

Open Source OAI Metadata Harvesting Tools

David Kellogg

August 27, 2004. Revised December 6, 2004.

Abstract

Open source OAI harvesting tools were surveyed for ease of use, installation difficulty and robustness. UMHarvester proved to be a solid harvesting tool. It places harvested content into directories. Other interesting tools include a Perl tool from Virginia Polytechnical Institute and PKP from the University of British Columbia. This is a solid command-line tool that fits well with scheduling tasks. These tools and others were surveyed.

1 Introduction

OAI harvesting is simple when compared to web-crawling.¹ There are only 6 verbs, the hierarchy of documents on the server is shallow, and a web interface easily retrieves data. Due to a lack of hierarchy, the burden of organizing the metadata falls on the harvesting institution. There are many choices for harvesters. Many try to handle more functions than necessary. A simple tool that only harvests and saves metadata is needed.

The University of Michigan's UMHarvester is a module that simply downloads metadata. It can be used as a module to PKP (public knowledge project) Open Archives Harvester handles the mass of data well, both downloading and presenting metadata.² Virginia Tech's Perl harvester only retrieves metadata.

It is important to show how simple metadata retrieval is to understand the necessary features of a harvesting agent. A sample OAI session will show the

¹see Evaluation of Open Source Spidering Tools at https://diva.cdlib.org/projects/harvesting_crawling/recall_crawl/spider_eval.pdf

²"PKP Open Archives Harvester", <http://www.pkp.ubc.ca/pkp-harvester>

tasks needed for harvesting. Harvesting difficulties are presented. A closer look will be given to PKP Open Archive Harvester and the Virginia Tech Perl Harvester. Other Harvesters, most of performed poorly with respect to installation ease, are also explored.

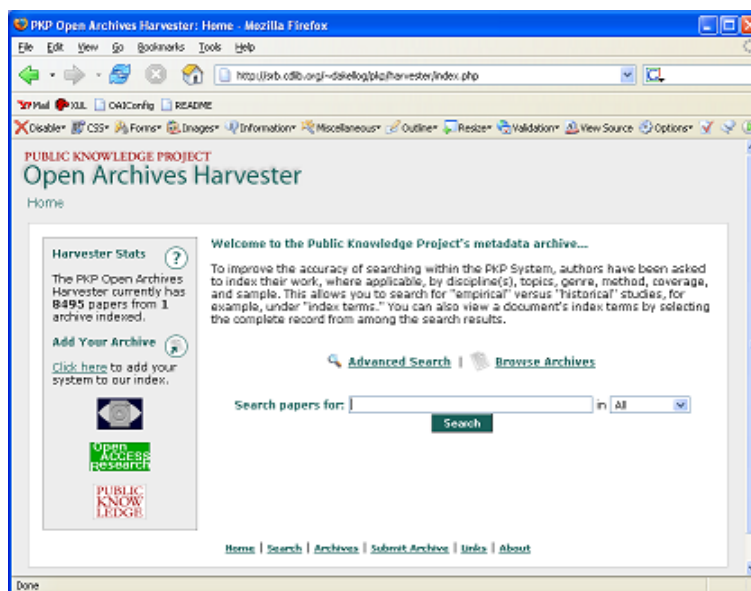


Figure 1: PKP

2 Sample OAI Session

The following shows an OAI harvesting session, which can be reproduced in a web browser, such as Firefox. Replies are all wrapped in XML markup.

Step 1. This returns information about the database.

`http://dlese.org/oai/provider?verb=Identify`

Reply: “‘DLESE’s mission is to improve’”

Step 2. This lists formats, e.g. oai_dc.

`http://dlese.org/oai/provider?verb=ListMetadataFormats`

Reply: “‘oai_dc adn’”

Step 3. This lists sets.

```
http://dlese.org/oai/provider?verb=ListSets
```

Reply: ‘‘nsdl . . . comet . . .’’

Step 4. This returns a long list of records.

```
http://dlese.org/oai/provider?verb=ListIdentifiers&from=1998-01-15&
```

```
metadataPrefix=oai_dc
```

Reply: ‘‘oai:dlese.org:COMET-74 . . .’’

Step 5. We can pick one out with this.

```
http://dlese.org/oai/provider?verb=GetRecord&metadataPrefix=oai_dc&
```

```
identifier=oai:dlese.org:COMET-74
```

Step 6. This returns a single Dublin Core record.

```
http://dlese.org/oai/provider?verb=GetRecord&metadataPrefix=oai_dc&
```

```
identifier=oai:dlese.org:DLESE-000-000-000-698
```

Reply:

...

<GetRecord>

<record>

<header>

<identifier>oai:dlese.org:DLESE-000-000-000-698</identifier>

<timestamp>2004-08-19T21:05:02Z</timestamp>

<setSpec>nsdl</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>

Anthony’s Nose, New York: A Review of Three Mineral Localities (title provided or enhanced by cataloger)

```

</dc:title>

<dc:subject>Mineralogy or petrology</dc:subject>

<dc:description>

This web site describes ...

...

Copyright 1998 John Betts and Bob's Rock Shop. All rights reserved.

</dc:rights>

</oai.dc:dc>

</metadata>

</record>

</GetRecord>

</OAI-PMH>

```

The sixth verb used in OAI harvesting is `ListRecords`. This is similar to `GetRecord`, but it retrieves multiple records instead of a single one. One verb not used in harvesting is `Document`, which is used for testing.

3 Harvesting Challenges

There are several difficulties in harvesting metadata. A metadata harvester needs to start and stop retrieval at arbitrary points in the set to allow parts of large collections to be downloaded. Much data is corrupted and does not parse well using a tool such as SAX. Invalid data should not crash or stop the parser, but often it does. It is important, therefore, to have access to the raw data in cases of poor formatting. A harvester can be created that is quite simple, as with the above. Most of the work requires transforming broken data into usable sets.

4 UM Harvester

University of Michigan's harvester meets the requirements for a harvester. These include simplicity, control, and easy access to all harvested data. The

UM harvester places metadata records in a directory structure, which is easy to browse.

It is important when configuring the program to change the *l* and *ll* directories written in the source code to local directories. The process appears to occur in the background. This is fine. A kill command will stop it if there are problems. A log file is generated to document searches. A configuration file holds only the metadata server name. According to the documentation, cumbersome verbs are shortened to two letters.

Records can be found at dot.cdlib.org in the directory, /home/dakellog/perl/lwp/harvester. They are broken into directories of record numbers 1-1000, 1001-2000, et al. One of the records in the location, /home/dakellog/perl/lwp/harvester/auburn/auburn1-1000/oai.dlese.org_NASA-SVS-001008.xml, holds information on Global Temperature and other subjects.

The record follows.

```
<record>
<header>
<identifier>oai:dlse.org:NASA-SVS-001008</identifier>
<timestamp>2004-10-22T19:28:13Z</timestamp>
<setSpec>svs</setSpec>
</header>
<metadata>
. . .
```

Code should be written so easy access to these records.

5 PKP

PKP was developed at the University of British Columbia for OAI metadata harvesting and retrieval. The entire site, including harvester and search engine works out of the box. The site installs easily into a LAMP-based³ server (Linux-Apache-MySQL-PHP⁴) without writing configuration files or installing non-LAMP dependencies. The code is readable and well-documented. This software is flexible, so it can be installed on Windows using open source tools alone.

³This is the chosen platform for Amazon, Paypal, Slashdot and Google. For origins, see <http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>

⁴To complicate matters, the P in LAMP can mean Python, Perl or Ruby. M can mean PostgreSQL

5.1 Harvesting in PKP

Harvesting in PKP is managed through a web page from a server running PHP and MySQL. In the browser adding an archive and running it is simple. In a test run, over 8000 metadata records from the DLESE database were found and indexed.

5.2 PKP Search Engine

The PKP search engine showed varied results. A search for “high school” did not always surface the words “high” and “school” in the metadata record. This appears to be an “Or” search by default. Over 4000 of the 8000 records were shown to contain at least one of these search terms. Using “high school course” reveals the same result. A quick scan of the linked record showed no evidence of the word, “course”.

5.3 GUI

Among GUI interfaces, this is an impressive design. It is easy to navigate the site and find the administrative tools to harvest archives and search them. Links lead to an advanced search. Administration access is password protected. With administration access, archives can be searched. Without administrative access, the user can browse the retrieved records.

6 Virginia Tech Perl Harvester

The Perl harvester from Virginia Tech⁵ performed well on tests. It was able to retrieve OAI records using the command line. This can be integrated into a cron job for daily updating.

7 Other Harvesters

Other harvesters were used. Most did not work out of the box. In these cases `README` and `INSTALL` files were both read and followed. Additional dependencies were met. Then still, most harvesters did not work. This contrasts

⁵<http://oai.dlib.vt.edu/odl/software/harvest>

sharply with web crawlers, where over 90 percent worked out of the box for what is in nearly all aspects a more difficult task. This is a multi-platform issue. Academics often insisted on a specific platform with numerous dependencies. These undocumented dependencies led to difficult compilations. Most harvesters required support to install them. In the interest of finding more capable harvesters, these were left aside. The following were additional harvesters that were evaluated.

ODL. Easy to use Perl tool. Downloads and arranges data well. It seemed not to be able to download data successively from the same set. No error message was shown.

UIUC Java Harvester. Difficult to install. Requires JDBC. Documentation required editing the server line in the configuration. It did not state which server to edit, since 2 to 4 server strings appeared in the configuration file.

UIUC VB Harvester. This MS Visual Basic Program requires MSXML. Based on Access DB, a database that commonly corrupts the data after passing the 1GB mark.

IVia. Requires many C libraries and a Debian system.

myOAI. Project website worked well. Local downloaded Perl script did not download data.

CPAN OAI::Harvester. This Perl module required many dependencies, such as packages HTML, HTTP, libwww, Net, SAX, URI and XML::SAX.

8 Conclusion

Two metadata harvesters showed promise to add as a module in a metadata retrieval and browsing program, PKP Open Archive Harvester and Virginia Tech Perl Harvester. PKP used an impressive GUI, and Virginia Tech created a very flexible command line harvester. Other harvesters were tested, but most were difficult to install. Because the greatest difficulty in these programs will be in organizing and repairing the metadata and XML, it is best to choose a flexible tool, which can be extended to handle non-standard data.

9 Location

The location of this document is

*[https://diva.cdlib.org/projects/crawling_harvesting/
oai_harvesting.pdf](https://diva.cdlib.org/projects/crawling_harvesting/oai_harvesting.pdf)*

This document is written in L^AT_EX.