Global Evaluation

The global evaluation of the prototype is performed after the treatment of all the 11 volumes of the Belgian Library catalogue, i.e. 4548 references. The volume of June is discarded because it was used in the first phase for the quality control evaluation. Performances are as follows:

- *OCR/ICR.* 6.69 doubts per reference for only the body of the bibliography and 9.87 doubts per reference if we include the main and secondary entries.
- *Structure Recognition.* 67% of the references were recognized automatically by the system.
- *Attribution of language and country codes.* 77.7% of the references have their codes created automatically by the system.

However, putting togeteher all the operations of correction provided for the automatic recognition of the structure and the code generation, as well as the corrections effected on references with "risk", only 47.5% of the references were fully recognized automatically without any manual intervention.

– *Execution Time.* The execution of the prototype takes about 1'30 per notice. This depends on the complexity of the structure and the correction procedures launched by the system.
Table 2 gives the time spent by the different modules of the system on the 4548 references.

Automatic Module	Time in hours	% total time
OCR/ICR	16.5	14%
Structure Recognition	99	83.5%
Others	3	2.5%

Table 2. Time spent by the Automatic Processing.

Manual Intervention

Table 3 gives statistics on manual interventions either for OCR correction or for re-treatment of the structure or the codes generation.

Module	Defect Cases	Manual interventions
OCR/ICR	44920 doubts examined	9.87 doubts per reference
Structure	1494 references unstructured totally or partially	33% of references
Codes Country Language	1014 references with in less one non-generated code	22.3% of references
Structure + Country Codes + language	2083 references corrected in less one time	52.5% of references
Anomaly after Quality Control	246 references returned to the Library	5.4% of references

Table 3. Statistics on Manual Interventions.

6 Comparing the two Methods

The two approaches studied revealed the importance of a model for structure recognition. They also showed that the model is insufficient if we do not have tools to extract pertinent feature (OCR or visual characteristics of the image). This is particularly so if the content of the document is rich and complex. Several references still remain unrecognized or ambiguous because of the complexity of such a task. The reasons for these failures are many.

First, as the model is built from non-normalized references (pre-ISBD), knowledge is incomplete and uncertain. Furthermore, a great number of sub-classes are represented in only one model which leads to difficulties during hypotheses generation and leaves ambiguities in the final structure. The principal difficulty encountered in these projects was obviously the model construction. The construction of models to cope with complex or ambiguous bibliographical information structures is indeed difficult to realize on account of the level details that the specification must be able to embrace and of the difficulty of determining appropriate weights for objects and attributes. An interesting prospect of this project concerns an automatic help for model learning. Such a system was elaborated by our team for scientific paper models [1]. It shows the interest in extending the learning to the micro-structure.

Second, recognition of the structure of this kind of document amounts to the problem of text understanding. The interpretation is not only based on character or word recognition, but also on the recognition of specific expressions and even complete sentences. Furthermore, the linguistic style of these sentences is not always fixed and known in advance (not regular). The problem here is not only syntactic recognition but also semantic understanding. For this, a perfect domain knowledge is necessary to understand the different specific expressions. Adapted tools and heuristics have to be investigated. This investigation can first be based on existing natural language processing techniques and linguistic models. Our recent work [29] is oriented in this direction.

7 General Conclusion

The aim of this paper is to enhance understanding of the issues involved in the retroconversion process and to show the advances in the field of character recognition and structure interpretation and their usefulness in the development of solutions to the retroconversion problem.

The prototypes produced in the MORE project constitute important results of the European approach and the different syntheses produced are very precious and should provide a broader basis for further work in the field of library automation.

Cooperation between libraries has produced valuable insights into practice of retroconversion of old catalogues. This has contributed to a better understanding of the problems involved and in establishing common standards and shared resources.

A number of specific problems remain to be tackled. These relate to:

– Character acquisition: the processing of an important proportion of non textual characters, such as punctuation (marks), does not take advantage of intelligent processing. There is a need to favour cooperation between different OCRs.
– Structure recognition: an important proportion of the microstructure codes depends on ambiguous characters or on the interpretation of very short textual

contexts, the structure being very rich and dense. Finally, the majority of content fragments are optional or repetitive and the number of cases is very high. The modelling of such a structure is very difficult to realize. Specific tools are not available for doing so.
- The creation of coded information: the creation of the publication language code and the country publication code is based on the analysis of the content of some areas and comparison with dictionaries. The results of automatic processing are consequently function of the characteristics of the collections presented in the catalogue, i.e. function of the number and the relative proportion of languages and publication countries and their relative ambiguities.

In the two projects, dictionaries are very important. They are used at the same time for character recognition, structure recognition and in creating coded information. These dictionaries are composed of specific and general tools for the definition of bibliographical information, and also for the country where the references are written. They therefore include the cataloguing rules and the tools of the local library.

The experimentation conducted in the framework of the two projects highlights how difficult a task it is to generalize a common retroconversion procedure to different libraries, because of the specificity of bibliographical information and catalogues.

Acknowledgements. We are very grateful to the European Commission for making the necessary documents available to us for this paper.

We are also very grateful to the JOUVE company for authorizing the publication of the technical information about the MORE project.

References

1. Akindele O. T. and Belaïd A.: Construction of Generic Models of Document Structures Using Inference of Tree Grammar, ICDAR'95, pp. 206–209, vol. I, Montréal, Canada, 1995.
2. Beaumont J., Cox J. P.: Retrospective Conversion. A practical Guide for Libraries. Meckler, Westport/London. 1989. 198 p.
3. Belaïd A. and Akindele O. T.: A Labelling Approach for Mixed Document Blocks, ICDAR'93, Tsukuba, City Science Japan, 1993.
4. Belaïd A., Chenevoy Y. and Anigbogu J. C.: Qualitative Analysis of Low-Level Logical Structures. In *Electronic Publishing EP'94*, volume 6, pages 435–446, Darmstadt, Germany, April 1994.
5. Belaïd A. and Chenevoy Y.: Document Analysis for Retrospective Conversion of Library Reference Catalogues, ICDAR'97, ULM, Germany, August 1997.
6. Belaïd A. and Chenevoy Y.: Constraint Propagation vs Syntactical Analysis for the Logical Structure Recognition of Library References, Lecture Notes in Computer Science, BSDIA'97, N. A. Murshed and F. Bortolozzi editors, vol. 1339, pp. 153–164, November 2–5, 1997.
7. Baird H. S.: The Skew Angle of Printed Documents, SPSE's 40th Annual Conference and Symposium on Hybrid Imaging Systems, pp. 21–24, 1987.
8. Chenevoy Y. and Belaïd A.: Low-Level Structural Recognition of Documents. Third Annual Symposium on Document Analysis and Information Retrieval, UNLV, Las Vegas - USA, 1994
9. Bokos G.: UNIMARC, CDS/ISIS and Conversion of Records in the National Library of Greece. In Program, (4)2, pp. 135–148, 1993.
10. CEC, DG XIII B: Report of the Workshop on Retrospective Conversion of Catalogues. Problems, Priorities and Projects under the Library Plan, Commission of the European Community, Directorate General XIII, B, Luxembourg, Printed as Draft, 1990.
11. CEC, DG XIII B: Libraries Programme, Telematics Systems in areas interest 1990-1994: Libraries, Synopses of Projects. http://www2.echo.lu/libraries/en/libraries.html
12. Chenevoy Y.: Reconnaissance structurelle de documents imprimés : études et réalisations, PhD Thesis, INPL, 1992.
13. Council of Europe: Guidelines for Retroconversion Projects prepared by the LIBER Library Automation Group, Council of Europe, Council for Cultural Cooperation, Working Party on Retrospective Cataloguing, 1989.
14. Crawford R. G., Lee S.: A prototype for fully Automated Entry of Structured Documents. In The Canadian Journal of Information Science, (15)4, pp. 39–50, 1990.
15. Harrison M.: Retrospective Conversion of Card Catalogues into Full Marc Format Using Sophisticated Computer-Controlled Visual Imaging Techniques. In Program, (19), pp. 213–230, 1989.
16. Hein M.: Optical Scanning for Retrospective Conversion of Information. In The Electronic Journal, (4)6, 1986.
17. ISBD (G): General International Standard Bibliographic Description: Annotated Text. Prepared by the Working Group on the General International Standard Bibliographic Description set up by the ILFA Committee on Cataloguing. London, 1977. 24 p.
18. ISO 5426: Extension of the Latin Alphabet Coded Character Set for Bibliographic Information Interchange. Second Edition. International Standards Organization. 1983.
19. ISO 6937: Information Technology - Coded Graphic Character Sets for Text Communication - Latin Alphabet. Second Edition. International Standards Organization. 1993.
20. International Standard Organization: Information processing, text and office systems, standard generalized markup language (sgml). Draft International Standard ISO/DIS 8879, International Standard Organization, 1986.
21. ISO 8859-1 to 7: Information Processing - 8-bit single-byte Coded Graphic Character Sets - Part 1-7: Latin Alphabet No. 1 to 7. International Standards Organisation. 1987.
22. Jensen H. E.: Error Analysis and Correction in Retroconversion. FACIT Technical Report no 3). Statens Bibliotekstjeneste, Copenhagen. October 1996.
23. Mackenzie C. E.: Coded Character Sets, History and Development. Addison-Wesley Publish. Co., Reading (Mass) / London, 1980. 513 p.
24. Mohr R. and T. C. Henderson T. C.: Arc and Path Consistency Revisited, Artificial Intelligence, (28), pp. 225–233, 1986.

25. Mohr R. and G. Masini, Good Old Discrete Relaxation, ECAI88, pp. 651–656, Munich, Germany, 1998.
26. Lib More: Marc Optical Recognition (MORE), Proposal No. 1047, Directorate General XIII, Action Line IV: Simulation of a European Market in Telematic Products and Services Specific for Libraries, 1992.
27. Ogg H. C. and Ogg M. H.: Optical Character Recognition: A Librarians Guide. Meckler, London. 1992. 171 p. ISBN 0-88736-778-X
28. Ortiz-Repiso V. and Rios Y.: Automated Cataloguing and Retrospective Conversion in the University Libraries of Spain. In Online & CD-ROM Review, (18)3, pp. 157–167, 1994.
29. Parmentier F. and Belaïd A.: Bibliography References Validation Using Emergent Architecture, ICDAR'95, vol. II, pp. 532–535, Montréal, Canada, 1995.
30. Parmentier F. and Belad A.: Logical Structure Recognition of Scientific Bibliographic References, ICDAR'97, Ulm, Germany, August 1997.
31. Schottlaender B.: Retrospective Conversion: History, Approaches, Considerations. Haworth Press, NY. (1992).
32. Simon B.: Recognita Plus: OCR with Strength in Hardware. In PC Magazine (10)50, April 1991.
33. Smith J. W. T. and Merali Z.: Optical Character Recognition: the Technology and its Application in Information Units and Libraries. Library and Information Research Report 33. British Library, London 1985, 125 p.
34. Sle G.: Bibliographic Standards for Retrospective Conversion. In IFLA Journal (16)1, pp. 58–63, 1990.
35. Valitutto V. and Wille N. E.: A Framework for the Analysis of Catalogue Cards. FACIT Technical Report no 2). Statens Bibliotekstjeneste, Copenhagen. October 1996.
36. Wayner P.: Optimal Character Recognition. In Byte, December 1993, pp. 203–210.
37. Wille N. E.: Optical Character Recognition for Retroconversion of Catalogue Cards: Hardware, Software and Character Representation. FACIT Technical Report no 1). Statens Bibliotekstjeneste, Copenhagen. October 1996.

Abdel Belaïd received his Ph.D degree in Computer Science in 1979 and his D.Sc. in 1987 from Université Henri Poincaré Nancy I, France. After a few years as an Assistant Professor he joined the National Center for Scientific Research (CNRS) as a Research Scientist in 1984. His areas of research include Image Processing, Pattern Recognition, Document Analysis and Character Recognition where he has authored over 70 articles. He is the co-author of a book, Pattern Recognition: Methods and Applications. He leads a research group at the UMR LORIA 7503 working on Document Analysis and Text Recognition.