A Novel Architecture of Agent based Crawling for OAI Resources

Shruti Sharma
YMCA University of Science & Technology,
Faridabad, INDIA
shruti.mattu@yahoo.co.in

J.P.Gupta
JIIT University, Noida , India
jp_gupta/jiit@jiit.ac.in
A.K.Sharma
YMCA University of Science & Technology, Faridabad, India
ashokkale2@rediffmail.com

***Abstract*—Nowadays, most of the search engines are competing to index as much of the Surface Web as possible with leaving a lurch at the OAI content (pdf documents), which holds a huge amount of information than surface web. In this paper, a novel framework for OAI-PMH based Crawler is being proposed that uses agents to extract the metadata about the OAI resources and store them in a repository which is later on queried through the OAI-PMH layer to generate the XML pages containing the metadata. These pages are further added to the search engines repository for indexing that makes in turn increases the relevancy of Search Engine. Agents are being used to parallelize the whole process so that metadata extraction from multiple resources can be carried out simultaneously.**

Keywords-OAI-PMH; Agents; Surface web;Hidden Web

I. INTRODUCTION

Search engines are not the answer to everybody's information need. They are just another tool to help find what user is looking for, but not the end-all, be-all solution. Due to the explosion in the size of World Wide Web, search engines are becoming increasingly important tool in locating relevant information. Such search engines rely on massive collections of web pages that have been crawled by web crawlers, that traverse the web by following hyperlinks thereafter storing downloaded pages in a large repository that is later indexed for efficient execution of user queries. Recent years demonstrate an unbroken trend towards end-user searching. Users expect search services to be integrated, up-to-date and complete in a sense that relevant information can be retrieved. However, a huge amount of data resides behind these search forms that is referred to as hidden web content.

Having indexed much of the Surface Web [1,4], search engines are now using various approaches to index the "deep" Web. At the same time, institutional repositories and digital libraries are adopting the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [13, 19] to expose their belongings of white papers, some of which are indexed by search engines. Academic and research institutions are expending enormous efforts to digitize their collections of theses, white papers, technical reports, maps, images, and historical documents to make them available in institutional repositories or digital libraries. The OAI-PMH began as grass-roots interoperability[15] effort for e-print archives that provides an application-independent interoperability framework based on metadata harvesting as shown in Figure I There are two classes of participants in the OAI-PMH framework:

- Data Providers administer systems that support the OAI-PMH as a means of exposing metadata;
- Service Providers use metadata harvested via the OAI-PMH as a basis for building value-added services.

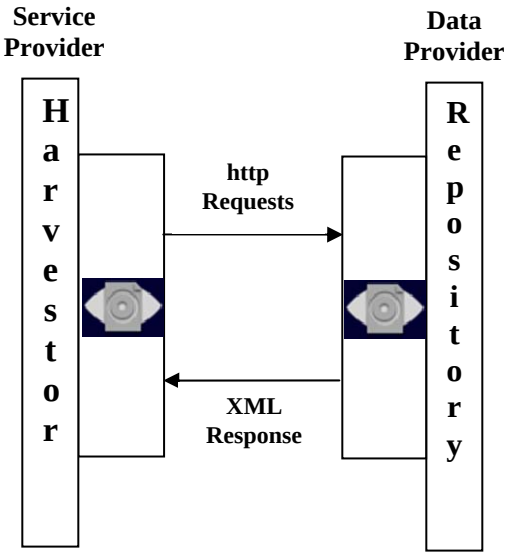

Figure 1: Functional Block diagram of OAI-PMH

By issuing an OAI-PMH request, an OAI compliant repository, a harvester can obtain an XML-encoded list of all the repository's metadata records.

A *harvester* operated by a service provider as a means of collecting metadata from repositories, is a client application that issues OAI-PMH requests. A *repository* is a network accessible server that can process the six OAI-PMH requests managed by a data provider to expose metadata to harvesters. To allow various repository configurations, the OAI-PMH distinguishes between three following distinct entities related to the metadata as shown in Figure 2

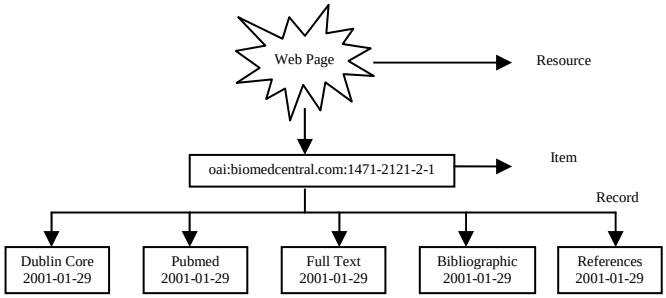

Figure 2. Data Model of OAI-PMH

- resource - A resource is the object that metadata is "about". The nature of a resource, whether it is physical or digital, or whether it is stored in the repository or is a constituent of another database, is outside the scope of the OAI-PMH.
- item - An item is a constituent of a repository from which metadata about a resource can be disseminated on-the-fly from the associated resource, cross-walked from some canonical form, actually stored in the repository.
- record - A record is metadata in a specific format returned as an XML-encoded byte stream in response to a protocol request to disseminate a specific metadata format from a constituent item.

II. RELATED WORK

Nelson et. al. [21] identified two problems associated with conventional web crawling techniques:

- Crawler cannot know if all resources at a non-trivial web site have been discovered and crawled ("the counting problem")
- The human-readable format of the resources are not always suitable for machine processing("the representation problem").

In [12,15], an approach that solves these two problems by implementing support for both the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) and MPEG-21 Digital Item Declaration Language (DIDL) into the web server itself has been introduced and Apache module "mod-oai", that can be used to address the counting problem has been presented by listing all valid URIs at a web server and efficiently discovering updates and additions on subsequent crawls.

Tang et. al. [22] proposed an automated, human-assisted process to extract metadata from documents in a large (>100,000 documents), dynamically growing collection. Such a collection may be expected to be heterogeneous, both statically and dynamically heterogeneous. In this work, process of first classifying documents into equivalence classes is proposed for which a rule-based approach to extract metadata has been provided. The rule-based approach differs from others in the sense that it separates the rule-interpreting engine from a template of rules. The templates vary among classes but the engine is the same.

A critical analysis of the available literature shows the following shortcomings in the existing work:

- The current work [18,12,20,21] requires the human-assisted process in order to extract the meta information from the documents in dynamically growing collection.
- Human-readable format of resources are not suitable for machine processing. Therefore, it becomes more important to identify the format of given resources.
- The current research in this area [12,13], is not fully able to find the relevant meta information.

Henceforth, in this paper, a novel framework for OAI-PMH based search service using agents has been proposed that efficiently retrieves the required Meta information from the documents using agents. Moreover, these agents are independent from each other and retrieve only the relevant information without any human assistance which is later used by the search engines. The proposed technique uses document classifier that automatically identifies the human-readable format of resources.

III. THE OAI-PMH BASED SEARCH SERVICE USING AGENTS

Amongst the enormous collection of white papers, some are indexed by the search engines and some are not, if the metadata about these pdf documents is provided to the search engines in the form of XML pages that can later be indexed and used to answer user queries, then the searching becomes more efficient.

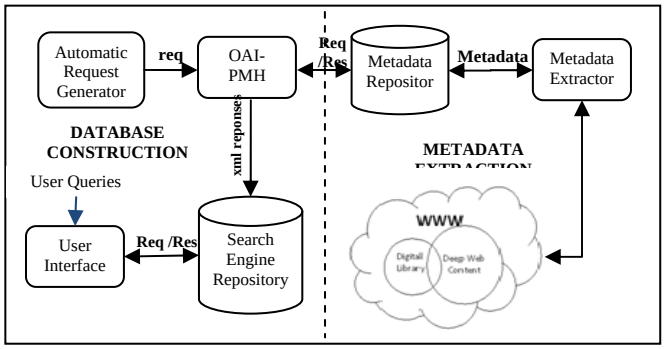

Figure 3. Block diagram of OAI-PMH based Search Service

Therefore the concepts of agents is introduced which will extract the metadata. The OAI-PMH is an important new infrastructure for supporting distributed networked information services. The architecture has been divided in two stages: the

metadata extraction and database construction as shown in Figure 3.

A. Phase I : Metadata Extraction

This is the first phase of the OAI-PMH based search service that is used for extracting the metadata related to the white papers and creating the metadata repository that is later used in the database construction phase. The metadata extraction process is shown in the Figure 3. It consists of the following functional components:

1) Dispatcher

This component reads the URLs from the database and fills the seedURL Queue. It may also get initiated by the user who provides a *seedURL* in the beginning. It sends a signal: *Something_to_Process* to the Agent manager. Its algorithm is given below:

```
Dispatcher ( )
Begin
 Do Forever
  Begin
   While (SeedURLQueue not Full)
    Begin
     Read URL from Database;
     if (digital library or listing of pdfs)
      Store it into SeedURLQueue;
    End;
   Signal (Something_to_process);
   End;
End;
```

2) AgentConf.txt

It is an agent configuration file that is used by the Agent Manager to store the agent's data. The contents of a sample file are tabulated in Table I.

3) Agent Manager

This component waits for the signal Something to Process. It then reads the seedURL Queue and creates multiple worker threads called agents and each agent is given one URL at a time from the SeedURL Queue as shown in Figure 4. These agents may reside on the same machine or on the different machine. The information about these agents (URL, IP, Date & Time stamp etc) is then stored in the AgentConf.txt file. Its algorithm is given below.

```
Agent_Manager ( )
Begin
      Do forever
  Begin
   Wait (something to process);
   While (Not end of SeedURLQueue)
    Begin
   Wait (request processed);
   Pickup URLs from SeedURLQueue;
   Create A new Agent. // Add all the information
   into the agentConf.txt
   Assemble and assign a URL to the newly
   created Agent;
  End;
 End;
End
```

TABLE I. AGENTCONF.TXT

Name	Value	Description
AgentId	Agent1	Unique Identification of the agent
URL	http://www.acm.com	The url of the website that is to be processed
URL-IP Address	10.12.122.15	IP address of the URL
Date & Time Stamp	21/06/2009 06:30:42	Date and time stamp at which the agent started processing the website.
Local Instance	Yes	The instance created on same machine (Local) or different machine.
MachineIP	122.12.13.12	IP address of the machine where the agent is created.

4) Agent

This component further has three more functional components namely as shown in Figure 5 :

- Link Processor
- Document Classifier
- Metadata extractor

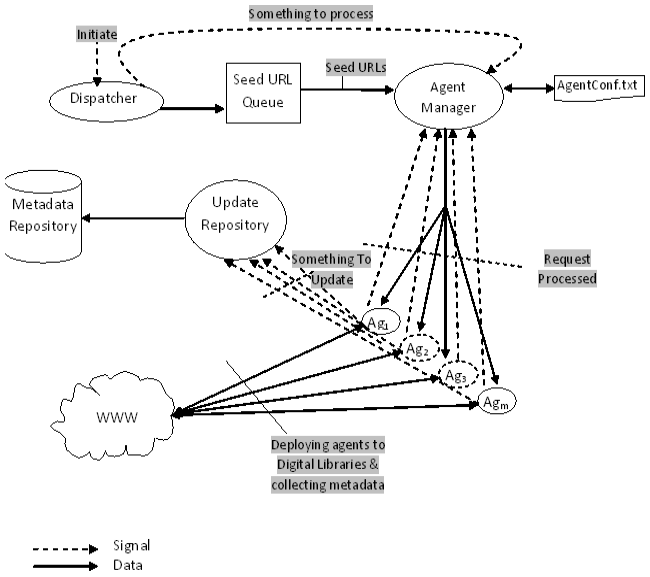

Figure 4. Proposed architecture of Agent Manager

The URL from the agent manager is given to the link processor. The link processor maintains a PDFLink Queue wherein all the links of the pdf documents whose meta data is to be extracted are stored. The PDFLink queue is then used by the document classifier to identify the type of the pdf document. It is used to distinguish document prior to metadata extraction. Various document classification algorithms are available which may be used. For example J. Hu [22] partitioned a page into an m*n grid and each cell was either a

text cell (more than half of it is overlapped by a text block) or white space cell. With partitions by absolute positions, this approach measure page similarity by blocks' absolute positions. Pages with same style but blocks with different size may be considered dissimilar. F. Cesarini[14, 22] encoded a document's cover page into a MXY-Tree and used it for document page classification. As an extension of XY-Tree an MXY-Tree recursively cuts a page into blocks by separators (e.g. lines) as well as white spaces.

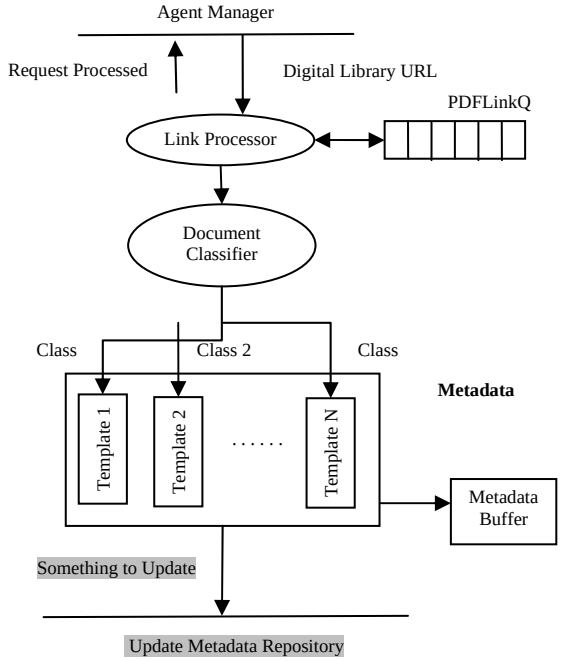

Figure 5. Architecture of an Agent

Various existing automated metadata extraction approaches can be used like rule-based systems, template based and statistical learning systems. These template-based systems produced good results. If classification is successful, the document is passed to the metadata extractor for the selected homogenous class(es). A template-based approach to encoding is a relatively straightforward set of extraction rules. For example, a rule might say: 'If the phrase is centered, bold and in the largest font, it is a title'. A template language allows changes to existing rules or the addition of new rules to a template for a class without having to modify the extraction engine. In a template-based approach, rules are decoupled from the extraction engine code by keeping templates in separate XML files. This decoupling contributes to an easily extensible system. For a new document class, just a new template file is created without modifying the code. The extraction engine understands the language in which templates are written. For an incoming document, the engine loads its corresponding template, parses the rules defined in the templates and extracts metadata from the document accordingly. Agent extracts the metadata as per the algorithm given below.

Agent ( )
Begin
Extract the pdf links from the Seed URL(from Agent Manager)
While (PDFLinkQ is not empty)
Begin
Pickup a pdf and pass it to the document classifier;
Identify the class of document;
Extract the metadata;
Signal (something to Update);
Signal (request processed);
End;
End.

5) Update Repository

This process waits for the signal something to update and on receiving the same, updates the metadata database with the contents of the Metadata Buffer. The algorithm of Update Repository process is given below:

Update Repository()
Begin
Set MaxSize to the maximum size of a batch;
Do forever
Begin
Wait (something to update);
No-of-records = 0;
While (No-of-records < MaxSize)
Begin
Pickup an element from Metadata Buffer;
Add to the batch of records to be updated;
No-of-records= No-of-records+1;
End;
Update batch to database;
If (Updation is unsuccessful)
Then write batch to Metadata buffer;
No-of-records = 0;
End
End

B. Phase II : Database Construction

In this phase the metadata repository created by the metadata extractor is used to construct the database for the search engine which is later indexed for relevant information retrieval.

1) Automatic Request Generator

The task of this module is to automatically issue the OAI-PMH requests based on the six verbs to the OAI-compliant repository and store the xml response pages to the database which is later indexed by the search engine. The http requests are generated on the basis of selective harvesting approach based on datestamps and sets.

TABLE II. OAI-PMH VERBS

Verb	http Request Format
GetRecord	http://arXiv.org/oai2?verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc
Identify	http://memory.loc.gov/cgi-bin/oai?verb=Identify
ListIdentifiers	http://an.oa.org/OAI-script?verb=ListIdentifiers&from=1998-01-15& metadataPrefix=oldArXiv&set=physics:help
ListMetadataFormats	http://www.perseus.tufts.edu/cgi-bin/pdataprov?verb=ListMetadataFormats&identifier=oai: perseus.tufts.edu:Perseus:text:1999.02.0119
ListRecords	http://an.oa.org/OAI-script?verb=ListRecords&from=1998-01-15&set=physics:hep&metadataPrefix=oai_rfc1807
ListSets	http://an.oa.org/OAI-script?verb=ListSets

To construct the database, six verbs or requests of the OAI-PMH are used, wherein the request is embedded in http request format, and the generated xml responses are added to the database. The six OAI-PMH requests with format are shown in Table II.

The protocol also supports selective harvesting that allows harvesters to limit harvest requests to portions of the metadata available from a repository. The OAI-PMH supports selective harvesting with two types of harvesting criteria that may be combined in an OAI-PMH request: datestamps and set membership[14].

a) Selective Harvesting and Datestamps

Harvesters may use datestamps to harvest only those records that were created, deleted, or modified within a specified date range. To specify datestamp-based selective harvesting, datestamps are included as values of the optional arguments, *from and untill*, in the *ListRecords* and *ListIdentifiers* requests. Harvesting is restricted to the range specified by the from and until arguments, extending back to the earliest datestamp if from is omitted, and forward to the most recent datestamp if until is omitted.

Example:

http://www.perseus.tufts.edu/cgibin/pdataprov?verb=ListIdentifiers&metadataPrefix=olac&from=2001-01-01&until=2001-01-01&set=Perseus:collection:PersInfo

b) Selective Harvesting and Sets

Harvesters may specify set membership as criteria for selective harvesting. To specify set-based selective harvesting, a *setSpec* is included as the value of the optional set argument to the *ListRecords* and *ListIdentifiers* requests, thereby specifying selective harvesting of records from items within the respective set.

Example:

http://an.oa.org/OAIscript?verb=ListRecords&from=1998-01-15&set=physics:hep&metadataPrefix=oai_rfc1807

In this paper, an OAI-PMH based search service is proposed that provides the facility to retrieve XML responses which contain metadata about the OAI resources that are stored in search engine's repository for indexing. The proposed scheme uses agent based approach that facilitates the metadata extraction from the pdf documents in parallel from different resources.

As the future work, some more efficient algorithm for document classification can be used. A rule based approach used for metadata extraction can also be modified for efficient metadata content extraction. Since in the proposed work, the multiple instances of the agents are running in parallel under agent manager, therefore same document can be extracted more than once, to avoid this some technique may be applied to identify the duplicate contents.

The Extensible Repository Resource Locators (ERRoLs) for OAI Identifiers project allows the creation of URLs that dynamically perform OAI-PMH queries against registered OAI repositories and generate HTML pages suitable for web crawling [10], can also be integrated with the automatic request generator module.

Moreover, inter-agent communication is also required so that agents working in parallel must not extract that same document.

V. REFERENCES

[1] Michael K. Bergman, "The deep web: Surfacing hidden value", Journal of Electronic Publishing, 7(1), 2001.

[2] V. Crescenzi, G. Mecca, and P. Merialdo. "Roadrunner: Towards Automatic Data Extraction from Large Web Sites,"

[3] VLDB Journal, 2001, pp. 109-118.

[4] P. G. Ipeirotis and L. Gravano, "Distributed search over the hidden-web: Hierarchical sampling and selection," In Proceedings of VLDB '02, 2002, pp. 394-405.

[5] X. Liu, K. Maly, M. Zubair, and M.L. Nelson, "DP9: an OAI Gateway Service for Web Crawlers," In Proceedings of the Joint Conference on Digital Libraries (JCDL), June 2002, pp. 283-284.

[6] M. L. Nelson, H. Van de Sompel, X. Liu, T. Harrison, N. McFarland, "mod_oai: An Apache Module for Metadata Harvesting," In Proceedings of ECDL 2005, Vienna, Austria, pp. 509-510.

[7] A. Ntoulas, P. Zerfos, J. Cho, "Downloading Textual Hidden Web Content by Keyword Queries," In Proceedings of the Joint Conference on Digital Libraries (JCDL), June 2005, pp. 100-109.

[8] S. Raghavan and H. Garcia-Molina, "Crawling the Hidden Web," In Proceedings of VLDB '01, 2001, pp. 129-138.

[9] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In Digital Libraries 06—The Third ACM Conference on Digital Libraries, pages 89-98, June 23-26 2007.

[10] AMeGA, Automatic Metadata Generation Applications, Retrieved April, 2005 http://ils.unc.edu/mrc/amega.htm

[11] Li X, Cheng Z, Sheng F, Fan X, and Ng P. A Document Classification and Extraction System with Learning Ability. Proceedings of the Fifth World Conference on Integrated Design and Process Technology, Dallas, Texas, June 2000.

[12] Liu X, Maly K, Zubair M, Nelson M. Arc: an OAI service provider for cross-archive searching. JCDL 2001: 65-66

[13] http://www.openarchives.org/OAI/2.0/

[14] Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. "Resource Harvesting within the OAI-PMH Framework," D-Lib Magazine, 10, 12, December, 2004, Corporation for National Research Initiatives. http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html

[15] Michael L. Nelson, Herbert Van de Sompel, and Simeon Warner, "Advanced Overview of Version 2.0 of the Open Archives Initiative Protocol for Metadata Harvesting, " ACM/IEEE Joint Conference on Digital Libraries, Houston, Texas, May 27 2003. http://www.cs.odu.edu/~mln/jcdl03/oai-2.0-adv.ppt

[16] Hussein Suleman and Edward Fox, "Beyond Harvesting: Digital Library Components as OAI Extensions" http://www.husseinsspace.com/publications/cstr_2002_odl_1.pdf

[17] Automated Building of OAI Compliant Repository from Legacy Collection Jianfeng Tang, Kurt Maly, Steven Zeil, Mohammad Zubair.

[18] H. Van de Sompel, M. L. Nelson, C. Lagoze, and S. Warner. Resource harvesting within the OAI-PMH framework. D-Lib Magazine, 10(12), 2004.

[19] H. Van de Sompel, J. A. Young, and T. B. Hickey.Using the OAI-PMH ... differently. D-Lib Magazine, 2003.

[20] Hu J, Kashi R, and Wilfong G. Document Image Layout Comparison and Classification. In Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR), 1999.

[21] Han H, Giles CL, Manavoglu E, Zha H, Zhang Z, and Fox EA. Automatic Document Metadata Extraction Using Support Vector Machine. 2003 Joint Conference on Digital Libraries (JCDL'03), Houston, Texas USA, May 2003.

[22] Cesarini F, Lastri M, Marinai S, and Soda G. Encoding of modified X-Y trees for document classification. In Proc. Sixth ICDAR, pages 1131–1136, 2001