DIGITAL LIBRARIES – GOOD OR BAD CHOICES ON ORGANIZING INFORMATION

Adi-Cristina Mitea, Daniel Volovici, Antoniu Pitic
"Lucian Blaga" University of Sibiu - Computer Science Department, Romania
adi.mitea@ulbsibiu.ro, daniel.volovici@ulbsibiu.ro, antoniu.pitic@ulbsibiu.ro

ABSTRACT

Digital documents as the real ones have to be classified and indexed in a library for proper future exploitation. Classification and indexation process is a hard one for librarians all over the world. A software system can ease their work and make the process more accurate. We present in our paper methods for classifying and indexing publications, suitable for such a system and analyze different storage and index database management systems capabilities in order to use them as support for classification, indexation and retrieval processes in an integrated software system for libraries. Furthermore, the problem of storing and retrieval of full content of a publication is taken into consideration.

Keywords: digital library, classification, indexation, storage and index structures.

1 INTRODUCTION

Significant progress was made in computers and information technologies in last decades. So, today we have computers at work everywhere, from technical to socio-human and services fields. Growth in computing and storing capabilities, also with the possibility to interconnect different computer systems determined a radical change in the way we perceive and interact with a lot of today's real world concepts. One of them is *the library* concept. Computers and information technology introduced a new concept, that of *digital library*. The classical management methods used in a public library had to be changed and improved so they can benefit from the new technologies. A digital library may permit not only to store, in a digital form, classical information about books and other publications like author, title, publishing house, publication year, ISBN, ISSN, table of contents, abstract/full text, etc, but also has to offer users an easily, rapidly and accurate method for retrieving desired publications.

Often readers do not know the title and author of a book or publication they need or they want a publication that covers a specific field or subject of interest. The librarian has to be able to deliver the right books for them. To be able to do that, the librarian might use classification and indexation methods.

Classifying and indexing publications in a library is a very important task for a librarian and it is essential for future successful exploitation of the library assets. Digital libraries can be interconnected, so it is very important to have and use a similar method for classifying and indexing documents from everywhere. Another very hard problem is that libraries do not used or even use today the same format to store their data. If data will be put in the same format, this will make possible a distributed search in all connected libraries. With a computer aid it is possible to automate the classification and indexation process and also to retrieve publications which match some particular criteria from different interconnected digital libraries. Libraries data, classification data, indexation data have to be stored in a database for future processing so it is very important to fully understand their characteristics in order to make the best selection.

Application designers must decide whether to store binary large objects, in our case the actual content of a digital library, in a filesystem or in a database. Generally, this decision is based on factors such as application simplicity, manageability or system performance.

2 CLASSIFICATION AND INDEXATION METHODS FOR LIBRARIES

Librarians developed over the years different methods to classify and index library publications to be able to manage more easily the library content and to deliver to readers the right books. Many of these methods lost over the time, because they were difficult to apply and laborious, but some of them are still known and applied in different libraries. Unfortunately, there is not in the present a unique method accepted and applied by everyone from every library in the world. This weakness makes the

information exchange between libraries very hard in practice. If such a method will be adopted, computers could be used to manage more easily library publications classification, indexation and retrieval aspects. Our first work consists of analyzing different classification and indexation methods developed for libraries and we identify the best solutions from the point of view of a future digital library which has to be connected with other digital libraries. Below, we present three classification and indexation methods suitable for digital libraries.

2.1 Universal Decimal Classification Method

A standard method for publications classification is *universal decimal classification* (UDC). UDC is a system of library classification developed by the Belgian bibliographers Paul Otlet and Henri La Fontaine at the end of 19th century. It is based on the Dewey Decimal Classification (DDC), but uses auxiliary signs to indicate various special aspects of a subject and relationships between subjects. This was developed as a classification system for all human knowledge [1]. It can be used for so called primary documents like books, periodicals, audio-video documents, or for secondary documents like catalogs, syllabuses, bibliographies, etc.

UDC offers the possibility to group together all the materials referring to the same subject, expressed and localized in an undoubting manner. Digits are used as universal decimal codes and this is very important because digits have the same meaning in entire world. In this case linguistic barriers do not exist and international information exchange is possible.

Universal decimal classification can be considered as a base for terminology comparisons and can be used as an international terms code in all domains. In essence, UDC is a practical system for numerical codification of information, so that information can be easily retrieved regardless of the way it is perceived. Human knowledge, seen as a unit, is divided in ten big classes symbolized by decimal fractions. Each of these classes is divided in ten subclasses by adding a new digit to the code. The rule of dividing a class in ten subclasses by adding a new digit to the code is extended with respect to the principle of deriving from general to particular.

In practice, the subject of a document which needs to be classified is not always simple and clearly delimited. This makes it necessary to have more than a main UDC index for a document. Auxiliary universal decimal classification indexes are also used.

The document subject can be a complex combination of multiple aspects and it will be represented by different classification indexes tied together, or it can be a particular aspect of a main universal decimal classification aspect which implies that auxiliary universal decimal classification indexes are used.

2.2 Subject Headings Indexation Method

Document indexation is the process that describes the content of a document with the aid of special terms called descriptors. The principles and rules for select and validate the descriptors and to index the documents are subjects of standardization with the aim of a consistent and similar information process. One language all descriptors are called linguistic thesaurus of that language.

The linguistic thesaurus of a language is a standard descriptors list, alphabetically ordered, which indicates the semantic, hierarchical and associative logic relationships between them.

Descriptors are in indexing unique accepted forms, they have authority, and this is the reason why vocabulary and linguistic thesaurus are called in librarian's literature authority lists.

Nowadays there is more than one linguistic thesaurus used for indexing purposes. The best known and used are LCSH (Library of Congress Subject Headings) in English [2] and RAMEAU (Repertoire d' Autorite-Matiere Encyclopedique et Alfabetique Unifie) in French [3]. Both are encyclopedic linguistic thesaurus and are for one and only one language. This is a constraint which limits for the moment information exchange.

Specialized linguistic thesaurus, with descriptors dedicated to a specific domain, was developed also by professional associations, research centers or international organizations. Some of them are multi-linguistic to facilitate information exchange, but this is still at the beginning.

A successful access to documents is determined in a great deal by a correct and complete analysis of its content. Access points to publications subjects are called in literature *subject headings*. Indexing through subject headings mean to classify publications by access points to publications subjects, principal subjects developed in publication content.

The process of indexation has to take in consideration the following aspects:

- *Subject headings concision* – a subject heading has to express one and only one idea. Document subject headings have to express in a concise and brief manner the document content.
- *Subject headings objectivity* – subject headings have to reflect only the document content, they do not issue value judgment.
- *Subject headings specificity* – a document will not be indexed in the same time at General Terms and Specific Terms for the same subject heading.
- *Subject headings coherence* **–** documents will be indexed using subject headings

which are conform to standard rules introduced by LCSH and RAMEAU for defining subject headings. They transform natural language in a special language for indexation.

Indexation has three phases:

- *Document analysis* – the document must be read (table of content, abstract, full text, bibliography) with the aim to understand its content and to identify the subjects which are treated in it and can be used for indexation.
- *Subject headings selection* –this process is governed by special information needs expressed by library's users and by library and users profile.
- *Subject headings validation* – for a successful indexation the selected subject headings have to be correct concepts. These subjects must be validated by comparison with widely accepted subjects who are present in librarian's specific literature or specialized scientific and academic databases. All those are considered *auxiliary indexation tools*.

Subject headings are made of a principal entry called *heading entrance* and one or more subheadings. The heading-entrance must express the essence of document content: the concept, the main notion, the phenomenon or the process. It is possible to have a simple document and in this case the heading-entrance is enough to express the document content, but usually documents are complex and the document heading-entrance is followed by several *subheadings* to be more accurate. Those subheadings express more information about the subject, space localization, time period and pattern of the document.

The unit *heading entrance-subheadings* is called *subject headings*. Internal parts are separated by double stars ** or double lines – like this:

subject subheading **space localization subheading **time period subheading**pattern subheading

- *Subject subheading* – means a document content aspect which is important and is not covered by the heading-entrance (for example, Automotive**Engine design)
- *Space localization subheading* – express the spatial localization implied by the document content if this exists (for example, Agriculture**Corn plant**Romania)
- *Time period subheading* – express the temporal localization implied by the document content if this exists (for example, Agriculture**Corn plant**Romania**19th century)
- *Pattern subheading* – express the presentation format of the document: article, poster, biography, bibliography, dictionary, etc. (for example, Agriculture**Corn plant**Romania**Poster)

2.3 UNIMARC Method

UNIMARC (UNIversal MAchine Readable Cataloging) is a standard format to tag library's publications in a machine readable form [4]. Using it will make possible an easy information exchange between different libraries, because they all speak the same language. UNIMARC is recommended by IFLA (International Federation of Library Associations) to all public and private libraries.

In UNIMARC format a publication is recorded by its content in a *block of subjects*. This block describes document content and is made of *classification fields* and *indexation fields*.

Usually libraries working with UNIMARC format use both methods: classification and indexation, for analyzing and tagging publications. These make the process of information retrieving more precise and efficient in case of both detailed and specific information research. UNIMARC format permits also to localize the publication in the library shelf if the quote indexes are used for publications [5].

In Romanian public libraries is very important to use in future parallel classification and indexation methods, because until now only classification through universal decimal codes was used. UNIMARC format uses subject headings method for indexation and is suitable for recording data in a library database. Pre-coordination in indexation suppose to access authorization lists and linguistic thesaurus created in advance and is very difficult to be done by librarians in absence of those indexation instruments in their native language. For example, we don't have yet a complete linguistic thesaurus and authorization lists defined in Romanian language. The Romanian National Library's specialists are working on it. The librarian has also to translate the subject headings from his native language to English or French to be able to validate it according with LCSH or RAMEAU, which are the best known and used linguistic thesaurus. This burden is a very difficult one for Romanian librarians and our project intend to develop a software tool to assist them.

A specialized information system would be developed for an automatic indexation and classification process of library assets: books, periodicals, audio-video documents, catalogs, syllabuses, bibliographies, etc. Specialized informatics systems could be used for an automatic indexation process if library publications are in a digital form.

3 DBMS EVALUATION FOR DIGITAL LIBRARIES SUPPORT

An automatic classification and indexation system implies undoubting a database. These systems have to be able to store and manage all the data needed in the process of classifying and indexing publications, such as universal decimal codes with their principal and auxiliary indexes, language linguistic thesaurus, subject headings with their entrance-heading and subheadings, in order to process libraries digitized documents. Classifying and indexing documents also produces outputs that had to be stored in a database for better management.

In order to develop our system, we made a study about today's database management systems characteristics at the physical level of the database architecture. Our goal was to fully understand these characteristics and to identify the best solutions to be implemented for a digital library support.

Today's software systems requirements are more and more complex and variable so database management systems (DBMS) producers have to face new challenges. This is the reason way they permanently improve their systems, implementing new capabilities. Storage structures and access methods of database management systems have been changed lately and today systems designers have to be able to choose the best solution for physical database model from a variety of possibilities. Informatics systems usually rely on a database and the success of future applications are in a great deal determined by database structure and the way data accesses are made. Systems performance in their operational phase is influenced in a big percentage by physical database design. If the designer makes the wrong choices during the database design process, the data could not be easily accessed later as they are needed and the success of the entire system will be compromised. So, it is very important to know and understand these new capabilities of DBMS with the aim to choose the right solution for the problem you want to solve.

We evaluated storage and index characteristics of three of the most important today's database management systems in order to support an automated classifying and indexing library system. Our analyze was made on Oracle 10*i*, Oracle Corporation product, SQL Server, Microsoft Corporation product, and DB2, IBM Corporation product. We choose these products because they are the best database management systems on the market today. We studied storage structures for data tables and index structures implemented in these DBMS.

3.1 Oracle 10i

In Oracle 10*i* data can be stored in different types of tables. Table structure characteristics are different from type to type and make it more suitable for a particular type of application then other.

Oracle 10*i* can store data in one of the following data tables [6]:

Heap tables store table rows in file data blocks as variable length records. It is the most common table structure. Data types can be system defined data types or user defined data types. System defined data types can be classical scalar types like: CHAR, NCHAR, VARCHAR2, NUMBER, DATE, etc., types for storing large data objects like: LONG, LONG RAW, LOB, BLOB, etc., collection data types like: VARRAY and TABLE (nested table), or reference type REF used for implementing object identity in an object-relational database. There are also data types extensions called Data Cartridges that can be used for complex data types like: text, audio-video, images, time series, spatial data, etc. These tables are suitable for storing publications data.

Partitioned tables store table rows in different data segments according with a partitioning method. All rows with the same partitioning method value are stored together in a data segment and these segments can be stored in different table spaces on the same or on different storage spaces. Partitioned tables are useful for large data tables with a lot of concurrent processing. System performance can be improved because queries can be directed only to those partitions containing data, or DML (data manipulation language) operations can be performed in parallel with a higher grade on different partitions. Also join operation between tables can benefit from partitioned tables if tables are partitioned on the same rule.

Oracle 10*i* offers several table partitioning methods designed to handle different application scenarios:

- *Range partitioning* uses ranges of column values to map rows to partitions. Partitioning by range is particularly well suited for historical databases or for large databases in which an old data package must be replaced from time to time with a new one.
- *Hash partitioning* uses a hash function on the partitioning columns to stripe data into partitions. Hash partitioning is an effective means of evenly distributing data.
- *List partitioning* allows users to have explicit control over how rows map to partitions. This is done by specifying a list of discrete values for the partitioning column in the description for each partition.
- *Range-hash partitioning* uses a mixture of range and hash partitioning methods to map rows to partitions.
- *Range-list partitioning* uses a mixture of range and list partitioning methods to map rows to partitions.

Partitioned tables are suitable for a distributed digital library system or in case of interconnected digital libraries for a central metadata repository.

Index-organized tables store table rows directly in an index structure. Leaf nodes of the B-tree index store table rows directly and this eliminates the additional storage required for ROWID (row identifier), which store the addresses of rows in ordinary tables and are used in conventional indexes to link the index values and the row data. Index-organized tables are build on primary key and provide fast access to table data for queries involving exact match and/or range search on the primary key. Queries involving other columns values are much slower. Also DML operations can be slower when they imply index structure reorganization.

Index-organized tables are suitable for storing universal decimal classification codes or linguistic thesaurus data or subject headings data of an automated classification and indexation system.

Clustered tables store table rows offering some degree of control over how rows are stored. Oracle server stores all rows that have the same cluster key value in the same block if this is possible. When data are searched by cluster key value all records are together and they could be obtained in a single disk access. A clustered table can be used also to store related sets of rows from different database tables within the same Oracle server block. This is very efficient when database queries imply joins on those tables on cluster key. The cluster can be an index cluster or a hash cluster according to the way the rows location is generated. For an automated classification and indexation system, universal decimal codes table and linguistic thesaurus table are good candidates for clustered tables.

Systems performance can be also improved if supplementary access data structures like indexes are used. An index is a tree structure that allows direct access to a row in a table. Indexes are built on an index key, which can be a single column key or a concatenated column key. An index can be a unique index or a non-unique one.

Oracle implements different types of index structures [6]:

Normal key B-tree index – is a single column key or a concatenated one with unique or non-unique values. Index can be created on ascending or descending values of the index key. This is the most common index structure and every database table could have several indexes created on index keys used in search criteria.

Reverse key B-tree index – index key bytes are reversed before the index is build. This structure is suitable for massive parallel data processing because it reduces concurrency conflicts.

Bitmap index – the leaf nodes of the index structure tree contain a bitmap not ROWID-s. Each bit in the bitmap corresponds to a table row and if it is set means that the row contains the key value. They are more compact, suitable for low-cardinality columns and are very useful when DML operations are seldom.

Function-based index – a function is applied to the index key columns before the index is created.

Bitmap join index – is an index structure which spans multiple tables and improves join operations performance on those tables. A bitmap join index can be used to avoid actual joins of tables, or to greatly reduce the volume of data that must be joined, by performing restrictions in advance. Queries using bitmap join index can be sped up via bit-wise operations. They are very useful for tables with frequently join operations between them.

Local partitioned index – is an index for a partitioned table which has the same index key as the partition key. If database tables are partitioned their existence is imperative.

Global partitioned index – is an index structure for a normal or partitioned table, which is partitioned and stored separately using a partition key. It is suitable for multiple concurrent accesses on the database.

Global non-partitioned index – is an index structure for a partitioned table. . If database tables are partitioned their existence is imperative.

3.2 DB2

DB2 offers two structure possibilities for storing data in a database [7]. These are:

Heap tables store table rows in no particular order in files data blocks. It is the most common table structure. Classical scalar system defined data types or user defined data types are possible. To manage new complex data types like text, audio, video, images, spatial data, etc., IBM introduced DB2 Extenders. These tables are suitable for storing publications data.

Partitioned tables are present also in DB2, but only hash partitioning method is available. This is a considerable limitation compared with Oracle partitioning capabilities.

Partitioned tables are suitable for a distributed digital library system or in case of interconnected digital libraries for a central metadata repository.

Indexing capabilities in DB2 are a little bit reduced than in Oracle. DB2 support following index structures [7]:

Normal key B-tree index – is a single column key or a concatenated one with unique or non-unique values. Index can be created on ascending or descending values of the index key. DB2 doesn't have reverse key indexes but it allows reverse scans on normal key indexes. This is the most common index structure and every database table could have several indexes created on index keys used in search criteria.

Clustered indexes are built like index-organized

table structures but they are an additional structure for a data table and columns are duplicated in both the table and the index. They provide fast access to table data for queries involving exact match and/or range search on the index key because table rows are stored in the leaf nodes. Only one clustered index per table can be created.

Bitmap index – DB2 supports only dynamic bitmap indexes created at run time by taking the ROWID from existing regular indexes and creating a bitmap out of all the ROWID-s either by hashing or sorting. For this reason, they do not provide the same query performance like static bitmap indexes and databases do not receive any of the space savings or index-creation time savings compared with static bitmap indexes.

Function-based index – the index can be created based on the expression used to derive the value of the generated column.

Local index – is a local index for a partitioned table which has the same index key as the partition key. Global indexes are not possible in DB2. If database tables are partitioned their existence is imperative.

3.3 SQL Server

SQL Server offers also two structure possibilities for storing data in a database [8], but it is much more restrictive then the other two DBMS systems.

Heap tables store table rows in no particular order in files data blocks. It is the most common table structure in SQL Server. Data types can be classical scalar system defined data types or user defined data types. For complex data SQL Server has new data types like: TEXT, NTEXT, IMAGE, etc. These tables are suitable for storing publications data.

Table 1: Analized DBMS characteristics

Feature	Database Management System		
	Oracle 10i	***DB2***	***SQL Server***
Heap tables	Yes	Yes	Yes
Partitioned tables	Yes	Yes	Partial
Hash partitioning	Yes	Yes	No
Range partitioning	Yes	No	No
List partitioning	Yes	No	No
Range-hash partitioning	Yes	No	No
Range-list partitioning	Yes	No	No
Index-organized tables	Yes	Partial	Partial
Clustered tables	Yes	No	No
Normal key B-tree index	Yes	Yes	Yes
Reverse key B-tree index	Yes	No	No
Bitmap index	Yes	Yes	No
Function-based index	Yes	Yes	Yes
Bitmap join index	Yes	No	No
Local partitioned index	Yes	Yes	Yes
Global partitioned index	Yes	No	No
Global non-partitioned index	Yes	No	No

Member tables store table rows in federated database architecture. SQL Server does not support partitioning as generally defined in the database industry. A federation of databases is a group of servers administered independently, but which cooperate to share the processing load of a system. The data are divided between the different servers and are stored in member tables. Because federation servers do not share the same system catalog, in fact each database server has his own system catalog, system performance and scalability is very low. When a user connects to a federated database he is connected to one server. If he requests data reside on a different server, the retrieval takes significantly longer than retrieving data stored on the local server and all remote servers has to be consulted. To improve a little bit this situation SQL Server introduces *distributed partition view* concept. A distributed partition view joins horizontally partitioned data from a set of member tables across one or more servers, making the data appear as if from one table. The data can be partitioned between member tables only on ranges of data values in one of the table column.

SQL Server implements much less index structures [8]:

Non-clustered index – it is a normal key B-tree index with a single column key or a concatenated one, with unique or non-unique values. Index can be created on ascending or descending values of the index key. This is the most common index structure and every database table could have several indexes created on index keys used in search criteria.

Clustered indexes are built like index-organized table structures but they are an additional structure for a data table. They provide fast access to table data for queries involving exact match and/or range search on the primary key because table rows are stored in the leaf nodes of the primary key index. Only one clustered index per table can be created.

Partitioned index - it is a local index on a member table. SQL Server does not support global indexes. If a federation database architecture is used their existence is imperative.

Function-based index – a function is applied to the index key columns before the index is build.

Table 1 presents a synthesis of found

characteristics on analyzed database management systems.

4 STORING THE CONTENT

The purpose of a digital library is to provide a central location for accessing information on a specific topic. An essential decision that has to be made in the process of designing a digital library is the choice on how to store the data.

4.1 File formats used in DL

In [9] we can find an overview on the main concepts surrounding file formats in a digital library environment, and the importance of choosing a file format that can suit the needs of such a system.

In the context of digital libraries, the file format is a set of specifications on how to represent information on a physical drive or in a database. File formats are targeted towards specific types of information, as for instance JPEG and TIFF for raster images, PDF for document exchange or TXT for plain text.

A number of factors have to be taken into account before venturing to choose one format or another. A few formats have gained a more considerable share of use due to certain advantages, also with this widespread use being an advantage in itself. However, all formats must be taken into account, also bearing in mind that acquisition and storage can be done in a different format than the distribution.

A series of criteria must be studied and correlated with the individual needs of the client. It is also important to keep in mind future requirements and prospects of expansion, so as to avoid the need for migration.

Migration is the transferring of data to newer system environments ([10], [11]). This may include conversion of resources from one file format to another (e.g., conversion of Microsoft Word to PDF or OpenDocument), or from one operating system to another (e.g., Windows to Linux), so the resource remains fully accessible and functional.

Migration can be necessary as formats become obsolete, or as files need to be transferred on another system. Resources that migrate run the risk of losing some of their functionality, since newer formats might be incapable of rendering all of it from the original format, or, more so, the converter itself may be unable to interpret the original format in its entirety. Conversion is often a concern with proprietary data formats. Therefore, migration is an undesirable process, and a good choice of file formats can reduce the risk of ending up in the need of migrating data.

Generalised use of a specific format can be an argument in favour of migrating data to that format, or against migrating data away from it. For example, even though Jpeg2000 is deemed superior to Jpeg, few migrate towards it, due to the wide adoption of the latter.

4.2 BLOBs and external files

We have the choice of storing large objects as files in the filesystem, as BLOBs (binary large objects) in a database, or as a combination of both. Only folklore is available regarding the right path to take – often the design decision is based on which technology the designer knows best. Most designers will tell you that a database is probably best for small binary objects and that that files are best for large objects. A good study on the subject can be found in [12]. The study indicates that if objects are larger than one megabyte on average, NTFS has a clear advantage over SQL Server. If the objects are under 256 kilobytes, the database has a clear advantage. Inside this range, it depends on how write intensive the workload is, and the storage age of a typical replica in the system. However, using different DBMS or file systems can change the results.

Filesystems and databases take different approaches to modifying an existing object. Filesystems are optimized for appending or truncating a file. In-place file updates are efficient, but when data are inserted or deleted in the middle of a file, all contents after the modification must be completely rewritten. Some databases completely rewrite modified BLOBS; this rewrite is transparent to the application. To ameliorate the fact that the database poorly handles large fragmented objects, the application could do its own de-fragmentation or garbage collection

Applications that store large objects in the filesystem encounter the question of how to keep the database object metadata and the filesystem object data synchronized. A common problem is the garbage collection of files that have been “deleted” in the database but not the filesystem. Operational issues such as replication, backup, disaster recovery, and fragmentation must be also considered.

Storing BLOB data in the database offers a number of advantages such as offering an easier way to keep the BLOB data synchronized with the remaining items in the row. BLOB data is backed up with the database. Having a single storage system can ease administration. Full Text Search (FTS) operations can be performed against columns that contain fixed or variable-length character data or against formatted text-based data, for example Microsoft Word or Microsoft Excel documents.

A well thought out metadata strategy can remove the need for resources such as images, movies, and even text documents to be stored in the database. The associated metadata could be indexed and include pointers to resources stored on the file system.

5 CONCLUSIONS

Computers era brought radical changes in our life. The classical management methods used in a public library had to be changed and improved so they can benefit from the new technologies. Classification and indexation process has to be tailored to be suitable for a computer aid. New methods are proposed but today's public libraries, Romanian or world around, do not have their data in a uniform format so it is a very difficult task to make information exchange between them work properly. If data will be put in a standard format, this will make possible a distributed search in all connected libraries. With a computer aid it will be possible to automate the classification and indexation process and to perform semantic searches in different interconnected digital libraries. Libraries data, classification data, indexation data have to be stored in a database for future processing, so it is very important to fully understand their characteristics in order to make the best selection. Our goal was to analyze different classification and indexation methods used today in public libraries and to identify the best suitable method for a computer automated system. We also evaluated storage and index characteristics of three of the most important today's database management systems in order to support an automated classifying and indexing library system and distributed semantic searches among interconnected libraries. *The good* and *the bad* choices were revealed for each particular data structure and access method. This study can be useful, too, for other software applications developers who had to make the best DBMS selection for their future software system.

The choice between a DBMS and a filesystem for storing usual DL data is considered also.

ACKNOWLEDGEMENT

This work was partially supported by the Romanian National Council of Academic Research (CNCSIS) through the grant CNCSIS no. 12099/2008-2011.

6 REFERENCES

[1] Universal Decimal Classification Handbook, Central Library of the "Lucian Blaga" University of Sibiu, (1995).
[2] Library of Congress Subject Headings, Library of Congress, USA, (1999).
[3] RAMEAU-Repertoire d 'Autorite - Matiere Encyclopedique et Alfabetique Unifie, France National Library, (2002).
[4] UNIMARC Handbook: Bibliographic format. Concise version, France National Library, (1994).
[5] UNIMARC Handbook: Authorities lists format, Manual UNIMARC : Format des notices d'Autorite, France National Library, 2004.
[6] Oracle 10*i* Technical Report. www.oracle.com
[7] DB2 UDB Technical Report. www.ibm.com
[8] SQL Server Technical Report www.microsoft.com
[9] D. Volovici, A.G. Pitic, A. C. Mitea, A.E. Pitic: An analysis of file formats used in digital libraries, First International Conference on Information Literacy in Romania, Sibiu, 2010
[10] J. Garrett, D. Waters, et all: Preserving digital information: Report of the task force on archiving of digital information, Commission on Preservation and Access and the Research Libraries Group, 1996
[11] H. M. Gladney: Principles for digital preservation, Communications of the ACM 49, 2006
[12] R. Sears, C. van Ingen, J. Gray: To BLOB or Not To BLOB: Large Object Storage in a Database or a Filesystem? , Technical Report, MSR-TR-2006-45, 2006