Analyzing Metadata for Effective Use and Re-Use

Naomi Dushay, Diane I. Hillmann
Cornell University, National Science Digital Library, USA
naomi@cs.cornell.edu
dih1@cornell.edu

Abstract

Using visual graphical analysis methods, the National Science Digital Library (NSDL) has developed techniques to expedite evaluation of large batches of metadata. These techniques allow efficient and thorough review of large quantities of XML metadata, thus enabling the focus of limited resources on evaluation and manipulation tasks that are most important in our context. In the NSDL, metadata is evaluated for aggregation, but these techniques are applicable to any situation where batches of metadata need to be evaluated. This paper discusses the motivations for these techniques and the techniques themselves.

Keywords: Metadata, Evaluation, Visual Graphical Analysis.

1. Evaluation of metadata in the past

Most experience with large-scale metadata aggregation has been acquired in major academic library environments with MARC data. The world of library MARC data sharing is a relatively controlled one, where content standards for metadata are widely understood and well documented. Records travel from libraries to bibliographic utilities and back, at each stage passing through various automated edit checks and often some level of human scrutiny. In this world, typographic errors and outdated headings are normal problems, but more serious quality surprises are few and most likely attributable to inadequate training. Managers of MARC databases rely heavily on each other to maintain a reasonable level of predictability, and for the most part, they are not disappointed.

In the MARC environment, sophisticated tools to evaluate existing metadata are neither generally available nor sorely missed. Library applications typically provide fairly comprehensive edit checks for controlled vocabulary or coded values and for valid tags when individual records are imported or created. For batch imports of records, random sampling for quality evaluation is the norm, though ordinarily some testing of the imported file in a non-production context is done as well. Since most batches of records are acquired from well-known suppliers, there is little impetus to invest in other techniques to assess record quality and adherence to standards.

The emerging metadata world is by contrast the Wild West. Instead of trained library workers applying well-documented content standards to describe a relatively small number of resource formats, there are untrained people working largely in isolation (and without adequate documentation) to describe an increasingly complex array of resources. At the other end of the new metadata spectrum from the human created records are those created by automated means, oftentimes with ill-documented methods, containing little or no provenance. It is no surprise that metadata available to aggregators like the National Science Digital Library (NSDL) varies strikingly in quality, and does not play well together. Without a predictable minimum level of quality, building services relying on this metadata is difficult, at best.

2. Overview of the NSDL

A primary goal of the NSDL is to transform the use of digital resources in science education, in the broadest sense. As part of this effort, we are creating a Metadata Repository (MR) by gathering large amounts of metadata pertaining to resources in the fields of science, technology, engineering and mathematics. In this respect, the NSDL functions as a metadata aggregator.

Because the NSDL has a strong focus on education, many of its funded projects are collecting or developing complex learning objects. Others are in early stages of metadata planning and are contemplating the use of a variety of formats. The NSDL determined that given this diversity, use of the Dublin Core in its qualified version, with the proposed education extensions, made the most sense for the NSDL Metadata Repository. This is primarily because it balances relative simplicity of use with some ability to express educational concepts.

The NSDL uses a two-tier model comprised of "collections" and their "items." An item may be large or small, and it may itself contain parts or smaller units; a collection is defined as an organized arrangement of items. Associating every individual item with a collection, though fairly primitive as an organizing principle, allowed for some simple assertions of quality based on the reputation of the entity responsible for the collection. This simple collection/item principle also facilitated "ingestion" of metadata into the MR with a minimum of intervention. One potential drawback to this approach is that collections "A" and "B" can contain duplicate items, but we decided that this must be accommodated for two reasons: 1) collections need to be able to update their metadata, and 2) we have no way of determining which record of several for the same resource is "best" and should be retained over others. Another difficulty with the two-tiered collection/item approach is that NSDL collections aggregate item-level materials at many granularity levels – from whole Web sites to individual image files to applets, and everything in between.

As we began processing item level records, we learned that evaluating these records was more difficult than we anticipated. In order to provide a reasonable level of quality and predictability for our developing search service and user interface, and to be able to expose more consistent, "normalized" records as an Open Archives Initiative (OAI) data provider [1], a process of evaluation and transformation of incoming metadata was necessary. In order to make such processing scalable, difficult choices about what constitutes an appropriate level of quality must be made and sensible priorities maintained.

3. Evolving DC Metadata environment at NSDL

We identified a number of requirements as we developed our first prototype of the NSDL in 2001--a small-scale exemplar of what we hoped to achieve on a larger scale, given longer term funding. One requirement was the ability to focus and limit searches in a variety of ways so that users could find what they needed without scrolling through many pages of search results. Another was the ability to highlight individual collections and the materials within them to create a lively and interesting user interface.

After our prototype was completed we began designing a robust MR for the NSDL — one that was expandable and flexible enough to survive into an unknown future. We continued to refine our metadata strategy as the OAI Protocol for Metadata Harvesting (OAI-PMH) [2] matured enough for us to use. We also focused on how to reduce the time spent evaluating and transforming each collection's metadata. Clearly, if we were to meet our goals for a large, useful repository of materials with minimal human effort, we needed to take seriously the limits of our resources and technology. We did not have the staff or funding to "perfect" the metadata coming in to the MR, nor that being served out from the MR via our OAI server.

As we began to move our old prototype metadata into our new MR, it became clear that the lack of tools to examine and to modify large batches of data was a significant problem. Looking at metadata newly available to us as an OAI harvester was not reassuring, as it also had significant problems, many of which random sampling could not begin to identify. These problems fell into several categories:

3.1. Missing data

Because we were interested in providing simple search limits based on resource type and format, data missing from the Dublin Core "format" and "type" elements were particularly problematic. In many cases, the entire collection consisted of materials in one format or of one type, and the missing information was deemed unnecessary for the collection's local purposes. In other cases, the metadata was very brief, or was taken from an earlier store of metadata that did not include the information.

3.2. Incorrect data

Incorrect data came in various flavors. Among other problems, we found creator names repeated in the language element, or the identifier for the metadata record repeated in the Dublin Core identifier element. In at least one collection's metadata, we found correct and fully formatted type metadata in the format element and vice versa. There were also many instances where misunderstanding or misreading of Dublin Core definitions resulted in very odd records. Some oddities were so consistent that we assumed they were coded into a crosswalk.

We saw various defaulted strings signifying 'no information available' -- generally something like "unknown" (sometimes abbreviated or misspelled), and we also saw elements comprised solely of stray characters such as dashes or hyphens. Some examples:

```
<dc:description>unknown</dc:description>
<dc:description> -- </dc:description>
<dc:description> … </dc:description>
<dc:description>No abstract available. </dc:description>
<dc:source>No source: created in machine-readable format</dc:source>
```

We speculated that much of this metadata bubbled up from computer programs converting database entries into XML metadata, or the metadata existed to support an application that required those text strings or symbols for display. Although the case could be made that this data was not technically "incorrect" (Dublin Core being notoriously agnostic on such matters), it had the potential to interfere with searches and to be a serious annoyance in aggregate.

3.3. Confusing data

In this category we encountered strings of names, sometimes inconsistently ordered or ambiguously separated (commas used to separate surname from forename and also to separate names). Some of these may have been citation or bibliography forms designed for efficient sorting of entry elements dumped without revision into a metadata record.

```
<dc:creator>Smith, John, George Jackson, Humphrey Little and Stanley Black</dc:creator>
<dc:contributor>Sanders, G.S., T.R. Brice, V.L. DeSantis, and C.C. Ryder.</dc:contributor>
```

Also in this category was a great deal of HTML tagging within values, most likely cut and pasted from Web HTML text, but difficult to interpret in the context of a search result. Often these records also included characters illegal in XML such as un-encoded ampersands, or bad UTF-8 encoding, or double encoding of XML entities. These had a tendency to cause problems in the NSDL user interface.

<dc:identifier>http://http://muffin.dog.org</dc:identifier>
<dc:description> …. < Mammals …</dc:description>

3.4. Insufficient data

As time passed it became increasingly clear that one critical disconnect was occurring in the juncture between the minimal OAI-PMH requirements for simple Dublin Core, and the need for simple and predictable ways to refine search results. With qualified DC, particularly using recommended encoding schemes, there was a greater possibility of interpreting the metadata provided to us. Search limits allowing users to specify results of a particular format, for instance, were much easier to manage if we knew that the provider was using the IMT encoding scheme for DC format values or the DCMIType encoding scheme for DC type. We could test for particular values in plain text with no specific encoding scheme for a small controlled vocabulary, but for large vocabularies and/or for large collections, this approach didn't scale well.

For textual resources, our text-based search indexing (which crawled the resource itself, when possible, to supplement the metadata provided) sometimes could supply information that could substitute for missing metadata. But where crawling was impossible, due to permission issues or because the resource was not text-based, we were sometimes left with very little to support our search service users.

4. The current NSDL metadata evaluation process

4.1. XML random sampling with formatting and color-coded syntax

Currently the NSDL harvests metadata almost exclusively using OAI-PMH, which requires XML encoded metadata and provides a convenient 'administrative wrapper' around metadata to support harvesting and re-use by others. OAI-PMH requires simple DC as a served metadata format. Additional metadata formats may be provided, but they must validate to an XML schema. Figure 1 shows an OAI-PMH response containing simple DC metadata for two records.

Since the NSDL harvests metadata in a wide variety of formats and in batches, creating a traditional cataloging interface for the purpose of evaluation and augmentation of metadata would be difficult and costly. Such an interface would be oriented towards one-by-one review of metadata records in a single format -- insufficient for NSDL needs as well as expensive to create. In the beginning, we sought to review raw XML files using a desktop XML application, XMLSpy [3], using random sampling to identify problems. While the color-coded syntax and tabbed visual display in XMLSpy was very useful (as in Figure 1), reviewing more than a handful of metadata records using this method was tedious at best and ultimately unsatisfactory, primarily because it provided no pattern of error, nor any convenient way of determining the extent of a discovered problem within a file.

4.2. The spreadsheet approach

After missing several problems using the random sampling method to review large quantities of XML metadata, we began using a spreadsheet program (Microsoft Excel), which allowed rearrangement of the information for better visual review see Figure 2. Sorting by element name, and then sub-sorting by the values within elements, it was possible to quickly scroll through the data, looking for elements that were empty, contained bad data, or ascribed to an incorrect namespace. This spreadsheet strategy, in short, optimized the human review process by re-organizing the information to take advantage of the human brain's pattern detection abilities. This approach was an improvement over random sampling, but had drawbacks. In particular, for large files, this method is unwieldy due to the size of the file to be reviewed and unreasonable attention demands on the evaluator.

The spreadsheet strategy involved several steps between harvest and review. To get the OAI-PMH XML responses into a format readable by a spreadsheet, the metadata needed to be transformed into a tab-delimited format. XSL stylesheets were created to do this for each incoming metadata format [4]. The tab-delimited metadata would display in Excel something like Figure 2, when sorted by element name by element value.

Note that the example in Figure 2 is for simple Dublin Core. Other metadata formats, such as Qualified Dublin Core, have multiple XML namespaces possible for elements, and may also have attributes and attribute values that would be included in table based analysis.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:oai_dc="http://
www.openarchives.org/OAI/2.0/oai_dc/">
    <responseDate>2002-11-26T23:10:38Z</responseDate>
    <request metadataPrefix="oai_dc" verb="ListRecords">http://www.open-video.org/oai2.0/index.cgi</request>
    <ListRecords>
        <record>
            <header>
                <identifier>oai:www.open-video.org:4293</identifier>
                <datestamp>2002-04-15</datestamp>
            </header>
            <metadata>
                <oai_dc:dc>
                    <dc:title>Science in Action: The Flow of Heat (Part I)</dc:title>
                    <dc:description>Kinescope of Fifties science TV program featuring discussions and demonstrations. Guest: Dr.
Harvey R. White(University of California). Animal of the Week: Gopher. Host: Dr. Earl S. Herald.</dc:description>
                    <dc:subject>Science;Television programs;Physics;</dc:subject>
                    <dc:creator>California Academy of Sciences</dc:creator>
                    <dc:date>1956-00-00</dc:date>
                    <dc:source>Internet Archive</dc:source>
                    <dc:relation>Science in Action: The Flow of Heat (Part I)</dc:relation>
                    <dc:contributor>Internet Archive</dc:contributor>
                    <dc:format>MPEG-4</dc:format>
                    <dc:identifier>http://ftp.archive.org/movies/divx/50508a.avi</dc:identifier>
                    <dc:language>English</dc:language>
                    <dc:coverage>Ephemeral films</dc:coverage>
                    <dc:rights>Unrestricted use except for resell or conversion to formats other than open-source MPEG-4 format.
See http://ftp.archive.org/html/conditions.html for more information.</dc:rights>
                    <dc:publisher>Open Video</dc:publisher>
                    <dc:type>video</dc:type>
                    <dc:identifier>http://www.open-video.org/segment.php?seg_id=4293</dc:identifier>
                </oai_dc:dc>
            </metadata>
        </record>
        <record>
            <header>
                <identifier>oai:www.open-video.org:4024</identifier>
                <datestamp>2002-04-18</datestamp>
            </header>
            <metadata>
                <oai_dc:dc>
                    <dc:title>Master Hands (Part I)</dc:title>
                    <dc:description>Classic 'capitalist realist' drama showing the manufacture of Chevrolets from foundry to finished
vehicles. Though ostensibly a tribute to the 'master hands' of the assembly line workers, it seems more of a paean to the designers
of this impressive mass production system. Filmed in Flint, Michigan, just months before the United Auto Workers won union
recognition with their famous sitdown strikes. Released the same year as two other films with which it shares similarities: MODERN
TIMES and TRIUMPH OF THE WILL.</dc:description>
                    <dc:subject>Automobiles: Manufacturing;Labor: 1930s;Occupations: Automotive;</dc:subject>
                    <dc:creator>Chevrolet Motor Company</dc:creator>
                    <dc:date>1936-00-00</dc:date>
                    <dc:source>Internet Archive</dc:source>
                    <dc:relation>Master Hands (Part I)</dc:relation>
                    <dc:contributor>Internet Archive</dc:contributor>
                    <dc:format>MPEG-1</dc:format>
                    <dc:identifier>http://ftp.archive.org/movies/vcd/07806a.mpg</dc:identifier>
                    <dc:language>English</dc:language>
                    <dc:coverage>Ephemeral films</dc:coverage>
                    <dc:rights>Unrestricted use except for resell or conversion to formats other than open-source MPEG-4 format.
See http://ftp.archive.org/html/conditions.html for more information.</dc:rights>
                    <dc:publisher>Open Video</dc:publisher>
                    <dc:type>video</dc:type>
                    <dc:identifier>http://www.open-video.org/segment.php?seg_id=4024</dc:identifier>
                </oai_dc:dc>
            </metadata>
        </record>
    </ListRecords>
</OAI-PMH>
```

Figure 1. Two sample records as XML

metadata record id	element namespace	element name	element value
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	title	Science in Action: The Flow of Heat (Part I)
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	description	Kinescope of Fifties science TV program featuring discussions and demonstrations. Guest: Dr. Harvey R. White(University of California). Animal of the Week: Gopher. Host: Dr. Earl S. Herald.
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	subject	Science;Television programs;Physics;
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	creator	California Academy of Sciences
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	date	1956-00-00
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	source	Internet Archive
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	relation	Science in Action: The Flow of Heat (Part I)
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	contributor	Internet Archive
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	format	MPEG-4
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	identifier	http://ftp.archive.org/movies/divx/50508a.avi
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	language	English
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	coverage	Ephemeral films
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	rights	Unrestricted use except for resell or conversion to formats other than open-source MPEG-4 format. See http://ftp.archive.org/html/conditions.html for more information.
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	publisher	Open Video
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	type	video
oai:www.open-video.org:4293	http://purl.org/dc/elements/1.1/	identifier	http://www.open-video.org/segment.php?seg_id=4293
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	title	Master Hands (Part I)
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	description	Classic 'capitalist realist' drama showing the manufacture of Chevrolets from foundry to finished vehicles. Though ostensibly a tribute to the 'master hands' of the assembly line workers…
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	subject	Automobiles: Manufacturing;Labor: 1930s;Occupations: Automotive;
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	creator	Chevrolet Motor Company
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	date	1936-00-00
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	source	Internet Archive
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	relation	Master Hands (Part I)
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	contributor	Internet Archive
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	format	MPEG-1
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	identifier	http://ftp.archive.org/movies/vcd/07806a.mpg
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	language	English
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	coverage	Ephemeral films
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	rights	Unrestricted use except for resell or conversion to formats other than open-source MPEG-4 format. See http://ftp.archive.org/html/conditions.html for more information.
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	publisher	Open Video
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	type	video
oai:www.open-video.org:4024	http://purl.org/dc/elements/1.1/	identifier	http://www.open-video.org/segment.php?seg_id=4024

Figure 2. Two sample records in spreadsheet

4.3. Visual graphical analysis

On April 12, 2002, two members of our group attended a talk by Ben Shneiderman of the University of Maryland, in which he demonstrated a number of PC applications with innovative user interfaces. One such tool was Spotfire DecisionSite [5], a visual graphical analysis application (hereafter referred to as "Spotfire"). Soon afterwards, the NSDL obtained a demo copy of Spotfire, with hopes that we could take advantage of a visual graphical analysis approach to metadata evaluation. Visual graphical analysis goes several steps beyond spreadsheets, capitalizing on the human brain's ability to recognize visual patterns in order to analyze large quantities of information.[1]

Spotfire is in some respects "spreadsheets on steroids" [6] – a way for the end user to view up to six data dimensions simultaneously. Spotfire also allows its users to select data for display based on relevant characteristics (such as "don't display empty elements" or "look for all values that start with 'http://'" or "indicate which elements have an encoding scheme of W3CDTF"). Because of its visual graphical approach, Spotfire can display large quantities of data on one screen or the user can zoom in on any portion of the data via the Spotfire user interface.

Full text searching is also provided, as well as a spreadsheet like "tables" view of a file. Thus, the visual graphical analysis approach allows an evaluator maximum flexibility to review large quantities of data efficiently and thoroughly while using a minimum of programming resources.

Because Spotfire does not yet support import of XML files, we used the same XSL stylesheets developed for the spreadsheet approach to transform the XML metadata into files of tab-delimited information. These tab-delimited files are then imported into Spotfire, and used to perform a graphical analysis on the metadata.

Figure 3 is a screen shot of a Spotfire scatter plot for a collection's metadata. The element names are on the vertical axis, and the collection's metadata record identifiers are on the horizontal axis. Figure 3 also illustrates the use of color and size: each encoding scheme is represented by a different color and a different size. In this view, for example, red is the DCMIType encoding scheme, applied to the DC Type element. Green is the URI encoding scheme, valid for a number of elements including Identifier, Relation, References, etc. Blue is the IMT encoding scheme, applied to the DC Medium element refinement. Note that since IMT is not valid for the DC Medium refinement to the Format element, but only for the Format element itself, this view shows a problem with the data, and also shows the extent of the problem – all "Medium" fields appear to be affected. Using Spotfire, we could also display elements *without* encoding scheme IMT, and see if there are any "Medium" elements without it. Note that axes, color, and size can all be reassigned with a few mouse clicks by the Spotfire user – the display instantly adjusts to reflect the new selections. This allows the evaluator to manipulate the data in different ways during the evaluation process without having to use any programming resource.

Scatter plots such as Figure 3 allow us to detect patterns: the presence or absence of fields in a collection's metadata, patterns within particular fields or within groups of records.

This presents a view of the overall structure of a collection's metadata—the forest, instead of the individual trees. From this view, an evaluator could easily focus on where it was useful to look more closely at the data, and ask the most relevant questions based on that view of the data. For example, why did some records in Figure 3 have Publisher, Rights, etc. and some not—was this fact relevant to the quality of the data or did it reflect instead a diversity of formats? If a few records were missing identifiers, were they actually physical objects, or did the missing identifiers in fact make those records problematic for NSDL purposes?

Spotfire also provides a table view of the data, which is similar to our spreadsheet approach with a few extra features. All metadata values could be examined one element at a time in the table view, as in Figure 4, and this is easily accomplished in Spotfire. Figure 4 shows the values for the date fields in a particular collection's metadata. Most of the values present are four digits representing a year, but there are some exceptions to this rule. We also can see from the table that the encoding scheme W3CDTF has been applied correctly to only those values adhering to the scheme.

[1] Although we have no specific research to cite here, we suggest two books by Edward R. Tufte "The Visual Display of Quantitative Information" (1983) and "Envisioning Information" (1990) as good places to start. A more recent work, "The Craft of Information Visualization: Readings and Reflections" (2003) written and edited by Benjamin B. Bederson and Ben Shneiderman, is more specific to the use of computers.

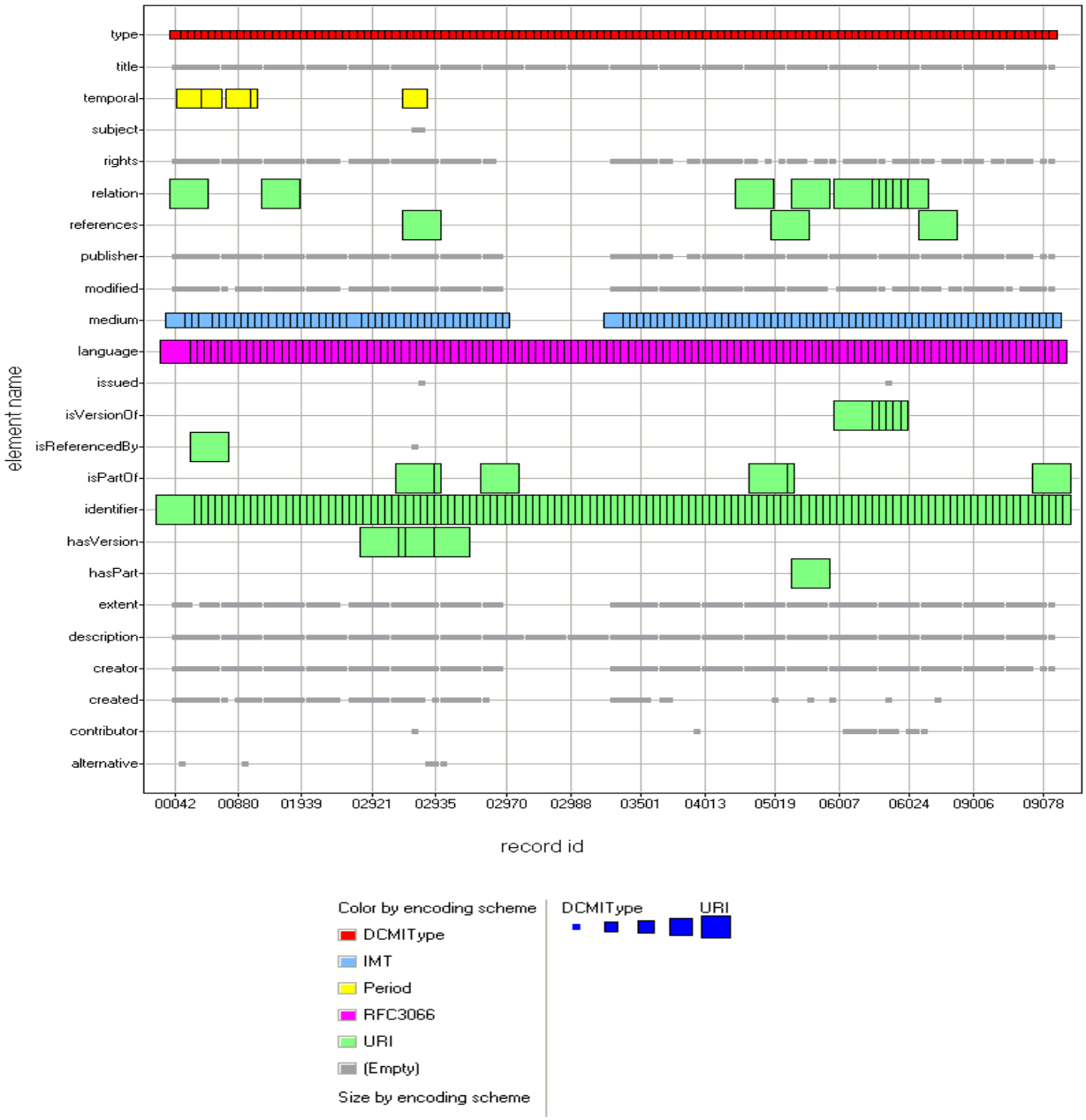

Figure 3. Spotfire scatter plot for a collection's metadata

metadata record id	element namespace	element name	element value	attribute name	attribute value
ScoutNSDL-821	http://purl.org/dc/elements/1.1/	date	1996	xsi:type	dct:W3CDTF
ScoutNSDL-826	http://purl.org/dc/elements/1.1/	date	1996	xsi:type	dct:W3CDTF
ScoutNSDL-842	http://purl.org/dc/elements/1.1/	date	1996	xsi:type	dct:W3CDTF
ScoutNSDL-822	http://purl.org/dc/elements/1.1/	date	1997	xsi:type	dct:W3CDTF
ScoutNSDL-847	http://purl.org/dc/elements/1.1/	date	1997	xsi:type	dct:W3CDTF
ScoutNSDL-840	http://purl.org/dc/elements/1.1/	date	1997	xsi:type	dct:W3CDTF
ScoutNSDL-860	http://purl.org/dc/elements/1.1/	date	1997	xsi:type	dct:W3CDTF
ScoutNSDL-818	http://purl.org/dc/elements/1.1/	date	1998	xsi:type	dct:W3CDTF
ScoutNSDL-828	http://purl.org/dc/elements/1.1/	date	1999	xsi:type	dct:W3CDTF
ScoutNSDL-833	http://purl.org/dc/elements/1.1/	date	1999	xsi:type	dct:W3CDTF
ScoutNSDL-83	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-820	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-838	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-839	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-856	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-858	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-859	http://purl.org/dc/elements/1.1/	date	2001	xsi:type	dct:W3CDTF
ScoutNSDL-829	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-832	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-850	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-817	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-831	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-841	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-844	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-851	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-854	http://purl.org/dc/elements/1.1/	date	2002	xsi:type	dct:W3CDTF
ScoutNSDL-849	http://purl.org/dc/elements/1.1/	date	[2002]		
ScoutNSDL-84	http://purl.org/dc/elements/1.1/	date	c1995 - 2001		
ScoutNSDL-85	http://purl.org/dc/elements/1.1/	date	c1995 - 2001		
ScoutNSDL-86	http://purl.org/dc/elements/1.1/	date	c1995 - 2001		
ScoutNSDL-819	http://purl.org/dc/elements/1.1/	date	c2000		
ScoutNSDL-816	http://purl.org/dc/elements/1.1/	date	c2001		
ScoutNSDL-82	http://purl.org/dc/elements/1.1/	date	c2002		
ScoutNSDL-835	http://purl.org/dc/elements/1.1/	date	c2002		

Figure 4. Spotfire table view for DC date field

Looking at values in a table presentation is a relatively easy way to check for conformance to a small controlled vocabulary or to a particular string pattern (such as strings beginning with "http://"). This approach makes possible a glance at *all* the values present in each field, a real improvement over random sampling techniques. Many anomalies in the data stand out in a table view, with relatively little time spent in the examination. Consistently (or inconsistently) applied defaults with typographic errors were often found by using this view. By quickly scrolling through the values for each field separately, we were better at finding bad values – they tended to stick out visually.

Some other graphical analysis techniques not illustrated here include plotting attribute names against element names to see which attributes have been applied to which elements. Using the table view to dynamically select all elements or a subset of one or more elements is another helpful feature.

Spotfire's full text search capability was often useful, particularly in locating HTML tags not valid in XML (e.g. "
" needs to be expressed as "
" in XML). Because table views only display the first few dozen characters of a value on a small screen, and because the HTML tags were often embedded in descriptive text, this was an important feature.

We also made use of the integration of data views in Spotfire – selecting a subset of the file in one view would select the same subset of the data in other Spotfire views. So if we selected records missing the identifier field in a scatter plot view, we could then switch to a table view with the same records pre-selected. Thus we could easily locate record identifiers or other information about records selected in the scatter plot. We also used XMLSpy in conjunction with Spotfire during evaluation, and would switch between Spotfire and XMLSpy to examine metadata in its native XML, which provided better context for individual records.

After the metadata files were evaluated, a simple specification was written by the evaluator and passed on to a programmer. The programmer then prepared XSL style sheets to transform the metadata. Common transforms included adding encoding schemes to fields (sometimes massaging the values slightly in the process), removing fields and values conveying no information, and adding missing information.

Visual graphical analysis allowed the efficient and thorough review of large quantities of XML metadata, and enabled the focus of our limited resources on the tasks that gave us the most payback. We were able to assess the submitted metadata as to its consistency and acceptability for our uses, and to specify necessary additions or improvements for our particular purposes. We could view data anomalies easily and determine whether what we called "standard transforms" would handle the problems they represented, or even whether they were worth bothering about. We spent less time testing and re-testing transform programs, because we had already seen all the quirks and gaps in the metadata. This was true for both the evaluator and for the programmer implementing transforms for metadata additions or improvements. The combination of scatter plot and table views allowed a view of both the structure of the data and its content, enabling us to move beyond the bottom line "does it validate?" question to assess instead how the data would behave in our portal, and how well it would function for our users.

Table 1. Some of the questions we posed using Spotfire

Question posed in Spotfire	**Errors discovered with this technique (examples)**
Which elements are present in the metadata file, and which namespaces are they in?	*Incorrect data:* the Audience element ascribed to the DC/1.1 namespace instead of the DC Terms namespace
Which attributes are present in the metadata file and which elements do they qualify?	*Incorrect data:* Encoding scheme IMT qualifying element refinement Medium instead of DC element Format
Are there any non-DC elements in the file?	*Incorrect data:* Local elements or refinements without valid namespaces found.
Which of the values of the "Type" element are actually valid DCMIType terms? Are there DCMIType values in other elements?	*Insufficient data:* Encoding scheme DCMIType should be applied to the DC Type element *Confusing data:* DC Format IMT values in DC Type fields designated as DCMIType, and vice versa
Which non-empty elements are present in the file?	*Missing data*
What identifier fields are present in this dataset? Which records have them, and what are their contents?	*Incorrect data*: double http://
Do all records have a title field?	*Missing data*

5. Conclusion

The use of data visualization software can significantly improve efficiency and thoroughness of metadata evaluation, both before and after transformation. The added predictability of transform input and results reduces the need for extensive testing, and allows the development of automated processes to proceed more quickly and with greater assurance. These evaluation techniques are potentially useful for evaluation of any type of metadata in any number of contexts.

Visual graphical analysis is most commonly used for statistical analysis. The NSDL may be able to take further advantage of these tools and techniques to evaluate information about the contents of the MR, the use of the MR, and similar information about other NSDL components, such as the Search service or the User Interface, in a manner that could be valuable to managers and funders as well as developers and designers.

In addition to the standard kinds of statistical analysis that might be useful for the NSDL, there are several non-traditional uses being considered as well.

As an aggregator, we necessarily must plan on managing data that is either continually or occasionally updated. Our harvest-transform-ingest model presumes that the difficulties seen with the data at first harvest will persist through any subsequent re-harvests, and that subsequent harvests will require the same transforms to enhance the metadata for NSDL purposes. Clearly this is an optimistic notion, already refuted by experience. If we could use the information Spotfire provides about the structure and content of the incoming metadata, and write software to compare previously harvested metadata with the present file, then significant changes in subsequent harvests from the same collection might be automatically detected.

Another possibility worth exploring is the use of Spotfire or similar tools to analyze automatically generated metadata in a similar fashion to what we already do with harvested metadata. Information collected via routine evaluation of automated metadata output could potentially assist in refining the metadata creation tool itself. Specific plans for this process do not yet exist, but given the complex statistical analysis performed with visual graphic analysis programs, we believe some experimentation along these lines would be fruitful.

Acknowledgements

This work was funded by the National Science Foundation under grant 0227648. The ideas in this paper are those of the authors and not of the National Science Foundation.

The authors would like to thank Anat Nidar-Levi for her help in preparing this paper for publication, Carol Terrizzi and Kizer Walker for their feedback, the NSDL Core Integration team, and all the people and organizations who have provided metadata to the NSDL.

References

[1] OAI Registered Data Providers, retrieved May 9, 2003, from http://www.openarchives.org/Register/BrowseSites.pl

[2] The Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0 of 2002-06-14, Document Version 2003/02/21T00:00:00Z, http://www.openarchives.org/OAI/openarchivesprotocol.html

[3] [Altova, Inc., XMLSpy Home Page], retrieved May 9, 2003, from http://www.altova.com/products_ide.html

[4] Dushay, Naomi, Stylesheets to Convert XML Metadata into .csv File, retrieved May 9, 2003, from http://www.cs.cornell.edu/naomi/DC2003/csv_xsl.htm

[5] [Spotfire DecisionSite Home Page], retrieved May 9, 2003, from http://www.spotfire.com/products/decision.asp

[6] Personal correspondence between Naomi Dushay and Lou Vitiritti (5/8/2002).