

# Efficient, Automatic Web Harvesting

Michael L. Nelson    Joan A. Smith    Ignacio Garcia del Campo

*Old Dominion University, Norfolk Virginia*

---

Herbert van de Sompel    Xiaoming Liu

*Los Alamos National Laboratory*



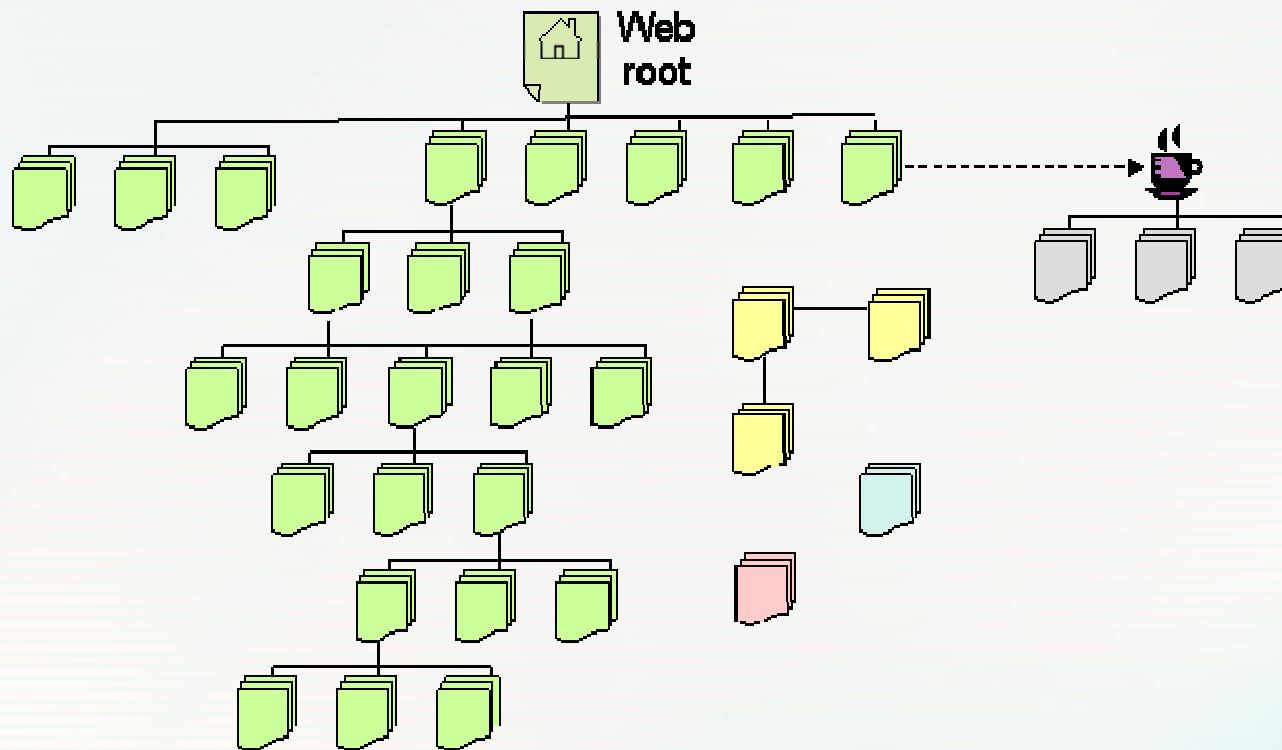
# Crawling Is Easy



- Billions of pages have been crawled
- Lots of search engines exist
  - A few “big boys” Google, Yahoo, MSN
  - Lots of interest in the technology
  - Interesting applications like targeted ads
- Specialty sites are out there too
  - fabfotos, findlaw, citeseer, netdoctor
  - Semantic engines are creating new concepts of links and web page relationships
  - There are even search engines about search engines:  
<http://www.search-engine-index.co.uk/>
- The search engines get around so quickly and so often that a cached copy is usually not too old
- So crawling must be pretty straightforward...

# Or is it?

- So why are we talking about making harvesting more *efficient* and *automatic*?
- How does a crawler work?
- HINT: It uses HTTP and it depends on *links* (URLs)



# HTTP is easy



- Make a request
  - GET blah.html
- Receive a response
  - blah.html

*sort of...*

Here's an actual GET request:

GET / HTTP/1.1

Host: www.modoi.org

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5) Gecko/20031007

Accept: application/x-shockwave-flash,text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,\*/\*;q=0.1

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://www.google.com/search?hl=en&q=modoi&btnG=Google+Search

If-Modified-Since: Thu, 17 Aug 2006 14:18:36 GMT

If-None-Match: "15b9b090-152c-51c72700"

Cache-Control: max-age=0

# Or is it?

- Now take a look at the response

GET / HTTP/1.1

Host: www.modoi.org

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5) Gecko/20031007

Accept: application/x-shockwave-flash,text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,\*/\*;q=0.1

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: <http://www.google.com/search?hl=en&q=modoi&btnG=Google+Search>

If-Modified-Since: Thu, 17 Aug 2006 14:18:36 GMT

If-None-Match: "15b9b090-152c-51c72700"

Cache-Control: max-age=0

The problem is, only a small piece of the page is loaded here



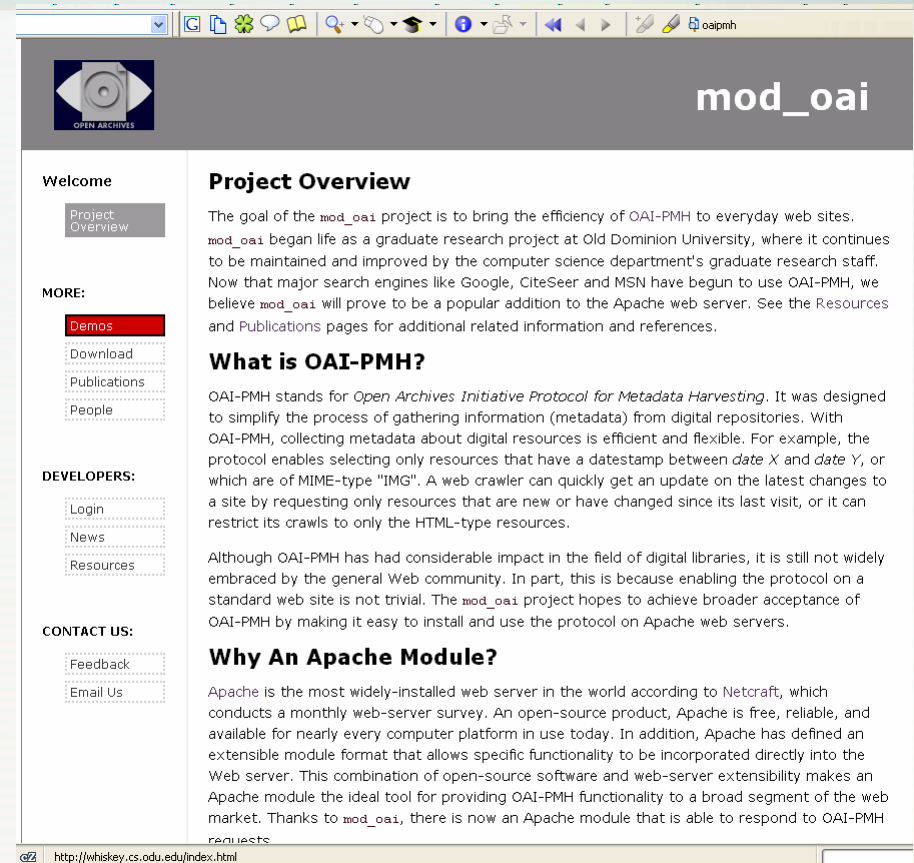
Images, style, come later...



# HTTP is limited

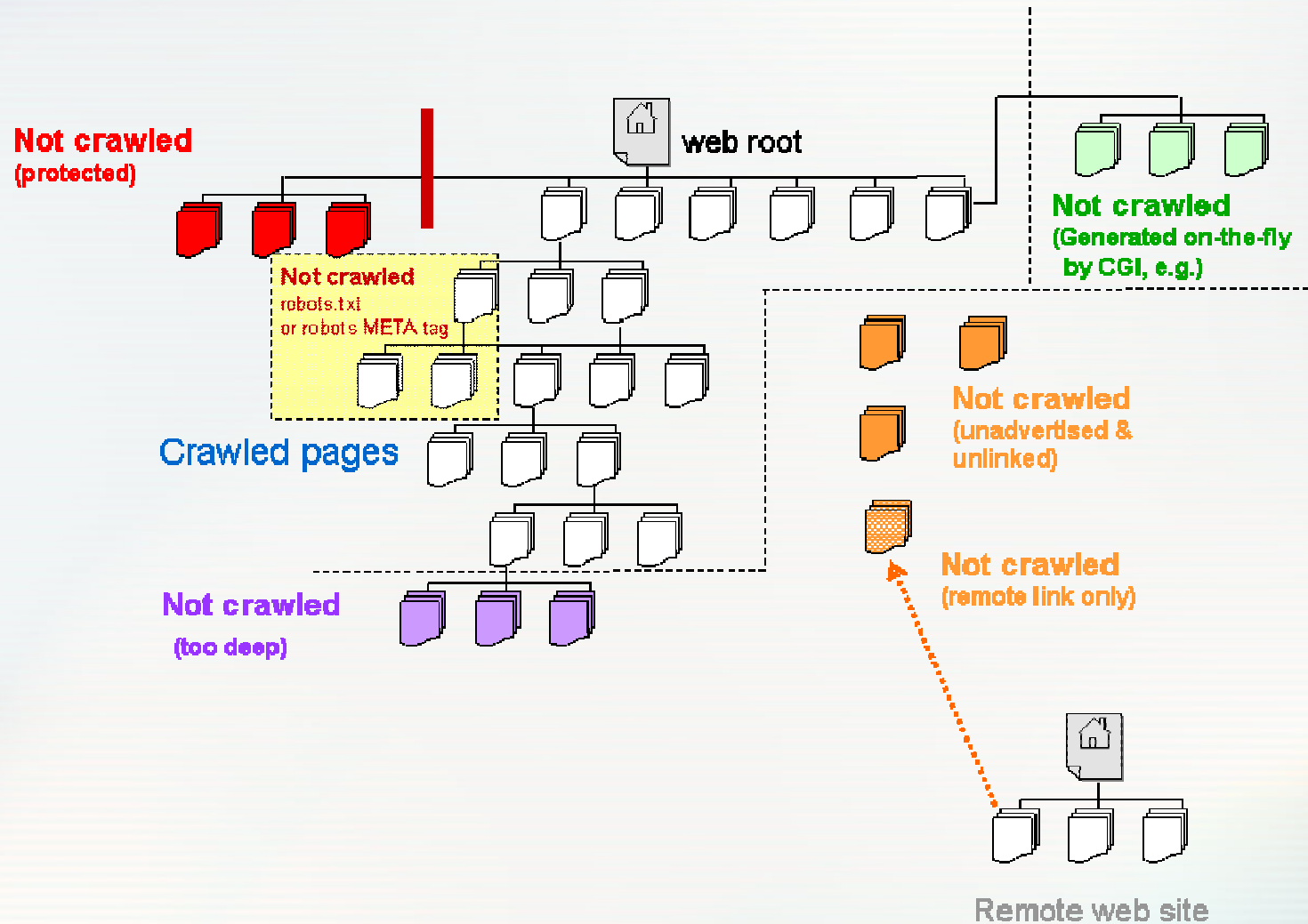


- 1 GET receives 1 resource
- Most URLs require many back-and-forth request-response exchanges just to load the single “page” that you see in your browser
- This home page for the mod\_oai project has several images, a CSS style sheet, a bunch of links, and the word content you see on the page.
- A browser or a crawler has to “read” the HTML of the basic page, figure out what else it needs to make the view complete, and go back to get each of those items.



**And that's just for ONE page!**

# Crawling is Complicated



# The Hard Life of a Robot



Results from our experiments watching crawlers: May-Sep 2005

- The google dance
  - About every 2 weeks
  - Thorough breadth, depth span
  - Heavy use of conditional GET (ex: "if-modified-since")
- The yahoo crawl
  - More sporadic, about every 30 days
  - Pretty deep, wide
  - Delayed visits meant it never saw short-lived pages
- MSN
  - Less deep, less broad
  - Hired out robots?
  - Little showed up in caches...
- Biggest problems with crawling
  - Getting everything crawled
  - Keeping new site pages linked
  - Updating search engine cache repositories
  - Time, time, time (and bandwidth and processing power)



# The Elevator Analogy



The Empire State Building



A Famous Visitor:  
He didn't need the elevator

- Really huge buildings are different from the usual
- Elevators do **not** go to every floor
- Some are “express” going to only a few floors, or directly to the top
- Higher floors may have *other* banks of elevators that go to more floors
  - Take elevator 1 to floor 31
  - Meet some people...
  - Go to elevator bank 2 and take a different set of elevators from floor 31 to floor 35.
  - Multiple routes to get back down to the first floor
- Crawling has a lot in common with this experience
- If there isn't a button for that floor, you can't get there from here!

What happened to the other floors?

Be careful which one you choose



# Isn't there a better way?



## Crawlapalooza vs. Harvester Home Companion

- World Wide Web

- A free-for-all
- Not organized
- Very little metadata
- Haphazard additions, deletions, modifications



- Digital Library

- Organized
- Groomed content
- Lots of metadata
- Structured changes



**It turns out that web crawling trick is hard to do after all**

# What if we could --



- Get a list of all URLs for the site
  - Including those not linked from root
  - Maybe even CGI-related links
- Get a list of everything new since last visit
  - Any pages that have changed
  - Any new pages added
  - Any pages that have been deleted
- Get a list of all <put your mime type here>
  - Images (specific subtype or all of them)
  - HTML pages only
  - PDFs only
  - Whatever mime spec you want...



# Libraries: Inspiration for a Digital Age



## Anatomy of a city library:

- Organized
  - Grouped
    - Topics
    - subtopics
  - Numbered
- Searchable
  - By author, title
  - By topic
  - By edition
- **Lots of metadata**



## Digital library is similar

- Expands on physical library concepts
- Special protocols let librarians organize and find resources & information
- **OAI-PMH** is one of these “library” protocols



# OAI-PMH: Empowering HTTP



We said we need a way to

- Get a list of all URLs for the site
- Get a list of changes (new, gone, altered) since last visit
- Get a list by some grouping we specify (e.g., MIME)

## **OAI-PMH gives us these options**

- Works a lot like CGI-style URLs you may see:
  - <http://www.foo.org/ask.php?pid=3244&uid=jsmith> (PHP-enabled web server)
  - <http://www.foo.org/oaiserver?verb=Identify> (OAI-PMH-enabled web server)
- It is designed for the robot, not the browser
  - Gives back valid, XML-formatted response
- **mod\_oai** is an Apache 2 module that allows OAI-PMH verbs to be used on the web site



# Overview of OAI-PMH Verbs



|                                     |  | Verb                | Function  |
|-------------------------------------|--|---------------------|---|
| metadata<br>about the<br>repository |  | Identify            | description of repository                             |
|                                     |  | ListMetadataFormats | metadata formats supported by repository              |
|                                     |  | ListSets            | sets defined by repository                            |
| harvesting<br>verbs                 |  | ListIdentifiers     | listing of all OAI unique ids contained in repository |
|                                     |  | ListRecords         | listing of N records                                  |
|                                     |  | GetRecord           | listing of a single record                            |

most verbs take arguments: dates, sets, ids, metadata formats  
and resumption token (for flow control)

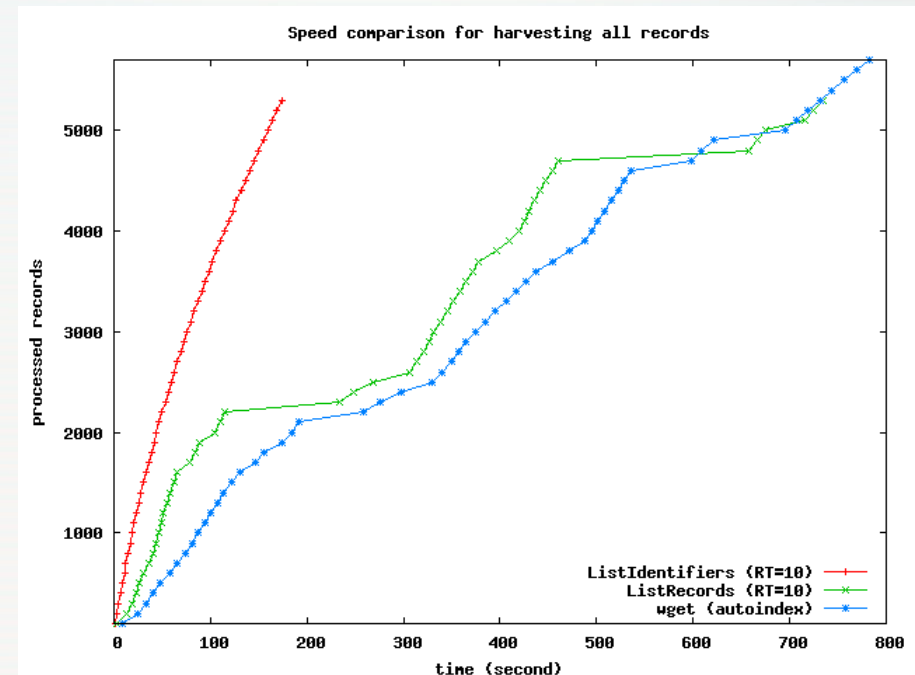
## A better way: using OAI-PMH to crawl a site

- Identify
  - Gives essential repository information
- ListRecords/ListIdentifiers
  - Lists all of the resources on the site
  - Can be “tweaked”:
    - Only those that are new since YYYY-MM-DD
    - Only those of MIME type <???
  - Streamlines crawling process
- ListSets
  - Tells the crawler what kind of groupings the site supports
- 6 Verbs in All
- Streamlined initial crawl, fast update crawls

# Performance Comparison: Initial Crawl



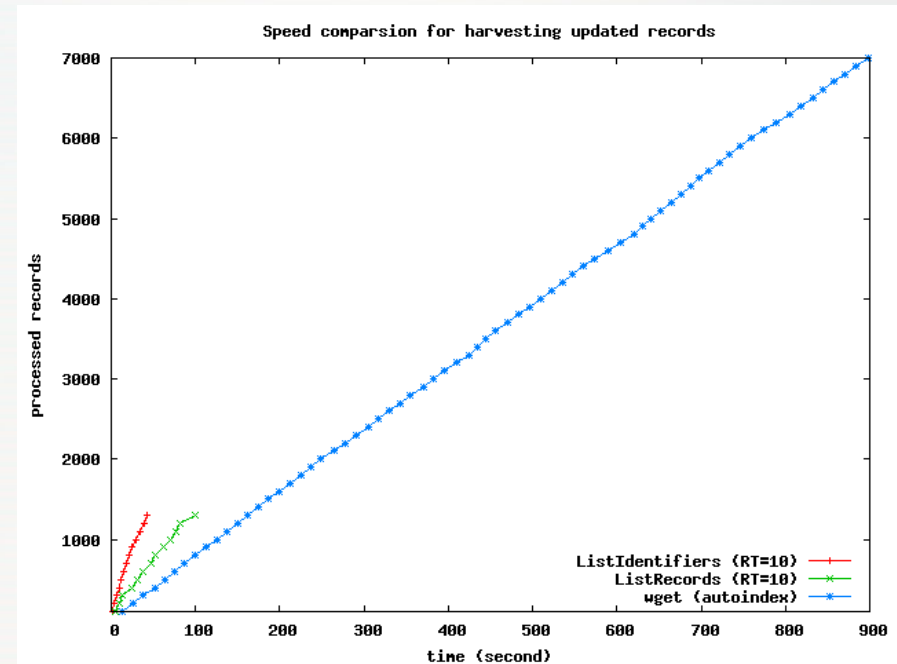
- All crawlers
  - Must ask for **every** resource
  - Discovery faster, automatic for mod\_oai mod\_oai
- ListIdentifiers
  - Only an OAI-PMH verb
  - Could be used to create an index of resource names
  - Gets unlinked **and** linked resources resources
- ListRecords
  - Only an OAI-PMH verb
  - Returns metadata **plus** resource
  - Gets unlinked **and** linked resources resources
- wget
  - Behaves like common crawler
  - Can only find linked resources



# Performance Comparison: Update Crawl



- Performance improved using mod\_oai (OAI-PMH)
  - Conditional request is streamlined
- If only new/changed pages are requested:
  - OAI-PMH crawler:
    - “GET from yyyy-mm-dd” (last (last visit date)
    - **One** request gets all the new new data
  - Standard crawler
    - “GET if-modified-since”
    - Must ask for **every** page



# OAI-PMH Verbs & Special Features



- Verbs:
  - Identify
    - Provides descriptive metadata about the DL
  - ListIdentifiers
    - Returns record headers only
    - Resumption token manages lengthy data set
    - Unique identifier for each site resource
  - ListMetadataFormats
    - Specifies types of metadata tracked by the site
    - Options include Dublin Core, MARC, DIDL, RFC1807, others...
    - Dublin Core is required by OAI specification
  - ListRecords
    - Sequential transfer of each record
    - Can limit to N records (flow control for crawler)
  - ListSets
    - Defined locally via scripts to aggregate common record groups
    - Facilitates selective harvesting of site
    - MIME-Type sets are automatically supported by mod\_oai
  - GetRecord
    - Selects specific, single record from site
    - Identified by the OAI unique identifier
- Special Features:
  - Datestamp harvesting
    - Example: Give me all records updated between 2005-10-05 and today  
“[http://www.xyz.us/oai?verb=ListRecords&from=2005-10-05&until=2006-06-11&metadatasprefix=oai\\_dc](http://www.xyz.us/oai?verb=ListRecords&from=2005-10-05&until=2006-06-11&metadatasprefix=oai_dc)”
  - Metadata only –or:
    - Full record; encapsulated as DIDL –or:
    - A complete package with all of this information
      - Akin to OAIS AIP



# Constructing an OAI-PMH Query

- Start with the site's main URL

<http://www.foo.org/>

- Add the baseURL location\*:

<http://www.foo.org/modoai>

- Add the OAI-PMH verb:

<http://www.foo.org/modoai?verb=GetRecord>

- Add the metadata prefix:

[http://www.foo.org/modoai?verb=GetRecord&metadataPrefix=oai\\_dc](http://www.foo.org/modoai?verb=GetRecord&metadataPrefix=oai_dc)

- Add any other qualifiers...

[http://www.foo.org/modoai?verb=GetRecord&metadataPrefix=oai\\_dc&identifier=  
<http://www.foo.org/bluebells.html>](http://www.foo.org/modoai?verb=GetRecord&metadataPrefix=oai_dc&identifier=http://www.foo.org/bluebells.html)

\*usually defined from root URL, but can begin at some other point in the site

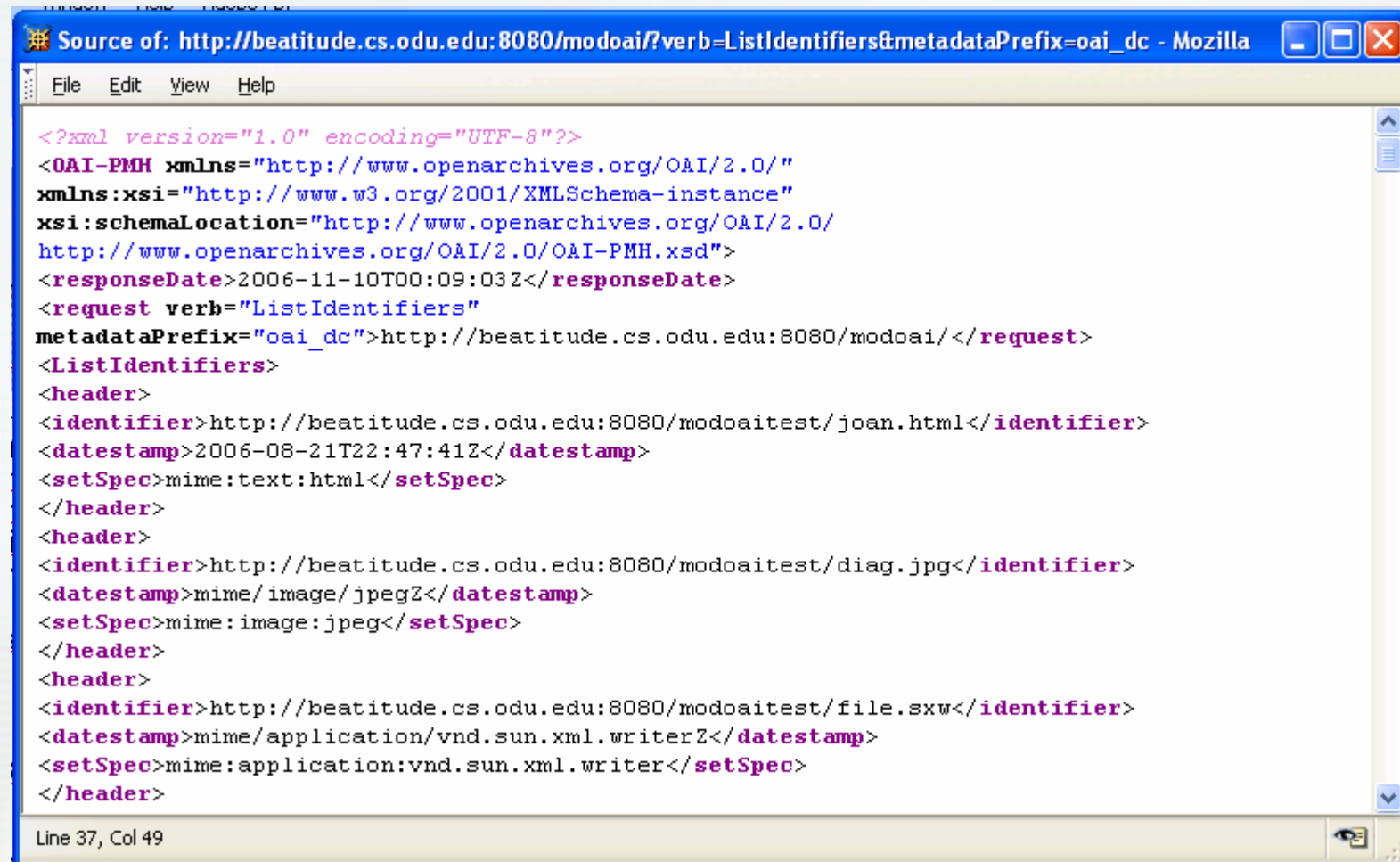
# The OAI-PMH Identify Verb



GET : <http://beatitude.cs.odu.edu:8080/modoai/?verb=Identify>

```
- <OAI-PMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-11-10T12:54:42Z</responseDate>
  <request verb="Identify">http://beatitude.cs.odu.edu:8080/modoai</request>
  - <Identify>
    <repositoryName>mod_oai Demo Repository</repositoryName>
    <baseURL>http://beatitude.cs.odu.edu:8080/modoai</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>jsmrit@cs.odu.edu</adminEmail>
    <earliestDatestamp>2005-08-03T12:00:00Z</earliestDatestamp>
    <deletedRecord>no</deletedRecord>
    <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
    - <description>
      <friends xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/friends/
http://www.openarchives.org/OAI/2.0/friends.xsd"/>
    </description>
    - <description>
      - <gateway xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/gateway/
http://www.openarchives.org/OAI/2.0/gateway.xsd">
        <source>http://beatitude.cs.odu.edu:8080</source>
        <gatewayDescription>http://www.modoai.org</gatewayDescription>
        <gatewayAdmin>jsmrit@cs.odu.edu</gatewayAdmin>
      </gateway>
    </description>
    </Identify>
  </OAI-PMH>
```

# ListIdentifiers Response Content



```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2006-11-10T00:09:03Z</responseDate>
<request verb="ListIdentifiers"
metadataPrefix="oai_dc">http://beatitude.cs.odu.edu:8080/modoai/</request>
<ListIdentifiers>
<header>
<identifier>http://beatitude.cs.odu.edu:8080/modoaitest/joan.html</identifier>
<datestamp>2006-08-21T22:47:41Z</datestamp>
<setSpec>mime:text/html</setSpec>
</header>
<header>
<identifier>http://beatitude.cs.odu.edu:8080/modoaitest/diag.jpg</identifier>
<datestamp>mime/image/jpegZ</datestamp>
<setSpec>mime:image/jpeg</setSpec>
</header>
<header>
<identifier>http://beatitude.cs.odu.edu:8080/modoaitest/file.sxw</identifier>
<datestamp>mime/application/vnd.sun.xml.writerZ</datestamp>
<setSpec>mime:application/vnd.sun.xml.writer</setSpec>
</header>
```

Line 37, Col 49

# Search Engine Use of OAI-PMH



- Google sitemaps: OAI-PMH or Do-It-Yourself
  - Via OAI-PMH
    - Just send them the baseURL!
    - Google does a ListRecords query on your site
  - Via Google's tool or manually constructed
    - XML-formatted file; URI/IRI compliant
    - Follow schema: <http://www.google.com/schemas/sitemap/0.84/sitemap.xsd>
    - ASCII and UTF-8 encoded (escaped quotes, ampersands, etc)
    - Limited size: 50,000 urls, 10mb max (per sitemap file)
- MSN Academic Live
  - Digital-library-centric (not general web)
  - Specifically states it can access OAI-PMH repositories
  - Unclear if role will grow to include MSN Search
    - [http://academic.live.com/Publishers\\_Faq.htm](http://academic.live.com/Publishers_Faq.htm)
- Yahoo
  - No sign-up guidelines for OAI-PMH-enabled sites
  - Yet... research showed good coverage of OAI-PMH Repositories
    - Outsourced OAI-PMH crawls [\[1\]](#)
    - OAIster (U Michigan Library) provides Yahoo with OAI repository information
- Professional Digital Libraries
  - Many support OAI-PMH
  - Many are not open to commercial search engines



# Google Sitemaps Using OAI-PMH

<http://www.google.com/support/webmasters/bin/answer.py?answer=34655&ctx=sibling>


**Webmaster Help Center**
Change Language: English

[Google Help](#) > [Help Center Home](#) > [Managing my Google Sitemaps files](#) > [Creating Google Sitemaps files](#)

**Documentation**  
[Webmaster Guidelines](#)  
[How do I add my site?](#)  
[About Google webmaster tools](#)  
[Webmaster Central](#)

**Tools**  
[Google webmaster tools](#)  
[Submit your content to Google](#)  
[Google services and tools](#)

## How do I submit my OAI-PMH path?

Google accepts the [OAI-PMH](#) version 2.0 protocol. You can't use this format for [Mobile Sitemaps](#). If you use this format for your site, simply [add](#) the baseURL of your OAI repository (for instance, <http://www.example.com/oaiserver>). When we query the baseURL, we automatically add query parameters (such as `?verb=Identify` or `?verb=ListRecords`), so you can simply add the baseURL itself. When we extract the URLs for your site, we expect the records in the repository to be formatted using [Dublin Core](#), with the URLs embedded in `<dc:identifier>` tags. Below is a sample record that includes the `<dc:identifier>` tag in bold. The URL listed in that tag is what we extract.

```
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title xml:lang="en">A title of extraordinary things</dc:title>
  <dc:creator>McCormack, Michael</dc:creator>
  <dc:subject>LCSH:Ausdehnungslehre; LCCN QA205.H99; Greatness:Amanda</dc:subject>
  <dc:publisher>J. Wiley & Sons</dc:publisher> <dc:date>Created: 1906; Available: 1991</dc:date>
  <dc:type>text</dc:type>
  <dc:identifier>http://example.com/physics/1796949</dc:identifier>
  <dc:language>english</dc:language>
  <dc:rights xml:lang="en">Public Domain</dc:rights> </oai_dc:dc>
```

As with other Sitemaps, the URLs must be within the same site and at the same directory location or lower than the baseURL. For instance, if you add <http://www.example.com/oaiserver> as the baseURL, the following URLs would be valid:

- <http://www.example.com/>
- <http://www.example.com/samples.html>
- <http://www.example.com/images/>

However, if you add <http://www.example.com/dataprovider/oaiserver>, then none of those URLs would be valid.

**You may also be interested in...**

- [How do I create a text file to submit as a Sitemap?](#)
- [What other formats can I use for a Sitemap?](#)
- [How do I submit a syndication feed?](#)

©2006 Google

XML Format info here:

<https://www.google.com/webmasters/sitemaps/docs/en/protocol.html#sitemapXMLFormat>



# What's A Dublin Core?



- Basic data set (fields) about something
  - Like the information on a library card catalog
  - Specifies certain elements
  - More than one “style” of DC: simple & qualified
  - Most people mean “simple” when they say DC
- Simple DC has 15 information fields:

|                |                |
|----------------|----------------|
| 1. Title       | 9. Format      |
| 2. Creator     | 10. Identifier |
| 3. Subject     | 11. Source     |
| 4. Description | 12. Language   |
| 5. Publisher   | 13. Relation   |
| 6. Contributor | 14. Coverage   |
| 7. Date        | 15. Rights     |
| 8. Type        |                |

# Improving Crawls Using mod\_oai



- Google sitemaps for OAI-PMH sites
  - currently harvests Dublin Core only
  - Uses your baseURL to crawl your site
  - Uses the date feature to get newest information
- Complex-object format/MPEG-21 DIDL
  - New OAI-PMH approach combines resource + metadata
  - Big files, but –
    - Could use gzip, deflate if server supports it (many do)
    - Still more efficient than traditional crawling
    - Can provide lots of useful metadata
  - Simplifies crawls
    - ListRecords gets everything
    - ListRecords + date range = fast updates
- Any crawler could request MPEG-21 DIDL format (oai\_didl)
  - Google could easily adopt it since they already use ListRecords
  - Any search engine looking for competitive edge could implement DIDL metadata metadata prefix to streamline crawls
  - Intranets could adopt this approach for archiving their internal web
  - Encoded base64 resource is also easy to decode for analysis or restoration restoration

# How does mod\_oai work?



- Code
  - Written in C
  - Designed to be platform-independent
    - Requires Apache 2
    - Uses APSX2 calls
    - Linux, MAC compatible
- Runs as a web server process
  - Installed like mod\_perl or mod\_deflate, for example
  - Config file handles module specifics (baseURL location, etc)
  - Enables OAI-PMH verbs to appear in the HTTP request
    - baseURL + verb gets OAI-PMH response
- The rest of the site works as normal
  - Users see no change
  - Standard crawlers can operate as usual

# Complex Object Formats: Characteristics

- Representation of a digital object by means of a wrapper XML document.
- Represented resource can be:
  - simple digital object (consisting of a single datastream)
  - compound digital object (consisting of multiple datastreams)
- Include datastream:
  - By-Value: embedding of base64-encoded datastream
  - By-Reference: embedding network location of the datastream
  - Descriptive metadata, rights information, technical metadata, ...
- MPEG-21 DIDL is one type of complex object format
  - Can be used in OAI-PMH
  - Metadata prefix for mod\_oai is “oai\_didl”

In other words:

- Instead of just looking at the index card about the book, we can actually get the book, too

Let's look at an example GetRecord verb for a very simple resource  
( <http://beatitude.cs.odu.edu/modoaitest/joan.html> )

# GetRecord: Get the Id *and* the Data



[http://beatitude.cs.odu.edu:8080/modoai?verb=GetRecord  
&Identifier=http://beatitude.cs.odu.edu:8080/modoaitest/joan.html  
&metadataPrefix=oai\\_didl](http://beatitude.cs.odu.edu:8080/modoai?verb=GetRecord&Identifier=http://beatitude.cs.odu.edu:8080/modoaitest/joan.html&metadataPrefix=oai_didl)

- oai\_didl metadata format (prefix)
- Complex object response
  - Encapsulates resource within the response
  - Encodes it as base64
- *Everything* known about the URL is in the response
  - All of the metadata types and the contents
    - Dublin Core
    - HTTP Headers
    - Any others that might be used by that server...



# Actual GetRecord Response (oai\_didl)



```
Source of: http://beatitude.cs.odu.edu:8080/modoai?verb=GetRecord&metadataPrefix=oai_didl&identifier=http://beatitude.cs.odu.edu:8080/modoaitest/...
File Edit View Help

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2006-11-10T00:48:04Z</responseDate>
<request verb="GetRecord" identifier="http://beatitude.cs.odu.edu:8080/modoaitest/joan.html"
metadataPrefix="oai_didl">http://beatitude.cs.odu.edu:8080/modoai/</request>
<GetRecord>
<record>
<header>
<identifier>http://beatitude.cs.odu.edu:8080/modoaitest/joan.html</identifier>
<datestamp>2006-08-21T22:47:41Z</datestamp>
<setSpec>mime:text/html</setSpec>
</header>
<metadata>
<didl:DIDL xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS http://purl.lanl.gov/STB-RL/schemas/2004-11/DIDL.xsd">
<didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">http://beatitude.cs.odu.edu:8080/modoaitest/joan.html</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<http:header xmlns:http="http://www.modoai.org/OAI/2.0/http_header/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.modoai.org/OAI/2.0/http_header/ http://purl.lanl.gov/STB-RL/schemas/2004-08/HTTP-HEADER.xsd">
<http:Content-Length>69</http:Content-Length>
<http:Server>Apache/2.0.49 (Fedora)</http:Server>
<http:Content-Type>text/html</http:Content-Type>
<http:Last-Modified>Mon, 21 Aug 2006 22:47:41 GMT</http:Last-Modified>
<http:Date>Fri, 10 Nov 2006 00:48:04 GMT</http:Date>
</http:header>
</didl:Statement>
</didl:Descriptor>
<didl:Component>
<didl:Resource mimeType="text/html"
encoding="base64">PGh0bWw+Cgk8Ym9keT4KCQ13b28taG9vCgkJPgk8bGk+YmxhaAoJCTwvdWw+Cgk8L2JvZHK+CjwvaHRtbD4K</didl:Resource>
<didl:Resource mimeType="text/html" ref="http://beatitude.cs.odu.edu:8080/modoaitest/joan.html"/>
</didl:Component>
</didl:Item>
</didl:DIDL>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

"joan.html"  
encoded in base64

# Summary: mod\_oai to the rescue!



- Search engines are taking a real interest in OAI-PMH as a means to improve crawling
- mod\_oai is an Apache 2.0 module that provides OAI-PMH interface for your site (currently Linux & Mac)
- You can send the baseURL to Google
- The module is relatively simple to install
- It won't affect regular site users and regular web crawlers
- Any changes to your site will be reflected by the mod\_oai server
- It makes crawling much faster, more efficient, more useful

# For more information



- A website with mod\_oai releases, demos and documentation is maintained by Old Dominion University and LANL:  
<http://www.modoai.org/>
  - New release next month
  - Improved installation process
- The Open Archives Initiative also maintains a web site:  
<http://www.openarchives.org/>
  - Forum, tutorials, news, research
  - OAI-PMH information
- There are active research projects at ODU using mod\_oai
  - Web preservation
  - Repository ingestion/handling