Generating Best-Effort Preservation Metadata for Web Resources At Time of Dissemination

Joan A. Smith and Michael L. Nelson
Old Dominion University, Department of Computer Science
Norfolk, VA 23529 USA
{jsmit, mln}@cs.odu.edu

ABSTRACT

HTTP and MIME, while sufficient for contemporary web page access, do not provide enough forensic information to enable the long-term preservation of the resources they describe and transport. But what if the originating web server automatically provided preservation metadata encapsulated with the resource *at time of dissemination*? Perhaps the ingestion process could be streamlined, with additional forensic metadata available to future information archeologists. We have adapted an Apache web server implementation of OAI-PMH which can utilize third-party metadata analysis tools to provide a metadata-rich description of each resource. The resource and its forensic metadata are packaged together as a complex object, expressed in plain ASCII and XML. The result is a CRATE: a self-contained *preservation-ready* version of the resource, created *at time of dissemination.*

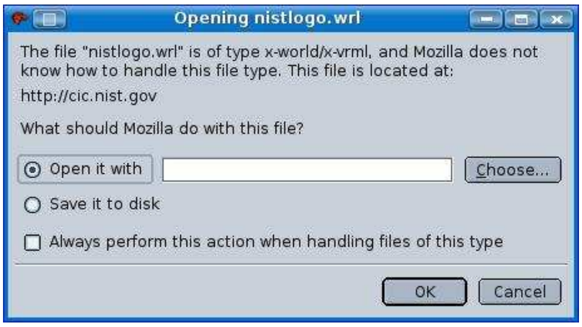
Categories and Subject Descriptors:H.3.5Information Storage and Retrieval Online Information Services [Web-based services]

General Terms Design, Documentation, Experimentation

Keywords Web preservation, OAI-PMH, mod_oai

1. HTTP, MIME AND HTML

HTTP, MIME, and HTML form the foundation of the web. Optimized for the "here and now", they have survived more than 10 years of web evolution. Once mostly plain ASCII text or Hypertext (HTML), many World Wide Web sites now contain application-specific files (Flash, Video, multimedia), non-hypertext documents (Adobe PDF, Word files, XML files) and enhanced hypertext content (XHTML, CSS). Successful access to this variety of resources is accomplished in part thanks to MIME typing, which identifies a resource as belonging to one of 8 major types, each of which has a variety of subtypes. Servers and browsers are individually configured to recognize various MIME types as defined by IANA, but the MIME Content-Type entity header sent by the server provides only bare-bones information about the resource, and in most cases relies on the file extension for identification. Problems can arise if the typing and content are mismatched. For example, the file `http://beatitude.cs.odu.edu:9999/falsePdf.pdf` is a UTF-8 encoded resource which has been renamed with the "dot-pdf" extension. Both the server and the client misidentify this file. Many browser brands attempting to access this file will generate an error. But if `falsePdf.pdf` is downloaded and examined with a more capable tool like the Unix `file` command, the "real" file format is recognized as "UTF-8 Unicode English Text". The MIME typing process was misled by the "pdf" extension.



The file "nistlogo.wrl" is of type x-world/x-vrml, and Mozilla does not know how to handle this file type. This file is located at:
http://cic.nist.gov

What should Mozilla do with this file?

● Open it with

○ Save it to disk

□ Always perform this action when handling files of this type

Figure 1: An unrecognized MIME type

In some cases, not enough information is given to access the resource once it is received. For example, a Content-Type of `application/octet-stream` could be an Open Office document, an Excel spreadsheet, or some other file format not recognized by the server. Another frequent scenario is where the server understands the type, but the client does not. Figure 1 illustrates the case where the server reports that "nistlogo.wrl" is an "x-world/x-vrml" file type, but the browser does not know what to do with it. VRML files, popular in the 1990s, are just one of many formats that have fallen into disuse. Travelling back in time, we might be able to get more useful metadata on the file: the best time to get information about a VRML file was about 10 years ago. Certainly, the minimal metadata generated by crawling the site for this resource is unlikely to prove sufficient for historians in the year 2100.

2. THE CRATE MODEL

For preservation, we need as much metadata as possible: keyword list, content summary, subject, structural details,

```
<Location /modoai>
   SetHandler        modoai-handler
   modoai_oai_active ON
   <modoai_plugin>
      label "jhove" %JHOVE
      exec  "/opt/jhove/jhove -m gif-hul %s"
      mime  "image/gif"
   </modoai_plugin>
   <modoai_plugin>
      label "ots" %Open Text Summarizer
      exec  "/usr/local/bin ots -summary %s"
      mime  "text/*"
   </modoai_plugin>
   <modoai_plugin>
      label "pronom" %PRONOM DROID tool
      exec  "java -jar DROID.jar -L%s"
      mime  "*/*"
   </modoai_plugin>
</Location /modoai>
```

Figure 2: Metadata Plug-In Implementation

copyright, authorship, application version, etc. The need for format information is especially important with image files which vary greatly in color-depth, compression algorithms, and other features. For the VRML file above, it would be useful to have a script which could use GDFR [1] or PRONOM [3] for a deeper inspection of the file. A variety of Open Source and commercial tools capable of analyzing files and generating preservation metadata can be found scattered over the web: Jhove, Open Text Summarizer, KEA, ExifTool, and others. The typical web server does not operate such tools; instead, they are applied by archivists to resources at the time of ingest. Crawlers move through sites, storing the resources they find, and applying analysis tools later. But the best time to analyze a file is at the point of dissemination, when the server is more likely to be able to provide preservation-related information.

What if metadata tools could be seamlessly integrated into the web server, to provide preservation support when the resource is crawled? The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) offers a solution to this dilemma since it supports complex object types such as METS and MPEG-21 DIDL [2]. We can use this format as a container, or CRATE, for our preservation-prepared resource. Since OAI-PMH was already implemented as an Apache web server module [4], *mod_oai*, we used it as our experimental prototype. Like other Apache modules, mod_oai activity is controlled through the web server configuration file (httpd.conf). A snippet from the mod_oai section is shown in Figure 2.

The webmaster uses the <Location> directive to define a *baseURL*, `http://www.foo.edu/modoai/`, as the entry point. Any incoming request using the baseURL will be handled like an OAI-PMH request. For example:

```
http://www.foo.edu/modoai/?verb=GetRecord
&identifier=http://www.foo.edu/barr.html
&metadataPrefix=oai_didl
```

The response is an XML-formatted document containing the HTTP-Header metadata (file type, modification date, etc.) and the resource itself, “barr.html”, converted to base64 and included within the XML-formatted response as part of the object description. In this case, the metadata is the *resource itself.* Note that mod_oai does not impact normal user access to the web server. Visitors to `http://www.foo.edu/` will still find the same site as before; `http://www.foo.edu/barr.html` returns the original, browser-friendly view of the resource.

Yet the oai_didl metadata format (MPEG-21 DIDL) *could* contain more information, if it were available on the server. We therefore expanded mod_oai to accept *plug-ins*, third-party tools which analyze the resource as it is being requested. These tools build the CRATE, a complex object consisting of the resource and its associated metadata. Plug-ins are implemented on a per-resource basis depending on file type; they can be applied to only a certain set of files or to every resource (Figure 2). This method lets each local web server have its own set of metadata extraction utilities, defined by the web master. The resulting metadata is wrapped – serialized – in the XML of the response, producing the preservation-ready resource (CRATE) we are looking for. An advantage with this approach is that the plain ASCII text XML-structured content that makes up the response is likely to survive in the long-term, enhancing the probability of future recoverability of the contents of our CRATE.

3. FINDINGS & DISCUSSION

We have tried several types of plug-ins with mod_oai: Jhove, KEA, Open Text Summarizer, MD5, and others. Anything that can run automatically is likely to be compatible. Scripts that further customize plug-in usage also work. For example, Jhove has a number of analysis or “HUL” modules (ASCII, TIFF, JPEG, etc.) targeted to specific file types. Rather than create a dozen `<modoai_plugin>` sections, the `exec` field could point to a script, `preJhove.sh`, which examines the file and then calls the appropriate HUL module.

There are two points we would like to emphasize. First, the CRATE process is *fully automated* – the metadata is not validated by the web server nor by any other administrative action. Second, the metadata is generated *at time of dissemination*; it is not pre-processed nor canned. The metadata thus reflects the best-information available at that point in time. This approach harnesses the web server itself to support preservation, moving the burden from a single web-wide preservation master to individual web servers, where detailed information about the resource is most likely to reside. It also moves preservation metadata from *strict validation at ingest* to *best-effort description at dissemination.* In other words, the web server acts as its own agent of preservation by providing the crawler with sufficient information to assist the preservation process at the time the site is crawled.

4. REFERENCES

[1] S. L. Abrams and D. Seaman. Towards a global digital format registry. In *World Library and Information Congress: 69th IFLA General Conference and Council*, 2003.

[2] J. Bekaert, E. De Kooning, and H. Van de Sompel. Representing digital assets using MPEG-21 Digital Item Declaration. *International Journal on Digital Libraries*, 6(2):159–173, 2006.

[3] J. Darlington. PRONOM - a practical online compendium of file formats. *RLG DigiNews*, 7(5), 2003.

[4] M. L. Nelson, J. A. Smith, I. Garcia del Campo, H. Van de Sompel, and X. Liu. Efficient, automatic web resource harvesting. In *Proceedings of WIDM 2006*, pages 43–50, 2006.