The FACIT Project

dr. Niels Erik Wille Library Advisory Officer
Statens Bibliotekstjeneste (National Library
Authority) Denmark

OCR/ICR in Retroconversion of Older Card Catalogues: The FACIT Prototype

1. Introduction

The objective of the FACIT project (Fast Automated Conversion with Integrated Tools) is to produce a working prototype for automatic formatting and automatic or computer assisted error detection and correction of scanned catalogue cards. The pro- totype is based on existing hardware and software for scanning and OCR, and the overall aim is to develop tools for fast and relatively cheap large scale conversion of typewritten and printed catalogue cards.

Apart from the prototype itself the key results of the project will be a series of methdological tools: Methods for formal analysis of catalogue cards, specially pre-ISBD cards; methods for analysis of typical errors in the character recognition process; methods for the assesment of the quality, speed and cost of retroconversion using OCR/ICR. These will be docu- mented in technical reports.

The prototype software, with accompanying draft manual, will be made availalble to libraries in Europe on a public domain basis for non-commercial use.

The project is supported by the EEC (DG XIII - E) under the Telematics programme, and is expected to be finished in September 1995.

Since the project is based on existing equipment for scanning and OCR it has not been part of the project to develop new or better methods to do this, but rather to assess the possibi- lities and shortcomings of equipment already in the market, especially in the lower price brackets.

The main problems facing libraries and other organizations wanting to use OCR for retroconversion of catalogue cards are the following:

1. The size of the cards (typically 7.5 X 12.5 cm) and the thickness of the cardboard means that only a few scanners may be used to feed the cards automatically. Photocopying the cards first or placing them manually on the glass plate of the scanner is not a practical solution because of the volume of cards (ranging from about ten thousand to several millions in one catalogue).

2. The catalogue cards will normally have been produced over a long period of time, using a variety of typewriters and other printing techniques. The quality of print may vary with the state of the ribbon in the typewriter, and wear and tear on the cards will have left *their marks. This will result in more mistakes in the* basic OCR, as well as less predictable errors. Better results may be achieved if the cards can be presorted *after visual criteria but this is normally neither possible* (cost of labour) or desirable (losing the original sequence of the cata- logue).

3. he character set used in a typical multilingual catalogue, including the range of accents and other diacritics, poses serious problems to most commercial OCR-packages. *The need to recognize text in several* languages at once, even within a single card, means that the typical post-processing modules of commercial OCR-packages, using monolingual dictionaries and linguistic rules to resolve errors produced by the OCR, are not very useful. The fine details of the diacritics - e.g. the need to distinguish correctly between á, à, â, ä and å with several other variations - means that lower (and faster) resolutions in scanned images cannot be used in most cases. The polytonal Greek character set has turned out to be a real hard case in this respect, apart from the fact that few commercial OCR packages support Greek character sets to any extend.

4. The vocabulary of names, titles, place names, publishers and bibliographic descriptions is very special compared to the language of ordinary running text, meaning that even if dictionaries are used, they will to a significant degree have to be created for this specific purpose.

What is needed most of all is a standard OCR-package with good performance in raw recognition of a wide range of characters (basic and extended Latin, basic and extended Greek, and basic and extended Cyrillic), and the ability to learn these charac- ters in "new" variations, like the ones found in older type- writers, and some older typefaces. Our best approximation to this so far has been the Recognita Plus 2.0A for Latin charac- ters and some but not all Greek characters, and a customized Greek OCR package for the extended Greek character set, but others may of course exist or come into existence soon.

For scanning only the Fujitsi 309x-series of scanners have the ability to handle catalogue cards in large numbers, if we look at scanners in lower price brackets. In the group of more ex- pensive scanners we have found a couple, the *Kodak Imagelink 900 and the Hybrid 4512, which are very* fast, but with a low resolution (200 dpi) as standard.

The FACIT prototype itself consists of a series of programs running under Windows on an IBM-compatible Personal Computer (PC).

The programs handle post-processing of the output from an OCR program in order to format bibliographic records and to detect and correct spelling errors in the records. The programs also handle storage and retrieval of intermediary files, editing of files and updating of dictionaries, tables etc. *for use in the formatting and error detection processes.*

Windows is selected as the operating system mainly for two reasons: 1) Most commercially available OCR packages for the IBM-compatible PCs nowadays run under Windows, making it easier to establish a good work-flow if the FACIT application runs under Windows too. 2) The graphic interface of Windows may be utilized to handle a user defined character set for the retroconversion, and to handle the images of the original cards in a fairly simple way.

The overall process is divided into four phases:

1. Scanning and OCR
2. Conversion into Parameter files internal format and
Dictionaries internal character set Tables
Formal descrip- tion of cards
3. Formatting and error Updating tables
detection/correction and dictionaries
4. Conversion into target format (UNIMARC) and
target character set

2. Scanning and OCR

Scanning and Optical Character Recognition (OCR) is not part of the FACIT application as such, but provided by third party hardware and software. Apart from the ability to handle basic needs such as scanning cardboard cards in large numbers and recognizing all characters occurring in the source with a low error rate, the OCR part will have to be able to provide input to the FACIT programs in the following form:

- Bibliographic records in one or more text files ("ASCII text files"), *consisting of printable characters and a few* control characters: NL (New Line), FF (Form Feed) and EOF (End of File). The FACIT application is able to accept 7-bit, 8-bit and 16-bit characters ("UNICODE") as input. Most OCR packages will be able to produce 8-bit characters, making it possible to rep- resent up to about 250 print-able characters simultaneously. The actual number may be less, since the OCR package may only allow letters, punctuation marks and other typographic characters from the ANSI character set or the Code Page system. But to represent the Latin characters (including diacritics) occurring in the European languages, as well as Greek characters and perhaps Cyrillic, one needs more than 250 characters. Recognita Plus 2.0A is able to differentiate between 242 Latin characters, 65 Greek charac- ers, 10 digits and 44 other characters (punctuation marks etc.), in all 362 characters (but of course not all simultaneously). Until the OCR packages are able to handle 16-bit characters, the characters in the source have to be "mapped" to the 8-bit character set of the input file, perhaps using character combinations to represent one character in the source (if the OCR package allows this). This will be translated into a true 16-bit character set used internally by the FACIT programs. One character, like "@" will have to be used to represent an unrecognized character. Form Feed as a separator between cards (pages) may be substituted by a uniqe sequence of characters like "$$", but this is not necessary.

- A series of digitized images in uncompressed TIFF-format, one image for each card (page) in the text-file. The images will have to be named and numbered in such a way that it is easy to establish the necessary links between the indivi-dual images and the pages. This information will have to be provided in the general parameter file (see below *section* 8).

The user will also have to provide three tables with informa- tion about the character set in the source file(s), and how to interpret this during the FACIT processing.

- **A Conversion Table** (in the form of a text file) showing the correspondance between the 7-, 8- or 16-bit character codes of the input file and the character codes of the internal 16-bit character set. The codes are given in hexidecimal numbers. The input character may also be designated as literals in quotes if unambiguous. The Conversion Table may be used to document the character representation of the input file in a manner independent of the character sets of printers and video displays by providing standardized descriptions (ISO 10646) after a "//" (double slash):

```
//  Sample Conversion Table
65, 0065 // Latin small letter a with acute
E1, 03B1 // Greek small letter alpha
"â", 00E2 // Latin small letter a with circumflex
"^a", 00E2 // Latin small letter a with circumflex
. . .
```

This table is used when converting an input text file into records in the internal card format and in the internal 16-bit character set. Other information about these characters are provided by specific font files (see section 4 below).

- **A Transition Table** *(in the form of a text file) showing* the relationship between the characters of the original source (the cards) and the characters of the input file, based on a statistical analysis of a representative sample, *using the same scanner and OCR package with the same* settings on the same type of cards. The characters are represented by the codes of the internal 16-bit character set, in hexadecimal form. Comments to provide interpretation for a human reader may be added after a "//" (double slash).

```
// Sample Transition Table
0061,0061,530 // Latin a - Latin a, 530 times
0061,0065,56 // Latin a - Latin e, 56 times
0061,0073,85 // Latin a - Latin s, 85 times
. . .
```

This table is used in the verification and error detection/correction processes (see sections 6 and 8).

- **A Substitution Table** (in the form of a text file) giving general substitutions to be carried out while converting the source file into the internal format. The table con- sists of two columns separated by commas:

Strings in the source file (may be specified as so-called regular expressions).

Resulting string.

Comments for documentary purposes may be added after a "//" (double slash).

```
// Sample Substitution Table
"Edltor", "Editor"
"ae", "æ"
"I."[a-z]+, "L"[a-z]+ // "I." is changed to "L" when
/  preceding a string of lower
//  case letters . . .
```

Additionally the user will have to provide a formal description of the bibliographic structure and data elements of the cards, but this will be taken up below in section 5 (Formal Specifications).

3. Internal Record Format

During the processing in the FACIT application all records are held in a custom-built object oriented database.

The basic elements of the records are the record identifier, the text from the input file, one record per card, and a link to the image file of the card. This link makes it possible to display the image on the screen when the human operator needs this in order to visually verify or correct the output from the OCR. As the formatting proceeds the record is expanded to incor- porate the various data elements found. The fields and sub-fields are also handled as objects that are established as needed, making the structure (field lengths, number of fields, repeating fields) very flexible.

The field and subfields to be used in the formatting process are determined by the formal description of the cards provided by the user. The internal record format does not put any gene- ral restrictions on the representation of the data elements. This means that the application is easy to customize, but the user will have to provide quite a lot of very specific infor- mation in order to do this.

The built-in database management system will handle creation, editing and deletion of records, and maintaining the link between individual records and the associated image file.

4. Internal 16-bit Character Set

In order to handle the range of characters to be expected in a typical multilingual European catalogue, an internal 16-bit character set is established, with the necessary string-mani- pulation routines, such as string-matching, string-sorting, display of strings etc. These routines are not as yet provided as standard in the programming languages.

The character set provided with the prototype will conform to the first plane of ISO 10646, UCS-2, sometimes called "UNI- CODE", but in principle it is completely customizable by the user. The set includes the correct graphical forms for display on screen for the purpose of monitoring the progress or edi- ting the records. A font editor is provided for customization of the character set and the display of the characters, making it possible for the user to determine exactly how the charac- ters will appear on screen.

The font editor also handles input and display of other infor- mation about the characters, like classification as capital or small letters, links between characters (e.g. telling that "A" and "a" are the same letter in capital and small versions), constituent elements of composite characters (like "^" and "a" for "â"), values for alphabetical sorting etc.

The fact that the characters on screen can be made to look very much like the characters in the source should make it easier for the human operator to visually verify and correct the results, both of the OCR and the formatting process.

5. Formatting Specifications

The formatting of the cards are carried out according to for- mal specifications provided by the user in a special formal language, based on the so-called Backus-Naur notation. The input takes the form of one or more text files (ASCII text files), giving information about dictionaries and the so-cal- led production rules (or just "rules") to be used in format- ting.

This information is the result of the analysis that the library will have to carry out on the catalogue to be converted, since no two libraries or even catalogues require the same rules. One catalogue may require two or more formal specifi- cations in order to cover all possiblities in the simplest way. The FACIT application can be told to make several runs through the records, each time using a different set of rules. In each run only the records rejected by earlier runs have to be processed.

The formal specifications are illustrated by a very simple case below. The space does not allow a more detailed presen- tation of this, but a technical report produced by the project outline the principles and presents the specification language in details. In order to produce such a specification a tho- rough knowledge of the catalogue is needed, as well as more than average understanding of the workings of a formal gram- mar. This will in many cases mean close collaboration between a librarian and a person with some training in computer science.

The sample Formal Specification is to be taken as an illustration of principles, not as a fully worked out, realistic sample.

```
// Sample Formal Specification File
[Dictionaries] PlaceNames = "PLACES.TBL" ;
FirstNames = "FSTNAMES.TBL" ; LastNames =
"LSTNAMES.TBL" ;
[Rules]
//************************* //*  Author  Rules        *
//*************************
Author   = AULastName Comma [Space] AUFstName
{[Space]
          AUFstName} ;
AULastName = InitialWord [ "-" InitialWord ]
|             DictLookUp(LSTNAME) ;
AUFstName  = InitialWord | Initial
            | DictLookUp(FSTNAME) ;
//*************************
//* Title Rules     *
```

```
//***********************
Title = "This is a title" : "Another title" ;
//*************************
//*  Imprint Rules      *
//*************************
Imprint        = ImpPlace [Comma]  Blanks ImpYear
[FullStop]
Pages ; ImpPlace   = (ImpCityName [ Comma Blanks
ImpStateName ] )
| DictLookUp(PLACES) ; ImpCityName = InitialWord |
( "[" InitialWord "]" ) ; ImpStateName = InitialWord
[FullStop] | Initial Initial ; ImpYear    = [ "cop." Blank ]
ImpActYear
| "(" ImpActYear ")"
| "[" ImpActYear "]" ; ImpActYear  = (("14" | "15" | "16"
| "17" | "18" | "19" )
Digit Digit )
| (("14" | "15" | "16" | "17" | "18" | "19" )
Digit Digit )
| "S.A." | "s.a."  ; Pages      = [ RomanNumber ("+" |
",") ] (( Number [ "+"
Number  ]  ("p."  |  "pages.")  Comma  "     "
(RomanNumber | "["  Number "]" ) " tab.")) ;
//*********************** //*
Series Rules     *
//***********************
Series    = "(" SeriesSpec ")" [FullStop] ; SeriesSpec
= SeriesTitle SeriesDelim SeriesNumber ; Series Title
= IntitalWord { Blank Word} ; SeriesDelim  = FullStop
Blank | Blank SemiColon Blank ; SeriesNumber =
Number | "Vol." Number
| ("Vol." | "Bd." ) Blanks Number Comma Blanks
"H. " Number
 | "Hft."  Blank Number |  "H. " Number ;
//****************************
//* Medium Level Rules      *
//****************************
RomanNumber = RomanDigit { RomanDigit } ;
RomanDigit  = "I" | "i" | "V" | "v" | "X" | "x" | "L" | "l"
| "C" | "c" | "D" | "d" | "M" | "n" ; InitialWord = UpperCase
{ LowerCars } ; Initial    = UpperCase FullStop ; Word
= Letter { Letter } ; Number     = Digit { Digit } ; Blanks
= Blank { Blank } ;
//*****************************
//*  Low Level Rules        *
//******************************
 Digit    = "$[Digit]" ; Letter   = "$[Letter]" ; AlphaNum
= "$[AlphaNum]" ; UpperCase  = "$[UpperCase]" ;
LowerCase   = "$[LowerCase]" ; NewLine    =
"$[NewLine]"  ;  CardSepar   =  "$[ESCAPE]"  ;
UnMatched  = "$[Any]" ; Blank    = " " ; Comma    =
"," ; FullStop  = "." ; SemiColon = ";" ; Colon    = ":" ;
Hyphen    = "-" ; Slash     = "/" ; LeftParen = "(" ;
RightParen = ")" ; LeftBrack  = "[" ; RightBrack = "]" ;
```

If needed a separate Formal Specification file may be provided by the user specifying areas of the card containing data ele- ments defined by position in the card, like Top Left Corner, Top Right Corner, Left Margin, Bottom Left Corner and Bottom Right Corner. Then the computer will first identify text or other characters in these areas and place them at a designated location in the sequence of the text. Further handling of this information is then carried out by sections in the Formal Spe- cifications file.

6. Dictionaries

For verification of strings etc. a set of dictionaries may be provided by the user. The dictonaries to be used are listed in the Parameter File (See scetion 8) and at the head of the Formal Specification (see sample above). The formal language includes orders to call dictionaries for specific operations Isuch as verification, substitution of strings or look-up of coded information.

Typical dictionaries could include: Lists of accepted Location Marks or Class Marks. List of publishing places (with country code). List of typical First Names. List of Typical Last Names or Family Names. List of Special Names, like names of kings and other princes, popes etc. List of publisher's names. List of typical words and phrases used in cataloguing: "Edited by", "Herausgegeben von", "S.L.", "S.A." etc. List of word typically used in titles (with indication of language(s) where that word occur).

The FACIT application will include tools to interactively update the dictionaries with new names, words, location marks etc. identified in the process of verification.

A dictionary is a simple text file with comments after a "//" (Double Slash). The entries do not have to be sorted, but alphabetical sorting is useful from the point of view of the human user.

```
//Sample First Name Dictionary
```

"Alexander"

"Alexandra" "Algernon"

"Anders"

"Anne"

"Arthur" ...

"Ben"

"Benjamin"

"Birthe"

....

When checking the dictionaries for verification of strings during formatting or other processes the information provided by the Transition Table may be used to determine possible matches, taking into account possible misreadings of charac- ters. If a match is not found further searching is to be made on strings constructed by substituting error prone characters with characters that could be the correct ones. The most error prone characters will be substituted first and the process carried on until a match is found or the possibilities exhausted. NOTE: These routines have not been worked out and tested yet, so it is not known whether they result in a too slow processing time on the equipment to be used. Research carried out by others seem to show that this may be the case.

If a match has been found after such substitution this may be taken as an indicator that the original string contained an error. The error may be corrected automatically or by user intervention depending on the circumstances and information provided in the Parameter File (se section 8).

7. Specification of Output Format

After the formatting and weeding out of "spelling errors" the resulting formatted records have to be converted from the internal record format into an output file in the target bibliographic format and a character set acceptable to the cataloguing system where the records are going to be used.

This - like the input - makes it necessary for the user to provide two tables:

1. A **Specification file** telling the computer how to tag the bibliographic data elements (kept in the internal record format as separate "fields" or "objects") to conform with the target format. This could be any bibliographic format suitable for exchange of bibliographic records, but UNIMARC is the target format aimed at by the project.

2. A **Conversion Table** telling the computer how the characters in the internal 16-bit character set are to be represented in the output files. UNIMARC is expected to support UNICODE characters in the foreseeable future, in which case no conversion should be needed.

8. Outline of the Overall Process of Using the Facit Prototype.

As indicated above the FACIT application includes a suite of programs to handle the diverse processes of input and output of record files, internal record management, editing of errors that require human intervention, updating of dictionaries and tables, editing of font files, and of course the overall management of the formatting and error detection process.

Dialogue with the user, both the system administrator main- taining the system, and the operator doing the actual conver- sions, takes place through a Windows based graphical inter- face. In the Prototype the language used will be English, but all texts in the interface may be translated into other lan- guages (using both Latin and Greek charac- ters).

The basic setup of the system for at specific conversion task is made by providing a general Parameter File (also an ordi- nary text file, that may be produced with a standard text editor).

The Parameter File tells the system which Dictionaries, Tables and Specification files to use for the session in progress. The Parameter File will also include information about values for certain parameters to be used, like threshold values for error statistics calculated on the basis of the Transition Table.

A formatting session will typically start by reading in the general Parameter file, then the first of the input files with accompanying image files, using the Conversion Table and the Substitution Table. Then the records will be checked for bibliographic records running over more than one card, using information provided in the Parameter file. The text will be concentrated on one record and the images of the cards linked to that record.

When the internal records have been established the formatting process will start. This may be a two step process, first de- composing the cards by extracting information from specific areas like corners and marging, then the actual formatting using the formal specifications provided. Each record is ana- lysed according to the specifications to see if it complies with the rules specified. In the course of this possible er- rors may be detected and presented to the operator for veri- fication and possible correction, or errors may be corrected automatically by the program. The image of the card may be displayed at any time alongside the text as represented in the internal record, so that the operator may check the original text without having to find the original card in the cata- logue. The internal record may be edited directly on the basis of this if needed. It will be possible to browse in the images in sequence in case an image was misplaced in the input process.

If the card is accepted by the formal analysis it is marked as O.K. It will have been updated so that is now includes fields and subfields with the data elements identified during the analysis. If the card is not accepted it will be marked as failed, and it will not be updated - apart from corrections made by the operator.

If the Parameter File specifies a series of Formal Specifica- tions to be used in a certain sequence, the next in sequence will be activated and used to analyse the failed cards. The process is repeated until the sequence is exhausted. Any cards still remaining as failed will then have to be corrected by the Human operator, normally using the image of the card as a reference point. This will be the case with cards that were badly recognized by the OCR for various reasons, and with cards that differ in format and bibliographic contents form the majority of the cards in the catalogue in question. Then the sequence will have to activated again. If cards are still remaining they will have to be output to an Error File in order to be handled "manually".

"Debugging" of the all files used to control the process may be carried out interactively using an editor.

After the formatting, which will include a substantial part of the error detection and correction, certain parts of the records like the title may then be submitted to further error checking using a speller checker or an analysis of character sequences based on n-grams (sequences of n-characters, n being 2, 3, 4 ...). This process may be combined with determining the language of the publication. The details of this part of the proces has not been worked out yet.

An article with more general information about the FACIT project, including extensive bibliographic information, is available from the Project Manager.

Copenhagen, December 1994

Project Partners, Contacts etc.

dr. Niels Erik Wille (Project Manager) Library Advisory Officer Statens Bibliotekstjeneste (National Library Authority) Nyhavn 31 E DK-1051 K¢benhavn K
DENMARK
Tel.: +45 33 93 33
Fax: +45 33 60 33
Internet: Niels.Erik.Wille@sbt.bib.dk

Mr. Hans Erik Jensen
Research Librarian Statsbiblioteket (State and University Library)
Universitetsparken
DK-8000 Århus C DENMARK

dr. Claudia Miconi
Librarian
Biblioteca Nazionale Centrale (National Library of Italy)
Piazza Cavalleggeri 1
I-50122 Firenze
ITALY

dr. Vera Valitutto
Senior Librarian
Biblioteca Nazionale V.E.III (State Library of Naples)
Palazzo Reale
Piazza Plebiscito Napoli
ITALY

dr. Georges Bokos
Head of Cataloguing Department
Ethnike Bibliotheke tes Hellados (National Library of Greece)
32 Panepistimio St.
GR-106 79 Athena
GREECE

Main Subcontractor:
Kim Mikkelsen
Managing Director
SYNERGI Bakkevej 13
DK-2950 Vedbæk
DENMARK

Associated Contractor:
Ivan Boserup
Senior Librarian
Det Kongelige Bibliotek (The Royal Library)
Christians Brygge 8
P.O.Box 2149 DK-1016 K¢benhavn K
DENMARK