Semester Project

Digital Libraries

Realized by

The Quang Nguyen

Supervisor

Martin Rajman

Assistants

David Portabella Clotet

Miroslav Melichar

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Summer Semester 2007

Contents

Acknowledgements

I would like to say a special thanks to:

- **David Portabella Clotet** and **Miroslav Melichar** (EPFL) for their guide during the whole semester.
- **David Aymonin** (EPFL) for his kind help on Nebis and Infoscience.
- **Patrick Jermann** (EPFL) for his help on Nebis.
- **Benjamin Barras** (EPFL) for giving us the access to MySQL server.
- **Jill Cousins** and **Sjoerd Siebinga** (The European Library) for giving us the access to their database.
- **The Index Data's people** for their help on YAZ and Z39.30.
- Dr. **Lee Giles** (CiteSeer) for his answer about CiteSeer's license.

1 Introduction

The amount of available ressources on the Web does not cease to increase. In order to provide them, there is a lot of factors we need to handle. We require the ressources to be well structures to facilitate the work of updating, sharing and distributing. Therefore, we need first analyze them before adopting any choice of available technologies.

In particular, analyzing data provided by online libraries raises a big challenge. Once diving into this environment, we can quickly notice its complexities, and its wealth. Along with a huge amount of data, we have to deal with diffrent protocols and metadata formats. This project will discuss about three standard formats and two protocols. As their direct application, we also intend to discuss about the retrieving and analyzing data from four different online libraries.

2 Format, protocol and tools description

In this section, we aim to give an overview of different formats, protocols and tools used in this project. For each of them, a description and some specifications along with examples will be given. It would not be necessary to dive into all details of each format or protocol because most of them are very complicated. The objective of this section is to give enough information to understand features used in this project.

2.1 The Dublin Core Metadata Initiative

The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems. [3]

The Dublin Core (DC) metadata standard provides a simple yet effective element set for describing a wide range of networked resources. It makes searching and retrieving resources simpler and faster. The success of DC can be testified by its adoption by governments, libraries, museums, archives, publishers, and more.

The DC standard includes two levels: Simple and Qualified. The Simple DC is a set of fifteen elements: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights. Also the Qualified DC includes three additional elements: Audience, Provenance and RightsHolder. Detailed description for each element can be found at [4]. The European Library studied in this project (section 3.4) uses four additional elements: Bibliographic Citation, Has Format, Temporal, Alternative.

Every DC element is optional and may be repeated and there is no prescribed order in DC for presenting or using the elements. The table 1 shows an example of the DC format within a XML document.

In the context of this project, DC format is used by OAI-PMH (section 2.4) and by many collections within the European Library (section 3.4).

```
<dc:title>  36 Problems for Semantic Interpretation </dc:title>
<dc:creator>  Gabriele Scheler </dc:creator>
<dc:subject>  36 Problems for Semantic Interpretation </dc:subject>
<dc:description>
  This paper presents a collection of problems for natural
  language analysis derived mainly from theoretical linguistics.
</dc:description>
<dc:contributor>
  The Pennsylvania State University CiteSeer Archives
</dc:contributor>
<dc:publisher>unknown</dc:publisher>
<dc:date>19930811</dc:date>
<dc:format>ps</dc:format>
<dc:identifier>http://citeseer.ist.psu.edu/1.html</dc:identifier>
<dc:source>
  ftp://flop.informatik.tumuenchen.de/pub/fki/fki17993.ps.gz
</dc:source>
<dc:language>en</dc:language>
<dc:rights>unrestricted</dc:rights>
```

Table 1: Example of a XML Dublin Core format from CiteSeer [1]

2.2 MARC 21

The MARC formats are standards for the representation and communication of bibliographic and related information in machine readable form. MARC is an acronym for MAchine-Readable Cataloging. The first version of MARC was developed at the Library of Congress beginning in the 1960s. MARC standards are organized by formats of records and each format is a set of fields.

MARC has five consise formats of records (sources from [10]):

- Authority records: provide information concerning the authorized forms of names and subjects to be used as access points in MARC records, the forms of these names, subjects and subdivisions to be used as references to the authorized forms, and the interrelationships among these forms.
- Bibliographic records: provide bibliographic information about printed and manuscript textual materials, computer files, maps, music, continuing resources, visual materials, and mixed materials. Bibliographic data commonly includes titles, names, subjects, notes, publication data, and information about the physical description of an item. The bibliographic format contains data elements for the following types of material: books, continuing resources, computer files, maps, music, sound recordings, visual materials, mixed materials.
- Classification records: provide information about classification numbers and the captions associated with them that are formulated according to a specified authoritative classification scheme.
- Community Information records: provide descriptions of non-bibliographic resources that fulfill the information needs of a community.
- Holdings records: provide holdings information.

As mentionned above, each format is a set of fields, so each record is divided logically into fields. A field is defined by a 3-digit tag, two nullable 1-digit indicators and a subfield code (a character).

An example is given in the table 2. Another example of a complete record in MARC format can be found at table 3.

Fomat	Field's ID	Field's Name	Example of values
Authority	670_♯_♯_a	Source citation	LC data base, 2/18/84
Bibliographic	245_♯_♯_a	Title	Linear Algebra
Classification	553_0_♯_h	Caption hierarchy	Transportation and communications
Community Information	110_2_♯_a	Corporate name	United States Marine Corps
Holdings	852_♯_♯_a	Location	ScCM

Table 2: Examples of formats and fields for MARC. The format for the field's identifier is [$tag\_indicator1\_indicator2\_subfieldcode$]. The symbol ♯ stands for undefined value (resources from *Marc* [10])

MARC Field & Data	Description
100 1♯ $a Arnosky, Jim	Author
245 10 $a Raccoons and ripe corn	Title
245 10 $c Jim Arnosky	Statement of responsibility
250 ♯♯ $a 1st ed	Edition statement
260 ♯♯ $a New York	Place of publication
260 ♯♯ $b Lothrop, Lee & Shepard Books	Name of publisher
260 ♯♯ $c c1987.	Date of publication
300 ♯♯ $a 25 p.	Pagination
300 ♯♯ $b col. ill.	Illustrative matter
300 ♯♯ $c 26 cm	Size
520 ♯♯ $a Hungry raccoons feast at night in a field of ripe corn	Summary
650 ♯1 $a Raccoons.	Topical subject
900 ♯♯ $a 599.74 ARN	Local call number
901 ♯♯ $a 8009	Local barcode number
903 ♯♯ $a $15.00	Local price

Table 3: Example of a record in MARC format. ♯ stands for undefined value (resources from *Understanding MARC Bibliographic* [11])

MARC 21 is a result of the combination of the United States MARC format (USMARC) and Canadian one (CAN/MARC). A framework for working with MARC data in a XML environment is also developed. The XML schema is available at [1].

In this project, Marc 21 format is used for records of Infoscience (section 3.2) and of Nebis (section 3.1).

2.3 EndNote

Different from two previous formats, Endnote is a commercial bibliographies and references management software package [5]. The current version of the software is 10 (EndNote X). An EndNote library is a collection of references. Each reference contains necessary information for creating a bibliography (author, title, description...) and can contain additional information such as keywords, general notes or summary.

Endnote gives the possibility to do searches for bibliographical items from online databases (for example from Nebis[9] or from National Library of Finland[2]) and to import the result into a local

[1] `http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd`
[2] National Library of Finland, `http://www.lib.helsinki.fi/english/`

library. The program already included a numerous connections to online databases. Any additional connection can be etablished either manually or with configured connection files (end with ".enz"). It is important to notice that this support is an amazing feature. Because in order to communicate with these online databases, EndNote has to handle different types of server protocols and record syntaxes.

An example of a record in XML EndNote format can be found at table 4. This example was extracted from an export of Infoscience [7]. The XML Document Type Definition (DTD) of EndNote can be found at [3]. The program includes pre-defined attributes. These attributes are listed in table 14, appendix A. Moreovers, user-defined attributes are also posssible.

One particular element in the example is *Reference_Type*. This element is not listed in the attributes table. Actually, this attribute specifies the document type (book, article,...) by the mean of an integer (identifier). Similar to other attributes, the values of *Reference_Type* are predified and are illustrated in table 13, appendix A. The program also gives the possibility to define a new document type. Each documnet type is associated with several attributes. For example, a book (reference type identifier = 1) is associated with {Author, Title, Publisher,...} when a computer program (id = 12) is associated with {Programmer, Title, Year,...}.

```
<RECORD>
  <AUTHORS>
    <AUTHOR>Kostic, T.</AUTHOR>
    <AUTHOR>Cherkaoui, R.</AUTHOR>
    <AUTHOR>Germond, A.</AUTHOR>
    <AUTHOR>Pruvot, P.</AUTHOR>
  </AUTHORS>
  <TITLE>
    Decision aid function for restoration of transmission power systems:
    conceptual design and real time considerations
  </TITLE>
  <SECONDARY_TITLE>
    IEEE Transactions on Power Systems   </SECONDARY_TITLE>
  <PAGES>923 - 9</PAGES>
  <NUMBER>3</NUMBER>
  <VOLUME>13</VOLUME>
  <REFERENCE_TYPE>0</REFERENCE_TYPE>
  <REFNUM>12</REFNUM>
  <KEYWORDS>
    <KEYWORD>
      decision support systems;     </KEYWORD>
  </KEYWORDS>
  <YEAR>1998</YEAR>
  <DATE>1998</DATE>
</RECORD>
```

Table 4: Example of a record in XML EndNote format from Infoscience [7]

With Endnote, one can choose several methods for adding records into the collection. The records can be created manually by specifying the value for each field, or they can be directly imported from an online database, or from files containing data via filters. For example, with the original Endote filter, if the tag "@A" is added at the beginning of a line of a text document, this line will be recognized as a "Author" field while importing the document into Endnote library. Endnote also support custummized filters. As the program has been widely used, there is a lot of custummired filters provided by universities, governments, hospitals [4].

[3] EndNote's Support, `http://www.endnote.com/support/ensupport.asp`

[4] Endnote filters, `http://www.endnote.com/support/enfilters.asp`

In this project, EndNote format was first used for Infoscience's record retrieval. However, due to loss of information, it was switched to MARC format (previous section).

2.4 The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)

(Sources from [13])

The OAI-PMH provides an application-independent interoperability framework based on metadata harvesting. There are two classes of participants in the OAI-PMH framework: *Data Providers* and *Service Providers*. *Data Providers* manage systems that support the OAI-PMH as a means of exposing metadata and *Service Providers* in its turn use metadata harvested via the OAI-PMH as a basis for building value-added services. For example, CiteSeer [1] is a data provider, it exposes data to harvesters, and *Directory and harvester of digital resources* [5] is a service provider, as it proposes a search engin based on metadata harvested via the OAI-PMH. An official list of data providers can be found at [6] and a list of service providers at [7].

A harvester is a client application that issues *OAI-PMH requests* and is is operated by a service provider as a means of collecting metadata from repositories managed by a data provider. *OAI-PMH reponses* in well-formed XML format will be sent back to the harvester.

2.4.1 OAI-PMH requests

OAI-PMH requests are expressed as HTTP requests. They consist of the base URL of the repository (*baseURL*), of a list of keyword arguments, which take the form of *key=value* pairs. Arguments are separated by ampersands (&) and each request must have at least one *key=value* pair that specifies the OAI-PMH request, where: *key*='verb' and *value* is one of the verbs defined below. An example of a request from CiteSeer[1] might be:

```
    http://cs1.ist.psu.edu/cgi-bin/oai.cgi?verb=GetRecord&metadataPrefix=oai_
dc&identif
```

where "'http://cs1.ist.psu.edu/cgi-bin/oai.cgi"' is the *baseURL* part. After the character "'?"' is the list of keyword arguments, and the first pair is "'verb=GetRecord"' specifying the type of the request. There are six type of verbs for OAI-PMH requests:

- **GetRecord**: This verb is used to retrieve an individual metadata record from a repository. Required arguments are *identifier* and *metadataPrefix*. The second argument is a string to specify the metadata format in OAI-PMH requests issued to the repository. Many metadata formats exist, such as Dublin Core [3], Open Language Archives Community [8] and RFC 1807 [9]. However, Dublin Core is the mandatory format for OAI-PMH.
- **Identify**: This verb is used to retrieve information about a repository. No arguemnt is required for this verb.

[5] Directory and harvester of digital resources, `http://roai.mcu.es/en/inicio/inicio.cmd`
[6] Registered data providers, `http://www.openarchives.org/Register/BrowseSites`
[7] Registered service providers, `http://www.openarchives.org/service/listproviders.html`
[8] OLAC, `http://www.language-archives.org/`
[9] RFC 1807, `http://www.ietf.org/rfc/rfc1807.txt?number=1807`

- **ListIdentifiers**: This verb is an abbreviated form of *ListRecords*, retrieving only headers rather than records. The only required argument is *metadataPrefix*, and there are three optional arguments *from, until, set*. An exclusive argument (which cannot be used with others) is *resumptionToken* . The value of this argument is returned by a previous *ListIdentifiers* and can be used to continue to harvest the imcomplete list.
- **ListMetadataFormats**: This verb is used to retrieve the metadata formats available from a repository. There's only one optional argument for this verb, which is *identifier*
- **ListRecords**: This verb is used to harvest records from a repository. It has the same list of argument as *ListIdentifiers*. The use of the argument *resumptionToken* is also the same.
- **ListSets**: This verb is used to retrieve the set structure of a repository, useful for selective harvesting. The argument *resumptionToken* is also available here.

2.4.2 OAI-PMH responses

An OAI-PMH response is in well-formed XML format and must validate against the XML schema for OAI-PMH responses [10]. An OAI-PMH response can be seen as three parts:

- The first part contains the XML declaration and informations related to namespaces and validating schema.
- The second part contains a *responseDate* indicating the time and date that the response was sent, and a *request* indicating the protocol request that generated this response.
- The last part contains either an *error* element or an element with the same name as the verb of the respective OAI-PMH request. In the case of an successful reponse, this element contains the metadata for the record itself.

Table 5 shows an example for the structure of a successful OAI-PMH response. Actually, this is the response from CiteSeer for the OAI-PMH request shown above. The metadata part of the record will be completed in table 6

```
<?xml version="1.0" encoding="UTF-8"?>
<OAIPMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAIPMH.xsd">
  <responseDate>20070610T15:57:24Z</responseDate>
  <request verb="GetRecord"metadataPrefix="oai_dc"
        identifier="oai:CiteSeerPSU:1">http://cs1.ist.psu.edu/cgibin/oai.cgi</request>
  <GetRecord>
    <record>
        ...
    </record>
  </GetRecord>
</OAIPMH>
```

Table 5: Example for the structure of a successful OAI-PMH response from CiteSeer[1]

A record is metadata expressed in a single format. There are three parts in a XML-encoding of a record:

[10] XML schema for validating responses to OAI-PMH requests, `http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd`

- **header**: contains the unique identifier of the item and properties necessary for selective harvesting. It consists of the unique identifier, the datestamp, zero or more setSpec elements and an optional status attribute.
- **metadata**: a single manifestation of the metadata from an item. The OAI-PMH supports items with multiple formats mentionned earlier (Dublin Core, OLAC, RFC1807)
- **about**: an optional and repeatable container to hold data about the metadata part of the record. It consists of rights statements and provenance statements

Table 6 gives an example of the structure of a record.

```
<record>
  <header>
    <identifier>oai:CiteSeerPSU:1</identifier>
    <datestamp>19930811</datestamp>
    <setSpec>CiteSeerPSUset</setSpec>
  </header>
  <metadata>
    <oai_dc:dc xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
       http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
       <dc:title>
       36 Problems for Semantic Interpretation
       </dc:title>
       <dc:creator>Gabriele Scheler</dc:creator>
       <dc:subject>Gabriele Scheler 36 Problems for Semantic Interpretation</dc:subject>
       <dc:description>
          This paper presents a collection of problems for natural language analysis
          derived mainly from theoretical linguistics.
       </dc:description>
       <dc:publisher>unknown</dc:publisher>
       <dc:date>19930811</dc:date>
    </oai_dc:dc>
  </metadata>
</record>
```

Table 6: Example of a record from CiteSeer[1]. This part completes the third part of table 5

In comparison with other protocols, with Z39.50 (section 2.5) for example, OAI-PMH has many advantages. One incontrovertible advantage is the support of the selective harvesting method. Selective harvesting allows harvesters to limit harvest requests to portions of the metadata available from a repository. The OAI-PMH supports this method with two types of harvesting criteria that may be combined in an OAI-PMH request: *datestamps* and *set membership*. The first criterion *datestamps* can be realized with the optional arguments *from* and *until* and the second criterion with the optional argument *set* within ListRecords and ListIdentifiers requests. This method gives the possibility of retrieving an exact portion of the collection. In order to obtain the set structure of a repository, a request using the verb *ListSets* can be emploied (e.g, a collection can be organized in sets "music", "litterature" and "computer science"). We could use this method for retrieving the whole collection, but that would move away from the purpose of the selective harvesting. For doing this, the argument *resumptionToken* can be used. This will be discussed in section 3.3, when we would like to harvest the whole collection from CiteSeer.

In this project, CiteSeer (section 3.3) and the The European Library (section 3.4) implement OAI-PMH.

2.5 Z39.50 protocol

Z39.50 is a protocol which defines a standard way (i.e by specifying data structures and interchange rules) that allow a client machine (referred to as the "origin" in the standard) to search databases on a server machine (referred to as "target" in the standard) and retrieve records that meet the criteria of the search request. The first version of the starndard waas approved in 1988 by the National Information Standards Organization (NISO), an American National Standards Institute (ANSI) accredited standards developer that serves the library, information, and publishing communities.

Z39.50 commuication and information retrieval specifications are built on distributed client/server architecture. The Z39.50 server manages one or more heterogeneous and distributed databases containing records and a set of access points (indices) that can be used for searching is associated with each database. These access points are used for searching. The Z39.50 client provides end-user interaction and display. The basic services defined by the protocol are {Init, Search, Present, Scan, Sort, Delete, Close}. The description for the three commun services are as follow:

- **Init** (Connection): Z39.50 is a stateful and connection-oriented application layer protocol. It requires a reliable full-deplux byte stream transport such as TCP. At the beginning, the client and server will exchange a serie of messages to etablish a connection, initiate a Z39.50 seesion and negotiate expectations and limitations (for example the maximum number of the records that will be be trasferred, the version of the protocol supported, the options for searching, scanning, deleting etc.). This is called "the Initialization Facility" by the standard. Once these agreements are negotiated, the client may send a request.
- **Search** (Searching records): When a end-user client submits a search request (as a query form), the Z39.50 client will translate the query into a standardized representation and pass it to the Z39.50 server (defined by the Search Facility). This latter will interrogate simultaneously one or more databases and produce a set of records, called a "result set" (defined by the Retrieval Facility), that are maintained on the server. The returned result of the search to the Z39.50 client is a report of the number of records comprising the result set. The result set can be combined with another result set or further restricted by subsequent searches. That is totally different from SQL servers, which do not employ result sets.
- **Present** (Retrieving records): Records from the result set can be subsequently retrieved by the Z39.50 client using "present" request. This request offers options for controlling the contents and format of the records that are returned (for example "usmarc" or "opac"). The client may process the records and display them to the end-user.

Z39.50 has a number of advantages. The protocol allow simultaneous searches on distributed and heterogeneous databases. It separates the user interface on the client side from the information servers, serach engines, and databases. Moreover, Z39.50 can be implemented on any platform. Because of this, the protocol enables different computer systems (with different operating systems, hardware, search engines, etc.) to interoparate and work together.

However, as mentionned in section 2.4, this protocol does not support the selective harvesting method. It means that the protocol does not giving the possibility to retrieve an exact portion of the collection by specifying one or more criteria. Another disadvantage of this protocol is the lack of the information about the time at which the record was inserted into the collection.

2.5.1 Z39.50 softwares

The list of commercial and free Z39.50 softwares can be found at [11]. A complete software which also implements Z39.50 client is EndNote. In this project, many searches on Nebis catalog have been performed using EndNote program.

Actually, many commercial and open source projects related to Z39.50 exist: ZOOM, YAZ and VBZOOM are two of them.

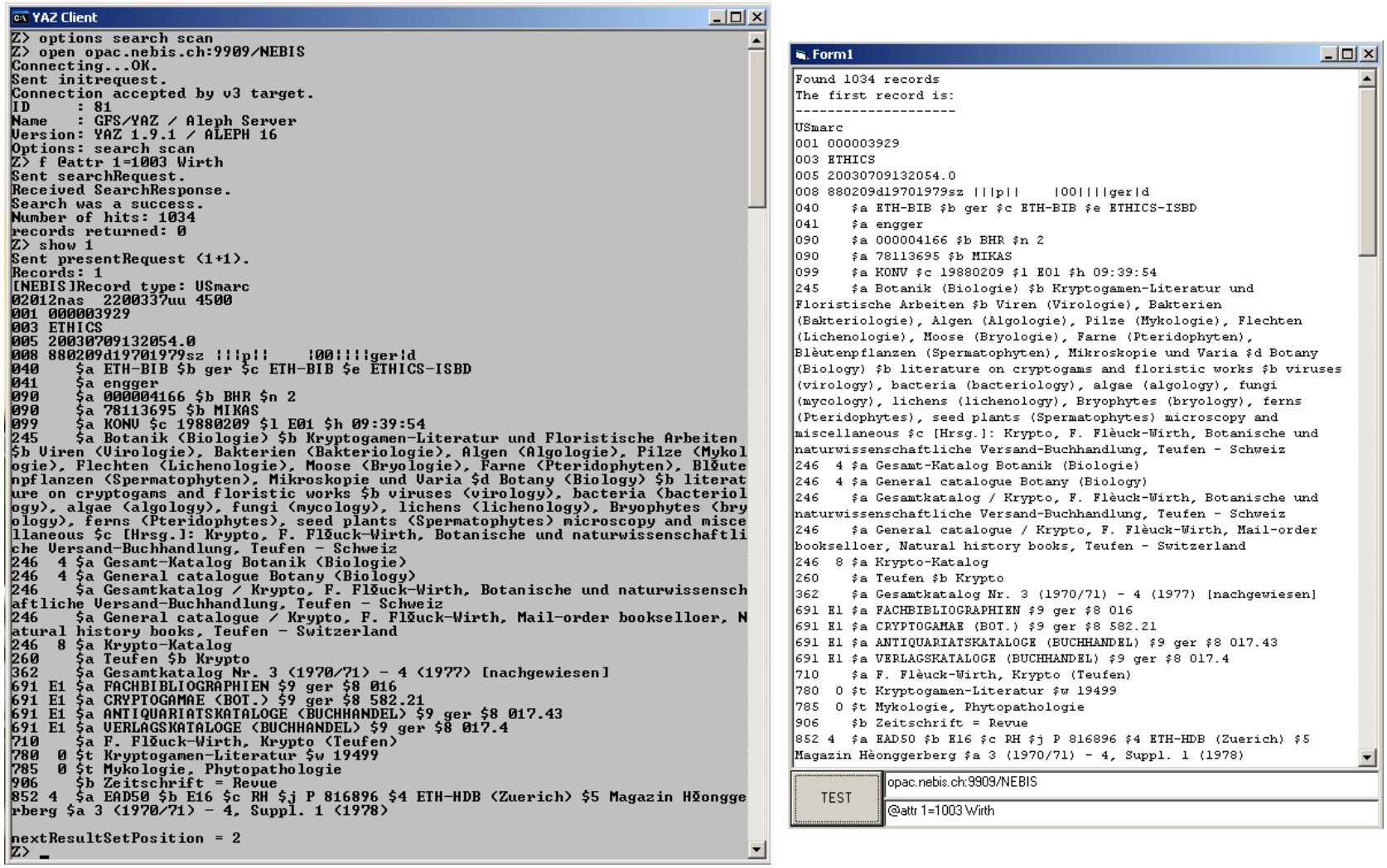
The ZOOM initiative [18] presents an abstract object-oriented API to a subset of the services specified by the Z39.50 standard. ZOOM can be considered as a part of the larger ZING initiative (Z39.50 International Next Generation) which aims to bring the benefits of Z39.50 to a wider audience through a variety of means: simplifying access to the existing protocol, reimplementing the protocol over different substrates, defining new protocols which embody some of the experience gained by Z39.50 workers, etc. (source [18])

YAZ of Index Data [17] is a programmers toolkit supporting the development of Z39.50 clients and servers. YAZ includes support for the industry standard ZOOM API for Z39.50. This API simplifies the process of writing new clients using YAZ, and it reduces your dependency on any single toolkit. YAZ can be used by itself to build Z39.50 applications in C (source [17]).

VBZOOM is a collection of ActiveX COMponents, written in Visual Basic, which implement the ZOOM Abstract API. The current VB ZOOM is a wrapper for the YAZ Toolkit from Index Data, plus a helper component for doing MARC-8 to Unicode character conversions. Even though the components are written in Visual Basic, they should also work from C++, Perl, Delphi, or any other programming language on the Windows platform that supports ActiveX COM Objects (source [16]).

An example of YAZ and VBZOOM is given in the figure 1. In this example, a connection to the Nebis Z39.50 server is etablished. Within YAZ client, the command "open opac.nebis.ch:9909/NEBIS" is used where "open" is the command, "opac.nebis.ch:9909" is the URL of the server and its port, then "NEBIS" is the name of the database. In VBZOOM test client, only the server's URL and port are required. The next step is about seaching the records whose author is "Wirth". The command line for YAZ client is "f @attr 1=1003 Wirth" where 1003 indicates the attribute "author", and the returned rapport shows that there are 1034 records that matched that criteria (line "Number of hits: 1034"), then the command line "show 1" performs the retrieval for the first record and display it in Usmarc format. In the VBZOOM test client, only the argument "@attr 1=1003 Wirth" is required then the rapport is displayed followed by the first matched record.

[11] List of Z39.50 softwares, `http://www.loc.gov/z3950/agency/resources/software.html`



(a) YAZ (b) VBZOOM

Figure 1: Example for Z39.50 clients. The figure illustrates the connection to the Nebis Z39.50 server, a search for records whose author is "Wirth" and a display of the record

3 Digital libraries

In this section, we would like to discuss about libraries whose databases are retrieved and analyzed in this project. The description of each library is first given, followed by services provided by each of them.

3.1 NEBIS

The Network of Libraries and Information Centers in Switzerland (NEBIS) comprises of over eighty university, technical college and research institute libraries from all language regions [12]. NEBIS is a member of the Informationsverbund Deutschschweiz IDS. The NEBIS catalog contains about 3 million titles, including books, serials, journals and non-book materials. Most documents may be ordered online. To borrow library materials you must register at one of the NEBIS libraries [9].

NEBIS implements the Z39.50 protocol. The website of Nebis provides several types of critera for searching records. Records can be searched by keywords either in all commun fields (title, author, etc.) or in one of them, and keywords can also be concatenated. Then, the search can be furthur limited by other criteria, such as the language, a specified library, the years from which and to which a record is published, and the document type. Each record in NEBIS belongs to a document type. The list of these types is provided in table 7. Each of theses document types comprises As illustrated in table 13, appendix A,

Articles	Graphic materials	Journals
Atlases and maps	Transparencies	Multimedia items
Collected works	Pictures	Games
Commemorative works	Slides	Language courses
Dissertations, theses	Photographs	Newspapers
Dissertations (ETH)	Laws, constitutions	Printed music
Electronic documents	Microform	Sound recordings
CD-ROM	Motion pictures	CD
DVD-ROM	DVD-Videos	DVD-Audio
Online documents	Videotapes	Tape cassettes

Table 7: Nebis document types. List extracted from the website http://www.nebis.ch

The NEBIS website [9] also provides an *Expert Mode* for searching records. This mode uses the Common Command Language (CCL) whose the codes are specified and given at the NEBIS website. In this mode, the command for searching a record whose the title contains "Computer Network" and the year of publication is 2006 might be *"WTI=Computer Network AND WYR=2006"*. The results can also be restricted by the same criteria like the previous seach mode.

The list of records which matched the search criteria is returned. When one of records in the list is selected, the full view of that record will be displayed. By default, the "Standard format" is selected, where fileds are in natural language (Title, Contents, Subjects, etc.). Moreover, other display modes are possible. NEBIS supplies "Catalog card", "Citation" and "MARC tags" display modes. However, the last mode does not display all MARC fields in their original version (please refer to section 2.2 for the format of the fields). For example, the field "001_#_#_#" is displayed as *"SYS"* in this mode.

As mentionned earlier, NEBIS implements Z39.50 protocol, hence it has Z39.50's advantages and disadvantages. An advantage is the network comprises of over eighty distributed and heterogeneous

[12] Nebis libraries, `http://www.nebis.ch/bibliotheken_e.html`

databases. NEBIS records can be searched and retrieved using Z39.50 clients. Hovever, there is a limited number of of records returned for each search request. For instance, this limit is 5000 records per search. In the same way, NEBIS does not support selective harvesting method and does not give the possibility to download the full-text version of several types of document yet (for example thesis and articles).

3.2 Infoscience

Infoscience is a database of the publications, research reports, PhD Theses, Master Theses, semester works, lectures, etc., of the Faculties, Laboratories and Researchers at the EPFL. The goals of Infoscience project are clearly defined from the beginning (source from [7]):

- Access to scientific resources: The purpose of the infoscience project is to facilitate access to scientific resources produced at EPFL: publications, preprints, research reports, projects, theses, students work, courses, posters.
- Access to the central catalogue: Some of these assets consist of the collections of books in EPFLs libraries. The Infoscience project intends to offer a central catalogue of these resources that supplement those produced in the Institute.
- Access to data about people: *people@EPFL* is a single interface letting all members of the community publish their CV, describe their projects, spotlight their important publications, add personal content, etc. Its use is suited to the requirements both of individuals and of groups: a lab, a service, a group or an association can use it to set up its portrait gallery simply and elegantly.

Infosciene still is in process. For instance, Infoscience counts more than 60 thousands documents in its database. There are more and more laboratories at EPFL choosing Infoscience as the main tool to archieve and publish their scientific resources. Resources integrated in Infoscience acquire a greater visibility, readability and analysis. This is because Infoscience does not only aim to centralize the information, but it also aims to offer a range of facilities of use to all members of the scientific community.

The main website of Infoscience [7] provides a search engine for its records. The user interface offers a large number of options for searching and retrieving records. The input criteria can be compared with any field, or only with a single field (author, title, abstract, keyword, report year, fulltext or reference). Then the result can be once again restricted to a type of document or filtered either by publication status, by origin or by fulltext availability. The list of document types used in Infoscience is given in table 8. Each document type comprises of a certain number of attributes. This is quite similar to Endnote's document types.

Publications	Monographs	Presentations & Conferences
Journal articles	Books	Posters
Conference Papers	Thesis	Presentations & Talks
Lectures & Teaching Material	Reports	Standards & Patents
Lectures	Book chapters	Standards
Teaching Documents	Proceedings	Patents
Student Projects	Reviews	

Table 8: Infoscience's document types. List extracted from the website http://infoscience.epfl.ch

Results of a search request are first diplayed and grouped into researcher's profiles and document types, followed by the details for each record. In addition to options for searching, another strength of Infoscience is the integrated features for displaying and exporting results. They can be sorted by different criteria (year, title, etc.) and can be split into lists or collection, and the last but not least option for displaying is the output formats. Infoscience supports a large number of output formats, such as HTML brief or detailed, HTML MARC, XML Dublin Core and XML MARC.

Morover, Infoscience also allows records to be exported, either in Endnote format or in BibTeX format. The first option was adopted at the beginning of this project for retrieving records from Infoscience. However, due to loss of information in some case, this option was then replaced by MARC format.

Enabling these formats are important from more than one perspective. An user would be delighted to find his usual output format, or, the resource could be directly imported into a reference managament software (e.g Endnote) rather than manually import from field to field.

There are still two other advantages of Infoscience that would be worth to mention. The first one is the possibility to import Infoscience interface into a website or into Jahia [13], and the second one if possibility to download the fulltext version of a document (when it is available and it may be with access reserve). As discussed above, the latter option is a filter while searching or displaying results, beacause not all records provide a fulltext version. Furthur, we can see that this possibility can be indentified by the presence of the a not-null value for the local attribute whose identifier is 52, name is "marc21_infoscience_filename" and the full identifier is "856_40_u". This attribute corresponds to an item in the local *Attributes* table (MySQL). Infact, the attribute contains the URL to the fulltext (usually in PDF format).

3.3 CiteSeer

(Source from [1])

CiteSeer is a scientific literature digital library and search engine that focuses primarily on the literature in computer and information science. CiteSeer aims to improve the dissemination and feedback of the scientific literature and to provide improvements in functionality, usability, availability, cost, comprehensiveness, efficiency, and timeliness in the access of scientific and scholarly knowledge. CiteSeer was developed in 1997 at the NEC Research Institute, Princeton, New Jersey, by Steve Lawrence, Lee Giles and Kurt Bollacker. The CiteSeer model was used to create a similar search engine, SmealSearch, for academic business documents. CiteSeer is also a non-profit service that has been considered as part of the open access movement that attemps to improve access to scientific literature by changing the method that the ressource is published.

CiteSeer was the first digital library and search engine to provide automated citation indexing and citation linking using the method of autonomous citation indexing (ACI). An ACI system can automatically create a citation index from literature in electronic format. Such a system can autonomously locate articles, extract citations, identify citations to the same article that occur in different formats, and identify the context of citations in the body of articles. The viability of ACI depends on the ability to perform these functions accurately.

Operating completely autonomously, CiteSeer works by downloading papers from the Web and converting them to text. It then parses the papers to extract the citations and the context in which the citations are made in the body of the paper, storing this information in a database. CiteSeer

[13] Infoscience's user documentation, Retrieving the data, `http://infoscience.epfl.ch/doc/Retrieving_the_data`

includes full-text article and citation indexing, and allows the location of papers by keyword search or citation links. It can also locate papers related to a given article by using common citation information or word similarity. Given a particular paper, CiteSeer can also display the context of how subsequent publications cite that paper. For instance, the site announces more than 760'000 documents.

Because CiteSeer is compliant with the Open Archives Initiative Protocol for Metadata Harvesting (section 2.4), it was choosen as an illustration of the protocol.

3.4 The European Library

The European Library (TEL) is a web service of Europes national libraries that offers access to the ressources of 47 national libraries. The ressources can be both digital or bibliographical (books, posters, maps, sound recordings, videos, etc.) and currently are in 20 languages. Its vision is clearly pronouced through the statement: "*Provision of equal access to promote world-wide understanding of the richness and diversity of European learning and culture*"[6].

Currently, the website gives the possibility to search through the resources of over 30 national libraries. According to the FAQ section[14], TEL contains around 250 collections of which around 150 are searchable and the rest are browse-only. A browse-only collection is not currently searchable through the TEL website and is often available on its own website.

The collections analyzed in this project was provided by Ms. Jill Cousins, director of The European Library. Actually, they represents only a portion of the accessible collections from TEL. Some informations about these collections are shown in table 9. According to Ms Cousins, the ressources are not centrally stored. Online databases are remotely interrogated via SRU (Search & Retrieve via URL) and Z39.50 protocols. Collections supporting OAI-PMH may be harvested and indexed centrally so that they can be accessed in a more efficient way. That's the case of the collections obtained in this project.

3.5 Licenses

One of the objectives of this project is analyzing the content of these libraries, and naturally, for doing this, the local copies are required. Before retrieving any data, we should find out if the data is protected, i.e consult their access right. Further, the same question should be addressed if the data is supposed to be distributed.

Nebis : According to Mr. David Aymonin, Director of Scientific Information and of the Central Library of EPFL, and to Mr. Egloff, lawyer, for instance there is no swiss law by which databases are protected in term of copyright. Therefore, for a private usage and when technically possible, everyone can legally retrieve data without any demande of authorization. One does not necessarily need to belong to an academic environment in order to benefit this advantage.

In the same way, it would also be possible to share the retrieved database partially or integrally, and this is done in a private circle (within the laboratory) or in a public one (public access to a website). However, even if the swiss law does not forbit this kind of action, it would be preferable to ask database's administrator or author for the permisson of sharing or publishing the

[14]FAQ section of The European Library, `http://libraries.theeuropeanlibrary.org/services/faqanswer_en.html`

content. Always accroding to Mr. Aymonin, an explicit request for the authorization should be addressed to the administrator or author in the case of making a commercial application using the database.

Therefore, in our case, it is entirely conformed to the swiss law to retrieve and perform an analysis for the Nebis collection.

Infoscience : As there is no additional specification of the copyright for the items, the same rule as Nebis aslo applies for Infoscience. Actually, when a user wants to add a full text version of his work to Infoscience database, Infoscience requests that it is the user's responsibility to make sure of the legal status of his work[15]. We have mentionned about the possibility of downloading the full text version of an item, which is indicated by the value of the attribtue "marc21_infoscience_filename" ("856_40_u"). However, the access to this file can be restricted and is specified by the user. The attribute "marc21_infoscience_document_type" ("856_40_x") indicates the access type for a given item ("public", "restricted", etc.)

CiteSeer : The content of the CiteSeer collection originates from the papars on the Web and the team shares this content by making it fully public. According to Dr.Lee Giles, responsible for the CiteSeer project, who is the David Reese Professor at the College of Information Sciences and Technology at the Pennsylvania State University, there is not any restriction in the use of the database.

The European Library : For instance the database is not publically open for harvesting or downloading. As mentionned earlier, the TEL collection was provided in XML documents by Ms. Jill Cousins, director of The European Library, for the purpose of the analysis of this project. The content cannot be either shared or distributed without an explicit request to Ms Cousins and her authorization.

[15] Infoscience's Copyright, `http://infoscience.epfl.ch/doc/Copyright`

Collection	Number of records	Language	Format
BNCF\arsbni1	5060	Italian	Dublin Core
BNCF\arsbni2	54964	”	”
BNCF\bertini	687	”	”
BNCF\europe	1	”	”
BNCF\manoscrittiinrete	33	”	”
BN\belasartes	30638	Portugese	TEL
BN\biblias	1083	”	”
BN\bibliografias	12900	”	”
BN\bnd	8204	”	”
BN\cartografia	6025	”	”
BN\cienciasartes	36000	”	”
BN\cienciassociais	141633	”	”
BN\espolios	171	”	”
BN\historiageografia	84378	”	”
BN\iconografia	28981	”	”
BN\impressosreservados	5279	”	”
BN\leituraespecial	3386	”	”
BN\literatura	168141	”	”
BN\manuscritos	3133	”	”
BN\musica	18535	”	”
BN\religiao	39660	”	”
BN\seriegeral	33600	”	”
BN\teses	44586	”	”
BNpol\polona	1697	Polish	Dublin Core
BnF\gallica	96070	French	”
NBS\decije	127	Serbian	TEL
NBS\doi\serbia	1110	”	”
NBS\pozorisni	559	”	”
NBS\svetogorska	127	”	”
NKP\kramerius	925	Czech	Dublin Core
NKP\manuscriptorium	83894	”	”
NUK\slobib	77932	Slovinian	”
ONB\bildarchiv	7849	Austrian	TEL_onbba
ONB\chmel	13	”	”
ONB\none	27420	”	”
ONB\portraet	43	”	”
ONB\rubeltBildDaten	4424	”	”
ONB\rubeltNegativArchiv	0	”	”
ONB\usis	15912	”	”
OSZK\HEL	4601	Hungarian	Dublin Core
OSZK\nda	514625	”	”
OSZK\corvina	35	”	dcx
OSZK\map	102	”	”
RR\digar	1621	Estonian	Dublin Core

Table 9: Collections provided by The European Library. They represent in total 1566164 records. Their size is around 1.82 GB (5.26 GB on disk)

4 Implementation and Analysis

This section will discuss first about the programming aspects, i.e. the programming languages and how they were used in order to harvest different databases, to parse from the original format into the required one, to insert information into local database and to analyze it. Then we'll discuss about the metrics and the results of the analysis. The Java source code along with the data retrieved from online databases are provided in the DVD support.

4.1 Programming Languages

4.1.1 Java

Java is the main programming language for this project. It was choosen beacause of its portability, available libraries, online tutorials and documentation. The official support center at Sun's website continously maintains and updates the documentation, and an incontrovertible advantage of Java comparing to other languages is its highly active community. Almost all answers to techinical problems in this project were found at the Java's official forum [16]. Along with Java's basical functions, two main technologies required are:

- Java API for XML Processing (JAXP)

  In order to retrieve information from XML documents, the JAXP API was used. It provides a common interface for creating and using the standard SAX, DOM, and XSLT APIs in Java. This API has the advantage that it does depend on vendor's implementation. This project uses the Document Object Model (DOM) approach (instead of Serial Access with the Simple API for XML (SAX)). A DOM has a tree structure, where each node contains one of the components from an XML structure. Each node has a type, such as *element* or *text*. DOM functions allow to create nodes, delete nodes, change their contents, and traverse the node hierarchy. We only need here the last option. The packages *javax.xml.parsers.DocumentBuilder* and *javax.xml.parsers.DocumentBuilderFactory* were first used to obtain the DOM instance from the XML document. Then, the DOM ,defined by XML-DEV group and by the W3C (package *org.w3c.dom*), is processed as a object tree(with Node, NodeList).

- Java Database Connectivity API (JDBC)

  Once necessary data is retrieved from the source document (in XML or other format), it will be inserted into the local SQL-based database. For doing this, JDBC API was used. This API allows to establish a connection with a database, to send SQL statements and to process the results.

4.1.2 MySQL

MySQL [12] is the most popular open source and SQL-based database. The project's database is located at the MySQL server of the IC faculty[17].

[16] Java Technology Forums, `http://forum.java.sun.com/index.jspa`
[17] IC faculty's MySQL server, `icmysql.epfl.ch`

4.2 Data retrieving

The architecture for the implementation of this part is given in the figure 2.

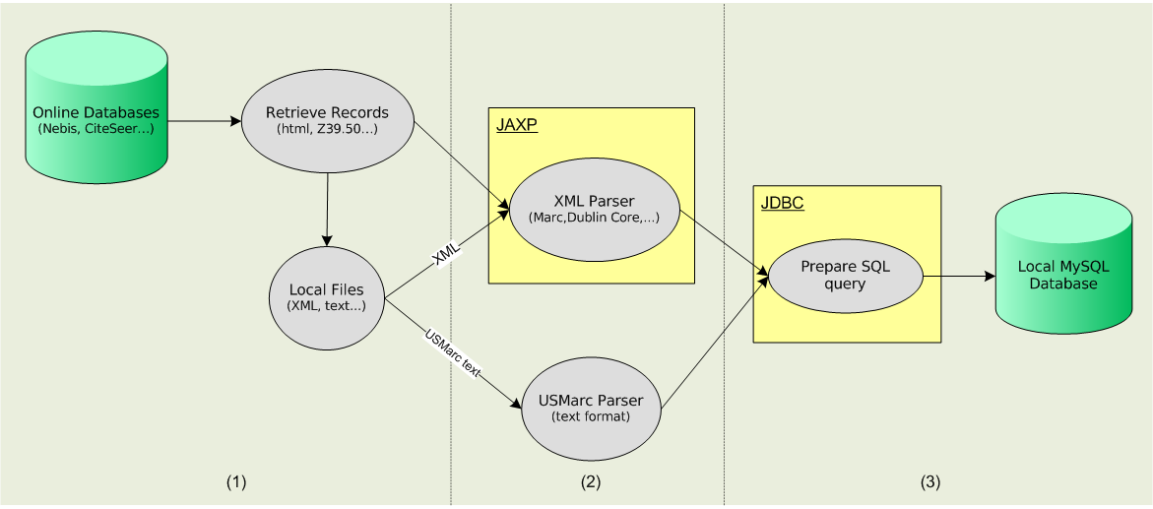


Figure 2: Architecture for the implementation is divided into three parts: (1) Connect to the online databases and retrieve records, (2) Parse the retrieved records (3) Insert the records into local database

The first part of the architecture consists in obtaining the resources. These resources could be located at online databases (like the cases of Nebis, Infoscience and CiteSeer) or in local files (The European library). Except for Nebis, all retrieved documents are in XML format. In this part, we have to deal with different server protocols in order to make a connection, send request and receive the response. The main difficulty here is how to construct the correct request to able to obtain the corresponding record, and which argument of the request can be used to apply a loop process. As mentionned before, OAI-PMH supports selective harvesting and full collection harvesting. Because of this advantage, the main task in this part was straight forward for CiteSeer, while the empirical methods have been performed for Nebis and Infoscience. This difficulty did not concern the European library, because the data was already supplied in local XML files.

The second part consists in parsing the retrieved documents (using JAXP API if the document is a XML format) and in constructing the corresponding SQL queries in order to insert data into the database. The main idea is to study in detail the structure of each library's data format, then create the appropriate parser. This approach is quite empirical. For certain library, different parsers were created because the library supports more than one data export format, and at the beginning, we are not sure which format to use.

There was an attempt of using JAXB for parsing the XML document. Using JAXP is supposed to have a more structured and clearer approach. However, this approach requires a well-defined XML schema, and it also requires that the XML documents are totally conformed to that schema. Although both MarcXML and Dublin Core propose an official XML schema, but in reality, the XML documents retrieved from the online databases are rarely conformed to it, or inside these documents, other structures are used so that the JAXB approach quickly becomes too complicated for a quite simple task. That's why JAXP remains the convenient choice.

The third part consists in inserting data into the local SQL-based database using JDBC API. For each library, a separated table with the library's name is created. Hence, there are four tables containing data of the records: Nebis, Infoscience, CiteSeer and EuropeanLib. Although it is the matter of different sources (with different formats), a global structure for all record's tables has been proposed [18] and is shown in figure 3.

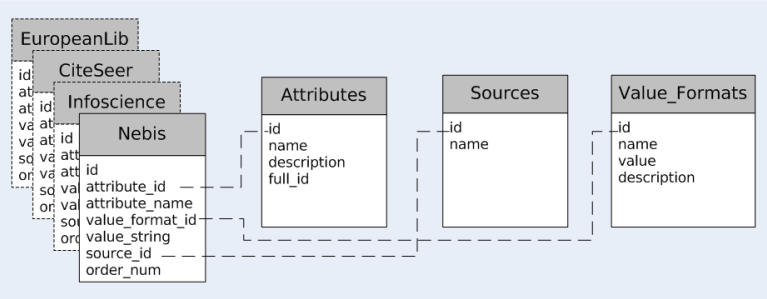


Figure 3: The SQL-based database's tables

The main idea is to create a flat table structure which can be used for all libraries. A new record retrieved from the online library will obtain a internal identifier (column *id*), which is simply the current largest identifier's value in the table incremented by 1, when this record is being inserted into the database. Then all fields of this record will have the same identifier (*id*) but will be specified by their attribute identifier (*attribute_id*). The only role of *attribute_name* is for an easier readability, and *value_string* constains itself the content of the field. The sequence order of all fields in a record is also stored by the mean of *order_num*.

In parallel, three other tables Attributes, Sources and Value_Formats are required. As their name indicates, the first table contains information of all attributes used in the database, and the second table the source of the data, i.e the name of the library. It is important to mention that as we have different data export formats (USMarc, Dublin Core, etc.), hence there are different types of attributes. In general, the prefix of the attribute name gives the type of data format. For example, attribute dc_ is a Dublin Core type, marc_ is a MARC type. Also, when available, other details such as description, fullname, are given for each attribute.

At the beginning of the processing of each library, the current Attributes table is somehow loaded into local hashtables, it permits to minimize the database access. As the attribute identifier is required while inserting the data into the database, it is necessary to first check if the attribute exists in the table Attributes, by the mean of local hashtables, then to request for its local identifier before any insertion. If the attribute does not exist, it will be first added into the table Attributes, local hashtables will be updated, then the new local identifier will be returned and used for the insertion.

The table Value_Formats is quite particular. It aims to give more specifications for the content of an attribute. For example, if a book is writen in English, then which of "en", "eng" or "english" would the value of its attribute "language" be? It would be necessary to specify the convention in order for the content of an attribute to be analyzed in a more efficient way. This table could be considered as an ontology's table.

An example of the structure is given in table 10. The example shows certain fields of a single record retrived from Infoscience. The record has a identifier 3, and its fields are then specified by the

[18] Thanks again to David Portabella and Miroslav Melichar for their help on the structue

id	attribute_id	attribute_name	value_format_id	value_string	order_num	source_id
3	3	marc21_document_type	0	CONF	1	1
3	1	marc21_unit	0	LASEN	3	1
3	16	marc21_author_name	0	Babusiaux, D.	4	1
3	16	marc21_author_name	0	Gnansounou, E.	5	1
3	16	marc21_author_name	0	Percebois, J.	6	1
3	18	marc21_title	0	Energy Vulnerability: the right indicators	7	1
3	23	marc21_publication_year	0	2007	8	1

Table 10: Example of a record retrieved from Infoscience in the local SQL-based database

attribute's identifiers. The example also illustres the sequence order for these fields. That means, although there are three authors in the example, we still can know exactly which of them is the first, second or the third author. This feature is also available for other attributes, but in reality, attribute like "publication year" appears rarely more than once in the same book. All values of *value_format_id* is 0 because the format was not specified. And, *source_id* indicates the source of the record, here Infoscience.

Except Nebis, these three tasks are performed within a separated class for each library (e.g, "Infoscience.java" performs all task for the Infoscience library), and all Java classes are managed by class *Digilib* ("Digilib.java") containing the main method. The only role of this main method is to create instances and to call their methods at the right moment.

Besides, the class *MySQL* ("MySQL.java") contains all necessary JDBC methods for the projects. One instance of this class is created within the main method and all other classes use this instance when they want to perform any database-typed action. This ensures that every class deals with the same and right database, and, the connection only needs to be etablished once, in the main method. Methods in *MySQL* allow to etablish the connection to the database (*getMySQLConnection()*), to create a statement in order to issue requests and to receive the set of results (*createStatement()*), and they also allow to close the connection (*close()*). As we use two types of queries, we need to create two methods. The first type of queries returns either the number of affected rows or nothing (*Insert, Update, Delete, Create Table, Drop Table, Alter Table* statements) and is defined by the *executeUpdate()* method. The second type concerns only the *Select* statement, which returns a set of results. This type is handled in the *executeSelect()* method.

The following will give some more details about the implementation of each library.

4.2.1 Nebis

The first part of the architecture shown in figure 3 could not be realized in Java for Nebis. This is due to the lack of a convenient API supporting the Z39.50 protocol. The only Z39.50 API in Java we could find is JZKit developped at Knowledge Integration [19]. However, the solution has a commmercial support, and it is very complete and complexe at the same time. Due to time limit, we had to find another solution.

The VBZOOM discussed in section 2.5 has been emploied and the schema is illustred in figure 4. Using its "Dynamic Link Library" (dll), a Visual Basic Script *Nebis.vbs* has been implemented in order to accomplish this first task. Within the script, an instance of the object *ZoomFactory*

[19]Knowledge Integration, Open Source Solutions for Libraries, Education and Information Management, `http://www.k-int.com/`

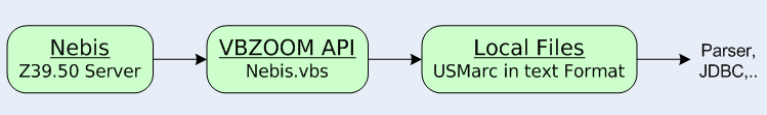


Figure 4: Schema for the Nebis records retrieving task using VBZOOM. The records are retrieved into local files and are in USMarc text format

etablishes a connection with the Nebis server via the URL `opac.nebis.ch` and the port 9909. Then, the attribute *12* which corresponds to the *Local number* is emploied to obtain the required record. The meaning of this attribute was found at [20]. One record is retrieved at once and is stored into a local text file named by its *Local number* (e.g, the file "14.txt" corresponds to the 14th Nebis record). By the mean of the variable *counter*, Nebis records are iterativement retrieved. This process accomplished with the following code lines:

```
Set zoom = WScript.CreateObject("VBZOOMC.ZoomFactory")
Set zconn = zoom.CreateZoomConnection("opac.nebis.ch", 9909)
zconn.SetOption "databaseName","NEBIS"
zconn.SetOption "preferredRecordSyntax", "USmarc"
...
Set zquery = zoom.CreateZoomQuery("@attr 1=12 " &counter)
```

The next steps are to parse these files and import them into the database that are included in the file "Nebis.java". The structure of all text files are similar and quite simple, that simplifies the task for the parser. The first three lines of the file contains configuration's information (such as the local number) and their format is *tag value*. The remaining lines follow the pattern *" tag indicator1 indicator2 ($subfieldcode value)$^+$ "* where the term $()^+$ means that the group can have one or more instances. An example for the content of a Nebis record is as follow:

```
001 006519424
005 19990902153100.0
008 990902n xx und d
999    $a Bestellhilfe zu Signatureintrag
852 4  $b E19 $j 33178 $4 SOZARCH (Zuerich)
```

Then, for importing into the database, we apply the common procedure discussed earlier.

4.2.2 Infoscience

The class *Infoscience* ("Infoscience.java") ensures the processing of the library Infoscience. The entire collection was once retrieved and analyzed in Endnote format. However, due to loss of information for the records because not all fields are exportable in Endnote format from Infoscience, we have switched to MARC21 format. Methods supporting these two formats are included in the class. When studying the table with Endnote format, we could notice that Infoscience's records did not employ all Endnote's reference types (and attributes). The list of these reference types and attributes is shown in table 15, appendix A. This list is issu from an analysis of the collection at a given moment, as Infoscience is in process, the possibility for this list to increase (or simply to vary) is not excluded at all.

Once again, the first task was done with an empirical approach, as we could not find any information about the Infoscience's architecture. We have studies requests issu from the Infoscience's website [7]

[20] Aleph Z39.50 server in the NL CR, `http://sigma.nkp.cz/web/Z39_NK_eng.htm`

while searching for items. As discussed earlier, Infoscience supports a lot of searching and exporting features, and we could profit these advantages. We noticed that the HTML request

> http://infoscience.epfl.ch/search.py?cc=Infoscience&as=1&ln=en&p1=&f1=
> &action=Search&sf=&so=d&rg=1&jrec=60845%20&sc=0&c=&of=xm

returned the last item in the collection. The pair *jrec=60845* is important here because 60845 corresponded to the documents counter displayed in the website at that moment, and when we replace that pair by *jrec=60846*, no document was returned. Hence, we deduced that *jrec* corresponded to the internal Infoscience document counter. Two other interesting fields are *rg=1* and *of=xm*. There is a limit for the number of records returned to a request, and naturally Infoscience does not allow to exceed that limit. Indeed, this number is specified by the field *rg* (i.e range) and the maximal value that Infoscience allows is 400. The field *of* corresponds to "output format", here, we expect it in MarcXML. With *jrec* as the starting point and *rg* as size of returned records set (respectively *jrec* and *range* in the Java code), the HTML request is contructed (figure 5) and the Infoscience collection is iterativement retrieved.

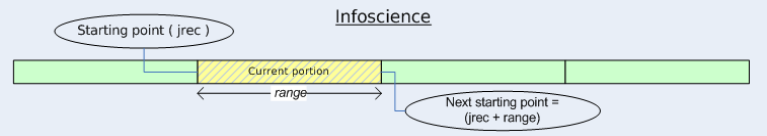


Figure 5: HTML request is constructed by the mean of *jrec* as the starting point and *rg* (range) as the size of the returned records set

As explained before, in the second task, the XML document will be parsed using JAXP API. We have two possibilities for building the DOM. One consists in storing the XML reponse in local file, then this file will be processed to build the DOM. Other possibility is to built directly the DOM from the URL (i.e HTML request). The advantage of the first approach is at the end, the collection will be stored in local files for any further study, and the advantage of the second approach is there are less steps to process, hence it reduces time and memory space. The second approach has been adopted. Once the DOM is built and knowing the structure of MARC format, the tree (DOM) is scanned in order to retrive the fields (composed by tag, indicator1, indicator2 and subfieldcode) and the corresponding values of the record. Then the data will be imported into the database, which is the objective of the third task.

The procedure is illustrated in figure 6. This procedure is repeated until the size of the reponse set is equal to 0, which means *jrec* has exceeded the number of documents wihin Infoscience collection.

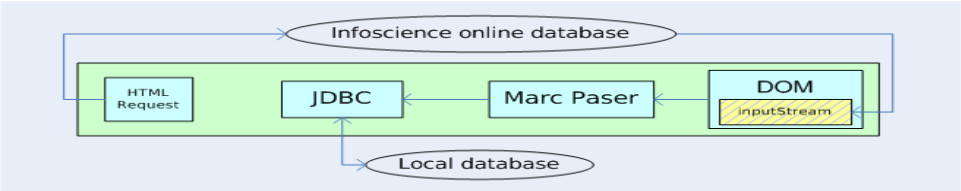


Figure 6: This figure illustrates the procedure for the second and the third tasks of Infoscience

4.2.3 CiteSeer

CiteSeer is compliant with the Open Archives Initiative Protocol for Metadata Harvesting, the procedure for searching and retrieving records is very well defined.

A friendly graphical user interface (GUI) was created (figure 7) for the purpose of choosing the type of verbs, gathering values for the required arguments and issuing all kind of OAI-PMH requests. This GUI corresponds to the *GUI.OaiGUI* class. By entering the URL of the OAI collection repository in *baseURL* text field, we can access to the specified collection. In this case, the CiteSeer's OAI-PMH base URL "`http://cs1.ist.psu.edu/cgi-bin/oai.cgi`" is inserted.

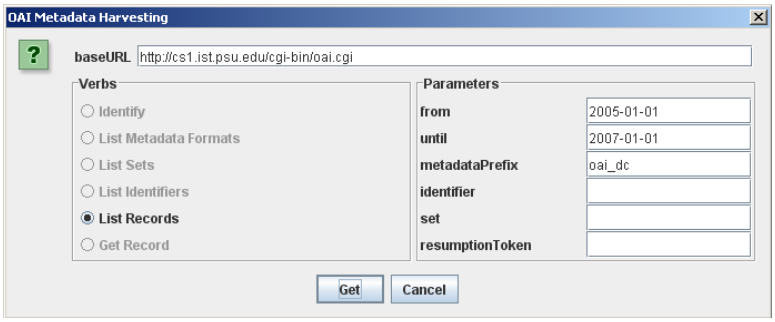


Figure 7: A simple OAI-PMH graphical user interface. It supports all types of verbs and their required arguments

The GUI also proposes all kinds of OAI-PMH verbs. Howerer, because in this project, we are only interested in harvesting the whole collection, so only the *ListRecords* verb is activated. If necessary, other verbs can be easily activated within the Java code. For reminding, each OAI-PMH verb requires a number of specific arguments, and they can also be optional. In the example in the GUI, for the *ListRecords* verb, the values of *from, until* and *metadataPrefix* are specified. But in our case, in order to harvest the whole collection, the two first arguments *from, until* would be empty. The very first OAI-PMH request issued from the application is:

`http://cs1.ist.psu.edu/cgi-bin/oai.cgi?verb=ListRecords&metadataPrefix=oai_dc`

This response to this request is a set of records, corresponding to a portion of the collection. The existence of the remaining portions is indicated by the presence of a not-null value for the *resumptionToken* argument. An example for this value is "!!!1001!oai_dc". We suppose that the meaning of this value is that the first record of portion is the 1001st record in the collection and the metadata is in Dublin Core format. Therefore, the next portion of the collection then can be harvested by issuing a new request:

`http://cs1.ist.psu.edu/cgi-bin/oai.cgi?verb=ListRecords&resumptionToken=!!!1001!oai_dc`

By this analogy, the CiteSeer's collection can be iterativement harvesting, portion by portion. The only different comparing to other libraries is this iteration is managed within the *main()* method of the *DigiLib* class. After a portion of the collection is processed by the *listRecords()* method of the *OAI* class, this method will return the value of the *resumptionToken*. If this value is not null, then the iteration will carry on, otherwise, the collection has been entirely retrieved and the iteration stops.

We have just described how the first task was accomlished. The two remaining tasks, i.e parsing and importing into the database, follow the rules discussed earlier. Nevertheless, as the Dublin Core is

mandatory for OAI-PMH, CiteSeer was harvested with this format. For instance, the parser works only with DC format, but the structure for other format (RFC1807, OLAC) has been prepared, but not implemented yet. All Dublin Core attributes have the prefix "dc_" in *Attributes* table. Similarly, as an OAI-PMH response consists in three parts: header, metadata and about, attributes of the first and the last parts are prefixed respectively "oai_header_" and "oai_about_".

4.2.4 The European Library

The collection of the European Library is provided in XML files (more than 1,5 million files). All the sub-collections, corresponding to a folder, have been considered as different sources and processed one by one. The principle format used in the library is Dublin Core, yet, other formats are also found and handled.

For each folder corresponding to a sub-collection, its name will be given in argument for the *processFolder* method of *EuropeanLib* class when called by the main method. All the XML files under this folder will be processed one by one. To prevent the files from being executed more than once, after each successful process, the current file will be removed from the folder. However, as we handle with a huge number of files, it would be necessary to keep track of processed and removed files by having a log storing their information, or at least their names. This is done by the mean of the *Log_EuropeanLib_Files* table. After a file is successfully processed, its path (including its name) will be stored in this table. In case of an error, in order to keep the process going on, the current file will be moved to another (predefined) folder and can be handled later. That's usually the case when we meet a new format or the XML document is not well conformed.

Aside from the fact the collection is stored in local files for the first task, the two last tasks follow the same procedure as explained for CiteSeer.

4.3 Data analysis

4.3.1 Metrics

Once the work of retrieving and importing data into local database is done, the next objective is to analyze this database. Keeping in mind that we intend to design a graphical user interface using these libraries, we focus in aspects that would improve the decision accuracy. That means, we would like to know for example which searching fileds would be proposed to the user by the mean of the interface, wether a separated view of the tables is necessary for querying while the information displaying will be done with another view, etc. Of course, we cannot ask the user to give the identifier of the item, even if that is the most direct way to select an item.

Different questions have been addressed at the begining: "How many authors per publication", "Is the fulltext version available and how to access to it" or "How many single value and multivalued attributes does the table have", etc. These questions have been studied by the mean of SQL queries, and even if the results did not give us all necessary informations, they still helped us to norrow the ideas and to define the approach of analyzing. Although they are different questions, but they all consist in analyzing the nature and the utilization of the attributes inside the database.

id	name	distinct_values	min	max	average	filter_score	null %	samples

Table 11: The fields in the analysis table

As it was the case for the database structure, a common structure for analyzing the attributes of a table (library) has been proposed, whose the fields are given in table 11. For each table, we select the set of attributes that are used in the given library and each attributes will be analyzed one by one. The restuls will be displayed with the following fields:

- **id, name:** These fields give the internal identifier and name of the attribute. These values are extracted from the ones in *Attributes* table.
- **distinct_values:** This value corresponds to the number of distinct values that the current attribute has in the collection. This is one of the important information we need in order to compute the desciminant score which will be discussed later. The SQL query to obtain these value is "*Select attribute_id, Count(Distinct(value_string)) As distinct_values From Table_Name Group By attribute_id Order By distinct_values,attribute_id*" where *Table_Name* is the name of the correponding table.
- **min, max, average:** These fields gives a statistic of the number of times that a attribute is used within a single item in the collection. They indicate respectively the minimal, the maximal numbers and their average. For example, we can usually meet a case in which every item has at least one author, but some of them have several authors. Please note that if the attribute's *min* value is 1, it does not apply that every items in the collection has used at least once this attribute, there may be items which do not use this attribute. Even if these values do not come to the computation of the filter score, they still have another importance. Indeed, for intance we have a flat database model. This model is very simple and has a great adaptability, yet, it generates a huge number of rows within a single table. So if we intend to convert this model to a more relationnal model, i.e. split the data into saparate tables, the values of these fields *min, max* and *average* will help us to conceive the new structure. They can help us to decide wether it is useful or mandaroty to move a given attribute into a separate table, or it can be remained as an attribute in the current table (sigle value if min=max=1, otherwise multivalued).

  These values can be extracted by the mean of the query "*Select attribute_id, min(number), max(number), avg(number) From (select id, attribute_id, count(*) As number From Table_Name Group By id, attribute_id) as Temp Group By attribute_id Order By attribute_id*".
- **null %:** This field gives the percentage of the null value used in the collection for the current attribute. If this value is 50%, it means that half of the times that this attribute appears in the collection, its value is not specified. An important value for the null value can signify that the current attribute is not well defined and hence not useful, because of an very low value for the filter score. Netherless, the inverse cannot be either affirmed. We cannot say that if this percentage is extremely low, then the attribute is useful. Suppose that every value of a given attribute in the collection is not null, but they are all different, then the filter score still remains very low. This field can be used as an measure to remove an attribute from a collection if all of its values are null.
- **samples:** This field gives some samples of the most used values for the given attribute in the collection. The percentage of each value's utilization is also shown. This percentage corresponds to the set of not-null values, i.e. the number of items in the collection exluding the number of null values for the current attribute.
- **filter_score:** This score indicates how useful an attribute can be in the choice of an item in the collection. It corresponds to the main criterion of this analysis. The inspiration comes

from the notion of information entropy and of the algorithm ID3 [21] then has been adapted and proposed by David Portabella Clotet. The idea is based on the number of distinct values of a given attribute in the collection.

The formula indicating how the score of a given attribute is computed is shown in figure 8, where: $\sharp distinct\_value$ is the number of distinct values of the current attribute, M corresponds to the number of items in the collection and $\sharp occurences(x)$ is the number of times in the collection that the current attribute has x as value.

$$score = \frac{-1}{\#distinct\_values} \sum_{x \in distinct\_values} \frac{\#occurences(x)}{M} \log_2\left(\frac{\#occurences(x)}{M}\right)$$

Figure 8: Formula for the computation of the desciminant score

4.3.2 Implementation

The *AnalysisID3* class ensures the procedure for the computation of the filter score.As usual, an instance of this class is created in the main method of the *Digilib* class then the appropriate method will be called from here. As the *AnalysisID3* class needs to access to the local database, the global instance of *MySQL* is also given as an argument.

The objective here is to compute the the filter score for each attribute in the given collection, in the same time, other informations such as the percentage of the null value, samples for the values of the attributes would also be extracted and writen into local files within the procedure. The algorithm 1 gives the structure of the *AnalysisID3* class.

```
M = number of items in the collection; currentSum = 0.0; totalNotNullFreq = 0; nullFreq = 0;
foreach id in attributeIds do
    Select the set of pair (value, ♯occurence(value));
    foreach valueX in the set do
        if value != null then
            currentSum += (♯occurence(value) / M) * log2(♯occurence(value) / M);
            totalNotNullFreq+= ♯occurence(valueX);
            write to file;
        end
        else
            nullFreq += ♯occurence(valueX);
        end
    end
end
nullFreq += M - totalNotNullFreq;
currentSum += (nullFreq / M) * log2(nullFreq / M);
write to file;
♯distinctValue = number of distinct values of the attribute id in the collection;
filterScore = -(currentSum/♯distinctValue);
```

Algorithm 1: Algorithm for the computation of the filter score

When we want to compute the score for a given collection, the method *computeFilterScore()* is called. It accepts as argument the name of the collection, the number of items in the collection and

[21] ID3 algorithm, `http://en.wikipedia.org/wiki/ID3_algorithm`

the output folder's path in order to write files for each attribute. The file's name is the attribute's identifier and the file can be divided into three parts: the *header*, the *data* and the *overview* parts. The *header* contains the collection's and attribute's names. Each line of the *data* corresponds to a distinct value of the attribute and has the format "*attributeId;;value;;frequency;;probability*". This is also valable for the null value. Among other auxiliary informations, the third part contains the value of the filter score. Writing out these informations may take a little more time but may also be very useful for further analysis. The samples and their corresponding percentage are also extracted from these files. Table 12 illustrates the structure of the output file which was extracted from the file of the attribute whose *id*=22 (*name*="marc21_publisher") and beloging to the Nebis table.

♯HEADER $table Nebis $attribute_id 22 ♯DATA 22;;Lang;;4061;;0.006721707 22;;Springer;;2004;;0.003316991	22;;Cambridge University Press;;1889;;0.0031266448 22;;Oxford University Press;;1307;;0.002163327 ... 22;;Null Value;;419856;;0.69493943 ♯OVERVIEW $theorical_distinct_values 28443	$processed_distinct_values 28042 $excluded_values $descriminantScore 1.6225982E-4 ♯END

Table 12: This examples illustrates the three parts (*header*, *data* and *overview*) of the output file. This file corresponds to the attribute (*id*=22, *name*="marc21_publisher") of the Nebis collection. Only a portion of the *data* part is shown due to its length. The first row of this part indicates that the publisher "Lang" appears 4061 times in the Nebis table, and its probability is 0.006721707. The score for this attribute "marc21_publisher" is 1.6225982E-4

A required element for the process is the set of all attributes used in the collection, and also their corresponding number of distinct values. These informations are obtained by the mean of SQL queries and are imported into the hashtable *attributeIdDistValues* whose keys are the attribute's identifers and contents are the numbers of distinct values. Following the algorithm 1, the set of pairs (*value, frequency*) for all values of each attribute will be extracted from the database and will be processed. The query permitting obtaining the set is "*Select Left(value_string,60) As value_string, Count(*) As frequency From Current_Table_Name Where attribute_id= Current_Attribute_Id Group By Left(value_string,60)*".

An auxiliary class was implemented for the purpose of facilitating the content's extraction from these files, its name is *FilterScore*, included in the package *Auxiliary*. This class contains three principle methods: *getScore, getSamples* and *getNull*. The first and the last method return respectively the list of filter scores and the list of the probabilities of the null value for all attributes (files) in the current folder. For each attribute, the second method *getSamples* returns a list of five most used values of the attribtue, along with which the probability of each value (excluding the null value's one).

Results obtained for each collection will be discussed in the following sections. The figures cannot show all values of samples, because of their length. The original Excel file is inlcuded in the DVD support (Documents\Analysis.xls). This file contains the results for all libraries, along with other details (attribute's description, etc.).

4.3.3 Infoscience

Results for Infoscience are given in figures 11 and 12(a). Table *Infoscience* currently counts 62191 items and 1019192 rows. We can quickly notice the very high rate of null values, which is a possible explanation for the very low values for the filter score in the collection. Even the score for the three "most interesting" attributes is between 0.28 and 0.4.

Many attributes are used at most once for every item in the collection. Therefore, when we want to switch to a separate tables model, we can keep them as single value attribute within a single table, while other attributes like "marc21_general_note" or "marc21_uncontrolled_keyword", we would move them to other tables. But of course it depends on the point of view of the developer. Because even if the maximal value is very high (130 times for "marc21_general_note"), its average is not at all (1.9173 times).

In general, we obtain a high score when we have a good tradeoff between (1) the number of distinct values, (2) the distribution of these values and (3) the percentage of null values. For example, if in Infoscience, we specify the value "doi" for the "marc21_doi" when searching an item, only 23.2% of the collection need to be searched (null value occupies 76.8%). Therefore, this attribute should be proposed to the user for searching items.

Naturally, in spite of their low score, common attributes like "title" or "author" remain mandatory.

4.3.4 Nebis

For instance, the Nebis table counts 218706 items and 9595032 rows. Because it is limited in time and in space, we could not import and analyze the entire Nebis collection, which is estimated to 5,4 billion items. Netherless, with only a small portion of the collection, we can already notice that the number of attributes in Nebis is much more than other libraries. With only more than 200000 items, we already count 561 attributes. The results are given in figure 13. Remarks for Infoscience are also applied for Nebis, in particular the very high rate for null values.

4.3.5 CiteSeer

Table *CiteSeer* currently contains 716772 items (12457184 rows). CiteSeer's results are given in figure 12(b). Attributes in CiteSeer are very well specified, we have a very low rate for the percentage of null values, except for one of them. However, due to the very high rate of distinct values, the filter scores remain very low.

A particular attribute in this collection is the *dc_relation* (*id*=66). This Dublin Core attribute is interesting before it indicates the relation of the current item with others, if available. For example, the item with *id*=62193 in table *CiteSeer* has two relations with other items which are identified respectively by "oai:CiteSeerPSU:97473" and "oai:CiteSeerPSU:154288" (these two values correspond to the *oai_header_identifier*'s one). A statistic for this attribute in table *CiteSeer* has been realized and shown in figure 9. The graphs give the number of items y that have x relation occurences. We can notice that there are a lot of items containing up to 5 relations, then the number of items decreases quickly as the relations increase. Items having more than 50 relations are not that rare. The query allowing to obtain these informations is "*Select relation, Count(*) as occurence From (Select id, Count(*) As relation From CiteSeer Where attribute_id=66 Group By id) As A Group By relation*". The informations provided by this attribute can be emploid to propose similar or related items to the current one, for example.

Besides, it is not necessary to propose another structure for the CiteSeer database model, unless we want to move the attribute "dc_relation" to a separte table. The current flat model is well apdated for the CiteSeer's content, because each attribute is present exactly once in each item (except "dc_relation").

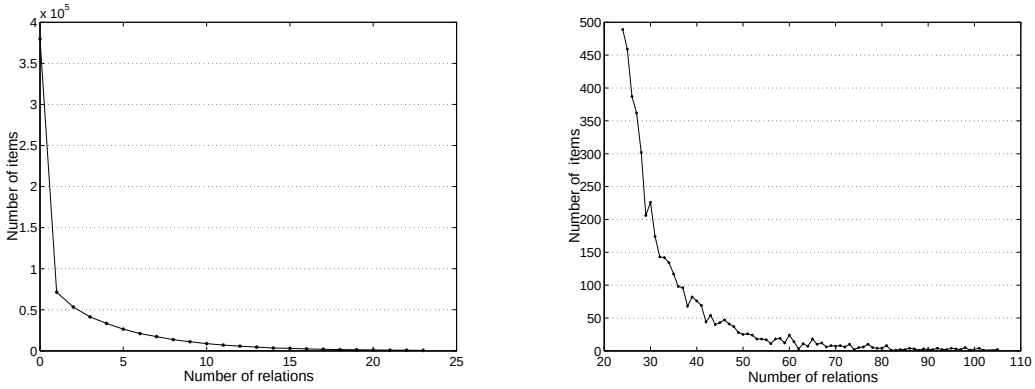


Figure 9: Number of items over the number of relations. Table *CiteSeer*

4.3.6 The European library

In order to keep a reasonable size for the tables, TEL collections have been split and imported into two different tables "EuropeanLib_Items" "EuropeanLib_Items_2". Like two libraries above, the filter scores here are also low. The rate for null values in the first table are much higher than the second's one. According to results of the two tables, two additionnal attributes that would be proposed to user are "dc_type" and "dc_language", because of their very tradeoff between the number of distinct values, the distribution of these values and the percentage of null values.

As the major part of TEL collections is in DC format, the same statistic for the attribute "dc_relation" as for CiteSeer has been realized for table *EuropeanLib_Items* and shown in figure 10.

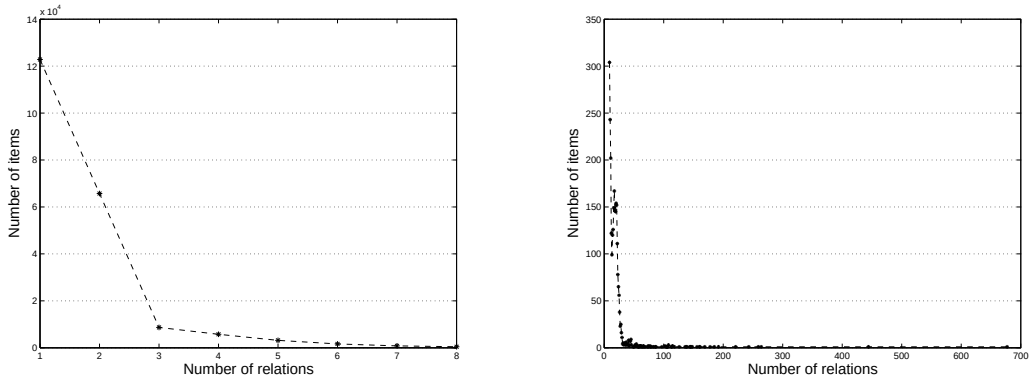


Figure 10: Number of items over the number of relations. Table *EuropeanLib_Items*

Analysis of the Infoscience's attributes (part 1)

id	name	full_id	distinct_values	min	max	average	filter_score	null %	Samples
13	marc21_doi	024_7_2	1	1	2	1.0008	0.3900827	76.8761%	"doi"(100%)
37	marc21_affiliation	973__a	2	1	2	1.0000	0.37265944	57.9473%	"EPFL"(94.08481%),"OTHER"(5.915191%)
39	marc21_reviewing	973__r	2	1	2	1.0001	0.28602257	75.7103%	"REVIEWED"(96.05456%),"NON-REVIEWED"(3.9454525%)
38	marc21_status	973__s	3	1	1	1.0000	0.26219577	59.6694%	"PUBLISHED"(97.58392%),"ACCEPTED"(1.29575%),"SUBMITTED"(1.1203253%)
17	marc21_role	700__e	2	1	9	1.2594	0.21909826	87.7908%	"dir."(51.99527%),"ed."(48.004753%)
75	marc21_endnote_import	856_40_i	1	1	1	1.0000	0.1903637	92.5986%	"EXTERNAL"(100%)
53	marc21_infoscience_document_type	856_40_x	5	1	2	1.1272	0.17412157	81.8430%	"public"(62.575268%),"restricted"(36.397446%),"PRIVATE"(0.74388945%),"icon"(0.2
3	marc21_document_type	980__a	16	1	1	1.0000	0.14380452	0.0643%	"BOOK"(40.953487%),"ARTICLE"(22.572445%),"CONF"(15.467169%),"THESIS"(7.6
11	marc21_oai_set	024_8_p	2	1	2	1.1788	0.13992606	92.7481%	"thesis"(84.8337%),"thesis:fulltext"(15.166298%)
51	marc21_description	856_4_z	7	1	3	1.0856	0.11180816	84.9946%	"Additional information"(40.998714%),"URL"(34.601376%),"Profil en français"(11.787
46	marc21_institution	852__a	3	1	1	1.0000	0.094996974	94.3867%	"BIBLMH"(65.36811%),"BIBLTT"(25.895166%),"BIBMCS"(8.736753%)
74	marc21_sumary_language	520__9	5	1	4	1.9514	0.045922216	96.3821%	"eng"(50.26664%),"fre"(41.0222%),"ger"(6.622219%),"ita"(1.91111%),"spa"(0.177777
32	marc21_super_year	773__y	84	1	1	1.0000	0.033933565	62.8017%	"2005"(7.901789%),"2004"(7.5819135%),"2003"(6.9205494%),"2002"(6.656868%),"2
1	marc21_unit	909_C0_p	254	1	6	1.0237	0.023771694	3.4426%	"BIBCEDEC"(10.557869%),"BIBCEAT"(7.4554534%),"BISCOM"(6.8459616%),"BIBL
23	marc21_publication_year	260__c	256	1	1	1.0000	0.022035195	4.4186%	"1996"(4.7574987%),"2004"(4.7305827%),"2005"(4.4900155%),"1998"(4.446276%),"
76	marc21_student_work_type	980__b	4	1	1	1.0000	0.021621663	98.8551%	"SEMESTER"(46.48868%),"DIPLOMA"(34.5505%),"MASTERS"(17.837048%),"OTHE
4	marc21_notice_status	980__c	3	1	1	1.0000	0.003873471	99.9357%	"DELETED"(100%)
26709		700__g	52	1	8	2.0132	0.002275906	99.0224%	"112562"(34.539406%),"117582"(5.592094%),"155767"(5.098674%),"112713"(4.934
35	marc21_super_number	773__n	740	1	1	1.0000	0.0022686	81.8414%	"1"(11.316743%),"2"(10.015053%),"3"(9.5368805%),"4"(8.031523%),"5"(5.800052%)
21	marc21_publication_place	260__a	1908	1	1	1.0000	0.001811722	57.0034%	"Lausanne"(27.22887%),"Paris"(13.066567%),"Berne"(9.027674%),"Zürich"(3.67988(
34	marc21_super_volume	773__v	1422	1	1	1.0000	0.00179325	77.7781%	"1"(3.3646889%),"2"(2.6845155%),"3"(2.1345878%),"4"(1.8017368%),"6"(1.6425471
29	marc21_super_issn	773__x	259	1	1	1.0000	8.84E-04	98.3535%	"0018-019X"(12.304672%),"0040-4039"(7.421866%),"0022-3263"(3.8085892%),"000
26711		856_4_2	1	1	1	1.0000	8.19E-04	99.9888%	"doi"(100%)
73	marc21_serie_title	440__a	484	1	1	1.0000	8.18E-04	97.0912%	"Lecture Notes in Computer Science"(11.719193%),"Proc. of IEEE"(4.0353823%),"Si
41	marc21_uncontrolled_keyword	653_1_a	23045	1	50	3.6337	8.00E-04	0.0000%	"Suisse"(2.2704248%),"aménagement du territoire"(1.8507501%),"LTS1"(1.2863598
22	marc21_publisher	260__b	7774	1	1	1.0000	7.86E-04	47.1644%	"EPFL"(15.791716%),"OFEFP"(2.7602787%),"UNIL"(2.3859522%),"IEEE"(1.8838065
5	marc21_conversion_notes	990__z	379	1	4	1.2550	6.38E-04	98.2988%	"promoting "Other" document to "Article""(10.491487%),"promoting "Other" document
25	marc21_conference_place	711_2_c	2184	1	1	1.0000	6.33E-04	90.8315%	"Lausanne, Switzerland"(1.9466853%),"Lausanne"(1.5433182%),"Paris, France"(0.96
16	marc21_author_name	700__a	49420	1	46	2.3811	5.98E-04	0.0000%	"Margaritondo, G."(1.0869739%),"OFFICE FEDERAL DE L'ENVIRONNEMENT, DES
27	marc21_page_count	300__a	10910	1	1	1.0000	5.45E-04	60.1695%	"30 cm"(0.9163941%),"1 vol. (pagination multiple) : ill. 30 cm"(0.3996609%),"2 vol. : i
28	marc21_super_title	773__p	10918	1	1	1.0000	5.13E-04	63.0123%	"Helvetica Chimica Acta"(1.3172195%),"Journal of the American Chemical Society"(0
26	marc21_conference_date	711_2_d	1811	1	1	1.0000	4.26E-04	95.2678%	"2004"(0.88345194%),"September 2004"(0.7475363%),"September"(0.71355736%),"
44	marc21_general_note	500__a	19717	1	130	1.9173	4.09E-04	44.6608%	"Fonds Cage_lions"(3.7540681%),"Fonds Lerch"(2.3884242%),"DIPLOMES"(2.08914
26708		440__v	179	1	1	1.0000	4.08E-04	99.5546%	"-"(4.693145%),"6"(2.5270782%),"8"(2.166067%),"97-28"(2.166067%),"21"(1.805055
12	marc21_DOI	024_7_a	5804	1	2	1.0008	3.79E-04	76.8922%	"NA"(58.687637%),"10.1016/S0022-328X(00)96166-7"(0.020875374%),"10.1007/BF(
45	marc21_thesis_note	502__a	3923	1	1	1.0000	3.71E-04	93.3576%	"Thèse sciences Ecole polytechnique fédérale de Lausanne EPFL"(24.473482%),"Th
31	marc21_super_page_count	773__c	14026	1	1	1.0000	3.58E-04	69.5294%	"-"(0.21108179%),"1"(0.1741425%),"1-4"(0.110817954%),"1-8"(0.089709766%),"6"(0
24	marc21_conference_name	711_2_a	2650	1	1	1.0000	3.49E-04	94.4879%	"None"(5.8343043%),"Academy of Management Annual Meeting"(0.32088676%),"IEE
7	marc21_isbn	020__a	8032	1	1	1.0000	3.27E-04	84.1006%	"0163-1829"(1.1529125%),"0003-6951"(0.88996756%),"0021-8979"(0.8596278%),"0
19	marc21_subtitle	245__b	10282	1	1	1.0000	3.09E-04	81.4716%	"rapport final"(0.39920163%),"Schlussbericht"(0.2777055%),"rapport de synthèse"(0.2
26706		245__n	8	1	1	1.0000	3.06E-04	99.9855%	"Volume 2"(22.221514%),"Volume 1"(22.221514%),"7"(11.110757%),"Deuxième éditi
40	marc21_abstract	520__a	15462	1	6	1.1908	3.00E-04	73.9737%	"./."(0.2347708%),"not available"(0.16063266%),"[on SciFinder (R)]"(0.10502904%),"-

Figure 11: Infoscience's analysis (part 1, table *Infoscience*)

Analysis of the Infoscience's attributes (part 2)

id	name	full_id	distinct_values	min	max	average	filter_score	null %	Samples
50	marc21_url	856_4_u	7174	1	3	1.1081	2.97E-04	87.8873%	"http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&d"(1.8717647%),"http:/
26710		773__k	23	1	1	1.0000	2.89E-04	99.9614%	"Applications of coordination chemistry"(8.333926%),"Semiconductor electrodes and
18	marc21_title	245__a	55353	1	1	1.0000	2.83E-04	2.2752%	"Tables annuelles de constantes de et données numériques de c"(0.05429775%),"Le
52	marc21_infoscience_filename	856_40_u	11127	1	2	1.1273	2.81E-04	81.8430%	"http://cvlab.epfl.ch/publications/publications/2004/Vacchett"(0.026567478%),"ftp://ftp
48	marc21_location	852__c	3170	1	1	1.0000	2.80E-04	94.8771%	"TTH01.1"(0.06277461%),"TME00.1"(0.06277461%),"TTH99.13"(0.06277461%),"TM
10	marc21_oai_id	024_8_a	3824	1	1	1.0000	2.79E-04	93.8480%	"oai:infoscience.epfl.ch:thesis-2169"(0.052273914%),"oai:infoscience.epfl.ch:thesis-2
15	marc21_extern_id	088__a	7045	1	1	1.0000	2.78E-04	88.6527%	"2005007"(0.028340658%),"2005013"(0.028340658%),"200433"(0.028340658%),"20
14	marc21_epfl_id	037__a	28661	1	3	1.0359	2.73E-04	53.8342%	"LBD-ARTICLE-1997-003"(0.006965971%),"LBD-CONF-1997-007"(0.006965971%),'
6	marc21_certificate_number	013__a	34	1	1	1.0000	2.71E-04	99.9453%	"WO2006102970"(2.9412255%),"WO2006066439"(2.9412255%),"WO2006047628 (A
2	marc21_batch_id	970__a	56685	1	1	1.0000	2.61E-04	8.8421%	"LEMA_Rius_Gonzales-Arbesu_Romeu_Cardama_Heldring_Ubeda_Mosig"(0.00352
26707		245__p	4	1	1	1.0000	2.09E-04	99.9952%	"Lectures on Mathematical Programming ismp97"(33.346004%),"Traité des matériaux
26705		024_7_u	1	1	1	1.0000	1.40E-04	99.9984%	"10.1145/258533.258573"(99.91451%)
26712		909__p	1	1	1	1.0000	1.40E-04	99.9984%	"LRMB"(100%)
33	marc21_super_logical_date	773__d	1	1	1	1.0000	0	100.0000%	

(a)

Analysis of the Citeseer's attributes

id	name	distinct_values	min	max	average	filter_score	null %	Samples
62	dc_format	2	1	1	1	2.993E-01	0.00349%	"ps"(68.63985%),"pdf"(31.360159%)
60	dc_date	5011	1	1	1	2.283E-03	0.00000%	"1970-01-01"(3.1793933%),"2002-07-11"(0.39329106%),"1997-04-26"(0.3666438%),"2002-03-27"(0.31209368%),"200
70	oai_header_datestamp	5011	1	1	1	2.283E-03	0.00000%	"1970-01-01"(3.1793933%),"2002-07-11"(0.39329106%),"1997-04-26"(0.3666438%),"2002-03-27"(0.31209368%),"200
66	dc_relation	196134	1	312	5.2353	1.856E-04	0.00000%	"oai:CiteSeerPSU:311874"(0.24275503%),"oai:CiteSeerPSU:328445"(0.21094576%),"oai:CiteSeerPSU:527057"(0.19
55	dc_creator	349325	1	1	1	4.803E-05	8.19270%	"Douglas C. Schmidt"(0.029025195%),"Subbarao Kambhampati"(0.025985908%),"E. Ycesan,C. -h. Chen,J. L. Snowd
54	dc_title	502372	1	1	1	3.677E-05	1.98459%	"Journal of Graph Algorithms and Applications"(0.040709022%),"Proceedings of the 2002 Winter Simulation Conferen
57	dc_description	609398	1	1	1	3.617E-05	0.06432%	"This report was prepared as an account of work sponsored by"(0.006142583%),"this document. The furnishing of this
56	dc_subject	544743	1	1	1	3.561E-05	0.00712%	"E. Ycesan,C. -h. Chen,J. L. Snowdon,J. M. Charnes Proceeding"(0.021905316%),"Douglas C. Schmidt,Steve Vinosk
64	dc_source	715481	1	1	1	2.980E-05	0.00349%	"http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-W"(0.15612204%),"http://www.aps.anl.gov/conferences/n
63	dc_identifier	716772	1	1	1	2.690E-05	0.00000%	"http://citeseer.ist.psu.edu/1.html"(1.3951438E-4%),"http://citeseer.ist.psu.edu/2.html"(1.3951438E-4%),"http://citeseer
69	oai_header_identifier	716772	1	1	1	2.690E-05	0.00000%	"oai:CiteSeerPSU:1"(1.3951438E-4%),"oai:CiteSeerPSU:2"(1.3951438E-4%),"oai:CiteSeerPSU:3"(1.3951438E-4%),"(
58	dc_publisher	1	1	1	1	0.000E+00	100.00000%	
59	dc_contributor	1	1	1	1	0.000E+00	0.00000%	"The Pennsylvania State University CiteSeer Archives"(100.0%)
65	dc_language	1	1	1	1	0.000E+00	0.00000%	"en"(100.0%)
68	dc_rights	1	1	1	1	0.000E+00	0.00000%	"unrestricted"(100.0%)
71	oai_header_setSpec	1	1	1	1	0.000E+00	0.00000%	"CiteSeerPSUset"(100.0%)

(b)

Figure 12: (a) Infoscience's analysis (part 2, table *Infoscience*) and (b) CiteSeer's analysis (table *CiteSeer*)

Analysis of the Nebis attributes

id	full_id	distinct_values	min	max	average	filter_score	null %	Samples
26747	003__	1	1	1	1	0.46906477	64.54%	ETHICS"(100.0%)
26717	040__c	2	1	1	1	0.46903256	64.55%	ETH-BIB"(100.0%)
26718	040__e	4	1	1	1	0.23453033	64.55%	ETHICS-ISBD"(99.99908%),"ETHICS-ISBDE"(4.6685344E-4%),"ETHICS-ISBD0"(4.6685344E-4%)
26720	090__b	7	1	4	1.9314	0.22725317	31.49%	BHR"(51.775383%),"GLIS"(48.122902%),"MIKAS"(0.09229038%),"EBI03"(0.008455925%),"Allegro/Schlag"(4.8319568E-4%)
26716	040__b	5	1	1	1	0.19207892	64.55%	ger"(99.28006%),"***"(0.6975278%),"fre"(0.021943646%),"7ger"(4.668861E-4%)
26721	090__n	8	1	3	1	0.11871847	64.79%	5"(94.91615%),"0"(2.3083215%),"7"(1.6037453%),"6"(0.960273%),"8"(0.20446342%)
26826	691_E1_9	6	1	22	3.0813	0.060675003	96.09%	ger"(98.65819%),"eng"(0.6730447%),"fre"(0.6688117%)
26770	906__a	8	1	2	1	0.051992122	92.58%	Hochschulschrift = Tháese/Mâemoire"(92.51052%),"Festschrift = Mâelanges"(5.985765%),"Briefe = Correspondance"(0.6960711%),'
26729	100__9	4	1	1	1	0.047402896	97.10%	ger"(99.84568%),"eng"(0.14863092%),"fre"(0.0057165734%)
26759	913__e	2	1	5	1.0829	0.04064532	98.99%	Veranstaltungsort"(100.0%)
26815	680_Z_i	1	1	3	1.0661	0.04045349	99.00%	Keine Beschlagwortung"(100.0%)
26756	913__a	4	1	5	1.0798	0.037878122	98.66%	Kongress = Congráes"(73.09072%),"Ausstellung = Exposition"(26.909267%)
26738	700__9	5	1	8	1.1521	0.033726074	98.12%	ger"(99.814896%),"eng"(0.14985898%),"fre"(0.035260938%)
26777	240__h	4	1	1	1	0.029666595	98.97%	Noten"(85.58825%),"Ton"(14.411822%)
26724	099__l	37	1	1	1	0.028342962	64.54%	Z01"(94.97036%),"E01"(2.534889%),"E24"(1.252742%),"E**"(0.7234539%),"E26"(0.20303383%)
26818	906__b	5	1	1	1	0.02544508	98.27%	Schriftenreihe = Collection"(99.244095%),"Zeitschrift = Revue"(0.52626544%),"Zeitung/Zeitschrift = Journal/Revue"(0.22007462%),"Z
26834	909_Z1_a	3	1	2	1.0008	0.023632044	99.15%	ZH"(99.961296%),"zbzaad200012l"(0.03882746%)
26741	852_4_b	96	1	393	1.6995	0.019705009	41.99%	Z01"(80.657005%),"Z05"(3.0091383%),"E39"(2.4970112%),"Z16"(2.1865969%),"E19"(1.5283923%)
26744	852_4_4	164	1	393	1.6995	0.019705009	41.99%	ZB (Zèurich)"(80.657005%),"ZB Musikabteilung"(3.0091383%),"FHNW-PH-S (Solothurn)"(2.4970112%),"UNI-RWI (Zèurich)"(2.1865
26784	906__d	4	1	1	1	0.019438861	98.96%	PM Partitur = Partition"(48.206722%),"PM Andere Ausgabeform = Autre forme"(43.21707%),"PM Klavierauszug = Râeduction pour p
26838	906__f	3	1	1	1	0.01728541	99.18%	MF Mikrofiche = Microfiche"(99.49639%),"MF Mikrofiche-Kassette = Cass. de microfiches"(0.48318768%),"MF Mikrofilmspule = Micr
26715	040__a	75	1	1	1	0.016152894	64.48%	ZBZ"(97.08983%),"UNI-RWS"(1.2594001%),"*******"(0.6462405%),"SzZuIDS NEBIS ZBZ"(0.28374943%),"ETH-BIB"(0.22550856%)
26859	680_Z_a	1	1	1	1	0.013631826	99.73%	Keine Beschlagwortung"(100.0%)
26780	245_0_h	9	1	1	1	0.013488059	98.92%	Noten"(83.3001%),"Ton"(14.851008%),"Tonaufzeichnung"(1.3445358%),"Musikdruck"(0.47364327%),"Gegenstand"(0.015278815%)
26722	099__a	247	1	2	1.0001	0.010909018	64.53%	ZKON"(21.099495%),"LIMA"(7.104723%),"ERBE"(6.0598283%),"HAAG"(4.887531%),"GRIM"(4.7792606%)
26973	909_E1_a	3	1	1	1	0.010123507	99.71%	N25"(87.762024%),"N18"(12.237958%)
26745	852_4_5	330	1	241	1.5309	0.010059857	48.29%	Freihand 03"(36.40586%),"Freihand 02"(26.88964%),"Magazin 05"(12.31539%),"Magazin 04"(7.700359%),"P1 Prèasenzbestand"(3.
26742	852_4_c	263	1	241	1.5312	0.009422887	48.28%	03"(36.63437%),"02"(26.883274%),"05"(12.312472%),"04"(7.698536%),"LSM"(3.8174264%)
26843	650_Z_i	1	1	1	1	0.009128565	99.83%	Keine Beschlagwortung"(100.0%)
26831	691_E2_9	4	1	25	1.4224	0.00834889	99.65%	ger"(99.66492%),"eng"(0.2869431%),"fre"(0.047823854%)
26871	710__9	4	1	4	1.0673	0.006423294	99.75%	ger"(99.74034%),"ita"(0.12995484%),"fre"(0.12995484%)
26787	245__h	22	1	2	1.0003	0.005819174	98.97%	Mikroform"(80.25847%),"Noten"(12.962098%),"Kartenmaterial"(2.8087237%),"Tonaufzeichnung"(1.7272036%),"Ton"(1.7110616%)
26808	700__h	5	1	6	1.6228	0.005626925	99.80%	Ton"(64.38166%),"Noten"(35.53533%),"Kl"(0.08361255%)
26761	072_7_2	195	1	4	1.0262	0.004713277	68.73%	Z01"(99.4489%),"E01"(0.2228787%),"Z19"(0.06246956%),"E45"(0.03705822%),"E86-19991231"(0.02594075%)
47	852__b	75	1	51	1.3038	0.004645166	96.12%	Z01"(47.310677%),"Z16"(20.387604%),"E65"(5.9591904%),"E01"(3.8931103%),"Z14"(2.8088448%)
26740	852__4	113	1	51	1.3038	0.004645166	96.12%	ZB (Zèurich)"(47.310677%),"UNI-RWI (Zèurich)"(20.387604%),"FH-HGK (Zuerich)"(5.9591904%),"ETH-BIB (Zuerich)"(3.8931103%)
26969	906__e	4	1	1	1	0.004621894	99.79%	SR CD"(89.29341%),"SR Tonband-Kompaktkassette = Cassette audio"(10.318006%),"SR Schallplatte = Disque 33 1/3"(0.31031597'
26909	773_A_9	3	1	4	1.0139	0.004528564	99.88%	Y"(98.770905%),"N"(1.2312163%)
26860	240__k	3	1	1	1	0.004257441	99.89%	Ausw."(97.4858%),"Ausw"(2.5148082%)
26833	044__a	67	1	4	1.8618	0.004067027	97.83%	sz"(45.287193%),"gw"(42.965168%),"xxu"(6.9202566%),"xxk"(1.3672471%),"au"(1.1686527%)
26760	072_7_a	957	1	4	1.0266	0.003544309	68.72%	M221"(1.976452%),"M058"(1.889668%),"M347"(1.8351635%),"M644"(1.7774839%),"M365"(1.7044584%)
26806	240__o	5	1	1	1	0.00319003	99.87%	KIA"(66.83279%),"Bearb."(30.662983%),"Bearb"(2.2527907%),"OrgA"(0.25031006%)
26925	856_EA_z	1	1	1	1	0.002549146	99.96%	Abstract / Index"(100.0029%)

Figure 13: Nebis analysis (table *Nebis*)

Analysis of the table EuropeanLib_Items's attributes

id	name	distinct_values	min	max	average	filter_score	null %	Samples
80	dcterms_hasFormat	1	1	1	1	2.16463200E-01	91.10586000%	"image/jpeg"(100.0%)
81	oai_onbba_ownerInstitution	5	1	1	1	9.83063200E-02	91.16760000%	"Österreichische Nationalbibliothek (ÖNB)"(49.25414%),"Österreichisches Institut für Zeitgeschichte (
82	oai_onbba_ownerInstitutionID	5	1	1	1	9.83063200E-02	91.16760000%	"342266"(49.25414%),"342261"(19.146582%),"346458"(18.025932%),"387911"(13.537722%),"53629
83	oai_onbba_ownerCollectionID	6	1	1	1	4.09830250E-02	95.93288000%	"342266"(99.89414%),"5362909"(0.07732375%),"342261"(0.020348355%),"387911"(0.004069671%)
85	oai_onbba_technique	39	1	1	1	6.20253760E-03	96.84125000%	"FOTOGRAFIE"(73.48567%),"Schwarz-Weiß-Negativ"(13.168104%),"Silbergelatineabzug"(4.433035
61	dc_type	1379	1	7	2.0453	4.40661560E-03	0.00000000%	"MovingImage"(62.795376%),"printed text"(30.662956%),"Fotografie"(26.772526%),"text"(22.284346
65	dc_language	535	1	6	1.0068	4.26736470E-03	61.95705400%	"ita"(25.022081%),"HUN"(18.7286%),"fre"(15.5055%),"ger"(14.776736%),"Magyar"(8.396673%)
370	oai_onbba_imageType	10	1	2	1.0061	3.62739060E-03	99.64678000%	"Einzel/Portrait"(62.69905%),"Rollenbild"(14.901568%),"Gruppenbild"(11.433907%),"Porträt"(7.73194
68	dc_rights	1045	1	4	1.3047	2.07585200E-03	61.72268300%	"free"(24.033005%),"domaine public"(21.995874%),"public domain"(21.995874%),"ÖNB"(11.392088%
77	oai_tel_recordId	55631	1	1	1	6.88987800E-04	90.78757500%	"http://www.bildarchivaustria.at/TELRequest.aspx?p_ImageID=13"(11.728774%),"http://www.bildarchi
6725	oai_onbba_view	6	1	1	1	4.21130530E-04	99.98113000%	"Großaufnahme"(55.258125%),"Vogelperspektive"(21.927826%),"Detail"(11.40247%),"Totale"(7.0169
67	dc_coverage	13553	1	10	1.8975	3.88153480E-04	42.16882300%	"Brno"(8.956078%),"Moravská zemská knihovna v Brn?"(8.947206%),"Zámek Kyn~vart"(6.890502%)
373	oai_onbba_placeDepicted	2127	1	6	1.6868	3.21353600E-04	94.44089500%	"Wien"(16.411602%),"Österreich"(15.759544%),"1. Wiener Gemeindebezirk"(3.942119%),"Innere Sta
62	dc_format	24847	1	9	2.0083	2.84397860E-04	0.00000000%	"PAL"(62.207287%),"MPEG-2, 8Mbps"(62.207287%),"Negatív film"(23.031528%),"html"(22.256392%
58	dc_publisher	22789	1	18	1.026	2.30049060E-04	48.51745200%	"Magyar Televízió 1"(11.663206%),"Duna Televízió"(9.170262%),"RTL Klub"(6.934844%),"TV2"(6.69
86	oai_onbba_event	478	1	9	1.0367	2.14566370E-04	99.36755000%	"Olympische Spiele: Berlin"(6.333413%),"Olympische Sommerspiele: Helsinki"(4.6846323%),"Wahler
59	dc_contributor	29928	1	71	1.8454	1.49412240E-04	67.43605000%	"Góczán Andrea"(4.316887%),"Drótos László"(4.02513%),"Balassa M. Iván"(2.3879354%),"Szabó Je
88	oai_onbba_personDepicted	3175	1	23	1.4987	1.45291170E-04	97.16814400%	"Kreisky, Bruno"(5.0382805%),"Schärf, Adolf"(2.1450684%),"Hitler, Adolf"(1.9171184%),"Figl, Leopol
56	dc_subject	68568	1	57	2.2763	1.19833140E-04	42.34890700%	"Magyar"(4.4555655%),"irodalom"(3.987873%),"Vers"(3.3565316%),"20. század"(2.1412208%),"19. s
64	dc_source	3163	1	4	1.1099	1.14935010E-04	97.66901000%	"Magyar Nemzeti Galéria, Budapest"(12.248825%),"Magántulajdon"(10.693756%),"Biblioteka Narodo
256	oai_onbba_dateAfter	2322	1	1	1	9.82897100E-05	98.64274600%	"1932-01-01"(4.2804894%),"1930-01-01"(3.5121963%),"1960-01-01"(3.3170745%),"1930"(2.036586
257	oai_onbba_dateBefore	2317	1	1	1	9.76969500E-05	98.65947000%	"1960-12-31"(2.8645558%),"1936-12-31"(2.259542%),"1957-12-31"(2.00025%),"1961-12-31"(2.0002
60	dc_date	152901	1	83	1.5566	9.63593500E-05	0.00000000%	"2006-11-28"(26.146313%),"1998"(5.7827888%),"1995"(5.13828%),"1994"(5.094059%),"1997"(4.352
566	oai_onbba_caption	3050	1	4	1.1969	9.02965900E-05	98.29350000%	"Rudolf Semotan"(8.147415%),"Otto Bartel"(3.7827287%),"Sozialistischer Verlag"(3.4723508%),"Her
55	dc_creator	107870	1	144	1.1789	8.75506700E-05	39.33415200%	"1956-os Intézet Közalapítvány"(2.3161082%),"Váli Dezs?"(1.741242%),"Destailleur, Hippolyte (1822
66	dc_relation	175901	1	677	1.8446	6.72619350E-05	35.35095000%	"http://www.neumann-haz.hu/scripts/webkat?infile=virt_keret.h"(8.37513%),"Elektronikus Periodika Ar
1185	dcterms_temporal	2	1	1	1	6.19557200E-05	99.99901000%	"1. Weltkrieg"(83.141914%),"Jahrhundertwende"(16.628384%)
63	dc_identifier	618756	1	6	1.5373	5.35561960E-05	0.00000000%	"http://www.bncf.firenze.sbn.it/cgi-opac/schedbib/schedbib.cg"(30.84747%),"http://www.neumann-haz
380	oai_onbba_dimensions	119	1	1	1	5.29524040E-05	99.96756000%	"10,9x14,2cm"(5.1018553%),"16,2x22cm"(3.571299%),"16,3x22,2cm"(3.0611134%),"16,5x22,4cm"(3
54	dc_title	421565	1	126	1.3326	4.90038000E-05	0.00000000%	"Reklám"(7271.164%),"Ajánló"(6111.003%),"Híradó"(2276.958%),"Degob jegyz?könyv"(1797.3837%)
57	dc_description	197542	1	520	1.5375	4.80345080E-05	0.00000000%	"Ezt a rekordot az NDA-tól független adatgazda készítette és"(3343.7961%),"lakóház"(45.906807%),""
79	dc_bibliographicCitation	1109	1	1	1	3.41702540E-05	99.81628000%	"Arch Oncol 10(3) 127-127"(0.18018037%),"Stomatol. glas. Srb. 50(1) 7-12"(0.090090185%),"Stomat
84	oai_tel_thumbnail	53733	1	1	1	3.40795740E-05	91.10586000%	"http://www.bildarchivaustria.at/Bildarchiv//524/B5946118T594"(0.005582952%),"http://www.bildarchiv

Figure 14: The European Library's analysis (part 1, table *EuropeanLib_Items*)

Analysis of the table EuropeanLib_Items2's attributes (log2)

id	name	distinct_values	min	max	average	filter_score	null %	Samples
65	dc_language	249	1	8	1.0239	6.94591500E-03	0.00000000%	"por"(75.99499%),"fre"(8.337186%),"eng"(6.7372518%),"spa"(4.1574874%),"lat"(2.8555994%)
61	dc_type	426	1	3	1.1545	3.47389840E-03	0.00000000%	"material textual, impresso"(87.64895%),"Texto policopiado]"(6.30579%),"material gráfico a duas dimensões"(4.955
60	dc_date	16647	1	2	1.0002	5.53094000E-04	0.28889155%	"2003"(1.6500229%),"2002"(1.6491199%),"2000"(1.6224799%),"1999"(1.5702535%),"2001"(1.5532461%)
26713	oai_mods_edition	9061	1	3	1.0002	1.63658450E-04	86.21650000%	"1ª ed"(23.71387%),"2ª ed"(16.596441%),"3ª ed"(7.20453%),"4ª ed"(4.092766%),"5ª ed"(2.5662804%)
58	dc_publisher	135373	1	10	2.0246	1.23327120E-04	0.00000000%	"Lisboa"(34.904255%),"[s.n.]"(13.908516%),"Porto"(10.047574%),"[S.l."(8.318726%),"s.n.]"(5.559924%)
56	dc_subject	189311	1	18	1.7606	8.33340900E-05	0.00000000%	"821.134.3-1"19", por"(1.4255486%),"821.134.3-31"19", por"(1.2901822%),"766(=1:469)"198"(084.5), por"(1.00009%
59	dc_contributor	94939	1	16	1.3592	7.71721240E-05	56.28883400%	"Portugal"(3.256485%),"Vieira, Ernesto, 1848-1915"(0.67739004%),"Universidade Técnica de Lisboa"(0.6104407%)
55	dc_creator	206719	1	14	1.0895	7.22097760E-05	9.58729700%	"Portugal"(1.6140599%),"Camões, Luís de, 1524?-1580"(0.17926621%),"Castelo Branco, Camilo, 1825-1890"(0.173
62	dc_format	233099	1	3	1.0009	6.66194200E-05	0.41120150%	", 30 cm"(0.9269138%),"1 v."(0.7010247%),"2 v."(0.40280288%),"1 v., 8º"(0.34885475%),", 8º"(0.34192285%)
67	dc_coverage	10676	1	5	1.2432	5.44569600E-05	96.76621000%	"Música sacra, Séc. 19"(5.7592354%),"Música sacra, Séc. 18"(4.292742%),"Música para piano, Séc. 19"(2.4642658
57	dc_description	196448	1	151	1.6508	5.42584130E-05	0.00000000%	"Monografia"(91.11565%),"Série"(4.1726446%),"1ª ed"(3.2800071%),"2ª ed"(2.2901223%),"Contém bibliografia"(1.8
26714	dcterms_alternative	44394	1	35	1.1239	4.67542100E-05	88.21968000%	"Leis, decretos, etc."(13.504976%),"Tratados, etc."(1.3147%),"Bíblia."(0.8076742%),"Liturgia e ritual"(0.77709985%)
64	dc_source	1145	1	2	1.0012	4.47904700E-05	99.74563000%	"Inclui facsimiles"(2.3598812%),"Rep. de desenho"(2.123893%),"Rep. de pintura"(1.4159286%),"Facsimiles"(1.1209
66	dc_relation	77545	1	3	1.0045	4.46147600E-05	82.16526000%	"Autores de língua portuguesa"(0.4106361%),"Literatura estrangeira"(0.3719286%),"Editora"(0.34668458%),"Romar
63	dc_identifier	1384727	1	412	2.5958	3.51347080E-05	0.00000000%	"Cota (Call-Number): UEBIB-Escola do Magist?erio Prim?ario"(3.22418%),"Cota (Call-Number): UEBIB-Fundo Teixe
54	dc_title	538523	1	2	1	3.48625180E-05	0.00180088%	"Estatutos"(0.44482598%),"Curriculum Vitae"(0.17468873%),"Relatório e contas"(0.13761991%),"Regulamento"(0.0
77	oai_tel_recordId	636386	1	2	1.0003	-4.22279180E-04	0.00000000%	"http://opac.porbase.org/ipac20/ipac.jsp?profile=porbase&uri="(100%)

Figure 15: The European Library's analysis (part 2, table *EuropeanLib_Items_2*)

5 Conclusion and future work

The principle part of this work concentrated on the understanding of the different protocols and formats, then on implementing solutions in order to retrieve data into the local database. The analyzing part was also addressed.

During the project, a lot of knowledge and experiences have been acquired, that varies from technologies (XML, JAVA, protocols, etc.) to human aspects (contact, request for information). Even if this work does not give a concrete solution or application, it would provide certain valuable tools for researchers on further analysis or people making user interfaces for these catalogs.

At this point of the project, we accomplished the following objectives:

- A generic software to harvest library collections has been implemented. It works with any library using (1) the Z39.50 protocol and (2) OAI-PMH protocol. Also, its parsers support (1) Dulin Core format, (2) XMLMarc format and (3) USmarc in text format. Moreover, this software provides methods to perform standard SQL queries.
- The same software provides methods for analyzing the nature and the utilization of attributes in the local database. Some analysis have been performed and discussed (filter_score, percentage of null values, etc.)
- The collections of Nebis, Infoscience and CiteSeer have been successfully harvested and imported into the local database. Among with them, The European Library collections have been imported. The four collection are also available in local files (in text or XML format).
- Standard protocols (Z39.50, OAI-PMH) and formats (DC, EndNote, MARC) have been studied. Their features, necessary for harvesting and analyzing libraries have been raised.

According to the results of the analysis, the items are not very well structured, in the sense that there is not a useful set of attributes used by all the items (too many null values), except for the case of CiteSeer. This makes the work of building an user interface and the decision on the set of attributes to filter on difficult. Therefore, here are some suggestions for the future works:

- Score function

  The filter function should be improved. The null values should not be taken into account. One suggestion for this improvement resides in mutiplying the percentage of not null values by the current *filter_score* function. Moreover, it is not enough to base only on the *filter score* to point out a set of attrutes to filter on.

- Protocol

  The fact that we could not reach all objectives (i.e. building an user interface) demontrates the complexity of the current information distribution system. In order to simplify and improve it, we think it is necessary to come to a standardized solution. And OAI-PHM would be a good candidate. It requires a lot of work for existing libraries using other protocols, but it would facilitate the access to these ressources and the work of researchers or people interrested in improving the accuracy when searching a ressource.

- Entity resolution [14, 15]

  Entity resolution is the task of determining which actual person, actor, or object a particular reference refers to by looking at the context. For instance, the following names "Mena, S.",

"Sergio Mena" and "S. Mena-de-la-Cruz" written in a publication citation could refer to the same teaching fellow of the University of York. On the other hand, the name "M. Schumacher" could refer to the Formula One driveror to a professor in Sierre. We usually need to take into account the context in where the name was found in order to resolve the entity. For instance, knowing the co-authors of the mentioned professor in Sierre may help in deciding that a publication citation is referring to him if we know that one of the co-authors is also present in the citation.

In the final word, I would like to thank David Portabella Clotet for his disponibility, his guide and his advices during the whole semester. The project has been a pleasure. It certainly required a lot of work, but considering the experiences and knowledge acquired, it was worth for.

References

[1] Citeseer Home Page, *Scientific Literature Digital Library*, http://citeseer.ist.psu.edu/citeseer.html

[2] The Z39.50 Information Retrieval Standard, *An article about Z39.50 from D-Lib Magazine*, http://www.dlib.org/dlib/april97/04lynch.html

[3] The Dublin Core Metadata Initiative Home Page, *Interoperable metadata standards*, http://dublincore.org/

[4] The Dublin Core Metadata Element Set, http://dublincore.org/documents/dcmi-terms

[5] EndNote Home Page, *A commercial reference management software package*, http://www.endnote.com

[6] The European Library Home Page, *A commercial reference management software package*, http://www.theeuropeanlibrary.org/portal/index.html

[7] Infoscience Home Page, *Scienctific information's portal of EPFL*, http://infoscience.epfl.ch

[8] The National Library of Netherlands Home Page, http://www.kb.nl/index-en.html

[9] Nebis Home Page, *Library catalog of swiss universities*, http://www.nebis.ch

[10] MARC Standards Home page, *Standards for the representation and communication of bibliographic and related information in machine-readable form*, http://www.loc.gov/marc/

[11] Understanding MARC Bibliographic, *A brief description and tutorial*, http://www.loc.gov/marc/umb/

[12] MySQL Home page, *The world's most popular open source database*, http://www.mysql.com/

[13] The Open Archives Initiative Protocol for Metadata Harvesting, *Specification*, http://www.openarchives.org/OAI/openarchivesprotocol.html

[14] Poon H., & Domingos, P. Joint Inference in Information Extraction, Proceedings of the Twenty-Second National Conference on Artificial Intelligence, 2007. Vancouver, Canada: AAAI Press, To appear, http://www.cs.washington.edu/homes/pedrod/papers/aaai07.pdf

[15] Richardson, M. & Domingos, P. 2006. Markov logic networks

[16] VB ZOOM Project, *An implementation of the ZOOM (Z39.50 Object-Orientation Model) Abstract API*, http://vb-zoom.sourceforge.net/

[17] YAZ of Index Data, *Programmers toolkit supporting the development of Z39.50/SRW/SRU clients and servers*, http://www.indexdata.com/yaz/

[18] ZOOM Initiative, *An abstract object-oriented API for the Z39.50 protocol*, http://zoom.z3950.org/

A EndNote reference types and attributes

Id	Name	Id	Name	Id	Name
0	Journal Article	13	Artwork	26	Chart or Table
1	Book	14	Encyclopedia	27	Equation
2	Thesis	15	Patent	28	Electronic Journal
3	Conference Proceedings	16	Electronic Source	29	Electronic Book
4	Personal Communication	17	Bill	30	Online Database
5	Newspaper Article	18	Case	31	Generic
6	Computer Program	19	Hearing	32	Government Report or Document
7	Book Section	20	Manuscript	33	Conference Paper
8	Magazine Article	21	Film or Broadcast	34	Online Multimedia
9	Edited Book	22	Statute	35	Classical Works
10	Report	23	ANET/COS	36	Legal Rule/Regulation
11	Map	24	Theological Dictionary	37	Unpublished Work
12	Audiovisual Material	25	Figure		

Table 13: Reference Type in EndNote (soucres from Endnote's Support at http://www.endnote.com/support/ensbl.asp)

Number	Name	Number	Name	Number	Name
1	Author	13	Tertiary Author	25	Accession Number
2	Year	14	Tertiary Title	26	Call Number
3	Title	15	Edition	27	Label
4	Secondary Author	16	Date	28	Keywords
5	Secondary Title	17	Type of Work	29	Abstract
6	Place Published	18	Subsidiary Author	30	Notes
7	Publisher	19	Short Title	31	URL
8	Volume	20	Alternate Title	32	Author Address
9	Number of Volumes	21	ISBN/ISSN	33	Image
10	Number	22	Original Publication	34	Caption
11	Pages	23	Reprint Edition		
12	Section	24	Reviewed Item		

Table 14: Attributes in EndNote. The data was extracted from the EndNote's tutorial of Arnaud Pelfrne at http://cid.ens-lsh.fr/aide/documents/ac_endnoteintro.htm

Id	Reference Type Name	Associated attributes
0	Journal Articles	Authors, Title, Secondary_Title, Pages, Refnum, Year, Date
1	Books	Authors, Title, Refnum, Publisher, Year, Date
2	Thesis	Authors, Title, Secondary_Title, Refnum, Abstract, Publisher, Keywords, Year, Date
3	Conference Papers	Authors, Title, Secondary_Title, Place_published, Refnum, Url, Abstract, Keywords, Year, Date
7	Book Section	Authors, Title, Secondary_Title, Refnum, Notes, Year, Date
10	Reports	Authors, Title, Refnum, Abstract, Year, Date
31	Book chapters	Authors, Title, Refnum, Notes, Publisher, Year, Date

Table 15: This table shows Reference Types and their associated attributes used in Infoscience. The number of reference types which are currently used is much inferior to the complete list, and so is the number of the associated attributes.