

The FACIT Project
LIB FACIT / 1-1044



FACIT Technical Report No 2.
A Framework for the Analysis
of Catalogue Cards

Vera Valitutto & Niels Erik Wille

Revised Version, October 1996

General preface to the FACIT Reports

The present report is one of five presenting the results of the FACIT project.

FACIT is short for Fast Automated Conversion with Integrated Tools. The project has been supported by the EU under the Libraries section of the Telematics Programme. It started in January 1993 and finished in February 1996.

The project has been concerned with two main questions:

1. To determine the feasibility of converting older card catalogues into modern OPACs using scanning, Optical Character Recognition, and automatic formatting into a bibliographic format (such as UNIMARC).
2. To develop a prototype system capable of handling automatic formatting, and automatic or semiautomatic detection and correction of the errors produced by scanning and OCR.

The first objective has been achieved in the sense that the project has shown that such retroconversion can only be expected to be feasible under certain conditions.

The Achilles' heel of fully automatic procedures in retroconversion is still the speed and quality of OCR. And this depends to a large extent on the state of the source material. The project was based on the assumption that commercially available equipment and OCR programs would be able to handle older typewritten and printed catalogue cards in a satisfactory way, so that the main effort could be aimed at formatting the cards, but very much more time has been spent on problems of OCR than was originally envisaged.

The results concerning OCR are based mainly on the use of commercially available scanners and OCR packages in the lower or middle price range (such as seem attractive to most libraries). The results may have been different if more sophisticated (and more costly) equipment had been used, or even custom built equipment. But in general the conclusion has to be that many older card catalogues are not suitable for

this type of methodology because of the state of the source material: Yellowed by age, worn, smudged, with handwritten additions, sometimes swollen or made uneven by dampness, written with a series of typewriters with varying typefaces and with ribbons that are more or less worn out, copied by stencilling etc.

The conclusion is not that the methodology is not feasible at all, but that its application is limited to fairly "well behaved" catalogues. A library wondering whether to apply scanning and OCR to retroconversion should carry out extensive tests in order to assess the suitability of this.

Formatting the cards after scanning and OCR does not, on the other hand, seem to present serious problems, if the output from OCR have a low level of errors. Based on a thorough formal analysis of the catalogue and the rules used in producing it, it will in most cases be possible to write a series of programs specific to that catalogue to do the job. This is confirmed by other projects.

The main focus of the project was to investigate the possibility of producing one application, able to handle a wide range of card catalogues as found in European libraries, avoiding the necessity of writing the formatting programs from scratch every time. This is done by feeding the application a formal description of the catalogue at hand, using a relatively simple formal language. At the same time the application should provide a set of integrated tools for the range of different procedures that go into retroconversion work. The project has demonstrated that this is in fact feasible.

But the work of formal analysis is quite demanding, both in terms of time and the necessary skills and knowledge. And it will have to be done again with each new catalogue, since no two catalogues are exactly alike. This process is needed in order to produce the necessary formal specifications for the formatting programs, both with a system like the FACIT Prototype and with custom built formatting programs. This is definitely a specialist job.

With automatic conversion lot of the costs go into setting up and testing the system with each new library and each new catalogue. This means that this methodology is not suitable for a small or medium size library to handle alone without expert assistance - from a commercial service or a large library that has already done some work in this area.

An important problem that has not been solved in a satisfactory way in this project, is the need for

detecting and correcting errors produced by scanning and OCR. The project has investigated various possible solutions, and it seems worth while to pursue this further. Meanwhile corrections will have to be done by the human operator with some support from the computer.

The second objective of the project, as stated above, has only been partly reached. A software package has been developed that is able to demonstrate the principles involved in automatic formatting of library catalogues and in customizing the procedures for use in libraries with widely different cataloguing practices as well as catalogues produced over time to different specifications. But the package does not include more advanced facilities for error detection and correction, and it still lacks a series of features that are necessary for use in large scale conversion of catalogues.

Nevertheless the results of the project are promising for further development work, and constitute a solid basis for future work by the partners and the subcontractors of the project as well as others. The aim of the published reports is therefore to make available the information generated by the project, in order to help making realistic judgements about the prospects of using the methodology described in a particular library for the conversion of a particular catalogue, and in order to make the information useful for other research and development projects.

The published reports from the FACIT project consists of the following:

Optical Character Recognition for Retroconversion of Catalogue Cards: Hardware, Software and Character Representation. By Niels Erik Wille. (FACIT Technical Report no 1). Statens Bibliotekstjeneste, Copenhagen. July 1996.

The report summarizes the experiences with scanners and OCR programs. Special treatment is given to the question of character sets and representation of characters, since this is normally of great importance in converting multilingual catalogues.

A Framework for the Analysis of Catalogue Cards. By Niels Erik Wille and Vera Valitutto. (FACIT Technical Report no 2). Statens Bibliotekstjeneste, Copenhagen. December 1995.

The report describes the problems involved in analysing a catalogue in order to evaluate the feasibility of converting it by automatic means,

as well as the formal language to be used in setting up the FACIT Prototype. This information should also be useful for someone aiming at developing similar tools for retroconversion.

Error Analysis and Correction in Retroconversion. By Hans Erik Jensen (FACIT Technical Report no 3). Statsbiblioteket, Aarhus. March 1996.

The report summarizes the issues involved in automatic or semiautomatic error detection and correction, and outlines plans for further development of the Prototype in order to incorporate more sophisticated handling of OCR errors.

The FACIT Prototype. Technical Documentation. By SYNERGI (FACIT Technical Report no 4). Statens Bibliotekstjeneste, Copenhagen. July 1996.

The report describes the Prototype in detail and the procedures to use when setting up the demonstration version. The level of information is highly technical. Due to a series of limitations the demonstration Prototype is not suitable for large scale conversion work, but using it with a smaller sample will provide a good grasp of the problems and procedures involved in automatic formatting etc.

Retroconversion of Older Card Catalogues using OCR and Automatic Formatting. Project Overview and Final Report. By Niels Erik Wille (FACIT Technical Report no 5). Statens Bibliotekstjeneste, Copenhagen. 1996.

This report presents the project as a whole and the main results reached. It includes a summary of the information included in the previous reports.

These reports are available free of charge.

A workable demonstration version of the FACIT Prototype is available. This is a combination of a suite of DOS programs and an interface produced as an application for Microsoft Access. The Prototype will run on a PC with Windows 3.11 or Windows 95 and Microsoft Access 2.0 or later versions. The Demonstration Prototype is available free of charge for use in European libraries.

All correspondence concerning the reports and the Prototype should be sent to:

Niels Erik Wille
Senior lecturer
Dept. of Computer Science, Communication and
Education
Building P4
Roskilde University
P.O.Box 260
DK-4000 Roskilde

or posted by e-mail to: new@snow.ruc.dk (Internet)

Information about the project and copies of the reports and the demo-version of the FACIT Prototype are also available on the World Wide Web at the address: <http://www.komm.ruc.dk/FACIT/>

Contents

I. Introduction	7
II. Terminology and Basic Concepts	10
III. General Approach	24
1. Analysis of lay-out	25
2. Structural analysis	29
3. Catalogue description	34
4. The use of existing cataloguing rules	38
IV. Methodological Recommendations	39
V. Presentation of Results	43
1. General Description of the Catalogue	43
2. Inventory of Elements	43
a. Character Set(s)	43
b. Bibliographic Elements	44
c. Bibliographic Vocabulary	45
3. Structural Description of Cards	45
(Cataloguing Rules Formalized)	
VI. References	52
Appendix 1: The FACIT Description Language	56
Appendix 2: Sample Formal Description, with Source and Result	74

I. Introduction

This framework has been developed for the purpose of analysing catalogue cards for processing by scanning/OCR and subsequent automatic formatting. The framework was developed as part of the FACIT project, but the principles involved are not limited to this project, since it reflects general principles of analysing structured textual information.

The principles adopted are those used in computational linguistics for the analysis of syntax, as well as in computer science for the construction of input analysers and compilers. The catalogue card is seen as an example of highly structured text, produced according to a few straightforward rules - relatively seen.

Section II lists the terminology used and the basic concepts of the framework. Section III gives the general approach and the justification for the recommendations. Section IV contains methodological recommendations, focusing on how to collect and process the necessary empirical data. Section V contains the recommendations on how to present the results of the analysis in order to make programming and customization of programs as easy as possible. The last section lists pertinent references, but is not meant as an full bibliography on the subject.

A full presentation of the FACIT Description Language is given in Appendix 1, and a sample description in Appendix 2.

The audience aimed at is both the librarian who has to carry out the analysis and the systems designer or programmer who has to use the analysis in the design or customization of an application package. In order to satisfy this double audience it has been necessary to explain many things that will be commonplace to either one group or the other.

The framework has been established on the basis of the experiences of the participating libraries analysing one or more catalogue sequences. The underlying material is included in the following reports:

Analysis of the Card Catalogue at Statsbiblioteket.
Prepared by Hans Erik Jensen & Dorete Larsen. (FACIT
Technical Report No. 2.1).

Analysis of Three Card Catalogues of Biblioteca
Nazionale Centrale di Firenze. Prepared by Claudia
Miconi & Gian Luca Corradi. (FACIT Technical Report
No. 2.2.1)

Descrizione formale delle schede del Catalogo
Palatino, Biblioteca Nazionale Centrale di Firenze.
Prepared by Databae Informatica. (FACIT Technical
Report No. 2.2.2)

Description of Italian Cataloguing Rules: 1886 -
1979. Prepared by Rosella Ruoppolo & Vera Valitutto
(FACIT Technical Report No 2.3.1)

Lex Rules in FACIT Format for BNN Catalogue - a first
approach. Prepared by StudioErre di Gianluigi Visco.
(FACIT Technical Report No 2.3.2)

Descrizione formale delle schede del Catalogo BNN.
Prepared by Stefano Tulini. (FACIT Technical Report
No. 2.3.3)

Analysis of the Card Catalogue of the National
Library of Greece. Prepared by Joanna Demopoulos.
(FACIT Technical Report No. 2.4)

Comment: These reports have been produced for use
in the project, and are not generally available.
Interested parties will have to approach the re-
presentative of the responsible partner if they
want more information about the reports.

While a lot has been written about the principles of
bibliographic description, the way to apply the In-
ternational Standard Bibliographic Description (ISBD)
to modern printed catalogues and how to create good
online catalogues (OPACs), surprisingly little has
been written about the practices found in older card
catalogues. The project has had to rely very much on
the empirical data elicited from the catalogues of
the participating libraries, supplemented by manuals
and cataloguing rules applied there.

It is obvious from this work that from the point of
view of a human investigator the same basic prin-
ciples are applied in most card catalogues. But one
finds such an abundance of variations and different
solutions to the same problem, that it will not be
possible to write one computer program to handle the
conversion of all card catalogues without very
extensive customization.

This means that each catalogue will have to be analysed in depth and then described in terms that can be interpreted and processed by a computer program. The present report passes on the experience of the FACIT project on how to do this.

II. Terminology and Basic Concepts

This section includes the basic terms used in the framework. Definitions have sometimes been left out when the use and meaning of the term should be obvious to the reader.

For the preparation of the list of terms Harrod's Librarians Glossary and Reference Book, Fifth Edition 1984, has proved invaluable. The terminology and the definitions adopted have also been influenced by the ISBD manuals and the UNIMARC manual, as well as the special needs of the present project, in which certain cross-linguistic problems have also been taken into account.

If not obvious from the wording, the terms are marked with a scope note, telling whether it belongs to the world of catalogues or the world of computers. Terms relating to language and writing are left unmarked.

Abbreviated Entry (catalogue): Usually an added entry (title, secondary author, translator or subject), but containing less than the full bibliographic information found in the Main Entry. See also: Added Entry; Main Entry.

Accent: See Diacritical mark

Access Point (catalogue): A unique heading, giving access to the books and other items in the library or bibliography. See also: Added Entry; Main Entry.

Accession number (catalogue): A number given the book or other item in the Accessions Register of the library in the order of arrival to the library.

Accessions Register (catalogue): The chief record of the items added to a library. The books and other items are normally registered in the order that they are added to the holdings.

Added Entry (catalogue): An entry supplementing the Main Entry to give extra access points. An Added Entry may be a duplicate of the Main Entry apart from the Heading or it may be an Abbreviated Entry. In retroconverting catalogues with added

entries special provisions will have to be made to reduce duplicate bibliographic descriptions without losing any information only found in the added entries. See also: Abbreviated Entry; Analytical entry; Main Entry.

Alphabet: A set of characters in a specific form, like Latin characters, Greek characters, Cyrillic characters etc.

Alphabetical catalogue: Catalogue organized alphabetically by Headings (author's names and titles), including Main Entries, Added Entries and Reference Entries). See also: Dictionary Catalogue; Systematic Catalogue.

Alphanumeric (computer): A contraction of "alphanumeric"; a sequence of characters which may be letters or digits (numerals), but not punctuation marks or special characters such as typographical or mathematical symbols.

Alternative Entry (catalogue): Use Reference Entry.

Analytical Entry (catalogue): An entry in the catalogue for part of a book, periodical or other publication, i.e. an article or a contribution of separate authorship to a volume of essays, festschrift, serial, volume of musical compositions etc. The entry includes a reference to the work containing it. This work is represented separately in a Main Entry. See also: Added Entry; Main Entry; Reference Entry.

ANSI: American National Standards Institute.

ANSI characters (computer): A set of character codes (8-bit) used as the standard character set of the Windows operating system. Includes the set of Latin characters termed "Latin-1". See also ASCII characters.

ASCII characters (computer): (American Standard Code for Information Interchange) A standard code for the representation of characters (letters, digits, punctuation marks and special characters) used in computers. Each character is nowadays represented by 8 bits (extended ASCII) making it possible to use 256 different characters. The first 128 characters are fixed and includes some non-printing "control characters" such as "Tab", "New Line" and "Carriage Return". The characters from 129 to 256 have variable meanings according to the "Code Page" used. This part includes the so-called "international characters", that is characters not included in the English alphabet.

Since the first character has code 0 (zero) the actual codes range from 0 to 255, with code 128 as the first of the variable set changing from Code Page to Code Page. The ANSI character set is a special implementation of the extended ASCII character set, but with codes in the range 128 - 255 that are completely different from the Code Page system.

ASCII file (computer): A machinereadable file that only includes the printing characters of the ASCII code together with "Tab", "New Line" and "Carriage Return". The interpretation of codes 128-255, e.g. in the display of the characters on the screen, depends on the set up of the viewing machine.

Author/Title Catalogue: Use Alphabetic Catalogue.

Authority File (catalogue): Use Authority List.

Authority List (catalogue): List of "authorized" forms of the names of personal and corporate authors, titles of works and terms for topical subjects established by the responsible cataloguing agency to get a more homogenous search tool. These are termed Uniform Headings. An authority list may also contain variant, non-preferred forms as access points.

Authorship statement (catalogue): Statement of the authorship or responsibility of the work described. Includes: Personal Author; Corporate Author.

Back-written card (catalogue): Use Front-and-back-written card.

Backus-Naur Form (computer): A notation for formal description of syntactical rules (rewriting rules) of a context free grammar, named after the computer science pioneers J.W.Backus and Peter Naur.

Bibliographic description (catalogue): The description of a book or other item in the library according to some set of cataloguing rules.

Bibliographic element (catalogue): The elements of a bibliographic description. This framework recognizes the elements of bibliographic description included in the ISBD(G): Title; Statement of Authorship (Responsibility); Edition Statement; Type of Publication; Statement of Publication and Distribution (or Imprint); Physical Description (or Collation); Series; Notes; Multilevel De-

scription. (cf IFLA, 1977 and IFLA, 1987). But the term also includes other elements typically found in library catalogues: Heading (including Subject Heading(s)); Location Mark(s); Accessions Number; Class Mark(s); Tracings.

BNF (computer): See Backus-Naur form.

Card (catalogue): The elements of a card catalogue: A piece of rectangular cardboard or paper of a standardized size containing bibliographic descriptions of the works in the library and ordered alphabetically or systematically according to some feature of the card (i.e. the Heading or a Class Mark). See also: Standard Library Card.

The contents of the card depend on the type of catalogue: Alphabetical Catalogue; Dictionary Catalogue; Systematic Catalogue; Topographical Catalogue etc, and also on the type of materials described: Books (monographs); periodicals (serials); maps, prints etc.

Card Catalogue: A catalogue whose elements are cards.

Catalogue: A list of bibliographic descriptions of items held by a specific library, set up to give access to the items. The description usually includes information about the location of the work in the library and the number of copies held. A Dictionary Catalogue or a Systematic Catalogue also includes information about the content (subject) of the work.

Cataloguing Rules: The formal and informal rules used in producing a specific catalogue. The rules determine what bibliographic elements to select, how to formulate the entries and how to represent them in the catalogue.

Character set: A full set of characters (letters, digits, diacritics/accents, punctuation marks and other characters) in one alphabet, disregarding size, typestyle and typeface (font).

Class Mark (catalogue): A sequence of characters representing a subject field in a specific Classification Scheme.

Classification Code (catalogue): Use Class Mark.

Classification Scheme (catalogue): A scheme of subject "classes" used to categorize the works in a collection according to the content or subject of the work. The subjects are organised in a hierarchy of classes and subclasses represented by specific Class Codes or Class Marks. These may be

composed of numbers, letters or combinations of numbers and letter and other characters. Two widely used Classification Schemes are Dewey Decimal Classification (DDC) and Universal Decimal Classification (UDC) both using numbers based on the decimal principle.

Classified Catalogue: Use Systematic Catalogue

Collation (catalogue): Use Physical Description

Corporate Author (catalogue): An institution, private firm or learned society etc. responsible for the content of the work, as opposed to a Personal Author. In some catalogues a Corporate Author is entered in the same place as the Publisher, usually after the Place of Publication, in others in the same place as a Personal Author.

Diacritic(al mark): Any character or mark, used to differentiate the phonological "meaning" of a basic character, i.e.: a: ä á à â; n: ñ; c: ç. The term is often opposed to accents that are used to mark stress or other features that do not strictly speaking affect phonological "meaning". Some letters look like a basic letter modified with a diacritical mark, (e.g. the letter å in the Scandinavian languages) but are actually basic characters in themselves. These distinctions are not respected in all cases in this report, since they are not important in this context.

Dictionary Catalogue: A catalogue with subject headings as well as author's names, titles and reference headings as access points, all organized in one alphabetical sequence. See also: Alphabetical Catalogue; Systematic Catalogue; Topographical Catalogue.

Digit: The characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

DOS (computer): Disk Operating System. Often used synonymously with Microsoft DOS, the operating system used for IBM-compatible personal computers, based in the Intel 80x86 series of CPU's. See also DOS/Windows Environment.

DOS/Windows Environment (computer): A hardware platform using Microsoft DOS and Microsoft Windows as the environment for application programs.

Edition (catalogue): One edition of a work is the collection of all copies printed from the same master at the same time. Two editions printed at different times may be identical as to content

and form. If changes have been made a revised edition is the result. If one edition is reissued at a later time it is called a reprint. In the bibliographic description the edition is indicated by a number or a phrase. A statement of authorship or responsibility related to a specific edition may be found in the Edition Area, as well as additional Edition Statements.

Electronic Catalogue: Catalogue produced using computer technology. Used in the report as a general term comprising among other things the OPAC (On-line Public Access Catalogue).

Entry Word (catalogue): Use Heading.

Field (catalogue): Part of a catalogue record representing a specific bibliographic element. May be divided into subfields (e.g. an author field divided into the subfields: authors' first names and authors surname.)

Field (computer): Part of a database record representing some specific type of information.

Field name (computer): The name of a field in a database. The name has to be constructed according to rules specific to the database management system in question.

File (computer): A sequence of digitally recorded data to be handled as a whole. A file has an identifying name and may be physically divided into blocks, records or other units needed by the storing device or other type of access means.

File name (computer): The name used to refer to a file. In the DOS/Windows environment a file name consists of one to eight printable characters (letters A-Z, digits 0-9, and '_' (Underscore), '^' (Caret), '\$' (Dollar sign), '~' (Tilde), '!' (Exclamation sign), '#' (Number sign), '%' (Percent sign), '&' (Ampersand), '-' (Hyphen), '{' '}' (Braces or Curly brackets), '@' (Commercial at), ''' (Single quotation mark) and '(' ')') (Parentheses)), optionally followed by a dot and one to three printable characters (with the same restrictions).

Follow-on Card (catalogue): The second, third etc. card when one entry takes more than one card. The existence of Follow-on Cards are usually marked in a special way on the first card of the sequence, i.e. by writing "See next card" in the bottom right corner. The Follow-on Cards may also be marked in a special way.

Form of Card (catalogue): Refers to several physical attributes of the cards in a catalogue: size; material used (paper, cardboard ...); production method (handwritten, typewritten, stencilled, off-set printed, type-set printed, photocopied, mixed); front-written or front-and-back-written.

Formatting (computer): The process of dividing a file or a string of characters into segments representing a specific entity or type of information.

Front-written card (catalogue): Card written only on the front side. See also: Front-and-back-written card.

Front-and-backwritten card (catalogue): Card written on both sides, so that the information on the front side is continued on the back side. Used in some card catalogues instead of follow-on cards.

Heading (catalogue): The sequence of characters (forming a number, a name, word or phrase) chosen as Access Point with the purpose of (a) arranging the cards in the sequence of catalogue and (b) grouping related entries together. It is usually written at the beginning of the catalogue entry - usually on a separate line and sometimes in larger type than the rest of the entry

Holdings (catalogue): The works held (owned) by the library.

Imprint (catalogue): Information about the printing, publication and distribution of the work. Publisher's Imprint: The name of the publisher and the date (year) and place of publication. Printer's Imprint: The name of the printer and the date (year) and place of printing.

Indentation: The distance from the left edge of a catalogue card at which various parts of an entry begin. A line is called indented if it begins one or more positions to the right of the first printing position on the card. The practice of using indentations varies from library to library. The Heading will often start at the first (leftmost) printing position, while the Main Body of the card uses indented lines, but there is no clear standard. Some libraries use three indentation positions: The first position is for the Heading, the second for the Title and the third for supplementary parts of the description.

Indexing Terms (catalogue): Terms indicating the content (subject) of the work using words in natural language, taken from the work itself, from a con-

trolled list of subject terms (thesaurus), or from the domain dealt with, at the discretion of the indexer.

International Standard Book Number: A number of 10 digits meant to uniquely identify a book by publisher, title, edition and volume number. The final digit is a check digit. The ISBN was adopted by ISO, the International Standards Organization, in 1970.

International Standard Serial Number: A number of 8 digits of which the last one is a check digit. An ISSN is meant uniquely to identify a serial publication such as a journal, magazine, yearbook etc.

ISBD rules (catalogue): International Standard Bibliographic Description. A set of rules for bibliographic description produced by IFLA, the International Federation of Library Associations. The first edition of the rules for description of monographs (books), ISBD(M), were issued in 1974 (revised 1987). The general framework, ISBD(G), was published in 1977. The ISBD rules aim - among other things - at presenting bibliographic information in such a way that it can be understood without being familiar with the language of the publication, typical names etc. To do this ISBD relies on a special punctuation as a lead in to bibliographic elements, such as a '/', Slash, in front of the first author and a ';', Semicolon, in front of each of the following authors.

Language Code (catalogue): The language of the publication as represented in a standardised code, e.g. the language codes of the UNIMARC manual.

Language of publication (catalogue): The language that the publication is written in, which may be different from the language of cataloguing.

Language of cataloguing (catalogue): The language used in the bibliographic description, which may be different from the language of the publication described.

Lay-out of Card (catalogue): The organisation of the written or printed elements on the card, with specific areas used for specific bibliographic elements: Main Body of the Card; Heading; Top (Top Left Corner, Top Center; Top Right Corner); Left Margin; Right Margin; Bottom (Bottom Left Corner; Bottom Center; Bottom Right Corner).

Letter: Characters such as: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, ß, t, u, v, w, x, y, z, æ, ø, å, and any composite characters produced by combining these with diacritics etc.. (And the equivalent in Greek, Cyrillic etc.) The total number of different Latin letters, including composite characters, recognized by ISO/IEC 10646-1 is around 750.

LEX (computer): Lexical analyser. A part of a suite of programs, LEX and YACC, used to generate parsers for syntactical analysis of computer input or computer programs. See YACC.

Literal (computer): An expression referring to a single character or a string of characters to be taken literally, as that character or string itself.

Location (catalogue): The place in the library where the material is to be found. This is indicated in the catalogue by a Location Mark. Location is used in a wide sense in this report.

Location mark (catalogue): A string of letters, digits and other characters used in a catalogue entry, book list or bibliography to indicate the library, collection or position on the shelves where the book or item in question may be found. Also called Location Symbol. Location Mark is used in a wide sense in this report to correspond with the term: "Signatur" (da,sw,no,de), "segna-tura" (it), "signatuur" (du), "signature" (fre) etc. used in other European languages.

Location Symbol (catalogue): Use Location Mark.

Main Entry (catalogue): The entry in the catalogue containing the fullest bibliographic description and the preferred Heading (main author or title according to the cataloguing rules used). See also: Added Entry; Analytical entry; Reference Entry; Unit Card.

MARC format (catalogue): MACHineReadable Cataloguing format. MARC formats exist in several variations: US-MARC, UK-MARC, danMARC, finMARC. UNIMARC is an attempt to standardize the MARC format, and is produced by IFLA, the International Federation of Library Associations. The MARC formats are based on tagged fields and subfields of variable length and with the possibility to repeat fields and subfields as needed. The tags to be used are defined in the manuals of each format. Basically the format is a tape format, an implementation of

ISO 2709, but in some cases a text format (line format) may also be used.

Multivolume Work (catalogue): A non-serial publication issued in a number of separate parts known to have been conceived and published as an entity; the separate parts may have different authorship and individual titles as well as a title covering the whole work.

Numeral: Use Digit.

OCR (computer): Optical Character Recognition. A technique for recognizing characters (letters, digits etc.) in a scanned image of a text. Used for converting printed or typewritten text into digital text.

OPAC (catalogue): Online Public Access Catalogue. Use Electronic Catalogue.

Parser generator (computer): A general program that will produce a parser on the input of some formal specifications.

Parsing (computer): The process of dividing a string of characters into syntactical units. A close analogy is the analysis of a natural language sentence (string of words) into grammatical units: Subject, Verb, Object etc.

Personal Author (catalogue): A person responsible for the content and form of a work. See also: Corporate Author.

Physical Description (catalogue): Pagination and/or number of volumes; Illustration statement; Size; Accompanying material. Also called Collation.

Plain text file (computer): A file containing only printable characters and the non-printable characters Tab, New Line and Carriage Return. This is also called an ASCII file because the characters codes used often are the ASCII 7-bit or 8-bit character set.

Production (computer): Another word for Rewriting rule.

Punctuation mark: The characters: , . : ; ! ? - () [] < > " ". In the ISBD character set also: / = and &.

Reference Entry (catalogue): An added entry in a catalogue referring the user to some other part of the catalogue. This could be a "See" entry

telling the user to use some other term or some other form of a name, or a "See also" entry telling the user to consult another access point as well as the current one.

Repertoire of characters (catalogue): The set of characters actually occurring in a given catalogue. It is important to make sure that the OCR-package is able to handle the whole repertoire of characters, while characters from the full character set(s) not occurring may be ignored, i.e. in training the OCR-package. The same considerations will have to be taken with the error detection/correction modules of the formatting package.

Reprint (catalogue): See Edition.

Revised Edition (catalogue): See Edition.

Rewriting rule (computer): A part of a formal syntactical description, where the structural entities (terms) are defined through possible substitution by other entities:

<Term1> => <Term2> <Term3> <Term4>

Rewriting rules are also called Productions or Generative rules.

Scanning (computer): The term has several meanings in the computer context. In this report it is used to refer to the process of transferring an image to digital form using a special type of "reader", a scanner.

Secondary Entry (catalogue) Use Added Entry.

Series Area (catalogue): Series statement; Sub-series statement; Numbering within series; International Standard Serial Number.

SGML (computer): Standard Generalized Markup Language. Standard proposed by the publishing industry for electronic manuscript preparation and markup, accepted by the International Standards Organization as ISO 8879, 1986. The standard uses linked tags formed according to a predefined syntax. Special implementations of the standard provides predefined tags for specific purposes, such as publication of journal articles.

Shelf list (catalogue): A list of the books in the library, arranged in the order of the books on the shelves. The entries in a Shelf List are often Abbreviated Entries.

Shelf Mark (catalogue): A string of letters and/or digits indicating the shelf on which a book or other item is to be found. A special case of Location Mark.

Short Cataloguing: Use Short Form Cataloguing.

Short Form Cataloguing: The style of cataloguing in which the entries give author, main title and publication year only.

Signature (catalogue): In many European languages a word like: "Signatur" (da, no, sw, de), "segnatura" (it), "signatuur" (du) and "signature" (fre) is used to signify information typically put in the top right or left corner of a catalogue card. This could be a Location Mark, a Class Mark or an Accessions Number. In English the word "signature" means only: 1) Parts of a book. 2) Marks (letters and/or numbers) printed at the bottom of the first page of a section as a guide to the binder. 3) A personal signature (name or initials) written in a persons own hand to authenticate a document. In this report Location Mark is used to translate the more general terms "Signatur" in Danish and "segnatura" in Italian.

Size of type: Characters in the same typeface but with a specific size.

Special characters: Characters such as: @ £ \$ & # % { } [] < > / \ and +.

Standard Library Card (catalogue): A card used for cataloguing of the internationally agreed size: 7.5 * 12.5 cm. (3 * 5 inches). The hole at the bottom of the card should be 7.9 mm in diameter and be positioned 4.8 mm from the bottom edge. Sometimes an A7 card is found, measuring 7.4 * 10.5 cm.

Subject Heading (catalogue): An expression in words, representing a subject field or group of subjects.

Systematic Catalogue: Catalogue organized according to a classification system such as Dewey Decimal Classification (DDC), Universal Decimal Classification (UDC) and Danish Decimal Classification (DK5).

Stock (catalogue): Use Holdings

Style (Characters) Use Typestyle.

Token (computer): (=Symbolic substitute) A string of characters "standing for" or acting as substitutes for one or more characters or strings of characters (Literals) in the formal descriptions of structured text and in the actual processing of the text.

Topographical Catalogue: Catalogue organized according to the geographical or topographical areas described in the works.

Tracing (catalogue): Information in a Main Entry card about alternative entries (access points) to be produced. This information is used to search (trace) the added entries if the Main Entry is revised. If a Unit Card is employed, tracings will be found in all entries. Tracings are usually found in cards produced according to the ISBD rules.

Type of Card (catalogue): The bibliographic type of cards found in the catalogue: Main entry card; added entry card (or secondary entry card); reference card (See ... See also ...); analytical entry (In ... It is in ...); follow-on card, that is the second, third etc card where one entry takes more than one card.

Type of Material (catalogue): The type of material represented in the catalogue: Books; periodicals; music scores, prints, manuscripts, microforms, sound tapes, video tapes, grammophone records etc.

Typeface (Font): Characters in one alphabet but with a specific form, i.e. Courier, Elite, Pica, Let_ter Gothic, Times Roman, Dutch, Helvetica etc. Characters in the same typeface may have different sizes and different typestyles.

Typestyle: Characters in the same typeface, but with a specific form: i.e. ordinary, bold, italic, underlined.

Uniform Heading (catalogue): The form of a Heading adopted for use in the catalogue for an author (personal or corporate), title, or for any other heading. See also: Uniform Title; Authority List.

Uniform Title (catalogue): The distinctive title by which a work, which has appeared under varying titles and in various versions, is most generally known, and under which catalogue entries are made. A uniform title may also be used to link a translation with the original. See also: Authority List.

UNIMARC (catalogue): A format for producing Machine-Readable Catalogues, issued by IFLA, the International Federation of Library Associations. The first edition was issued in 1977. The UNIMARC format is an attempt to standardize the various MARC formats used in the U.S.A. and in several European countries. The MARC formats are specially constructed to handle bibliographic information and enables the use of fields of variable length, a hierarchy of fields and subfields and repetition of fields. See also MARC format.

Unit Card (catalogue): A basic catalogue card, in the form of a Main Entry card, used to produce all entries in the catalogue. An added entry is created by adding any heading necessary to a copy of the card.

UNIX (computer): Computer operating system designed to support the use of machine-independent software. UNIX is multi-user and multi-tasking. It is programmed using the C language and provides many tools that can be integrated with new C applications. UNIX is implemented on a wide variety of platforms from PCs to mainframes.

Windows (computer): Computer operating system produced by Microsoft. It runs on computers based on the Intel 80x86 series of central processing unit (CPU), the so-called PCs or IBM-compatible PCs. The interface is graphical, based on "windows" and interaction with a combination of a keyboard and an "mouse". Together with the DOS operating system it provides a standard environment for personal computers.

YACC (computer): "Yet Another Compiler Compiler". A tool originally created for the UNIX environment to help produce "parsers" or syntactical analyzers for computer input. It works with another program called LEX. (Bennet, 1990. Mason Brown, 1990).

III. General Approach

The FACIT project aims at developing a working prototype for - among other things - automatic formatting (structuring) of catalogue information. This application is produced as a tool for retroconversion of catalogue cards using scanning and optical character recognition, but can also be used with copies of the existing catalogue produced by direct keying to a plain text file.

The first step of the retroconversion process is the transformation of the existing catalogue into a machinereadable file in a standard text format - in this project a plain text format is assumed ("ASCII format"). This is to be produced using fast scanning and optical character recognition. The problems of handling catalogue cards and recognizing the text is the subject of another report: FACIT Technical Report no 1: Optical Character Recognition for Retroconversion of Catalogue Cards: Hardware, Software and Character Representation.

Other parts of the project will investigate problems of cost and planning for efficiency. The aim of the present report is to focus on the bibliographic data in the existing catalogue and the way to capture these data in the retroconversion process. The question of error analysis is treated in FACIT Technical Report No 3. Error Analysis and Correction in Retroconversion.

The records in the text file will be character-by-character copies of the original catalogue entries, retaining as far as possible the original lay-out of the card. The formatting program will work on the information present in the copy, that is character strings and lay-out.

For the formatting program to work it is necessary to provide a formal description of the catalogue in which the bibliographic elements to be recognized by the program are linked to the character strings and the lay-out.

Even with libraries formally using the same cataloguing rules the variations in the actual application of these rules in cards turn out to be so many that is not possible to produce a single program that is

able to handle all cards from all catalogues without extensive customization.

And most existing card catalogues will have been produced over a period of time with changing cataloguing standards and more or less conscious variations and deviations in the application of these, so that even with one catalogue a single, custom built program will not be enough.

Experience from this and other projects using scanning, OCR and automatic formatting of records shows that each library will have to invest a considerable effort in describing the existing catalogue in a way that is useful for processing by a computer. This may then be used as specifications for a suite of custom built formatting programs, or - as with the FACIT Prototype - as input to a set of very general formatting programs requiring customized and very detailed formal descriptions specific to the job at hand.

1. Analysis of lay-out

A useful approach seems to be first to analyse the general lay-out of the cards in the catalogue, which could be as in the sample shown below:

Top Left Corner		Top Right Corner	
Left Margin	Heading		
	Main Area: Title and Statement of Authorship - Edition - Imprint - Collation		
		Supplementary Area: Series - Notes - ISBN - Binding - Price - Number of copies.	
Bottom Left Corner	Hole	Bottom Right Corner	

Different libraries might use different lay-outs, i.e. using only the main area, or having a separate area in Top Center. The important thing is to describe the catalogue as it is actually organized by the library.

The following examples show some variations in layout and cataloguing rules from actual card catalogues, illustrating what to look for.

disp. 85-84

Meuser, Michael
Stärkung von Handlungskompetenz zum Verhältnis
von verstehender Soziologie und Sozialpädagogik
/ Michael Meuser.
Bonn, 1983.
V, 363 s.

Disputats, Bonn 1982

(hole)

Ars

E
O c
27

The dilemmas of government expenditure.
Essays in political economy by economists and
parliamentarians. [By] Robert Bacon [a.o.].
Publ. by the Institute of Economic Affairs.
(London) 1976. XI + 110 s.

(IEA readings, 15).

TV

70.917 Vom Klang der Bilder. Die Musik in der Kunst
des 20. Jahrhunderts. Herausgegeben von Karin
v. Maur. München, Prestel, 1985. 480 sider, il-
lustreret (s/h-f).

Udgivet i forbindelse med udstillingen af
samme navn i Staatsgalerie Stuttgart 6. Juli
bis 22. September 1985.
Litteraturhenvisninger.

70.917-78-70.99-red

(hole)

r

86/1489

61.642
(38.43)

Keen, Ernest
Three faces of being. Toward an existential
clinical psychology.
New York (Cop. 1970).
XII + 367 s.

(The century psychology series)

DSH

(hole)

8-80/945

Marryat, F.
Die Kinder im Neuwald / in einer gekürzten
Fassung nach Frederick Marryat ; [aus dem Däni-
schen von Elke Pirck] ; Ill. von Edward Mortel-
mans.
Reinbek bei Hamburg : Carlsen, [1979].
62 s. : ill.

(Carlsen Abenteuer-Bücher ; 16).

Originaltitel: The Children of the New-Forrest

83-196

(hole)

K

139085

CR 4.22 Kem

Kemeny, J.G. and T.E. Kurtz
BASIC programming. 2.ed. New York, Wiley, 1971.
150 s.

(hole)

881-524

International Symposium on Steroid Induced Uterine Proteins, 1979, Marburg
Steroid induced uterine proteins : proceedings of the International Symposium on Steroid Induced Uterine Proteins held in Marburg, West Germany, 28-29 September, 1979 / M. Beato, ed.
Amsterdam : Elsevier, 1980.

376 s. : ill.

(Developments in endocrinology ; vol. 8)

ISBN 0444802037

Kongr.: Biog 48, (hole) 68

U2

See next card

TUREN gar til Paris, in italiano.
Parigi. /Testo Ermann Dedichen/. Nuova ed., rist.
Milano , A. Vallardi, 1979.
95 p. ill. 20 cm (Guida del turista,27).
Ed. f.c.
I. Parigi--Guide I. Dedichen, Ermann II. Tit.

914.436

E M E R S O N RALPH WALDO.

The Conduct of Life. By R.W.E. The
Riverside edition.

London. G.Routledge and sons(W.Clo=
wes and sons), 1898, 16°(mm.186x122),
p.308.

(Emerson's complete works, VI).

(Hole)

An analysis such as this might show that some of the bibliographic elements to be identified by a formatting program are always associated with a certain area of the card, such as the top left corner. Any-

Most of the white space has been reduced to nothing or to empty lines, containing only a New Line character "␣". The spaces retained in the copy are illustrated using the character: "■" The left pointing arrow: "◀" represents New Line. The letters "XYZ" at the bottom represent the noise produced by the hole at the bottom of the card. The dollar signs: "\$" mark the beginning and end of the record.

The errors which will be found in any actual example are left out for the sake of the presentation. They are assumed to be handled by a series of error detection and error correction modules.

The picture of the computer file is still somewhat misleading. Another way of representing it is as a stream of words, punctuation marks etc.:

```

@CardSeperator
@NewLine
@Space * 41
"disp"
@Dot
@Space
"85"
@Hyphen
"84"
@NewLine
@EmptyLine
@Space * 6
"Meuser"
@Comma
@Space
"Michael"
@NewLine
@Space * 8
"Stärkung"
@Space
"von"
@Space
"Handlungskompetenz"
@Space
"zum"
@Space
"Verhältnis"
@Space
@NewLine
@Space * 8
"von"
@Space
"verstehender"
@Space
"Soziologie"
@Space
"und"
@Space
"Sozialpädagogik"
@NewLine
@Space * 8
@Slash
@Space
"Michael"
@Space
"Meuser"
@Dot
@NewLine
@Space * 8
"Bonn"
@Comma
@Space

```

```

"1983"
@Dot
@NewLine
@Space * 8
"v"
@Comma
@Space
"363"
@Space
"s"
@Dot
@NewLine
@EmptyLine
@Space * 8
"Disputats"
@Comma
@Space
"Bonn"
@Space
"1982"
@EmptyLine
@EmptyLine
@EmptyLine
@Space * 27
"XYZ"
@NewLine
@CardSeperator
@NewLine

```

"Words" are given as strings enclosed in double quotation marks, while all other information is represented as "tokens", marked by a "@" (Commercial at). This is the type of information that the formatting program has to rely on.

The next step in the process will depend on the application actually used. The example given below illustrates some general principles.

To produce a formatted record the application will have to recognize the structure of the card using only the sequence of characters including Space, Tab and New Line characters.

To the human reader - especially one who knows the conventions used when the original card was produced - it is obvious that "disp. 85-84" in the top right corner represents information about the location of the publication in the library. "disp." means that it is a dissertation, "85-84" means accession number 84 in 1985. The location is in the collection of dissertations where the publications are ordered by year and accessions number.

The computer will be able to recognize this as a string of characters placed at the end of the first line on the card (after more than 30 Spaces), matching a general description like: the string "disp." followed by a Space and a new string consisting of two digits (range 00 to 99), a hyphen and one or more digits, and terminated by a New Line character.

The next field of information is the Heading, holding an author's name in inverted form. The Heading field starts 6 Spaces from the left edge of the card. The structure is one or more Surnames separated by Spaces, followed by Comma, Space and one or more First Names separated by Spaces, and terminated with a New Line.

If the computer has access to a "dictionary" of Surnames and First Names, it can verify that "Meuser" is a Surname and "Michael" a First Name. Otherwise a Surname may be recognized as a string of letters starting with a capital letter followed by small letters only and terminated with a Space or a Comma. A First Name is a string of letters starting with a capital letter followed by small letters only and terminated with a Space or a New Line. All names encountered before the Comma are taken as Surnames, and all names after the Comma as First Names.

The next field is the Main Title. It starts 8 Spaces from the left edge of the card and consists of a sequence of letters, digits and punctuation marks. It may include one or more New Line characters followed by 8 Spaces, since the title may run over more than one line. The termination of the Title is clearly marked with a '/', Slash, which actually is the most important clue to the structure from the point of view of the computer. The New Line characters and the leading 8 Spaces are not part of the title and should be discarded.

A Subtitle would have been separated from the Main Title with a ":" (Colon) and a Parallel title with a "=" (Equal sign). This means that it would be easy to recognize something as either a Subtitle or a Parallel Title.

The string of characters starting from the Slash and terminated with a "." (Full Stop or Dot) and a New Line is the authorship statement which will contain the names of one or more authors in direct form, separated with Commas, and perhaps initiated with a string like: "by", "af", "Herausgegeben von" etc. This is fairly easily recognized by the computer, but the distinction between First Names and Surnames may require another dictionary check, or an arbitrary decision to always take the last part of a personal name as the Surname and the rest as First Names. This will work in the majority of cases, but not with certain non-European names and not with early European names.

The string "Berlin" which starts the next line (again after 8 Spaces) is easily recognized as a Place of Publication using a dictionary check. After Comma and Space comes 4 digits with "19" as the first two and then a Full Stop and a New Line. This must be the Year of Publication: "1983".

The next line again contains the Pagination: "V, 383 s.". This can also easily be described to the computer, using the techniques outlined above.

Then comes a Note: "Disputats, Bonn 1982".

And at the end some Noise: "XYZ" which can be discarded.

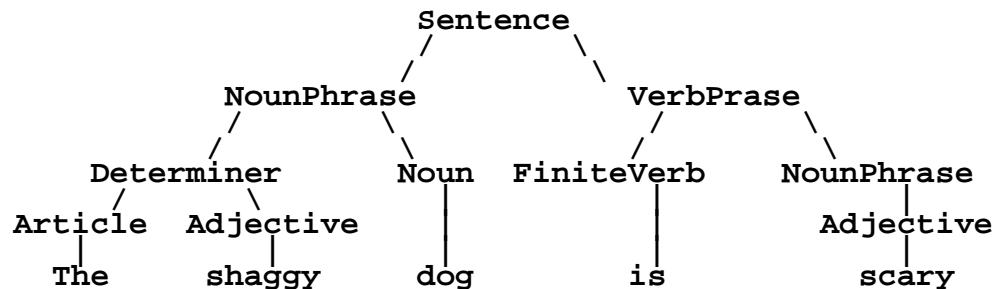
Since this card was produced according to a modified version of the ISBD rules the punctuation is an important clue to the bibliographical structure. In general spaces and empty lines, punctuation marks and the special vocabulary of bibliographies, as well as lists of names, publication places, publishers etc. will be used in the automatic structuring (formatting). This is why these features will have to figure prominently in the formal analysis of cards.

In the example below the structure of the card is made explicit using SGML styled tags. The tags are in angled brackets: <Tag>. The end of the scope of a tag is marked using the same tag with an added slash at the beginning: <Tag> - </Tag>.

```
<Record>
<LocMark>disp. 84-85</LocMark>
<AuthorHeading><LastName>Meuser</LastName><FirstName>
Michael</FirstName></AuthorHeading>
<MainTitle>Stärkung von Handlungskompetenz zum
Verhältnis von verstehender Soziologie und
Sozialpädagogik</MainTitle>
<AuthorshipStatement><FirstName>Michael</FirstName>
<LastName>Meuser</LastName></AuthorshipStatement>
<Imprint><PlaceOfPublcation>Bonn</PlaceOfPublication>
<YearOfPublication>1983</YearOfPublication></Imprint>
<PhysDescript><Pageination>V, 363 s.</Pageination>
</PhysDescript>
<Note>Disputats, Bonn 1982</Note>
</Record>
```

The tags bring out the hierarchy of the bibliographical elements.

The catalogue record can also be described in the same kind of tree-structure used in analysing sentences in natural and formal languages:



This kind of structure may be also be described using "rewriting" rules or "production" rules:

```
Sentence := NounPhrase VerbPhrase
NounPhrase := Determiner Noun
Determiner := Article Adjective
VerbPhrase := FiniteVerb NounPhrase
Article := "the" | "a" | "an" | ...
Adjective := "shaggy" | "big" | "dirty" | "scary" |
...
Noun := "dog" | "cat" | "horse" | "human" | ....
FiniteVerb := "is" | "was" | "has" | "did" | ...
```

The card example could be described in this style using the following rules:

```
Card := LocMark Heading MainArea Notes
Heading := AuthorHeading | SubjectHeading |
TitleHeading
AuthorHeading := LastName Comma FirstName
MainArea := Title AuthorshipStatement Imprint
PhysDescript
Title := MainTitle [Colon Subtitle]
```

```

[Equal ParallelTitle]
AuthorshipStatement := Slash FirstName LastName
Imprint := PlaceOfPublication [Publisher]
                YearOfPublication
PhysDescript := Pagination Illustration Size
LocMark := "disp. 84-85" | ...
LastName := "Meuser" | "Smith" | ...
FirstName := "Michael" | "William" | "Johannes" | ...
MainTitle := CapitalLetterWord {Word}
SubTitle := CapitalLetterWord {Word}
ParallelTitle := CapitalLetterWord {Word}
UniformTitle := CapitalLetterWord {Word}
PlaceOfPublication := "Bonn" | "Berlin" | "London" |
                ...
YearOfPublication := Digit Digit Digit Digit
Word := CapitalLetter{SmallLetter}
Word := SmallLetter{SmallLetter}
Word := CapitalLetter{CapitalLetter}
CapitalLetterWord := CapitalLetter{SmallLetter}
CapitalLetter := "A" | "B" | "C" | "D" | ...
SmallLetter := "a" | "b" | "c" | "d" | ...
Digit := "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
        "8" | "9"
Colon := ":"
Equal := "="
Comma := ","
Dot := "."
Slash := "/"

```

If a group of cards can be described in this way any skilled programmer will be able to write a program - a so-called "parser" - which can handle the analysis automatically. Tools - the so-called "parser generators" or less aptly "compiler compilers" - exist to help in the construction of such programs. A well known example is the set of programs called LEX and YACC, which were originally developed in the UNIX environment but are also available for the DOS/Windows environment.

The real problem is not to write a parser fitting a specific set of cataloguing conventions, but to create a suite of programs that will be able to handle the variations found within one catalogue as well as the catalogues of new libraries, without having to write a totally new program every time.

The solution selected in the FACIT Prototype is to produce a very general "parser". This program will read a formal description of the cataloguing rules applied in the catalogue to be converted. The description also tells the program which bibliographic elements (fields) have to be included in the internal representation of the formatted record. The program then reads the input (the source file) and transfers information (strings of characters) from the source to the specified field when the strings conform to the structure associated with each field.

The formal description is provided by the user and formulated according to a set of fairly simple rules: the FACIT Description Language. This is presented in detail in Appendix 1.

3. Catalogue Description

The necessary formal analysis will have to be based on an in-depth description of the catalogue. This description also serves the purpose of evaluating

whether it actually possible or cost-effective to convert the catalogue using scanning and OCR.

As said above it would be quite easy to construct an automatic formatting system if all card catalogues were homogenous, and represented one standardized and clear set of rules for bibliographic description. But in real life this is not the case for the following reasons:

- The card catalogue will usually have been produced over a long period of time, in which the cataloguing rules applied have changed.
- The card catalogue will in many cases contain cards produced by different cataloguing agencies or bibliographic agencies each applying their own rules and variations of shared rules.
- The individual cards will have been produced by different individuals and are affected by subjective, sometimes idiosyncratic interpretations of the rules, material mistakes and unavoidable errors.
- Many catalogues to be converted contain many different types of cards: Main Entry cards with headings representing authors or titles. Added entries by secondary authors, title, subjects etc. Entries covering more than one card. Analytical entries on articles etc. The system will have to be able to differentiate between these types and handle the information according to the type.
- The errors produced in scanning and OCR may blur the information used in structuring the record.

There are important differences between a card catalogue and an electronic catalogue which sometimes makes it difficult to establish a one-to-one-relationship between a card and a machinereadable record. In some - actually many - card catalogues information from several cards will have to be collected to produce one final record, since a new card had to be produced for each added entry to a particular title.

In the electronic catalogue only one record is needed for each title. Extra access points are provided through the search registers and through supplementary information added to the record itself.

The card catalogue is searched in the alphabetical or systematic sequence based on the heading of the card, while the electronic catalogue is searched via character strings or combinations of character strings extracted from the records and integrated into the search registers.

A low level of redundancy is aimed at in the electronic catalogue, while a high level of redundancy is needed in the card catalogue if many access points are to be provided to the user. We do not want this high level of redundancy in the retroconverted catalogue if we can help it, and on the other hand we do not want to lose any information in the original catalogue explicitly or implicitly provided through the organization of the cards.

Of course the simplest and most efficient would be to find one sequence of the card catalogue in which there is only one entry per title and in which the card contains the fullest possible bibliographic and administrative information about that title in the library.

A shelf list may come near to this ideal if it is produced using copies of main entry cards including tracings. A shelf list would also be less used, that is less worn and smudged, than the main alphabetical or systematic catalogue. Unfortunately the shelf list in many libraries do not fulfill the criteria. It may be handwritten or in a bound volume. It may not include information only found in added entries. Etc.

Since sorting the cards prior to scanning and OCR means using costly human labour, the conversion package will have to take care of the duplication of information and the resulting redundancy in the final result.

This could also be a quite simple matter, if all added entry cards were copies of a main entry card with just different headings to give extra access points. Then the program could discard all added entry cards using some formal criteria for selecting the records to be kept and the records to be discarded.

If certain cards are to be discarded the computer will have to be able to identify some distinct feature of those cards, i.e. a specific structure or a distinctive element such as the character string: "See", "See also", "In", the absence of a Location Mark in the top right corner etc.

If no cards can be discarded at an early stage all the different structures will have to be recognized and acted upon by the computer. In other words they will have to be described in all formal details to the program, using elements that the computer is able to "understand": letters, digits, punctuation marks, special characters, recurrent character sequences with a specific function, spaces, line breaks etc.

A card saying

Virgilio, Publio Maro see Vergilius, Publios Maro

will have to be recognized as an entry concerning preferred forms of personal names, a reference entry. This will then be converted into a record for the Authority File.

An added subject entry may contain information about the subject of the title in question not contained in the main entry card. A machine-readable record will have to be produced on the basis of the added entry, then the added entry and the main entry will have to be matched using a search for duplicate records, and the subject information transferred to the record produced using the main entry. After this the added entry record may be discarded.

An added entry using a secondary author as heading may contain a form of that author's name not found in the main entry. The inverted form used in the heading could be used to as a supplementary key to the subdivision of that name into i.e. last name and

first name(s). Again a procedure involving searching for duplicate records will have to be used before the rest of the information from the card may be discarded.

But it is also important to acknowledge the fact that even with automatic procedures questions of economics arise. Using the computer to analyse and handle special cases automatically involves a considerable effort in specification, programming and debugging. If a special procedure is only needed very seldom in a given catalogue, it will often be less costly to have the computer characterize the card as a deviation not dealt with, and then present it to a human operator to do whatever is necessary.

In the analysis of the catalogue it will be important to distinguish between main cases occurring a lot of times, and special cases occurring very seldom.

A good knowledge of the catalogue to be converted is necessary to do this, established through a close analysis of the catalogue itself, if possible supplemented with information about the cataloguing rules used in producing the catalogue.

This analysis will aim at:

- Identifying the different types of cards.
- Describing the structure(s) of each type of card, focusing on the physical and logical lay-out of the card, and the sequence of bibliographic elements (or fields) in the cards.
- Describing the structure(s) of each bibliographic element, focusing on constituent elements, character strings unique to each element (i.e. a special bibliographic vocabulary such as "by", "von", "af", "edited by", "herausgeben von" etc. to signal a responsibility statement), delimiters etc.
- Describing the set of characters: letters, including diacritics, digits, punctuation marks and special characters found in the catalogue. The focus will again be on characters helpful to identify elements and borders between elements.
- Describing recurrent word or expressions useful for the identification of structure.

In most card catalogues the heading is an important bibliographic element which is also problem ridden. It will often be easy for the machine to identify the heading, but difficult to identify the information contained in the heading and its relations to the rest of the card.

The function of the heading in the card catalogue is to order the cards alphabetically or systematically: The (last) name of an author or person as subject, the first word of the title (excluding articles), the first significant word in the title, the first word of a "uniform" title, the first word of a geographical term, a subject heading, a keyword or a class mark.

This information will often have to be translated or transferred into some other bibliographic form which is useful in the electronic catalogue.

4. The Use of Existing Cataloguing Rules

Knowledge of the cataloguing rules used to produce the catalogue is important. In a sense the analysis mentioned above is a "reconstruction" of the cataloguing rules. But this does not necessarily mean knowledge of the actual, codified rules which the cataloguer had at hand when cataloguing the works.

If such codified rules exist it will be very helpful to the person doing the analysis. But even without them it will be necessary to figure out the rules followed by the cataloguer as evidenced by the actual catalogue. And the cards may contain rules or principles obviously honored by the cataloguer without being covered by the codified rules, as well as systematic deviations from the prescriptions.

The analysis should give an outline of any codified cataloguing rules focused on the following aspects:

- What are the origin of the rules: The adaptation of national or international rules or a system developed by the library itself?
- What types of entries (cards) are covered?
- What do the rules say about the lay-out of the cards?
- What are the prescriptions for the heading?
- What do the rules say about author's names and the form of the name in the heading and any other statements of authorship? Do the rules recognize corporate authors as well as personal authors?
- What bibliographic elements are recognized by the rules and what is the sequence and the delimiters (punctuation marks etc.) prescribed?
- Do the rules prescribe any information not of a specific bibliographic nature, like tracings and notes. What about accessions numbers, statements of holdings and other administrative information?
- What do the rules say about multivolume works and analytical entries?

The codified rules may be used to understand features in the cards that are not immediately comprehensible, and to find out which features seem to represent the norm and which the deviations or variations. The codified rules may also be used to check whether all important instances have been met in the sample on which the analysis is based. Thus a brief outline of the cataloguing rules used by the library over the ages could act as reference handbook for the analysis of the card catalogue. But in the last instance the corpus of the cards themselves is the authority, not the codified rules.

IV. Methodological Recommendations

The main sources for the analysis of a catalogue to be converted are the cards themselves, any written (codified) rules used by the cataloguing department of the library, and the personal expertise and experience of the staff of the library.

Normally it will not be possible to base the analysis on an inspection of all cards. Some kind of sampling procedure will have to be employed. The important thing here is to make sure that highly frequent types of cards will be represented, while variations with low frequency are of less concern.

A good starting point could be to draw a small sample, 200 - 500 cards for a small library, 2.000 - 3.000 cards for a large library, at a random point in the catalogue sequence. From this sample, supplemented with written rules and the know-how of an experienced cataloguer, the main types of cards should be established and described preliminarily.

Then a larger random sample is drawn, 1.000 - 2.000 cards for a small library, about 10.000 cards for a large library, in order to verify and supplement the first description. This sample is worked through in all detail in order to catch all types and variations present in the sample. The description is checked against the written rules and the know-how of staff to see if any important cases are missing in the sample. When cases are added from other sources than the cards themselves the catalogue has to be scanned more or less at random to see if these cases actually turn up.

To guarantee an exhaustive and detailed description of the catalogue all the types of cards have to be carefully documented using preferably a photocopy of an example of the type in question. The different cases will have to be systematically registered and categorised according to the specific parameters present.

On the basis of the sample the different proportions of the types will also have to be established.

All information will have to be established using the actual cards. It is important to distinguish between what should be in the catalogue and what is actually there. In the end the analysis will have to be verified through the confrontation of the application with the cards. Errors and misunderstandings will of course be unavoidable, but a lot of costly debugging effort may be saved if the main principles of the catalogue are grasped correctly at this stage.

This will of course have to be an iterative process, since it will not be possible from the very beginning to know exactly what will be found in the catalogue. The procedure outlined above shows how this may be planned from the start.

A more detailed list of the things to look for are included in Section IV. Presentation of Results. Repetition of this should not be necessary at this point. But the following steps are well worth stressing:

1 Establish the types of cards that you will have to describe, sorting them by functions, rules applied and by producers (if this is a significant information for the catalogue in question).

The types of cards to look for are: Main Entry Cards; Added Entry Cards with secondary authors, titles or subject terms as access points; Follow-On Cards; Reference Cards for authors names or subject terms.

If the rules affect the way the different types of cards are structured, this will have to be noted too. For instance in the Italian bibliographic tradition the sequence of the physical description used to be: <size> cm. <pages> p. [or pp.]. After introduction of the RICA rules in 1979 the sequence is: <pages> pp. <size> cm. The forms are equivalent and both useful for the identification of the Physical Description Area.

Another example: In the oldest rules the Series Area was delimited with angled brackets: < and >. Later this was changed into parentheses (curved brackets): (and). Both cases correspond to codified rules and may be used to identify the Series Area.

If the library have acquired catalogue cards from different cataloguing agencies the rule applied by these will of course affect the structures found in the cards. If the different sources of cards can be identified by a simple procedure, this will be very helpful for the application.

As an example of this: In the catalogue of Biblioteca Nazionale V.E.III, Napoli, cards from 4 sources are to be found, all marked with the date and "BNN", "BNCF", "CAT.UN." or "BNI" at the bottom of the card. This may be captured and used to determine the best strategy for structuring the information in the card.

2 List the standard sequence for each type of card.

The bibliographic elements to be found will vary from library to library and from period to period within the library. Also the sequence and the lay-out of the card may vary.

The use of top and bottom, as well as corners and margins should be noted. It is very important to note if certain information in these area have to be read as columns with the character strings belonging together distributed over more than one line.

Elements found in most cards, especially Main Entry Cards, are:

Location Mark (Ital: "segnatura", Da: "signatur")

Heading (representing authors, beginning of title or subject terms)

Main area including Title area, Area of authorship statement or responsibility and Edition area.

Imprint area including Place of Publication, Publisher and Year (Date) of Publication.

Physical Description area including Pagination, Illustrations, Size and Additional Material.

Notes area including multivolume statement.

Tracings area

Other information: Holdings statements; accessions number; class marks; subject terms; extra locations marks.

3 Stress the key features useful to recognize each of the areas in the card, using features that the computer can "understand".

To do this look for the significance of features like: punctuation marks; slashes and dashes; brackets; other special characters; specific words and abbreviations; use of capital letters; use of spaced characters; use of bold characters and underlined characters; use of coloured characters (normally red in typewritten cards); use of line breaks (New Line + Carriage Return); use of empty lines; use of shorter or longer indentations; length of fields; position in relation to other fields. Comment: Not all OCR packages are able to identify and retain information about bold characters and underlined characters. Some scanners are blind to certain colours, i.e. blue, green or red.

Note also elements that may occur in different contexts with different functions or meanings. For instance a comma, ",", in the heading may be used to separate the Last Name and First Names of an author, be part of the beginning of a title, and separate the Main Term in a Subject Heading from any Specifying Terms. A "P.P." (= pope) occurring in the Heading will help to identify this as an authorship statement, while the sequence "@Number p. @Number cm." or "cm. @Number pp. @Number" will help to identify the Physical Description area.

Sometimes more than one area will have to be taken into account. The Heading may for instance in some cards be identified as a string of characters beginning with a Capital Letter or a Digit following a string of characters that have been identified as a Location Mark by dictionary look-up, left justified on the card, and followed by an indented line.

One way to keep track of this kind of information could be to set up a free text database with samples and explanations of the structures found in the catalogue. Searching across the samples for specific characters could help identify and highlight such features in the cards.

4 Point out the identifiers for each bibliographic element, trying to answer the following questions:

- Where in the cards are the standard bibliographic elements: Authorship statements, Titles, Edition statements, Place of Publication, Publisher, Year of Publication etc. found?
- How are they represented in the cards? Are there specific typographical conventions? A standard sequence of constituent elements (i.e. First Names, Last Names, Call Names, Family Names, Titles etc.) ?
- Is the same bibliographic element present in more than one area? What differences and equalities are there in the representation?

The same features as listed under 3. above will be useful here.

5 Take special care to identify possible sources of ambiguity in the Heading of the cards.

The bibliographic element represented in the Heading will in many cases be difficult to identify for the computer, so special care has to be taken in the description of this area, and its relations to the rest of the card.

Look for features that will uniquely identify the bibliographic elements as well as ambiguous features. In many cases it will be possible to provide the machine with clues that enables it to make a shrewd guess, even if the evidence is inconclusive, in other cases the question will have to be resolved by a human operator.

Examples of features could be:

- The occurrence of articles like "the", "a" and "an" (in English) in brackets or parentheses may indicate the beginning of a title.
- An indentation of the line following the Heading followed by a line or a dash, "-", may also indicate the beginning of a title.
- A full stop after the Heading will normally indicate that the Heading is not the beginning of a title.
- Words and abbreviations usually found in names may indicate an authorship statement.
- A comma after the first word of the heading may indicate an inverted personal name, that is an authorship statement.

V. Presentation of Results

1. General Description of the Catalogue

Describe the catalogue or the selected catalogue sequence in the following terms:

Type of catalogue: Alphabetic author-title-catalogue, alphabetic subject catalogue, systematic (classified) catalogue, shelf list etc.

Contents of the catalogue: The works (specific collection) covered by the catalogue. Works giving rise to different types of bibliographic descriptions are of special importance: Monographs, periodicals, microforms, prints, sound recordings, videograms, manuscripts etc.

Physical description of the catalogue:

Size of cards: Height by length in centimeters. Note variations in size.

Material used: Cardboard/paper ...

Estimated number of cards:

Other relevant information: Backwritten cards (estimated proportion). Smudged and worn cards.

Quality and readability of text: Proportion of handwritten, stencilled, typewritten or printed cards; number of different typewriters used; number of different fonts used; do they occur in the same card? Note handwritten additions and other additions like rubber stamps.

Types of card in the catalogue: If more than one, note the estimated proportion and their function:

Main Entry Cards; Added Entry Cards; Analytical Cards; Multivolume cards; Follow-on Cards.

General Layout of cards: The use of corners, margins, headings, main area, supplementary areas etc.

Cataloguing Rules used in the catalogue: Listing of the cataloguing rules applied in the catalogue/catalogue sequences, with a brief outline of the principles.

2. Inventory of Elements

Make an inventory of the different elements found in the cards, based on a large sample of cards.

a. Character Set(s)

Characters actually occurring in the catalogue are noted, not just general characterisations like Latin

or Greek alphabet. A catalogue produced with the Latin alphabet will in all probability contain only a specific subset of all possible Latin letters, including also combinations with diacritical marks.

To characterize letters outside the basic set of Latin letters found in the ASCII or the ANSI character set (A-Z and a-z), the verbal descriptions used in ISO/IEC 10646-1 will secure unique identification of the letters: 'Latin small letter a with ring', 'Latin small letter a with circumflex', 'Greek letter alpha', etc. A list of selected Latin and Greek characters with standard description and 16-bit codes is found in Appendix 2 and 4 of FACIT Technical Report no. 1 Optical Character Recognition for Retroconversion of Catalogue Cards: Hardware, Software and Character Representation.

If transcriptions or substitutions are used, i.e. because not all needed characters can be produced with the typewriter used, this is noted so that the right character may be inserted during the conversion process. (Example: ae used for æ)

The list should be organized as follows:

Letters: Include all combinations of basic letters with diacritical marks etc.

Digits (numbers): Note if l (Latin small letter l) is used for 1 (Digit one) and o (Latin small letter o) or O (Latin capital letter O) is used for 0 (Digit zero).

Punctuation marks: Mark any significant the use of Space (i.e. Double Space at certain points).

Other characters

Note if certain subsets of characters only occur in certain types of cards or with certain fonts. This could be used to pin-point a type of card needing special treatment, but is could also be a source of errors in the OCR process.

Handwritten characters or diacritical marks may have been added when not available on the typewriter; this will in all probability be a source of unsystematic errors in the OCR process.

More information about character sets in the context of OCR is found in the FACIT Technical Report No 1 as mentioned above:

b. Bibliographic Elements

Note all bibliographic elements occurring in the cards, using ISBD (or UNIMARC) as a point of reference. For elements not occurring in these standards give a short definition or description.

Each element is described concisely using the following template:

Name of element

Abbreviation of element (as used in the structural analysis; cf. next section)

Definition (as needed)

Position: Is described relative to other elements (before/ after) or in terms of the general layout of the card (corners, margins, main area, supplementary areas etc). If the element described may occur in different positions, this is noted. Any links between positions and types of cards etc. are noted.

Delimiters: Sequence(s) of characters signalling change from one element to the next. List all possible variations. Note if the delimiters may be discarded when information is represented in the target (like the '/' marking the boundary between Title and Authorship Statement in the ISDB rules).

Structure of the element: Note any sub-elements occurring in the element and the sequence of these sub-elements.

Repertoire of characters occurring in the element: Note any special meaning of characters (or types of characters) occurring in the element.

Special vocabulary incl. abbreviations occurring in the element: Note vocabulary etc. that may be useful for identifying the element or the boundaries between elements (cf Section c. below).

Notes: Any other comments deemed pertinent.

c. Bibliographic Vocabulary

Alphabetic list of all bibliographic terms and abbreviations occurring in the cards with indication of the area in which it is used.

3. Structural Description of Cards (Cataloguing Rules Formalized)

This section is a formal description of the catalogue cards to be used as an input to the programming specifications. This may be done in several ways, but they will all be variations on the basic techniques for writing formal grammars developed in computational linguistics and computer science. An outline of this is found in Section III above.

If a program or application for automatic formatting has already been selected or has been developed for the conversion project, the description should of course conform to rules stipulated for the program.

For the FACIT Prototype a special Description Language as been developed, based on a set of simple rules. The FACIT Description Language is presented in detail in Appendix 1, and an example of its use is given in Appendix 2.

This and most other formal descriptions are based on so-called "rewriting rules" or "productions rule" which give a top-down analysis of the entity - in this case a catalogue card - to be processed. Using these rules the computer program will analyse the input - e.g. the source file - to decide whether it is constructed according to the rules. If the answer is yes, the program will initiate appropriate actions,

such as writing information into the fields of a database or tagging the information. If the answer is no, the input will be marked as faulty and the user given a message that says so.

A "rewriting rule" or "production rule" links one term (and one term only) on the left hand side with one or more terms on the right hand side. The link may be indicated in several ways. The most common are arrows, e.g. '->' or '=>' and equal signs or equal signs following one or two colons: '=', ':=' or '::='. The choice of notation has no real significance.

<Term1> := <Term2> <Term3>

This means that the structural unit represented by <Term1> can consist of the sequence of structural units represented by <Term2> and <Term3>. One may also say the <Term1> can be substituted by ("rewritten" as) the sequence <Term2> followed by <Term3>. The name "production rules" refers to the notion that the entity signified by <Term1> can be "produced" by substituting the sequence <Term2> <Term3>.

To one term like <Term1> several alternative structures may be associated. In this case a vertical line, "|", is used to separate the alternatives:

<Term1> := <Term2><Term3> | <Term4><Term5><Term6>

Alternative structures could also be represented by several rules with the same left hand side:

**<Term1> := <Term2><Term3>
<Term1> := <Term4><Term5><Term6>**

Some systems like the FACIT Description Language only allows a term to be defined once, so only the form with '|' can be used.

One of the terms in the sequence may be optional, that is sometimes it will be present in the sequence, sometimes not. This is represented using square brackets: "[" and "]":

<Term1> := <Term2> [<Term3>] <Term4>

In this case the rule could also be represented using the notation for alternatives:

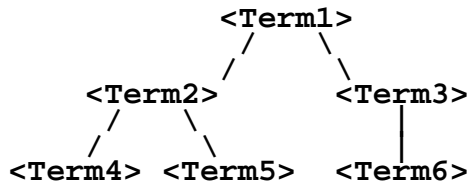
<Term1> := <Term2><Term3><Term4> | <Term2><Term4>

If an optional element may be repeated any number of times curly brackets, '{' and '}' are used instead of square brackets.

<Term1> := <Term2> {<Term3>}

Some systems use a plus-sign, "+", added after the right square bracket: [<Term3>]+

The terms occurring on the left hand side of a rewriting rule have to be defined using other rewriting rules. In this way a hierarchical as well as sequential structure is defined, like in a natural language sentence:



At one point the chain of definition will have to terminate with entities that do not have to be defined themselves. In the case of catalogue cards these will be strings composed of the character set used in the cards.

These are called the terminal terms, while the terms representing the syntactical structure are called non-terminal terms. A terminal term can be of two types: tokens and literals.

Tokens are terms that stand for one or more literals. Examples are 'Comma' representing the comma character, ','; 'Number' representing any number; 'Year' representing a number with four digits in the range 1000 - 1999. 'Word' representing any sequence of letters consisting of first a capital letter, then only small letters, or all capital letters or all small letters. In some systems tokens are marked in a special way to differentiate them from other terms, like: @Comma, @Number, etc.

Literals are actual characters or sequences of characters. They are normally represented as the character sequence itself surrounded by double (or single) quotation marks: "London", "1993", "herausgegeben von".

If a non-terminal term can be substituted by one out of a closed set of literals this is represented by a list of literals on the right hand side separated by a vertical bar, '|':

<Term1> := "by" | "von" | "af" ...

The three dots "... " belong to the metalanguage and means that not all elements of the set have been specified.

The description language will also need some notation to indicate actions that have to be initiated when a certain structure is found in the input.

For the FACIT Prototype it was found that only one such action was needed as part of the Description Language: This action is to establish a named field in the internal record format when a specified structure is recognized by the parser, and to write the string conforming with the structure to that field.

The technique used is:

<FieldName> : <Term1> = <Term2> <Term3> ...

All strings conforming to the structure of <Term1> is written to the field named <FieldName> in the internal format.

Other systems will include their own way of specifying actions.

A general parsing program does not have to know anything about catalogue cards. The user will provide that knowledge through the writing of suitable rules. And the terms used in such rules can be completely arbitrary. From the point of view of the computer the only "meaning" attached to the terms have to do with sequences and groupings of characters found in the input stream.

Even if any name can be used to designate a term, for the benefit of the human user they should be constructed in such a way that they are mnemotechnic as well as not too long.

The convention used in the presentation of the FACIT Description Language in Appendix 1 and 2 is that all non-terminal terms begin with a capital letter and the rest are small letters, capital letters and digits. If the term is constructed as an abbreviation i.e. of a bibliographic term, the beginning of each abbreviated word is marked with a capital letter, the rest are small letters. But this is just a matter of convenience. Examples: 'AuthorField', 'Imprint', 'PlaceOfPublication'.

Other systems may allow the use of the Underscore character, '_' in the construction of term names: 'Place_of_publication'.

The names of the tokens should also be chosen to reflect what is represented: 'Digit', 'Colon', 'WhiteSpace' (= Spaces and Tabs), 'Newline'

Most systems will have the possibility to insert notes or comments, telling the reader more about what is intended. Such notes will be indicated by a special string of characters. like '//', ';' or '/* ... /*'. It is good practice to comment freely, since it helps even the author of the description to understand what is meant.

Often several attempts have to be made before the description is right. There may be several ways of reaching the same goal, and some of them may be more simple or more effective than others. A good way to test the description is to have the computer program validate it against some fairly simple test data representing important cases. (This of course presupposes that you already have a working program.)

The description starts from the top, by identifying elements that may easily be identified by the computer, and then workin all the way down to the basic elements, the individual characters and strings of character met in the input stream:

The following example shows an extract from the beginning of such a description.

```

Card := LocMark AuHeading MainSection [Suppl] |
        LocMark TitHeading MainSection [Suppl] |
        SecAuHeading "vedi" AuHeading
        MainSection [Suppl] LM
    ...

    // A card (the entity as a whole) can be struc-
    // tured in at least three different ways (the
    // three dots show that the list is actually
    // longer).

LocMark := UDC Alpha
    // The Location Mark consists of a UDC notation
    // and an alphabetical part (beginning of the
    // surname of the author or the beginning of the
    // title.

...

AuHeading := LName Comma FName
    ...
    // Heading with the name of the primary author.
    // Other variations of the name form are not
    // specified.

TitHeading := CapWord [Word] ...
    // Heading with the beginning of the title.

SecAuHeading := LName Comma FName
    ...
    // Heading with name of secondary Author. This
    // will be followed by the string 'vedi' and then
    // the name of the primary author. Otherwise the
    // structure is like any personal name.

MainSection := Title Author Imprint PhysDescr
    ...
    // The main body of the bibliographical
    // description.

Imprint := PlPubl Publ YearPubl
    // The Imprint section consisting of Place of
    // Publication, Publisher and Year of Publi-
    // cation.

PhysDescr:= Size Pag PhysNote
    // Physical Description consisting of Size of
    // Publication, Number of Pages, and other
    // information about the physical form and
    // appearance.

Suppl := Notes Tracing
    // Supplementary information, consisting of Notes
    // and Information about heading used in other
    // cards representing the same publication
    // ("tracings").

```

No standard can be established for the names of terms used in such a structural analysis, since conditions differ from catalogue to catalogue and even within the same catalogue. A solution will have to be found according to the elements actually in the catalogue,

and the demands of writing effective as well as understandable rules for the given system.

The following list includes often needed elements, and may be taken as a point of departure and inspiration:

Card: The whole card = one bibliographic record. (In the FACIT Description Language the top entity has to be called 'Main'.)

LocMark: Location Mark (In many European languages (apart from English) as word like 'signature' is used.)

UDC: Universal Decimal Classification notation (Class Mark)

DDC: Dewey Decimal Classification notation (Class Mark)

Heading: General term for a heading, not indicating content.

AuHeading: Authorship Heading

SecAuHeading: Secondary Authorship Heading

TitHeading: Title heading (normally first part of title)

MainSection: Main body of the card

MainTitle: Main Title

SubTitle: Subtitle

ParTitle: Parallel title

Author: Authorship Statement in the main body of the card (includes primary and secondary authorship)

Lname: Last name

Fname: First name

PersTitle: Personal title used to specify an author (e.g. 'king', 'pope', 'prof.' 'dr.')

Imprint: Imprint: Place of Publication; Publisher; Year of Publication

PlPubl: Place of Publication

Publ: Publisher

YearPubl: Year [and date] of publication

PhysDescr: Physical Description of publication

Size: Size of publication

Pag: Pagination, number of pages.

PhysNotes: Notes to the physical description

Suppl: Supplementary Area

Notes: Notes

Tracing: Tracings, that is indications of headings used in other cards representing the same publication.

Year: String of digits representing a year

RomanYear: String of characters representing a year as a Roman number (e.g. 'MCMLXXXV').

CapWord: Word with a capital letter as the first letter

Word: Any word with all small letters, all capital letters or first letter capital, the rest small letters.

Alpha: String of letters

Number: String of Digits

Dot: '.'

Comma: ','

Colon: ':'

Semicolon: ';'

Slash: '/'

In many cases it will not be possible to write one structural description covering all the types of cards found in one catalogue sequence (a sequence of cards that have to process in one continuous run). In this case alternative descriptions will have to be provided.

How to handle such alternative descriptions will depend on the system used. The FACIT prototype includes the facility to select parsers on the basis of information present in the individual cards, such as the presence of a specific string of characters. Cards made according to the ISBD rules could for instance be identified by the occurrence of typical ISBD punctuation like: '/' '.' '-' etc.

The structural descriptions provided to the formatting programs (the "parsers") should of course be as comprehensive as possible. But it may in fact be impossible or very expensive in terms of staff time to create structural description that will cover all possible cases. The best strategy is to cover the types of cards with the highest frequency first, and then include more and more cases, again on the basis of frequency of occurrence.

Some of the cards not accepted by the parser are actually the results of errors in cataloguing or in OCR conversion. They may be corrected and then accepted. Others represent special cases that will have to be converted manually. A reasonable aim is to cut down the number of "correct" cards (cards produced according to the cataloguing rules) not accepted by the formatting program to 5% of the total number of cards or less.

VI. References

- Aho, Kernighan & Weinberger, 1988
Aho, Alfred V., Brian W. Kernighan & Peter J. Weinberger: The AWK Programming Language. Addison-Wesley Publishing Company, New York ... 1988. 210 p.
- Allen, 1987
Allen, James: Natural Language Understanding. The Benjamin/Cummings Publishing Company Inc. Menlo Park, Calif. 1987. 574 p.
- Avram, 1972
Avram, Henriette D.: RECON Pilot Project. Library of Congress, Washington D.C. 1972. 49 p.
- Beaumont & Cox, 1989
Beaumont, Jane & Joseph P. Cox: Retrospective Conversion. A Practical Guide for Libraries. Meckler, Westport/London. 1989. 198 p.
- Bennet 1990
Bennett, J.P.: Introduction to Compiling Technoques. A First Course using ANSI C, LEX and YACC. McGraw Hill Book Company, London ... 1990. 242 p.
- Bokos, 1993
Bokos, George: "UNIMARC, CDS/ISIS and conversion of records in the National Library of Greece." In: Program vol.27 (2), April 1993. Pp. 135 - 148.
- Boserup & Holtse, 1992
Boserup, Ivan & Lisbet Holtse: "Automatic Conversion at The Royal Library, Copenhagen. A Progress Report." Paper given at The Second International Conference on Retrospective Cataloguing, München, 28.-29.January 1992.
- CEC, DG XIII B, 1990
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. 1990. [var.pag.] Printed as draft.
- Chapman, 1987
Chapman, N.P.: LR Parsing. Theory and Practice. Cambridge University Press, Cambridge ... 1987. 228 p.
- Council of Europe, 1989
Guidelines for Retroconversion Projects prepared by the LIBER Library Automation Group. Council of Europe, Council for Cultural Co-operation, Working Party on Retrospective Cataloguing. 14 July 1989 (revised). 13 p.
- Crawford & Lee, 1990
Crawford, R.G. & Susan Lee: "A prototype for fully automated entry of structured documents."

- In: The Canadian Journal of Information Science/Revue canadienne des sciences de l'information.
Vol. 15, No. 4. December 1990. Pp. 39 - 50.
- Deutsches Bibliotheksinstitut, 1993
Retrokonversion: Methoden. Verfahren. Kosten.
Edited by Kirsten Weber. (dbi-materialien, 128).
Deutsches Bibliotheksinstitut, Berlin. 1993. 411 p.
- Gough, 1992
Gough, John K.: Syntax Analysis and Software Tools. Addison Wesley Publishing Company, Reading, Mass./Wokingham (U.K.), 1992. 400 p.
- Harrison, 1985
Harrison, Martin: "Retrospective Conversion of Card Catalogues into Full MARC Format Using Sophisticated Computer-Controlled Visual Imaging Techniques." In: Program 19 (July 1989). p. 213-30.
- Harrod, 1984
Harrod's Librarians Glossary of terms used in librarianship, documentation and the book craft and Reference Book. Fifth Edition. Revised and Updated by Ray Prytherch. Advisory Editor Leonard Montague Harrod. Gower, Aldershot, Hants. 1984. 861.
- Hein, 1986
Hein, Morten: "Optical Scanning for Retrospective Conversion of Information." In: The Electronic Library, December 1986. Vol.4, No.6. P. 328 - 331.
- IFLA, 1977
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.
- IFLA, 1987
ISBD (M). Revised Edition. IFLA Committee on Cataloguing. International Federation of Library Associations, London. 62 p.
- IFLA, 1987
UNIMARC Manual. Edited by Brian P. Holt with the assistance of Sally H. MacCallum & A.B. Long. IFLA Universal Bibliographic Control and International MARC Programme/British Library Bibliographic Service, London. 1987. 482 p.
- IFLA, 1990
IFLA Journal 16 (1). Special issue devoted to an overview of projects and approaches to retrospective conversion, providing an international perspective.
- IFLA, 1991
ISBD (S). Revised Edition. Joint Working Group of the IFLA Committees on Cataloguing and Serial Publications. London 1991. 62 p.
- IFLA, 1991
UNIMARC/Authorities : Universal Format for Authorities. recommended by the IFLA Steering Group on a UNIMARC Format for Authorities.

- Approved by the Standing Committees of the IFLA sections on Cataloguing and Information Technology. K.G.Saur, München ... 1991. 80 p.
- ISO/DIS 10324
Information and Documentation - Holdings Statements - Summary Level. ISO/DIS 10324:1991. International Standards Organisation. 1991.
- Jain, 1991
 Jain, Raj: The Art of Computer Systems Performance Analysis. Techniques for Experimental Design, Measurement, Simulation and Modeling. John Wiley and Sons, Inc., New York. 1991. 685 p.
- Jennings, Newman & Wilkinson, 1982
 Jennings, Newman & Wilkinson: "Data capture by optical scanning of published material for database enhancement." In: Program 16 (1). January 1982. Pp. 17 - 27.
- Jensen, 1986
 Jensen, Hans Erik: Problemer i forbindelse med retrospektiv inddatering af kortkataloger og optisk læsning. [Problems of retrospective conversion of card catalogues and optical character recognition.] Statsbiblioteket, Århus. 1986. 52 p.
- Kristensen, 1990
 Kristensen, Jens Thyge: Konstruktion af indlæseprogrammer. [Construction of programs for input control.] Teknisk Forlag, København. 1990. 119 p.
- Mason & Brown, 1990
 Mason, Tony & Doug Brown: lex & yacc. O'Reilly & Associates Inc, Sebastopol, CA. 1990. 216 p.
- Schottlaender, 1992:
 Schottlaender, Brian: Retrospective conversion: history, approaches, considerations. Haworth Press, Binghamton, NY. 1992. 167 p. (Also: Cataloging and Classification Quarterly 14 (1992), 3/4.
- Smith & Merali, 1985
 Smith, John W.T. & Zinat Merali: 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.
- Süle, 1990
 Süle, Gisela: "Bibliographic Standards for Retrospective Conversion" In: IFLA Journal 16 (1). 1990. P. 58 - 63.
- Syré, 1987
 Syré, Ludger: Retrospektive Konversion. Theoretische und praktische Ansätze zur Überführung konventioneller Kataloge in Maschinenlesbare Form in den USA, Grossbritannien und der Bundesrepublik Deutschland. Deutsches Bibliotheksinstitut, Berlin. 1987. 231 p.

Character sets

ISO/IEC 10646-1:1993 (E)
Information technology - Universal Multiple-Octet
Coded Character Set (UCS) - Part 1: Architecture
and Basic Multilingual Plane.

Appendix 1

The FACIT Description Language

The following description is based on the system implemented in the FACIT Prototype by Jakob Darger of SYNERGI and documented in FACIT Technical Report no 4: The FACIT Prototype. Technical Documentation.. Anne Katrine Wille has kindly provided her expertise towards making the description consistent and complete.

The presentation of the FACIT Description Language starts with the basic building stones of structured texts, the Literals (single characters and strings of characters). Then comes the more complex expressions, linking the Literals together to form a hierarchy of textual elements, like the elements of a bibliographical description.

The first four sections are informal, introducing and exemplifying the elements of the Description Language together with any actions associated with the descriptions. Section 5 covers the same ground, but this time in a strictly formal way, using a BNF notation. Section 6 present two supplementary description techniques used in the FACIT prototype.

The actual descriptions of catalogue cards are created by the user according to the rules presented here. To illustrate how this may be done a fully worked out example is given in Appendix together with a small sample of cards. The resulting analysis is presented through a print-out made with the FFORM-program from the FACIT Prototype.

1. Literals

Literals are expressions that refer to one character or a string of characters, taken as that character or string of characters itself (taken "literally").

1.1 Single Characters

A printable character in the basic ASCII or ANSI character set (code 32 - 126) may be referred to by the character itself inclosed in double quotation marks:

"a" : the character 'a'

This covers the characters:

A a B b C c D d E e F f G g H h I i J j K k L l M m N
n O o P p Q q R r S s T t U u V v W w X x Y y Z z
(Codes 65-90 and 97-122)

0 1 2 3 4 5 6 7 8 9 (Codes 48-57)

" " : <Blank> or <Space> (Code 32)

! : Exclamation mark (code 33)

: Number sign (code 35)

% : Percent sign (code 37)

& : Ampersand (code 38)

' : Apostrophe or Single quotation mark (code 39)

(: Left parenthesis (code 40)

) : Right parenthesis (code 41)
 * : Asterisk (code 42)
 + : Plus sign (code 43)
 , : Comma (code 44)
 - : Minus sign or Hyphen (code 45)
 . : Full stop or Dot (code 46)
 / : Solidus or Slash (code 47)
 : : Colon (code 58)
 ; : Semicolon (code 59)
 < : Left angle (code 60)
 = : Equal sign (code 61)
 > : Right angle (code 62)
 ? : Question mark (code 63)
 @ : Commercial at (code 64)
 _ : Underscore (code 95)
 ` : Grave accent (code 96)
 { : Left curly bracket (code 123)
 | : Vertical bar (code 124)
 } : Right curly bracket (code 125)
 ~ : Tilde (code 126)

" " : the empty string (the NULL character, code 0).

The following characters have a special meaning (cf. below):

" : Double quotation mark (code 34)
 \$: Dollar sign (code 36)
 [: Left square bracket (code 91)
 \ : Back slash (code 92)
] : Right square bracket (code 93)
 ^ : Caret or Circumflex accent (code 94)

They can only be referred to using the Back Slash character as a prefix:

"\" : Double quotation mark
 "\\\$" : Dollar sign
 "\\[" : Left square bracket
 "\\\" : Back slash
 "\\]" : Right square bracket
 "\\^" : Caret or Circumflex accent

It is also possible to refer to characters using the code number in decimal or hexadecimal using one of the following expressions:

```

${<decimal>}
${&<hexadecimal>}

```

where <decimal> is substituted by an integer in the decimal system, and <hexadecimal> by an integer in the hexadecimal system. The '&' tells the system to interpret the number as hexadecimal.

When this type of expression is used to refer to a character it has to be enclosed in double quotation marks too:

"\${65}" : the character 'A'
 "\${&41}" : also 'A'

This method makes it possible to refer also to characters in the range 128 - 255 (decimal) in an 8-bit character set or to all characters in a 16-bit character set, such as UNICODE or the internal character set of the FACIT prototype, without using an often misleading graphic representation.

Characters with codes in the range 128 - 255 (decimal) may also be referred to using the character itself in double quotation marks, e.g. "ä" (Latin small letter a with dieresis). But the reference will actually be interpreted according to the code number of that character in the active display character set, such as ASCII (PC8, Code Page 431, 850 and 865): Code 132, ANSI: Code 228, ROMAN-8: Code 204 of UNICODE: Code 244 etc. If for instance a source file was produced using a different coding of characters in the range 128 - 255 than the one used in the description, the reference could be wrong or misleading.

In the internal character set, all values between 1 and 9999 (decimal) may be used. The user may arbitrarily assign any character to any value, but it is recommended that the code values of UNICODE (a subset of ISO 10646-1) is used for the basic and extended Latin, Greek and Cyrillic alphabets. The value Zero (0) is reserved for use as a NULL-character, or empty string (""). A survey of characters needed for retroconversion of catalogues in European languages (with 16-bit hexadecimal code) is included in the FACIT Technical Report No. 1: Optical Character Recognition for Retroconversion of Catalogue Cards: Hardware, Software and Character Representation., as Appendix 2 - 4.

A set of predefined expressions ("wild card" expressions) may be used to refer to one character or a group of characters that may occupy a specific position in a string of characters:

`{Any}` : Any character
`{Alpha}` : Any₁ letter (range 65 - 90 or 97 - 122 (decimal))¹
`{NoAlpha}` : Any character except a letter
`{Digit}` : Any digit (0 1 2 3 4 5 6 7 8 9)
`{NoDigit}` : Any character not a digit
`{AlNum}` : Any letter or digit (an "alphanumeric" char.)
`{NoAlNum}` : Any character except letters and digits
`{UpCase}` : Any upper case letter (range 65 - 90)
`{NoUpCase}` : Any character except upper case letters
`{LowCase}` : Any lower case letter (range 97 - 122)

¹) The restricted built-in range of codes representing letters is a consequence of the limited implementation of the internal character set in the present version of the FACIT Prototype. In future versions it is planned to supply a utility for providing information about the internal character set, among others which characters should be classified as letters as well as differentiating upper case and lower case for all letters.

`${NoLowerCase}` : Any character except lower case letters

`${Newline}` : A New Line character (Line Feed, code 10)

`${NoNewline}` : Any character except a New Line character.

`${WS}` : "White Space", that is Spaces (Blanks) or Tabs

`${NoWS}` : Any character except "White Space"

`${ESCAPE}` : Represents the card separator, normally '\$'. Equal to "\\$"

These wild card expressions (character tokens) may be extended by the user introducing other letters or other characters in addition to the above mentioned (cf. section 2, below).

Another way of referring to a group of characters that can substitute each other in one character position, is to specify a set of alternative characters using square brackets: '[...]':

"[abc]" : Any one of the characters 'a', 'b' or 'c'.

The character '^', Caret, functions as a negation sign:

"[^abc]" : Any character except 'a', 'b' or 'c'.

The character '-', Minus sign or Hyphen, indicates a range of characters:

"[a-z]" : Any one of the small Latin letters in the range of codes starting with 97 ('a') and ending with 122 ('z'). This is equivalent to "\${LowerCase}".

"[0-9]" : Any one of the digits (codes 48 to 57). This is equivalent to "\${Digit}".

A range is always interpreted according to the current coded representation of the characters. In ASCII (PC8, Code Page 850) "[A-Ã]" would mean all characters from code 65 ('A') to code 199 ('Ã') (which may not be the intention of the user).

It is possible to indicate two or more ranges at the same time:

"[A-Za-z]": Any one of the capital Latin letters or the small Latin letters; equivalent to "\${Alpha}".

"[0-9A-Za-z]" : Any Latin letter or any digit; equivalent to "\${AlNum}"

The character '\\', Back slash, is used to make sure that the following character is taken literally, making it possible to refer to the special characters '"', '\\', '-', '[' and ']' as Literals:

"[\\[\\]]" : one of the characters: '[' or ']'.

It is possible to use "Wild Card" expressions inside square brackets:

"[^\${Newline}]" : Equivalent to \${NoNewline}

"[\${32}]\${9}]" : Equivalent to \${WS}

1.2 Strings of characters

Any string of characters may be referred to by enclosing the string itself in double quotation marks. The string may include character tokens in the form '\${132}':

"string" : The literal string 'string'.

"\${115}\${116}\${114}\${105}\${110}\${103}" : The string
'string'

"str\${105}n\${103}" : The string 'string'.

Instead of referring to a specific character in the string, it is possible to refer to a group of characters that may occupy the same position in the string (alternatives), using "wild card" expressions (as defined above) or characters enclosed in '[']' (also defined above):

"[Vv][Ee][Dd][Ii]" is shorthand for the following group of strings: "VEDI", "Vedi", "vedi", but also "VEDi", "VeDI", "vEDI", "VeDi", "vEdI", "Vedi", and all other possible combinations.

2. Assigning literals to expressions (tokens)

Any literal (one character or a string of characters) may be assigned to an arbitrary identifier (a "token"), to represent or refer to that literal, using an assignment statement:

```
<TokenName> = <String> .
```

where <TokenName> is any string consisting of alphanumeric characters (letters and digits), in the code range 48-57, or 65-90 or 97-122, and with a letter in the first position. Spaces are not allowed. Only strings of up to 32 characters are allowed.

<String> is an expression referring to a literal or a group of literals, that is an expression enclosed in double quotation marks, and constructed according to the rules given above for Literals (section 1). All strings have to be enclosed in double quotation marks.

All assignment statements are terminated with a '.' (Full Stop).

The spaces (blanks) are optional. One or more space characters are simply ignored. New Line and Carriage Return are also ignored, meaning that an assignment statement may run over more than one line.

Two slashes '//' mark the start of a comment or other addition not meant to be processed by the program. Everything following a double slash is ignored.

Often used literals may be assigned to tokens that are independent of the current character set:

```

Any = "${Any}" .
Digit = "${Digit}" .
Alpha = "${Alpha}" .
AlNum = "${AlNum}" .
UpCase = "${UpCase}" .
LowCase = "${LowCase}" .
NewLine = "${Newline}" .
NoNewLine = "${NoNewline}" .
Blank = " " .
Dot = "." .
Comma = "," .
SemiColon = ";" .
Colon = ":" .
Quotation = "\"" .
Hyphen = "-" .
LParen = "(" .
RParen = ")" .
LBracket = "[" .
RBracket = "]" .
LCurly = "{" .
RCurly = "}" .
Slash = "/" .
AE = "${146}" . // Character 'Æ'
ae = "${145}" . // Character 'æ'
OE = "${157}" . // Character 'Œ'
AA = "${143}" . // Character 'À'

```

NB! The tokens listed are not part of the basic language, but examples of the use of the language. The list has to be constructed and provided by the user.

2.1 Alternatives

Alternative strings may be assigned to the same Token, using the character '|' (Code 124):

```
<TokenName> = <String1> | <String2> .
```

means that TokenName represents either String1 or String2.

This may for instance be used to extend the range of the Upper Case letters represented by the Token Name: UpCase:

```
UpCase = "${UpCase}" | AE | OE | AA .
```

This may also be written on more than one line:

```
UpCase =  "${UpCase}"
          |  AE
          |  OE
          |  AA .
```

The multiline form is especially useful when the literal expressions are very long.

2.2 Sequence (concatenation of strings)

A token may be assigned to a sequence of literals, represented as tokens or literals:

```
<TokenName> = <String1> <String2> .
```

means that TokenName represents a sequence of String1 and String2, put together with no intermediate characters.

NinthCent = "18" Digit Digit .

represent a string of numbers starting with '18' and followed any two digits (= a year in the 19th century).

2.3 Repetitions

The possibility of repetitions of types of strings are expressed using square brackets, '[' and ']', or curly brackets, '{' and '}'.

<TokenName1> = <String1> [<String2>] .

means that TokenName1 represents String1 followed by zero or just one occurrence of String2.

<TokenName2> = <String1> { <String2> } .

means that TokenName2 represents String1 followed by zero or one or more than one occurrence of String2.

2.4 Groupings

Expressions on the right hand side may be grouped using parentheses, '(' and ')', in order to make sure that the expression is interpreted in the correct way:

<TokenName1> = <String1> (<String2> | <String3>) .

is different from:

<TokenName2> = (<String1> <String2>) | <String3> .

Example:

Year = ("18" | "19") Digit Digit .

means any year in the 19th or the 20th century, while

Number = "18" | "19" Digit Digit .

means either the number '18' or any year in the 20th century.

2.5 Dictionary look-up

The right hand side of an assignment statement may also be a dictionary look-up, using the expression:

dictlookup(<DictionaryName>)

or

dictcheck(<DictionaryName>)

In this case the left hand side refers to a group of literals, strings of any length, found in a dictionary file. Each string in the file is on a separate line terminated with a New Line character.

A dictionary file and the name of the file is supplied by the user.

```
Surname = dictlookup(Surnames) .
```

means that Surname refers to any string found in the dictionary file named 'Surnames'.

Dictionary look-up may in practice be used only in certain cases (in grammars for the DEC program of the FACIT Prototype. The dictionary has to be declared in the head of a grammar file (cf. the section below on grammars).

2.6 Concluding remarks on token-assignment statements

Only one expression is allowed on the left hand side. If the user wants an expression with more than one word - e.g. for the sake of intelligibility - capital letters may be used when linking the words together to form one expression:

```
TwoWords = <String1> .
```

Two or more assignment statements cannot have the same left hand side. If different literals are to be linked to the same token, the literals are all written on the right hand side of one assignment statement and separated with the alternate sign: '|'.|

3. Higher order expressions

Literals and Tokens (representing literals) may be organized into more complex expressions, using the same basic language as the one used in token-assignment statements:

```
<Identifier> = <Expression1> .  
    // Simple assignment  
  
<Identifier> = <Expression1> <Expression2> .  
    // Sequence  
  
<Identifier> = <Expression1> | <Expression2> .  
    // Alternation or Selection  
  
<Identifier> = <Expression1> [ <Expression2> ] .  
    // Zero or one occurrence  
  
<Identifier> = <Expression1> { <Expression2> } .  
    // Zero or one or more occurrences  
  
<Identifier> = <Expression1> ( <Expression2> ) .  
    // Grouping of expressions
```

where the left hand side, <Identifier>, may be any string consisting of alphanumeric characters (letters and digits), in the code ranges 48-57, 65-90 or 97-122, and with a letter in the first position. Spaces are not allowed. Only strings of up to 32 characters are allowed.

The expressions on the right hand side are either expressions of the same type as on the left hand side (= identifiers), or tokens or literals. A literal is an expression enclosed in double quotation marks, and otherwise constructed according to the rules given above (section 1.). A token is an expression referring to one or more literals according to some

assignment statement constructed according to the rules given above (section 2). It is actually a special case of an identifier.

All assignment statements are terminated with a '.' (Full Stop).

The spaces (blanks) are optional. One or more space characters are simply ignored. New Line and Carriage Return are also ignored, meaning that an assignment statement may run over more than one line.

Two slashes '//' mark the start of a comment or other addition not meant to be processed by the program. Everything following a double slash is ignored.

Two or more left hand sides cannot be the same. Alternatives have to be expressed using the operator '|'.
'|'.

An expression cannot occur on the right hand side in the same assignment statement where it occurs on the left hand side. (Recursive assignments are not allowed.)

A special type of higher order assignment statement is only allowed in grammars (control files for the DEC program of the FACIT prototype):

```
<FieldName> : <Identifier> = <RuleExpression>
```

This assignment statement will establish a field in the internal record format of the FACIT Prototype with the name given in front of the colon (<FieldName>), and then write everything fulfilling the criteria specified in the statement following the colon to that field.

Apart from this special case only one expression is allowed on the left hand side of an assignment statement.

Higher order assignment statements are used to write formal descriptions, also called grammars, of entries to the FACIT prototype, typically copies of catalogue cards, produced with image scanners and optical character recognition.

4. Grammars

Grammars consist of two sections, a dictionary section and a rules section.

4.1 Dictionary Section

The dictionary section declares the dictionaries to be used in connection with this specific grammar. It is optional, but if it exists it has to be the first section of a grammar file (apart from lines of comments starting with double slashes '//').

The dictionary section always starts with the line:

```
[Dictionary]
```

or

```
[Dictionaries]
```

After this follows a list in dictionary declarations of the form:

```
<DictionaryName> = "<FileName>" .
```

The <DictionaryName> is an identifier to be used in the rules section as an argument for the dictionary look-up function:

```
dictlookup(<DictionaryName>)
```

An identifier is a string of letters and digits (in the code ranges 48-57, 65-90 or 97-122), with the first one being a letter, and not exceeding 32 characters.

The <FileName> is the name of an existing dictionary file². The name has to conform to the normal conventions for file names in a DOS/Windows environment: up to 8 characters, possibly followed by a dot, '.', and an extension of up to 3 characters.

All declaration has to be terminated with a dot, '.'.

4.2 Rules section

The Rules section declares the rules of the grammar, using assignment statements as defined above (Sections 1 - 3). The Rules section is mandatory and always starts with the line:

```
[Rule]
```

or

```
[Rules]
```

Each rule has the form:

```
<ElementName> = <RuleExpression> .
```

where <ElementName> is an identifier which is used in the formulation of rule expressions (the right hand side of other rules). Each rule is used to define one element using other elements, tokens or literals.

An <ElementName> is a string of letters and digits (in the code range 48-57, 65-90 or 97-122), with the first one being a letter, and not exceeding 32 characters.

²) The present version of the FACIT Prototype assumes that a dictionary file is a plain text file ("ASCII file"). Each entry occupies one line, terminated by New Line. The searching algorithm is not sensitive to case. The file does not have to be ordered, e.g. alphabetically.

This organisation is not efficient with larger dictionaries. The searching algorithm should also be enhanced in order to provide more sophisticated matching, e.g. taking into account probabilities of errors produced in OCR. Future versions of the FACIT Application are planned to incorporate such improvements.

The <RuleExpression> is an expression formulated in the FACIT Description Language as specified in section 1 - 3 above, using Identifiers (Element Names or Tokens), Literals, the following characters: '|', '{', '}', '[', ']', '(', ')', and the reserved words: 'dictlookup' and 'dictcheck'.

A Token is actually just a special case of an Element Name, where the defining rule has only literals on the right hand side, not other elements needing definition.

The first rule always has to have the name (identifier): 'Main':

Main = <FirstRuleExpression> .

Eventually the chain of rules has to end up in rules with only Literals on the right hand side. No Element Names can be left undefined.

And it may be worth repeating that an element can only be defined once, that is an Element Name can only occur on the left hand side of one rule.

No recursive rules are allowed: An Element Name cannot occur as part of its own definition.³

As described above (section 3) special rules establish names of fields in the internal records of the FACIT prototype and assign any text that matches the defining rule to that field:

<FieldName> : <ElementName> = <RuleExpression> .

A Field Name and the associated Element Name can be the same.

Comments may be placed at the end of lines, after a double slash, '//', or on separate lines, to help the human reader understand what the grammar is supposed to do.

³)This is not the whole truth. Recursive definitions are allowed, but only if the recursive element is not the first to be evaluated:

Integer = Digit | Digit {Integer} .

is allowed, but not

Integer = Digit | {Integer } Digit .

The same goes for indirect recursion, where A is defined by B, which uses A in the definition.

But since the best thing is to avoid recursion in order to escape using the bad forms inadvertently, the rule is formulated more categorically than absolutely necessary.

The rules taken together describe a hierarchically structured text such as an entry in a card catalogue or a bibliography. They may be read as definitions of the elements of the text, but are often called "re-writing rules", "production rules" or simply "productions", because they are able to "generate" all the possible texts matching the set of definitions.

The actual names of the elements and tokens are provided by the user. They can be completely arbitrary, but will normally be constructed in such a way that they remind a human reader of the reality that is being referred to.

The field names may be arbitrary as well, but will normally be constructed so as to reflect the fields of a bibliographic description. The field names also have to be used to refer to the fields in other programs in the FACIT Prototype, such as the programs producing the output files.

A sample grammars for analyzing bibliographic records is shown in Appendix 2.

5. Formal specification of the description language

The following is a formal specification of the language presented in sections 1 - 4, using the so-called BNF notation (Backus-Naur Form):

```
Grammar ::= DictionariesSection RulesSection .
DictionariesSection ::= DictionaryHeading
    DictionaryDeclaration {DictionaryDeclaration} .
DictionaryHeading ::= '[' 'Dictionaries' ']' |
    '[' 'Dictionary' ']' .
DictionaryDeclaration ::=
    DictionaryName '=' ' "' FileName ' "' '.' .
DictionaryName ::= Identifier .
FileName ::= FileName .
    // FileName represents a file name constructed
    //according to the conventions of the DOS/Windows
    //environment.
RuleSection ::= RulesHeading FirstRule {Rule} .
RulesHeading ::= '[' 'Rule' ']' | '[' 'Rules' ']' .
FirstRule ::= 'Main' '=' RuleExpression '.' .
Rule ::= [FieldName ':' ] ElementName '='
    RuleExpression '.' | TokenName '=' Literal
    { '|' Literal } '.' .
RuleExpression ::= Sequence | RuleAlternation .
RuleAlternation ::= Sequence '|' Sequence { '|'
    Sequence } .
Sequence ::= SequenceSection { SequenceSection } .
SequenceSection ::= SequenceElement | Repetition |
    ElementAlternation .
```

```

Repetition ::= '[' SequenceElement { SequenceElement }
            ']' | '{' SequenceElement { SequenceElement } '}'.

// A Sequence containing only a Repetition, allow-
// ing zero occurrences of the entity defined,
// should not be used, since it will in all pro-
// bability lead to formatting errors.

ElementAlternation ::= '(' SequenceElement '|'
                    SequenceElement { '|' SequenceElement } ')'.

SequenceElement ::= ElementName | TokenName |
                  Literal | DictionaryLookUp .

DictionaryLookUp ::=
    'dictlookup' '(' DictionaryName ')' |
    'dictcheck' '(' DictionaryName ')' .

FieldName ::= Identifier .

ElementName ::= Identifier .

TokenName ::= Identifier .

Identifier ::= Alpha { Alpha | Digit } .
            // Not more than 32 characters long

Literal ::= '"' CharElement { CharElement } '"' .

CharElement ::= AnyChar | '[' ['^'] AnyChar { AnyChar }
              ']' | '[' ['^'] AnyChar '-' AnyChar { AnyChar '-'
              AnyChar } ']' .

AnyChar ::= Char | '$' | '{' CharExpr '}' | '\ ' '$' |
           '\ ' { '&' Integer } | '\ ' '^' | '\ ' '$' |
           '\ ' [ '\ ' '^' ] .

// Char represents any character in the code
// range: 32 - 127, except '\ ' '[' '^'
// or '$'

CharExpr ::= 'Any' | 'Alpha' | 'NoAlpha' | 'Digit' |
            'NoDigit' | 'AlNum' | 'NoAlNum' | 'UpCase' |
            'NoUpCase' | 'LowCase' | 'NoLowCase' | 'Newline'
            | 'NoNewLine' | 'WS' | 'NoWS' .

Integer ::= Digit { Digit} .

Alpha ::= 'A' | 'a' | 'B' | 'b' | 'C' | 'c' | 'D' |
         'd' | 'E' | 'e' | 'F' | 'f' | 'G' | 'g' | 'H' |
         'h' | 'I' | 'i' | 'J' | 'j' | 'K' | 'k' | 'L' |
         'l' | 'M' | 'm' | 'N' | 'n' | 'O' | 'o' | 'P' |
         'p' | 'Q' | 'q' | 'R' | 'r' | 'S' | 's' | 'T' |
         't' | 'U' | 'u' | 'V' | 'v' | 'W' | 'w' | 'X' |
         'x' | 'Y' | 'y' | 'Z' | 'z' .

Digit ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' |
         '7' | '8' | '9' .

```

6. Supplementary descriptions

The FACIT system allows some other types of descriptions, using somewhat different language constructions. It would be desirable to harmonize these descriptions with the main description language, but this has not been possible within the time allotted to the project.¹

6.1 Formal analysis by area

For some information found on a catalogue card it may be simpler to describe the location on the card, rather than the formal features of that information in terms of character sequences.

An example of this is the following: In one library information about the location of the publication is always found in the top right corner of the card, but the location marks ("signatures") shows a lot of variation, which makes it difficult to write the necessary "grammar rules". In this case the user can specify that all information found in the top right corner should be allocated to the field "Location Mark".

The language constructs to do this cannot be easily incorporated into the basic FACIT Description Language, and the actual processing of the cards, eliciting information found in a specific area, should be carried out before processing the main body of information in the card. This is done by the BDEC program (Basic Decomposition) in the FACIT Prototype using special control files.

The syntax is somewhat different from the Basic Description Language, partly because the job is different, partly because it was formulated at an earlier stage of the development process and has not been revised in view of later developments. A detailed description is found in Section 6 of the FACIT Technical Report no 4: The FACIT Prototype.

The information needed to write the specifications for the BDEC program is the following

- Position of the information in terms of the number of lines in the card and the number of "columns" (writing positions) in the lines.
- Easily recognizable elements in the character strings found in a specific area of the card.

The basic syntax is:

```
[<FieldName>] <FieldExpression
```

The <FieldName> is a string of alphanumeric characters with a letter as the first character and not exceeding 32 characters. The Field Name establishes a field with that name in the internal card format.

The <FieldExpression> formulates the criteria. Everything matching the criteria is written to the specified field in the internal record.

The individual rules start with a '[' and end when the next '[' is encountered or when End of File is reached. The rules are not delimited by a '.', Dot.

Comments may be added at the end of a line or on separate lines following a '//', Double Slash.

Several criteria may be formulated, linked with the operators 'and', 'or' and 'not' (Boolean operators).

Criteria expressions may be grouped with '(' and ')' to indicate the correct logical relations.

Information matching the criteria is moved to the specified field, while the rest of the information is

left in the default field, TEXT, for further processing.

The individual criteria may relate to strings or to positions in a given area of the card.

6.1.1 Strings

The character strings are evaluated "word" by "word", that is the card is analysed into "words" which are continuous strings of characters delimited by Space, Tab or New Line characters. Four string-operators are available: 'like', 'contains', 'length' and 'rlength'.

The expression

```
like("<StringExpression>")
```

evaluates to true, if the "word" being evaluated matches the <StringExpression>. The string expression is formulated as defined above in Section 1 and 2, using characters in the range 32 to 126, character tokens ({<number>}) or wild cards ({Alpha}, {Digit}, {Newline} etc.). The String Expression has to be enclosed in Double Quotation marks.

The expression

```
contains("<StringExpression>")
```

evaluates to true, if the "word" being evaluated contains a substring that matches the <StringExpression>. The String Expression is formulated as above.

The expression

```
length <ComparisonOperator> <NumberExpression>
```

evaluates to true if the length of the current "word" holds the relation specified in the <ComparisonOperator> to the number given in the <NumberExpression>. The possible operators are:

= : Equals

< : Less than

<= : Less than or equal

> : Greater than

>= : Greater than or equal

The <NumberExpression> may be any expression that evaluates to a number

The expression

```
rlength <ComparisonOperator> <NumberExpression>%
```

is like 'length', but results in the "relative length" of the current "word" measured against the longest line in the card. The value is given as a percentage (rounded to the nearest integer) and the Number Expression has to evaluate to a number between 0 and 100.

6.1.2 Areas

The Area criterium specifies whether the current "word" is located in a certain area.

The general structure is:

<Position> <ComparisonOperator> <NumberExpression>
[%]

or

<Position> between <NumbExpress> [%] and <NumbExpress>
[%]

The <Position> is the position of the current "word". The following expressions are available:

'col' or 'colleft' : The column (character position) of the first character in the "word".

'rcol' or 'rcolleft' : The relative position of the first character in the "word", compared to the longest line on the card and expressed as a percentage (rounded to the nearest integer).

'colright' : The column (character position) of the last character in the "word".

'rcolright' : The relative position of the last character in the "word", compared to the longest line on the card and expressed as a percentage (rounded to the nearest integer).

'row' : The row number (line number) of the current "word".

'rrow' : The relative position of the row (line) of the current "word", compared to total number of lines on the card, and expressed as a percentage (rounded to the nearest integer).

The Comparison Operators are as for strings: '=', '<', '<=', '>' and '>='.

The Number Expressions (<NumberExpression> or <NumbExpress>) are any expression that evaluates to a number (rounded of to the nearest integer). They may be constructed using the normal mathematical operators: '+' (sum), '-' (difference), '*' (multiplication) and '/' (division). When the relative Position is given the Number Expression has to be a percentage between 0 and 100.

Two special expressions are available:

'rowcount' : The total number of rows (lines) on the card (= lines with characters and empty lines just terminated by a New Line character). The number of lines refer to the plain text copy of the card produced by OCR, not to the number of possible line on the original cardboard card.

'columncount' : The total number of columns (character positions) on the card (= the longest line of characters, including spaces and terminated by a New Line character). The number of columns refer to the plain text copy of the card produced by OCR, not to the number of possible columns on the original cardboard card.

6.1.3 Examples

Example 1: Keywords of maximum 10 characters are placed in the lower left corner in the last two lines and on the leftmost third of the card:

```
[Keyword]
(row >= rowcount - 1) and (length <= 10) and
  (rcol <= 33%)
```

Example 2: Subject terms are placed in the upper left corner in the first line and on the leftmost half of the card:

```
[Subject]
(rcol <= 50%) and (row = 1)
```

Example 3: An accession number is placed in the upper right corner in one of the first three lines. The number consists of two digits followed by a hyphen and then one to four digits. All other information in this area is ignored:

```
[AccessNum]
( row <= 3 ) and ( col >= 20 ) and
  (like("${Digit}${Digit}-${Digit}") or
   like("${Digit}${Digit}-${Digit}${Digit}") or
   like("${Digit}${Digit}-${Digit}${Digit}${Digit}"
    ${Digit}" ) or like("${Digit}${Digit}-${Digit}
    ${Digit}${Digit}${Digit}" ) )
```

6.2 Grammar Selection

In the control file for the DEC program it is possible to specify which field - in the internal card format - that are to be processed, which grammar (syntax file) to use and under what conditions.

The general structure is:

```
[<FieldName>]
  <GrammarFile0> : .
  <GrammarFile1> : <Conditions1> .
  <GrammarFile2> : <Conditions2> .
.
.
```

All records in the internal card format includes a field called TEXT, which holds the text not yet processed. All the text from the card is written to this field, when the record is created. The TEXT field is assumed by the program if the <FieldName> is omitted.

One and only one Grammar File has to be listed as the default grammar - that is without Conditions after the colon.

The File Name of the Grammar File has to follow the normal conventions in the DOS/Windows environment.

The Conditions are one or more String Patterns combined with the logical operators: 'and', 'or' and 'not' and if needed grouped with parentheses.

"<StringPattern>"

not "<StringPattern>"

"<StringPattern1>" and "<StringPattern2>"

"<StringPattern1>" or "<StringPattern2>"

The String Patterns are formulated according to the rules in Section 1 and 2, using characters in the code range 32 to 126, character tokens ($\{\langle\text{Number}\rangle\}$), wild cards ($\{\text{Alpha}\}$, $\{\text{Digit}\}$, $\{\text{Newline}\}$ etc.) and regular expressions ($[abc]$, $[\^abc]$, $[A-Z]$ etc). The String Pattern has to be enclosed in Double Quotation marks.

The DEC program will check whether the specified field (default TEXT) contains a string or strings matching the conditions given and then execute the associated grammar with the text in the field as input.

Example 1: If the card includes the string 'ISBN' the Grammar File 'isbn.syn' is executed, otherwise the Grammar File 'default.syn':

[TEXT]

```
"default.syn" :  
"isbn.syn" : "[Ii][Ss][Bb][Nn]" .
```

Example 2: If the card includes the string 'see also' the card is skipped, otherwise the Grammar File 'default.syn' is executed:

[text]

```
"default.syn" : .  
: "[Ss]ee also" .
```

Example 3: A special Grammar File has been constructed to post process the author field ('AU') after the main processing of the card:

[AU]

```
"author.syn" : .
```

A more formal presentation of these language constructs is given in Section 7 in the FACIT Technical Report no 4: The FACIT Prototype.

Appendix 2

Sample Formal Description with Source File and Resulting File

Grammar file

```
//
//      text2.syn
//
//      1995-06-13 NJD   Created
//      1995-08-01 NEW   Corrected and modified
//

[Dictionary]
Places      = "tables\places.tbl" .
ForeNames   = "tables\forenames.tbl" .
SurNames    = "tables\surnames.tbl" .
CallNames   = "tables\calnames.tbl" .

[Rules]
Main = OptBlanks Signature OptBlanks Author OptBlanks
      TitleSection OptBlanks ImprintSection OptBlanks Notes .

SI:Signature = [ "(" Number ")" Dot ] SignPart
               { Dot SignPart } [ Spaces RomanNumber
               [ Spaces "aa" ] ] .

SignPart = AlphaString | Number .

AU:Author = Author3 | Author2 | Author1 .
Author1 = CallName OptSpaces ForeName2 { Spaces ForeName2 }
          [ Dot ] OptSpaces NL .
Author2 = ForeName1 OptSpaces "DE" Spaces RegName2 [ Dot ]
          OptSpaces NL .
Author3 = RegName1 OptSpaces NameSupplement [ Dot ] OptSpaces
          NL NL .

CallName = UpCase " " { UpCase " " } .
ForeName1 = UpCase " " { UpCase " " } .
RegName1 = UpCase " " { UpCase " " } .

ForeName2 = UpCase { UpCase } .
RegName2 = UpCase { UpCase } .

NameSupplement = "(" AlphaString { Spaces AlphaString } ")" .

TI:TitleSection = TitleLine { NL TitleLine } TitleEnd .
TitleLine = NoNL { NoNL } .
TitleEnd = NL NL .

ImprintSection = [ ( "In" | "A" | "A'") Spaces ] Place
                 Comma OptSpaces Publisher Comma OptBlanks
                 Year Comma OptBlanks Size Comma OptBlanks
                 ( Pagina | Volume ) Dot .

PL:Place = DictLookup(Places) .
PU:Publisher = { AlNum | "*" | " " | "'" | "=" | "-" | "(" |
                ")" | Space | NL } .
YR:Year = ActYear [ "-" ActYear ] .
ActYear = ( "15" | "16" | "17" | "18" | "19" ) Digit Digit .
SZ:Size = Size1 | Size2 | Size1 OptBlanks "(" Size2 ")" .
Size1 = "4" [ "o" ] | "8" [ "o" ] .
Size2 = "mm" Dot OptBlanks Number "x" Number .
PG:Pagina = "p" Dot OptSpaces [ "[" ] ( RomanNumber | Number )
           [ "]" ] OptBlanks [ Supplement ] .
VL:Volume = "vol" Dot OptSpaces Number OptBlanks
```

```

Supplement = [ Supplement ].
Supplement = Word { OptBlanks Word } .
NT:Notes = { Any } .
OptBlanks = { Blank } .
Blanks = Blank { Blank } .
Blank = Space | NL .
OptSpaces = { Space } .

Word = [ "(" | "]" | ( UpCase | LowCase ) { LowCase } [ Punct ] .
Punct = "." | "," | ";" | "?" | "!" | "(" | ")" | "*" .
RomanNumber = RomanDigit { RomanDigit } .
RomanDigit = "I" | "V" | "X" | "L" | "C" | "M" .
Number = Digit { Digit } .
AlphaString = Alpha { Alpha } .
Digit = "${Digit}" .
NonDigit = "${NoDigit}" .
Alpha = "e" | "a" | "${Alpha}" .
AlNum = "e" | "a" | "${AlNum}" .
UpCase = "${UpCase}" .
LowCase = "e" | "a" | "${LowCase}" .
NL = "${Newline}" .
NoNL = "${NoNewLine}" .
Spaces = Space { Space } .
Space = " " .
Dot = "." .
Colon = ":" .
SemiColon = ";" .
Comma = "," .
Any = "${Any}" .

```

Source file

782.15
T O S C A N E L L I GIUSEPPE.

Discorso del deputato di Pontedera G.T. pronunciato nella tornata del 14 giugno 1872 contro il progetto di legge: Convenzione per l'istituto di studi superiori in Firenze.

Roma, Tip. Eredi Botta, 1872, mm.204x144, p.32.

§

Misc.57.21
T O S C A N O AGNELLO.

Istruzioni per conoscere le principali malattie del bestiame da macello da servire precipuamente di guida a' primi eletti municipali di A. T.

Napoli, Tip.del "Fibreno", 1835, 8° (mm.216x135), p.56.

§

3.G.7.1
T O S I GIOVANNI.

Apologia Accademica e forense per l'abate Giovanni Tosi in una causa di aucupio di pettirossi a civetta.

In Firenze, Nella stamperia di Gio. Battista Stecchi, 1748, 4° (mm.340x220), p. LXXVI con una incisione nel frontespizio.

§

4.7.2.1
T O U R N A C H O N DE MONTVERAN.

Histoire critique et raisonnée de la situation de l'Angleterre au 1.er Janvier 1816, sous les rapports de ses finances, de son agriculture, de ses manufactures, de son commerce et sa navigation, de sa constitution et ses lois et de sa politique extérieure; par m. De Montvéran.

A' Paris, Chez Barrois l'ainé libraire, 1819-1820, 8° (mm.202x125), vol.5.

§

10.8.5.3

T O U S S A I N T CLAUDE JACQUES.
Traité de géométrie et d'architecture théo-
rique et pratique, simplifié, ouvrage clas-
sique présenté à S.ex. monseigneur le Mi-
nistre de l'Interieur par C.I.Toussaint...

A Paris, Chez l'Auteur (De l'imprimerie
de Hocquet e C.e), 1811-1812, 4° (mm.281x215),
vol.2 con tavole.

§

(11).C.9.5.6 XLV aa
T O R R I G I A N I ANTONIO

S. Pietro Apostolo. Orazione detta dal p.
A.T. nella chiesa di S. Michele Visdomini
il giorno 29 giugno 1855.

Firenze, Tip. G. Mariani, 1855, 8° (mm.
243x168), p.16.

§

289.21

T O S C A N E L L I GIUSEPPE.

Discorsi pronunziati alla camera
nelle tornate del 21 e 22 dicembre
1870 da G.T. deputato dl Pontedera
contro i progetti di legge. Accetta-
zione del plebiscito romano, traspor-
to della capitale in Roma.

Firenze, Tip. Eredi Botta, 1870,
mm. 230x150, p.47.

§

(11).C.5.3.3 XLVII

T O S C A N A (Granducato di).

Aggiunta et ampliacione de privilegi
concessi à gl'archibusieri à Cavallo di
Romagna, cauati da rescritti di diversi
tempi esistenti nella banca di S.A,S.

In Firenze, Nella stamperia di Zanobi
Pignoni, 1629, 4° (mm.200x140), p.[4].

§

2.C.6.5

T O R R I A N I FRANCESO.

Epistola Francisci Tyrriani sacerdotis
societatis Jesv. De definitione propria
peccati originalis, ex Dionysio Areopagi=
ta, et de Conceptione Virginis et matris
Dei, sine peccato, ex scriptura Angelicae
Salutationis et testimonijs antiquorum
Patrum...

Florentiae, Apud Bartholomaeum Sermartel=
lium, 1581, 4^o (mm. 203x149), p. 44.

Resulting file

The example does not show the direct result of applying the grammar file, since this will be in the internal (16-bit) format. The output file was produced by applying an output procedure corresponding with the grammar file, using tags in UNIMARC style.

```
001 TEST-00000432
200 $aDiscorso del deputato di Pontedera G.T. pronunciato
nella tornata del 14 giugno 1872 contro il progetto di
legge: Conven- zione per l'istituto di studi superiori in
Firenze.
210 $aRoma $cTip. Eredi Botta $d1872 $emm.204x144 $fp.32
700 $aT O S C A N E L L I GIUSEPPE.
910 $a782.15
$
001 TEST-00000433
200 $aIstruzioni per conoscere le princi= pali malattie del
bestiame da macel= lo da servire precipuamente di gui da a'
primi eletti municipali di A. T.
210 $aNapoli $cTip.del *Fibreno* $d1835 $e8° (mm.216x135)
$fp.56
700 $aT O S C A N O AGNELLO.
910 $aMisc.57.21
$
001 TEST-00000434
200 $aApologia Accademica e forense per l'abate Giovanni Tosi
in una causa di aucupio di pet- tirossi a civetta.
210 $aFirenze $cNella stamperia di Gio. Batti- ta Stecchi
$d1748 $e4°(mm.340x220) $fp. LXXVI con una incisione nel
frontespizio.
700 $aT O S I GIOVANNI.
910 $a3.G.7.1
$
001 TEST-00000435
200 $aHistoire critique et raisonnée de la si- tuation de
l'Angleterre au l.er Janvier 1816, sous les rapports de ses
finances, de son agriculture, de ses manufactures, de son
commerce et sa navigation, de sa constitution et ses lois
et de sa politi- que extérieure; par m. De Montvéran.
210 $aParis $cChez Barrois l'ainé libraire $d1819-1820
$e8° (mm.202x125) $gvol.5
700 $aT O U R N A C H O N DE MONTVERAN.
910 $a4.7.2.1
$
001 TEST-00000436
200 $aTraité de géometrie et d'architecture théo- rique et
pratique, simplifié, ouvrage clas- sique présenté à S.ex.
monseigneur le Mi- nistre de l'Interieur par
C.I.Toussaint...
210 $aParis $cChez l'Auteur (De l'imprimerie de Hocquet e
C.e) $d1811-1812 $e4°(mm.281x215) $gvol.2 con tavole.
700 $aT O U S S A I N T CLAUDE JACQUES.
910 $a10.8.5.3
$
```

001 TEST-00000437
 200 \$aS. Pietro Apostolo. Orazione detta dal p. A.T. nella
 chiesa di S. Michele Visdomini il giorno 29 giugno 1855.
 210 \$aFirenze \$cTip. G. Mariani \$d1855 \$e8° (mm. 243x168)
 \$fp.16
 700 \$aT O R R I G I A N I ANTONIO
 910 \$a(11).C.9.5.6 XLV aa
 \$

001 TEST-00000438
 200 \$aDiscorsi pronunziati alla camera nelle tornate del 21 e
 22 dicembre 1870 da G.T. deputato di Pontedera contro i
 progetti di legge. Accetta= zione del plebiscito romano,
 traspor- to della capitale in Roma.
 210 \$aFirenze \$cTip. Eredi Botta \$d1870 \$emm. 230x150 \$fp.47
 700 \$aT O S C A N E L L I GIUSEPPE.
 910 \$a289.21
 \$

001 TEST-00000439
 200 \$aAggiunta et ampliacione de privilegi concessi à
 gl'archibusieri à Cavallo di Romagna, cauati da rescritti
 di diversi tempi esistenti nella banca di S.A,S.
 210 \$aFirenze \$cNella stamperia di Zanobi Pignoni \$d1629 \$e4°
 (mm.200x140) \$fp.[4]
 700 \$aT O S C A N A (Granducato di).
 910 \$a(11).C.5.3.3 XLVII
 920 \$a,
 \$

001 TEST-00000440
 200 \$aEpistola Francischi Tvrriani sacerdotis societatis Jesv.
 De definitione propria peccati originalis, ex Dionysio
 Areopagi= ta, et de Conceptione Virginis et matris Dei,
 sine peccato,ex scriptura Angelicae Salutationis et
 testimonijs antiquorom Patrum...
 210 \$aFlorentiae \$cApud Bartholomaeum Sermartel= lium \$d1581
 \$e4°(mm.203x149) \$fp.44
 700 \$aT O R R I A N I FRANCESO.
 910 \$a2.C.6.5
 \$