A Novel Design of Information Retrieval System for Digital Libraries

Nitin Gupta[1], Dr. Komal Kumar Bhatia[2], Arundhati Walia[1]

[1]Assistant Professor
HRIT,
Ghaziabad, India

[2]Associate Professor
YMCA
Faridabad, India

ABSTRACT

With the rapid growth of the world-wide web,the general purposecrawler and search engine poses scaling challenges. In this content, digital libraries plays vital role as the information available in Digital links belongs to almost every domain. In this paper a Novel Design of Information Retrieval System for Digital Libraries is being proposed. The goal of proposed design is to selectively download pages that are relevant to a pre-defined set of Digital Library. Thus, these areneeded to download this type of information that contains various research documents and other multimedia data.

Keywords: WWW, OAI-PMH, DC, DDL,URLs, BOF, EOF, FIFO.

1. Introduction

World Wide Web, abbreviated as **WWW** or **W3** and commonly known as **the Web**, is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks. Digital information retrieval is a process of extracting large amount of information from World Wide Web. At CERN in Geneva, Switzerland, Berners-Lee and Belgian computer scientist Robert Cailliau proposed in 1990 to use Hyper Text to link and access information of various kinds as a web of nodes in which the user can browse at will",[2] and publicly introduced the project in December.[3]

"The World-Wide Web was developed to be a pool of human knowledge, and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project."[4]

1.1 General Web Search Engine

A convention search engine retrieves the information from WWW [6]. The major modules of a web search engine are a Crawler, an Indexer, a Query Engine and a Ranking Engine.

- **crawlers:** A crawler is a software that traverses a web automatically and downloads the Pages for search engine. It downloads the pages by using seed URL.
- **Repository:** All the pages downloaded by crawler is kept in a temporary storage called As repository.
- **Indexer module:** Indexer module does the indexing. Indexing creates the compressedDescription of pages.The crawling and indexing process is a query independent process.
- **Query module:** User fires the query to search engine using this module. This module is aQuery dependent module.
- **Ranking:** Ranking module places the most relevant page on the top. It is calculated by theFormula PR(a)=q+(1-q)∑PR(pi)/C(pi).

The conventional search engine is not able to search digital information like digital files, e-prints/pre-print files. The amount of the digital information is growing tremendously since last 10 years. The proposed method helps to search indexed digital information which is beyond the reach by conventional search engine.

1.2 Review of OAI-PMH

The interoperability of e-print/pre-print and digital data servers can be increased by usingOpen Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Such a servers host a scientific and technical papers. The rising cost of technical journals head lead to the development of pre-print servers. This allows the scholars and researchers to deposit their articles into these servers which allow rapid extraction of information compared to the traditional print journals. The growth of e-print/pre-print repository creates the problem of information overload as well as some other general problem like:

- The end-users/scholars may not be able to know the existence of a repository.
- Overlapping of coverage in terms of subjects.
- Multi-disciplinary nature of subjects needed the documents to be kept at a number of repositories.
- Discipline-specific and institution-specific archives created duplication efforts.
- The end-users/scholars had to search individual repositories to get documents of his interest.
- Also, it was undesirable to require scholars to deposit their work in multiple repositories.

To solve the above motioned problems, there is need to from a framework to integrate these e-print/pre-print archives. The major need is to define an interface to permit e-print servers to expose their metadata for the articles it held so that search services could harvest its meta data.

1.3 Digital Data Link (DDL)

Digital data linkis defined as ‘'data about data'' describing the information about an object. The NSDL Metadata Primer [5] defines metadata as "Structured, standardized descriptions of resources. The resources may be digital or physical resource which aids in retrieval and use of these resources for example indexing in book is a good example of metadata describing the details about the chapter, title in each chapter and page number respectively. Metadata can be produced for all sorts of objects like Books, Journals, Images, and Learning Materialsetc. Moreover various metadata standards MARC for material in library catalog, MPEG for Images, LOM for Learning Materials etc.Metadata therefore allows a precise and standardized way of describing content in discrete packages called metadata records.

The Dublin Core is a metadata standard for describing a range of digital objects, and contains a set of 15 metadata elements (e.g. Title, Creator, Subject, Description, Publisher, Contributor, Date, etc.). Dublin Core is important as it is often mandated as a minimum metadata requirement [7][8].A *simplified* example of a Dublin Core (dc) metadata record describing this article is included below.

Table 1: The Dublin Core Elements[1]

THE DUBLIN CORE ELEMENTS
1. TITLE The name given to the resource by the CREATOR or PUBLISHER.
2. CREATOR The person(s) or organization(s) primarily responsible for creating the intellectual content of the resource.
3. SUBJECT The topic of the resource: keywords or phrases that describe the subject or content of the resource, including controlled vocabularies or classification schemes.
4. DESCRIPTIONS A textual description of the content of the resource, including abstracts in the case of document-like objects or content descriptions in the case of visual resources.
5. PUBLISHER The entity responsible for making the resource available in its present form, such as a publisher, a university department or a corporate entity.
6. CONTRIBUTOR Person(s) or organization(s) in addition to those specified in the CREATOR element who have made significant intellectual contributions to the resource but whose contribution is secondary to the individuals or entities specified in the CREATOR element (for example, editors, transcribers and illustrators).
7. DATE The date the resource was made available in its present form.
8. TYPE The category of the resource, such as home page, novel, poem, working paper, technical report, essay, dictionary. It is expected that TYPE will be chosen from an enumerated list of types.
9. FORMAT The data representation of the resource, such as text/html, ASCII, Postscript file, executable application or JPEG image.
10. IDENTIFIER String or number used to uniquely identify the resource. Examples for networked resources include URLs and URNs (when implemented). Other globally unique identifiers, such as International Standard Book Numbers (ISBN) or other formal names, would also be candidates for this element.
11. SOURCE The work, either print or electronic, from which this resource is derived, if applicable.
12. LANGUAGE Language(s) of the intellectual content of the resource.
13. RELATION Relationship to other resources. The intent of specifying this element is to provide a means to express relationships among resources that have formal relationships to others, but exist as discrete resources themselves.
14. COVERAGE The spatial and temporal characteristic of the resource. Formal specification of COVERAGE is currently under development.
15. RIGHTS The content of this element is intended to be a link (a URL or other suitable URI as appropriate) to a copyright notice, a rights-management statement or perhaps a service that would provide such information dynamically.

1.4 Some Existing ServiceProviders[11]

The job of service providers is similarly to web crawler of search engines. Service provider harvest the metadata exposed by data provided by individually going to their repositories. This harvested metadata is later on collected in database in XML format and then passed to provide and integrated search interface and browsing indices

1.4.1.OAIster

Description: OAIster is a project of the University of Michigan Digital Library Production Services, originally funded through a Mellon grant. The goal is to create a collection of freely available, difficult-to-access, academically-oriented digital resources that are easily searchable by anyone.

Homepage: http://oaister.umdl.umich.edu/o/oaister/

1.4.2. Networked Computer Science Technical Reference Library

Description: The Networked Computer Science Technical Reference Library (NCSTRL - pronounced as "ancestral") is an international collection of computer science research reports made available for non-commercial use from over 100 participating organizations worldwide.

The organizations that participate in NCSTRL include Ph.D. granting computer science departments, research laboratories, ePrint repositories, and electronic journals. The documents in NCSTRL are almost all textual, ranging in size from 100-plus page doctoral dissertations to short technical reports.

Homepage: http://www.ncstrl.org

1.4.3.iCite: CITATION INDEXING

Description: iCite is a citation indexing service based on OAI-PMH by ScuolaInternazionaleSuperiore di StudiAvanzati (SISSA, International School for Advanced Studies), Italy. It allows searching 3613394 citations in 150984 documents (as on February 20, 2003).

Homepage: http://icite.sissa.it:8888/icite/

1.4.4. Electronic Thesis/Dissertation OAI Union Catalog

Description: This is a service built by harvesting metadata from Open Archives of electronic theses and dissertations. The underlying technology is based on layered Open Archives with data being harvested from source archives and then stored in a Union Catalog. This Union Catalog is then front-ended with a search engine for demonstration purposes, but the data is just as easily accessible to other service providers, both local and remote.

1.5 Problem associated with the above maintained service providers.

The mentioned search technologies are not able to index the digital libraries as the crawler used by them does notdownload them; we proposed a framework, a novel architecture which will help in searching the shared digital libraries with the help of OAI-PMH and Dublin Core (DC).

2. Proposed Framework for Searching the Digital Information

To search the e-print files, we need to collect the data from different sources like web sites or from different digital libraries, content of Books, Journals, Images and Learning Materials etc. Thiscollected information from different web sites links is stored in DDL-Link as shown in Fig. 1. The crawler pick one by one links and deposits in web repository where it is to be first converted into text file and then check for any duplicate data. Duplication of data is computed [9, 10] by using fingerprint [10] for each web pagethat is a succinct (say 64-bit) digest of the characters of that page. If fingerprints of two web pages are equal then we test whether the pages themselves are equal, and if ithappens to be equal then we declare one of them to be a duplicate copy of the other. If the page contains the duplicate data then live that page and then go for the next one. In such a situation, we only index one copy of web page. After getting the 100% fresh page we send it to the Dublin Core for Metadata (the different 15 element of the metadata) and for more detail we get download 512kb of file from beginning of file (BOF) and end of file (EOF) each (1 MB of metadata from each file).

From the text we make the Inverted Index after applying the Tokenization by which we get the token from the text.

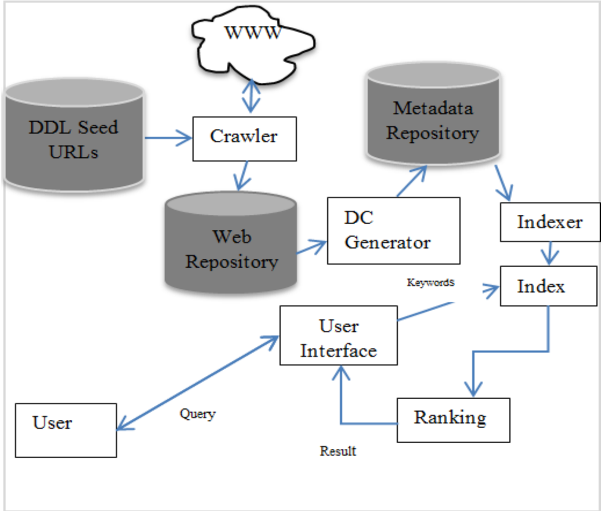


Fig.1: Framework of Digital Data Information Retrieval System

2.1 Component Wise Details

2.1.1. Digital Data Link (DDL) Seed URLs

Digital data link seed URLs consist of pre-defined links which is updated time to time, from which we select one link at a time and send it to the Crawler to fetch the data and send it to web repository as shown in Fig.2. Digital seed URL is data structure that contains all URLs to be downloaded DDL seed URLs is implemented by collection of FIFO sub queue in which it is placed is determined by URLs and canonical host name.

2.1.2. Crawler

Crawlersare software programs that traverse the World Wide Web by following hyperlinks extracted from hypertext documents. Crawlers are associated by web search engine to download documents for indexing and ranking. It traverses the web by downloading documents and following embedded links from page to page.

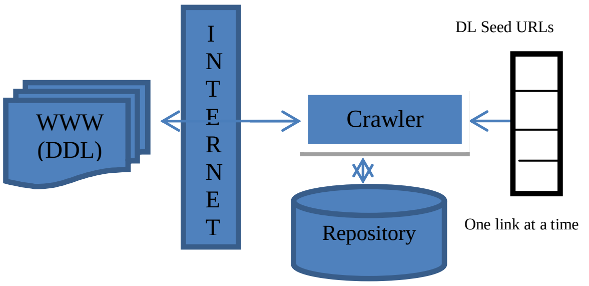


Fig.2: Working of DL Seed URLs

2.1.3. Web Repository

The web crawler downloads the same document multiple times,More over in many cases documents are mirrored on multiple servers to prevent processing of document more than once, web crawler performs fingerprint test to decide if document has already been downloaded or not. In fingerprinting we check whether the document is pre-download or fresh one by checking it with the previse file.If it is not fresh we leave that file and go for the next file and again do the same process till we get the fresh file. Later the file is send to the DC generator for further processing.

Fig. 3 shows a data structure called document fingerprint set storing a 64 bit checksum of the contents of each download documents.

The data structure maintains two independent set of fingerprints

- A small hash table kept in memory.
- A large sorted list kept in a single disk file.

2.1.4. Dublin Core

In this we take the fresh file and fetch the data (as per the 15 element required) and store it in the metadata as well as 512 kb of file from BOF and EOF (1 MB). The 15 element are taken as per Table 1.

Algorithm-The Design of Information Retrieval System for Digital Libraries

```
1) Read a file form the Web Repository.
2) Convert the file in the Text format.
3) Take the finger print of the file.
4) Get it check from previous finger print,
        If it is their go to step-1 for the next record.
     Else
        Add it in the finger print record.
5) Pass the file to DC to get the 15 element of the DC.
6) Store it in the metadata.
7) Download the text from the file 512 KB from both BOF
   and EOF.
8) Get it store in the metadata.
9) Get all the metadata indexed for further used.
```

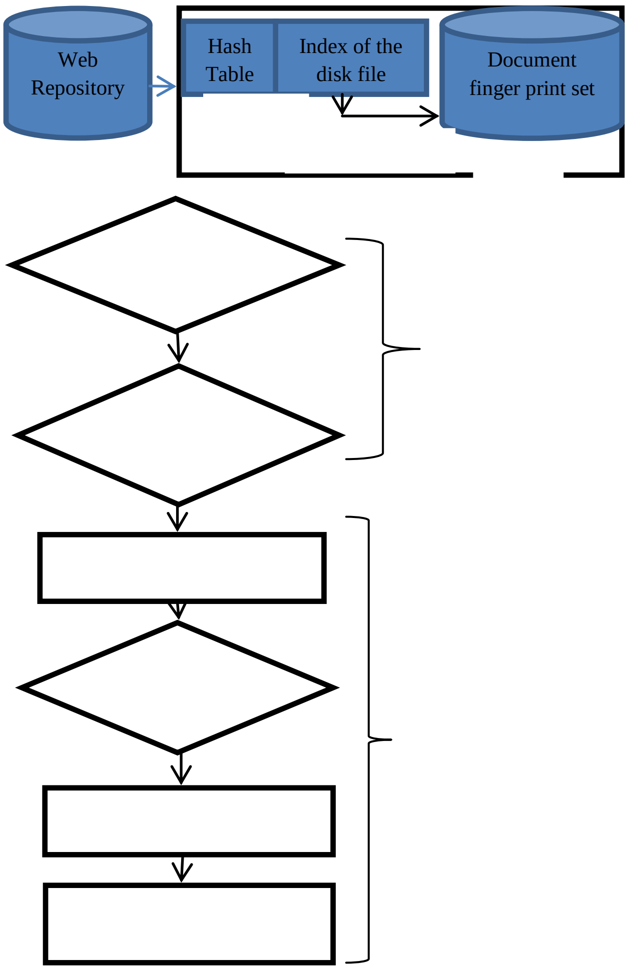


Fig. 3Fingerprint test flow

The algorithm shows the process which we have proposedin the framework, in which we have download the file from the digital link seed URL from which we take the URL one by one and get download the data and send to the Web Repository in which the file is first converted into text and then we apply the Finger print on the text file to see that the file is previously downloaded or not.If the fingerprint match with the previous finger print we live that file and for the next file and see its finger print until we get the new file.

When we get the new file it is send to the Dublin Core to fetch the 15 element and the 512 kb of data from both side(from the BOF and EOF) by which we get the 1 MB of file, then we store it in the metadata for further use. After it we send it to get the index (inverted index).

When the user make any request from search engine in which it can search with many choice or single word, which fetch from the indexer to index with various types of ranking methods which gives the best result from all metadata. The flowchart explains the complete process as shown in the fig. 4.

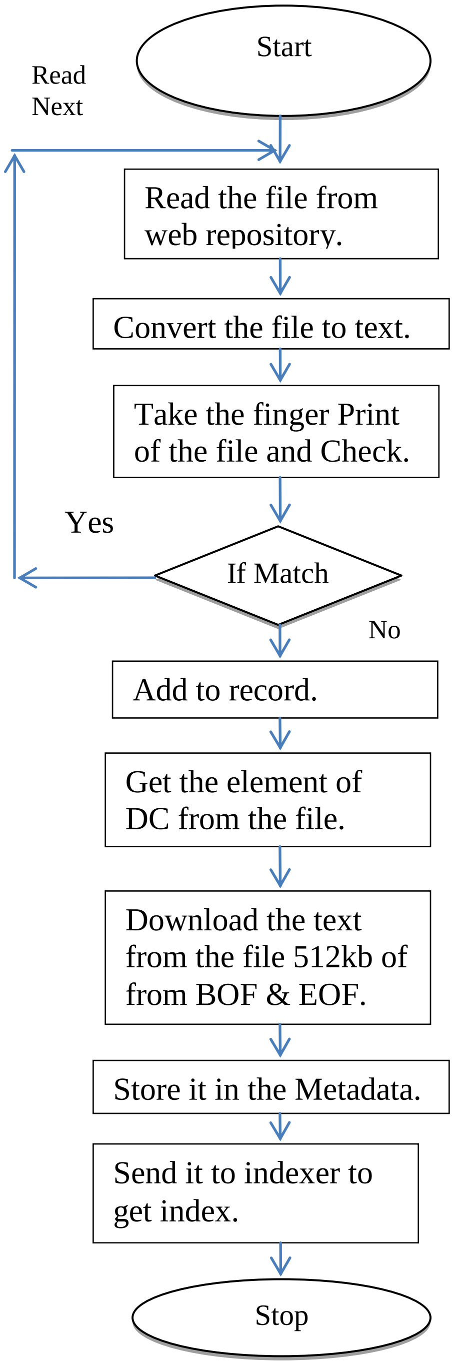
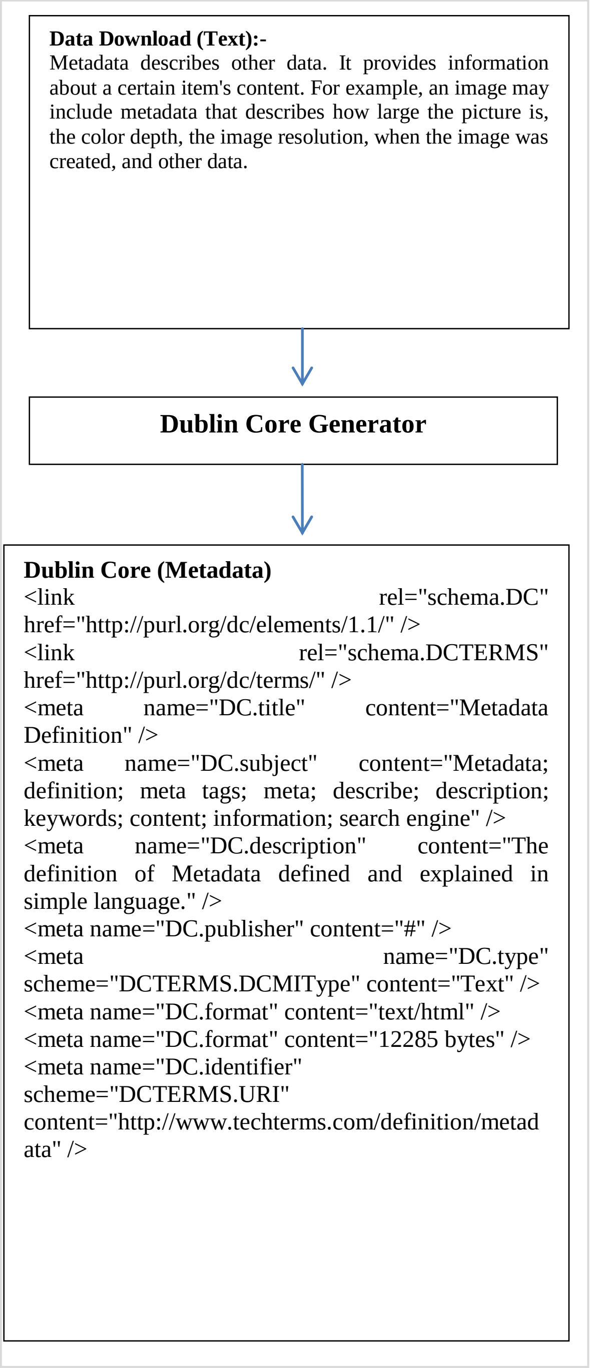

Start
Read Next
Read the file from web repository.
Convert the file to text.
Take the finger Print of the file and Check.
Yes
If Match
No
Add to record.
Get the element of DC from the file.
Download the text from the file 512kb of from BOF & EOF.
Store it in the Metadata.
Send it to indexer to get index.
Stop

Fig 4: Process Flowchart

Example :1

(http://www.techterms.com/definition/metadata)



Data Download (Text):-
Metadata describes other data. It provides information about a certain item's content. For example, an image may include metadata that describes how large the picture is, the color depth, the image resolution, when the image was created, and other data.

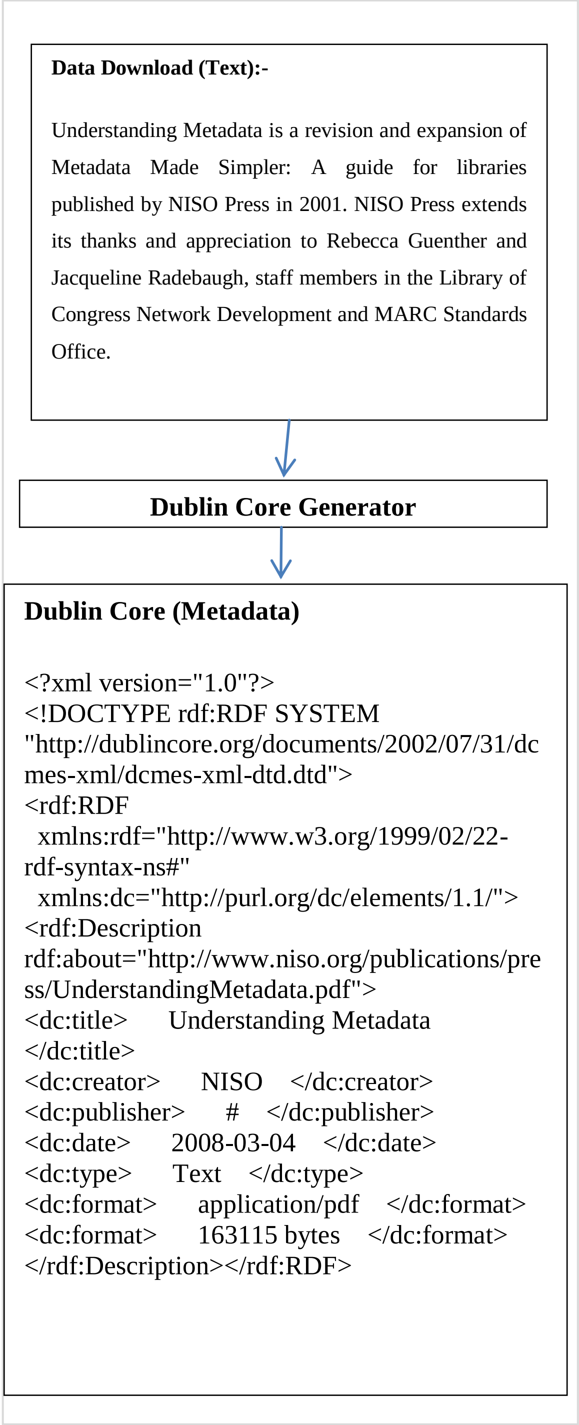
Dublin Core Generator

Dublin Core (Metadata)

```
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
<meta name="DC.title" content="Metadata Definition" />
<meta name="DC.subject" content="Metadata; definition; meta tags; meta; describe; description; keywords; content; information; search engine" />
<meta name="DC.description" content="The definition of Metadata defined and explained in simple language." />
<meta name="DC.publisher" content="#" />
<meta name="DC.type" scheme="DCTERMS.DCMIType" content="Text" />
<meta name="DC.format" content="text/html" />
<meta name="DC.format" content="12285 bytes" />
<meta name="DC.identifier"
scheme="DCTERMS.URI"
content="http://www.techterms.com/definition/metadata" />
```

Example :2

(www.niso.org/publications/press/UnderstandingMetadata.pdf
)



Data Download (Text):-

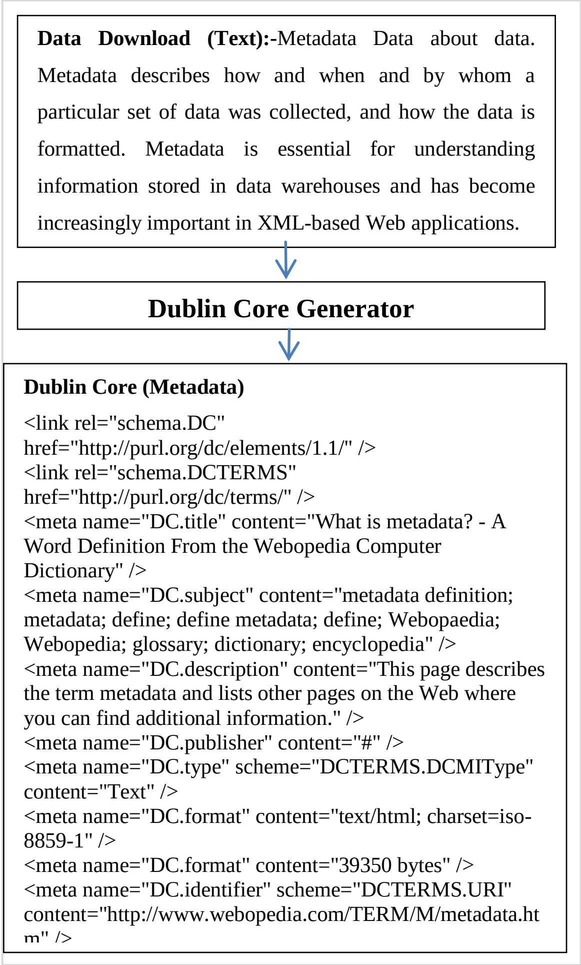
Understanding Metadata is a revision and expansion of Metadata Made Simpler: A guide for libraries published by NISO Press in 2001. NISO Press extends its thanks and appreciation to Rebecca Guenther and Jacqueline Radebaugh, staff members in the Library of Congress Network Development and MARC Standards Office.

Dublin Core Generator

Dublin Core (Metadata)

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF SYSTEM
"http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-dtd.dtd">
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
rdf:about="http://www.niso.org/publications/press/UnderstandingMetadata.pdf">
<dc:title>    Understanding Metadata
</dc:title>
<dc:creator>    NISO    </dc:creator>
<dc:publisher>    #    </dc:publisher>
<dc:date>    2008-03-04    </dc:date>
<dc:type>    Text    </dc:type>
<dc:format>    application/pdf    </dc:format>
<dc:format>    163115 bytes    </dc:format>
</rdf:Description></rdf:RDF>
```

Example :3
(www.webopedia.com/TERM/M/metadata.htm)



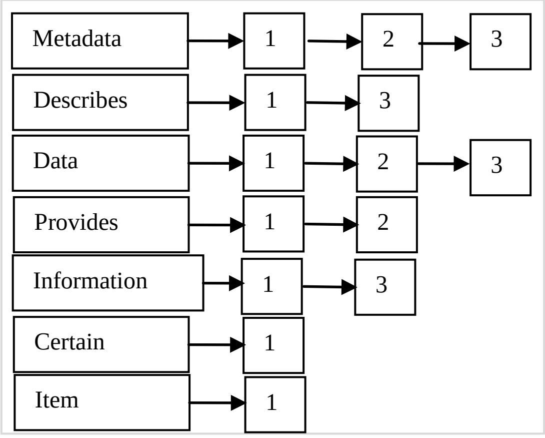
Data Download (Text):-Metadata Data about data. Metadata describes how and when and by whom a particular set of data was collected, and how the data is formatted. Metadata is essential for understanding information stored in data warehouses and has become increasingly important in XML-based Web applications.

Dublin Core Generator

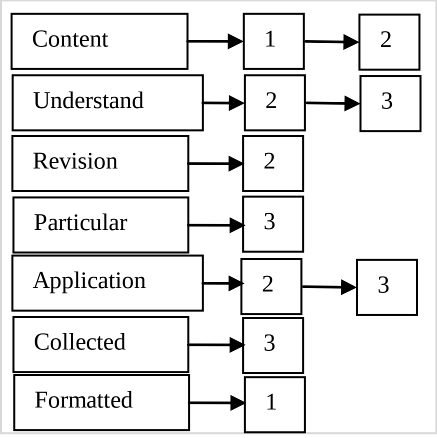
Dublin Core (Metadata)

```
<link rel="schema.DC"
href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.DCTERMS"
href="http://purl.org/dc/terms/" />
<meta name="DC.title" content="What is metadata? - A
Word Definition From the Webopedia Computer
Dictionary" />
<meta name="DC.subject" content="metadata definition;
metadata; define; define metadata; define; Webopaedia;
Webopedia; glossary; dictionary; encyclopedia" />
<meta name="DC.description" content="This page describes
the term metadata and lists other pages on the Web where
you can find additional information." />
<meta name="DC.publisher" content="#" />
<meta name="DC.type" scheme="DCTERMS.DCMIType"
content="Text" />
<meta name="DC.format" content="text/html; charset=iso-
8859-1" />
<meta name="DC.format" content="39350 bytes" />
<meta name="DC.identifier" scheme="DCTERMS.URI"
content="http://www.webopedia.com/TERM/M/metadata.ht
m" />
```

After applying the entire thing we get the Metadata from the text:-



Metadata	1	2	3
Describes	1	3	
Data	1	2	3
Provides	1	2	
Information	1	3	
Certain	1		
Item	1		



Content	1	2
Understand	2	3
Revision	2	
Particular	3	
Application	2	3
Collected	3	
Formatted	1	

After getting the Metadata send it to indexer to get the token [11] (as soon in example). And make the interface in which we can have search with different level (15 element of DC) and also with the simple content. When the user sends the query for search any file he simply goes to that interface and type the choices and send it for search(as in the flowchart) which is send the request to the index from which we get the result after processing it with ranking them in decreasing order to the user.

3. CONCLUSION

In this paper we have reviewed the general working of the search engine and its variousComponents and discussed that the digital servers search engine used by OAI-PMH fails to search the indexed digital information. We proposed a framework which makes crawler associated with search engine and DC to search e-print/pre-print document more efficiently.

4. FUTURE WORK

The proposed work can be modified as:

- By exploring the unsaved data in the digital from by scanning them and convert them into text from which we can have many new data which is never explored because they are in hard copy as well as in the hidden Web.
- To search topics in video and audio tape format this is not still perfectly searched.
- Improving the working of the antivirus software faster and not affecting the working of the system.

5. REFERENCE

[1] Saiful Amin, The Open Archives Initiative Protocols for Metadata Harvesting : an Introduction in DRTC workshop, Bangalore march 2003

[2] "Berners-Lee, Tim; Cailliau, Robert (November 12, 1990). "WorldWideWeb: Proposal for a hypertexts Project". http://w3.org/Proposal.html. Retrieved July 27, 2009.

[3] Berners-Lee, Tim. "Pre-W3C Web and Internet Background". World Wide Web Consortium. http://w3.org/2004/Talks/w3c10-HowItAllStarted/?n=15. Retrieved April 21, 2009.

[4] Wardrip-Fruin, Noah and Nick Montfort, ed (2003). The New Media Reader. Section 54. The MIT Press. ISBN 0-262-23227-8.

[5] NSDL Metadata Primer. URL: http://metamanagement.comm.nsdlib.org/outline.html

[6] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. Searching the Web. *ACM Trans. Inter. Tech.*, 1(1):2-43, 2001.

[7] http://dublincore.org/documents/2002/10/06/current-elements/ [/documents/2002/10/06/current -elements/].

[8] http://purl.org/metadata/dublin_core_elements for further information

[9] Bin He, Kevin chen-chuanchang; "Automatic complex schema matching across web query interfaces: A correlation mining approach"; ACM Transactions on Databases Systems; Vol. 31; No.1; Pages 1-45; March 2006

[10] By Christopher D. Manning, PrabhakarRaghavan&HinrichSchütze, Pages 438-441; April 1, 2009.

[11] By S Amin - 2003 - Cited by 2 - Related articles Paper: H. *The Open Archives Initiative Protocol* for. *Metadata Harvesting: An Introduction*. SaifulAmin.x