Application profiles: Exposing and enforcing metadata quality

Diane I. Hillmann
Cornell University
dih1@cornell.edu

Jon Phipps
Cornell University
jp298@cornell.edu

Abstract

In this paper, we explore a range of issues yet to be addressed in the large-scale use of application profiles. While considerable attention has been paid to human-readable application profiles, there is a growing need for machine-readable application profiles that can support quality control mechanisms including, but not limited to, data validation. We examine these issues in the context of the evolving Semantic Web and the DCMI commitment to RDF and the challenges presented. We frame the discussion in terms of select functions to be served by application profiles and our notion of data profiles. While much remains to be done to address these issues, positive movement toward solutions is dependent on the appropriate framing of those issues in terms of the needs of large-scale applications such as metadata aggregators.

Keywords: metadata quality; metadata validation; machine-readable application profiles.

1. Introduction

Since the groundbreaking article by Heery and Patel introducing the idea of "Application Profiles," considerable effort has been expended in discussing the specification of application profiles in the context of Dublin Core (Heery & Patel, 2000). Two important early documents sponsored by the European Committee on Standardization (CEN) set the stage for technical specification (CEN, 2003 & 2005). In this rapidly changing context, several communities bravely created extensive APs for their communities (DCMI Libraries Community, 2004; DCMI Collection Description Community, 2007).

But although considerable progress has been made in the area of technical specifications for APs in general, as well as the conventions for expressing human-readable APs as HTML documents, little discussion or experimentation has been expended on the considerable problem of machine-readable Application Profile implementation. As the early phase of work with APs draws to a close, this gap looms large. To a great extent we have been caught flat-footed: expressing the value of APs largely in the context of their human usable form, and assuming that they will evolve naturally to function as machine-readable information. However, the specific functions that might be addressed by machine-readable APs have remained vague and largely unexplored.

Perhaps Heery & Patel were more prescient than they knew, when they said in the introduction to their 2000 article:

> The experience of implementers is critical to effective metadata management, and this paper tries to look at the way the Dublin Core Metadata Element Set (and other metadata standards) are used in the real world. Our involvement within the DESIRE project reinforced what is common knowledge: implementers use standard metadata schemas in a pragmatic way. This is not new, to re-work Diane Hillmann's maxim 'there are no metadata police', implementers will bend and fit metadata schemas for their own purposes (Heery & Patel, 2000).

During the same period, several developments in the general digital library arena caused the idea of Application Profiles to be enthusiastically received. Early implementations of Dublin Core

were largely used within projects—data sharing was not yet easily accomplished. By 1999 early data sharing efforts based on cross repository searching were producing disappointing results:

> Digital library experience suggested that cross searching does not scale well, at least partly because the search service degrades to the level of the slowest and least reliable server in the cross search set. For example, NCSTRL found that distributed searching of a small number of nodes was viable, but that performance was very bad over 100 nodes. In the UK, the Resource Discovery Network (RDN) was finding that even with only five subject gateways in its cross search there were problems of poor performance and in the provision of a browse interface, and developers were looking for a feasible centralized database solution. The more servers are cross-searched, the higher are the chances of encountering one or more slow or unreliable servers (Open Archives Forum, n.d.).

After a short flirtation with a very small element set optimized for preprints, the newly launched Open Archives Initiative specified Simple Dublin Core as their minimal element set, and metadata harvesting was born. The OAI community, originally organized around the narrow goal of enabling better searching of scientific preprints, realized quickly that a broader focus on general resource sharing was politically attractive and practically within reach. During the same period, the Dublin Core community, responding to requests for richness beyond the initial 15 elements, approved the first group of qualifiers and, perhaps more importantly for this discussion, introduced "encoding schemes" as the method for enabling controlled vocabulary use within Dublin Core (DCMI, 2000).

Once large scale data sharing and aggregation became a reality with the birth of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), and Dublin Core grew beyond its simple origins, communities and implementers needed a way to express their intentions and expectations beyond the technical specifications of XML. Application Profiles, as described by Heery and Patel, did two things: they broke the perceived boundaries of established metadata schemas, and recognized that implementers both needed and demanded more flexibility to achieve their aims.

2. Metadata Quality Criteria and Measurement

Early conversations about metadata quality, particularly in the library community, were based on experience with MARC data distributed via bibliographic utilities. The library community was an early adopter of computer technology and data distribution standards, but as with many highly evolved, early adopting communities, found it increasingly difficult to accommodate the high rate of change as other metadata standards joined the fray. Bruce and Hillmann (2004), attempting to re-start the quality conversation to include newer models of metadata aggregation, defined seven criteria for determining metadata quality: completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness, and accessibility. Although the criteria provide opportunities to converse about quality, without ways to measure that quality, they remain frustratingly beyond reach. With the Application Profiles added to the mix as a template for expectation, we can begin to see the potential to compare actual data to that template and quantify the results.

One example where potential for quantifying exists is with the criteria of *completeness*. Without a notion of expectation it is difficult to determine how complete an individual metadata record or an aggregation of records might be. But if, for example, the Application Profile requires Title, Identifier and Description, and the metadata lacks any Descriptions, it is, by definition, incomplete. In this case the use of Obligation data from the Application Profile allows for a simple but quantifiable determination of completeness, though not necessarily weighted. Related to *completeness* is *conformance to expectations*, where the determination of quality results is a bit more complex to define. If an Application Profile includes descriptions of a *condition* when a value should be present, and that condition cannot be expressed programmatically, it would be difficult to determine by machine whether or not the metadata conforms in that regard to the

Application Profile, although presumably a human with adequate tools could determine conformity in some of those cases. Bruce and Hillmann provide some insight into the important differences between the two criteria:

> Element sets and application profiles should, in general, contain those elements that the community would reasonably expect to find. They should not contain false promises, i.e., elements that are not likely to be used because they are superfluous, irrelevant, or impossible to implement. Controlled vocabularies should be chosen with the needs of the intended audience in mind, and explicitly exposed to downstream users. Sometimes problems with conformance to expectations appear in disguise. Moen et al. correctly point out that problems with omitted metadata frequently occur because users see the particular element as irrelevant or unnecessary, so that what appears at first blush to be a completeness problem is in fact a problem with conformance to expectations.
>
> Finally, metadata choices need to reflect community thinking and expectations about necessary compromises in implementation. It is seldom possible for a metadata project to implement everything that anyone would want; most often, the metadata provider cannot afford to make a project unimpeachable by making it comprehensive. It is therefore important that community expectations be solicited, considered, and managed realistically. Better an agreed-upon compromise that is well executed and documented than an approach that aspires to be all things to all people and ends up poorly and unevenly implemented (Bruce & Hillmann, 2004).

Provenance is difficult to determine with most metadata unless there is a data wrapper (such as provided by OAI-PMH) which contains provenance information, and that information is maintained properly. Provenance is to some extent administrative in nature, and its presence and reliability depends on the policies of the data provider, and potentially a whole chain of data providers that may have touched the metadata in past transactions. At one level, the presence of provenance information is a good beginning point, but without better tracking of where metadata has been and how it has been modified (not really possible using the methods provided within OAI-PMH) there are significant limits to what can be assumed about the quality and integrity of data that has been shared widely.

The areas of *data accuracy* and *logical consistency and coherence* are perhaps the most subjective of the quality criteria, and very difficult to determine (much less measure) even in the best of circumstances. Some measurement of accuracy might be possible as an outcome of other processes: for instance, a high level of invalid vocabulary terms, when a vocabulary is specified and a schema available to the validator, might be interpreted as an accuracy problem. Simple validation of XML determines whether the data is "well-formed" and this is also an indication of accuracy in data. But where text strings are specified there are no real methods to measure the accuracy of the keying and many typographical errors may be introduced as part of the normal processes of creation and maintenance. Since *logical consistency and coherence* are being considered as one of the characteristics of the review process DCMI offers to Application Profile creators, one might want to assume that metadata conforming to a reviewed Application Profile was therefore consistent and coherent, but this would be a stretch. Particularly if a data creator practiced wholesale defaulting of missing data—called "promiscuous defaults" by Dushay and Hillmann (2003)—such an assumption would be false.

Currency of data as a function of *timeliness* is measurable when administrative data is available and the available policies of the provider give confidence that regular maintenance has taken place, but the question of how timely is timely enough emerges fairly quickly. Application Profiles do not generally define benchmarks for freshness of data, and in general the context of the acquisition and use of the data would determine the answer to the freshness question. Bruce and Hillmann also consider "lag" an aspect of timeliness, defined as the difference between the provision of the content and the metadata, or vice versa. *Lag* would only be possible to determine

in the second instance (when the metadata was available and the content was not), but it would be almost impossible to determine the difference between a lag and other technical problems.

Bruce and Hillmann (2004) include a number of *accessibility* issues for metadata:

> Metadata that cannot be read or understood by users has no value. The obstacles may be physical or they may be intellectual. Barriers to physical access come in several forms. Metadata may not be readily associated with the target objects, perhaps because it is physically separated, comes from a different source, or is not properly keyed or linked to the object being described. Or it may be unreadable for a wide variety of technical reasons, including the use of obsolete, unusual or proprietary file formats that can only be read with special equipment or software. In some cases, metadata is considered "premium" information that is accessible only at extra cost to the user, or proprietary information that is not released publicly at all, often because it represents a competitive advantage that the creator or publisher wishes to retain. In other words, the barriers may be economic or trade-related rather than technical or organizational.

Some common accessibility problems stem from content providers' understandable desire to track usage of their content. To do so, they either require individuals to register before viewing or using content, or they prevent direct URL access to content by routing access through a single portal. Other sites prevent search engine indexing of their content by using robots.txt barriers or allow only limited indexing designed to route potential users through bespoke search pages. These strategies, although they may seem to fulfill immediate goals for content providers, ultimately prevent effective aggregation of resource metadata, and should be considered quality problems. Some of these problems can be detected by machine, particularly if the same URL is used for all resources, or there is an intention to provide combined full-text indexes in combination with metadata, and effective access to indexing is prevented. Such issues are, however, not related to Application Profiles in any obvious way.

3. Functions of Application Profiles

An important function of Application Profiles is the development and documentation of community consensus. There were two communities that broke into Application Profiles early: the Open Language Archives Community (OLAC), and the DC Libraries Working Group (DC Libraries). The OLAC community created an application profile around their early adoption of OAI-PMH, without the specific intention to create an AP. Their main goal was to build tools to enable the smaller, less technically supported language archives to participate fully in community data sharing activities, and they realized early that creating a community consensus around elements and vocabularies was vital to their task (Bird & Simons, 2004). The DC Libraries community was struggling with the disconnect between Dublin Core and USMARC, and attempting to go beyond crosswalks, recognizing that much of their future revolved around digital materials that would not be cataloged using USMARC. In both cases, the communities used a community participation process to achieve agreement around data usage and expectations. In the absence of real technical underpinnings around Application Profiles in the first few years of discussion, the "movement" towards APs grew almost entirely based on the desire and need for communities and projects to come to formal agreements about expectations.

An additional function grows from the needs of communities to establish practical guidelines for the creation of metadata intended for sharing. These guidelines generally attempt to provide a basis for decisions about usage of elements, data normalization, and use of controlled vocabularies. Application profiles in this context are to a certain extent a precursor, or template, for guidance documentation. In communities such as OLAC, the decisions in their AP were specifically related to their need for specific tools required for data sharing amongst community members, and indeed, the consensus building aspects of APs function well as a basis for tool building.

4. Expectations for Machine Readable Application Profiles

Clearly, the effort of building Application Profiles is justified even if they function only as the basis for human-readable documentation. But it is the promise of machine readability, built upon this human-created foundation, which begins to address the real potential of Application Profiles to improve and support metadata quality. The challenges for this transition from a primarily human-readable focus to the fuller functionality of human- and machine-readable Application Profiles will require the addition of supporting technology, including functioning registries that can help manage AP expression in the form of multiple data validation engines and templates, as well as manage the ongoing change cycle and distribution of technical information for a variety of purposes.

Machine-readable expressions of the more generally available human-readable guidelines expressed by an AP are complicated by the fact that APs must often define expectations for data that may be generated in the form of both XML and RDF. XML and RDF require very different notions of data validation. In fact, the notion that a single RDF statement, or a set of statements expressed in a description set, is or is not valid in the syntactic sense of XML or general data validation does not apply to RDF at all.

RDF "validation" does not extend beyond the fundamental "well-formed" semantic constraints of RDF itself. Even when enhanced processing and inference rules are supplied in the form of RDFS and OWL, RDF data cannot be "schema-validated" in the sense that most data providers and data consumers think of when they think of valid data as being syntactically correct (W3C Semantic Web Best Practices and Deployment Working Group, 2004).

Further complicating the issue for data consumers seeking RDF data validation is RDF's reliance on the assumption of an "Open World" of data in which data that is not present in a description set can not actually be considered to be missing, but simply isn't able to be dereferenced or retrieved at the present time. "Missing" data may also not be considered missing if its existence can be inferred based on conditions provided by an OWL or RDFS definition. These assumptions are highly useful when dealing with data that is intended to live and be accessible in the wide and deep ocean of the Open World of the Semantic Web, but can be less useful when trying to support interoperability among more terrestrial data-driven applications.

4.1. Data Validation

> RDF is a strategy for *principled decentralization* in a world where unanticipated data re-use, [and] unanticipated data extensions, are valued (Brickley, 2005).

There are many potential approaches to at least partially solving some of the challenges posed by RDF data validation. One promising approach involves an implementation of the rules-based approach embodied by Schematron, described in "An XML structure validation language using patterns in trees" ("Schematron," 2007). Based on the earlier work "Schemarama" by Dan Brickley, Leigh Dodds and Libby Miller, "Schemarama 2" (2005) uses a set of validation rules expressed as SPARQL CONSTRUCT queries that are intended to indicate the presence (or absence) of data in a description set. Failure of any of the queries indicates that the data is not "valid" in the sense that it doesn't return valid results from the query rules.

> [S]ome constraints are difficult or impossible to model using regular grammars. Commonly cited examples are co-occurrence constraints (if an element has attribute A, it must also have attribute B) and context sensitive content models (if an element has a parent X, then it must have an attribute Y) (Dobbs, 2001).

XML presents a different set of validation challenges. Like RDF, metadata encoded in XML can be parsed by a machine to confirm that it is "well-formed," and with an W3C XML schema or DTD and a grammar-based validating parser the XML data can be syntactically validated—checked for conformance to a pre-determined structure and the presence or absence of particular encoding features. Rules-based validators, such as Schematron, offer somewhat greater potential

to validate XML data based on predefined constraints, as in the Schemarama approach to RDF cited above.

For metadata aggregators, the challenges are considerable, even when metadata is created and aggregated in a community context in which all members of the community agree to play by an agreed-upon set of rules and expectations as expressed in an Application Profile. Even when a metadata provider agrees to produce AP-conforming metadata for subsequent harvesting, the provider will often have local requirements and constraints, or even a local AP, dictating a different description set than the one that will ultimately be shared with the community. Local metadata formats may differ from the ultimate format of the shared metadata as well. Each metadata provider therefore must ideally validate their data in the context of their own local domain and then should also validate their data against the community AP before distributing it.

As statements are aggregated and shared within the community, prudence requires that each consumer of the aggregated metadata validate the incoming data against the community AP before it is crosswalked as necessary into local data structures. In the context of the rules and conventions for metadata validation expressed in a community AP, this usually means applying the limited grammar-based approach to validation of XML or the even more limited notion of valid RDF.

What these validation rules and conventions cannot do is validate data content. With an Application Profile, it should be possible to validate much of the content of a record as well.

5. Application Profiles

Application Profile structure has been developed extensively within the Dublin Core community (Baker et al., 2005). The structure is based on the inheritance of some of the element attributes from the metadata schema, the addition of basic usage information, and the association of specific vocabularies to encode the values themselves or from which to choose values. The Guidelines (currently under revision) divide the attributes into groupings according to their function: Identifying, Definitional, Relational, and Constraints.

Thus, *Date* may be associated with the W3CDTF encoding rules, but *Subject* may require the use of a specific controlled vocabulary in the context of a particular Application Profile.

The DC Properties are being updated to include Domains and Ranges for each of the DC properties (Powell, 2007). As currently stated in the Dublin Core Abstract Model (DCAM) and in other DC documentation, property domains and ranges are most useful in defining constraints in the context of RDF-based metadata. In its simplest form the DCAM/RDF notion of Domain says that when a statement is asserted that contains a predicate defined by DCMI as 'date' (http://purl.org/dc/terms/date), it can be inferred (but not required) that the subject of the statement is a member of the class 'resource' (http://www.w3.org/2000/01/rdf-schema#Resource). The DCAM/RDF notion of Range for that predicate says that the object of the statement may be inferred (but not required) to be a member of the class 'period' (http://example.org/dc/terms/Period).

Application Profiles may modify or override these constraints as necessary. As discussed earlier, validation of 'correct' application of Domain and Range presents some challenges for RDF data validation, and is difficult at best to express these parameters in ways that can be validated by typical grammar-based validating parsers.

Thus, the best potential for content validation occurs in the Relational Attributes and Constraints. Relational attributes at present consist of:

a) Refines: The described term semantically refines the referenced term
b) Refined by: The described term is semantically refined by the referenced term
c) Encoding Scheme For: The described term, an Encoding Scheme, qualities the referenced term

d) Has Encoding Scheme: The described term is qualified by the referenced Encoding Scheme

e) Similar To: The described term has a meaning the same as, or similar to, that of the referenced term.

The relationship between the property and an encoding scheme, expressed reciprocally in c) and d), would allow validation of the presence of appropriate encoding schemes, and, particularly in the case of controlled vocabularies available in a form usable by the validator, the vocabulary terms themselves. This should be possible whether or not the terms exist in the instance metadata as text strings or as URIs. When the encoding scheme is a syntax encoding scheme, the validator should ideally be able to determine whether the information is expressed properly. For instance, a date expressed as "January 13, 2005" should be detectable by a validator as not conforming with W3CDTF, but the validator may not be able to detect that a date expressed as YYYYDDMM is incorrect, unless the MM portion is more than 12 or the DD more than 31.

Property URI	**http://purl.org/dc/elements/1.1/rights**
Qualified Name for Property	dc:rights
Defined By	The Dublin Core Metadata Element Set, v1.1 http://purl.org/dc/elements/1.1/
Type of Term	Property
Subproperty Of	[n/a]
Source Label	Rights
Label in this DCAP	[n/a]
Source Definition	Information about rights held in and over the resource.
Usage in this DCAP	A statement of any rights held in/over the collection.
Comments for this DCAP	Either a value string or a value URI or both may be provided. To record information in multiple languages, use a single statement with multiple value strings and associate each value string with the appropriate language tag.
Uses Vocabulary Encoding Scheme	[Not specified]

Value URI	**Value String**	**Syntax Encoding Scheme(s)**	**Rich Represent**
Optional	Optional		Not permitted

Obligation	Optional
Condition	[n/a]
Minimum Occurrences	0
Maximum Occurrences	unbounded

FIG 1. A human readable fragment from the DC Collections Application Profile, showing some of the more granular expressions of intention used in newer APs.

The Constraints area of the AP provides perhaps the most critical (and most complex) area of potential use in validation. In the current CEN documentation, constraints are expressed as four parameters:

a) Obligation: whether the element is required to be always or sometimes present. Examples of values include "mandatory," "conditional," and "optional."

b) Condition: describes the condition or conditions when a value should be present

c) Datatype: indicates the type of data expected as the value of the element

d) Occurrence: indicates limits to repeatability of the element

More recent Application Profiles, based on the Abstract Model, are providing more detailed expectations for encoding scheme values, specifically whether value strings or value URIs are allowed or required.

6. Data Profiles: A Tool for Evaluation and Measurement of Metadata Quality

Validation of data using Application Profiles needs to be viewed in the context of an overall metadata management strategy. Particularly in an OAI-PMH-based aggregation environment, where data is harvested from various sources and re-exposed for harvest by others, a strategy that uses machine processing as much as possible is essential. Phipps, Hillmann and Paynter (2005) envision an environment where data harvest and management is largely automated with a scheduling function available to manage most of the work. Critical to this kind of effort, particularly when automated transformation is part of the picture, is to be able to create and maintain a "footprint" or "data profile" of what is actually harvested, so that changes in the data initiated by the data provider can be recognized and re-evaluation of the transformation routines initiated.

To a great extent, a data profile is the beginning step as well for determining whether the data matches an Application Profile, particularly when the data provider asserts that the data was built to conform to a specific Application Profile. In order for the data profile to work as a precursor step, it must record the following:

1. Properties used for all records within a data set
2. Properties used only for some records within the data set, and the characteristics of that subset as distinguished from the larger set
3. Datatypes and encoding schemes used in the context of each property used
4. Validation rates of URIs and text strings used as values when encoding schemes are declared
5. Earliest and latest dates of creation and updating within the set.

7. Sample Use Case for Making Quality Assertions Based on Machine Evaluation of Metadata

Metadata is harvested from open repository, with an OAI About container specifying that the Collections Application Profile was used to create the data (DCMI Collection Description Community, 2007). The data is examined using Spotfire (n.d.), a data visualization tool. It is determined that:

1. All records include the following properties: dc:title, dc:type
2. Only 102 of the 245 records in the set include mandatory dcterms:abstract
3. All Type properties include the mandatory value "Collection" as a text string, not a URI (which is mandatory)
4. Only 102 of the 245 records in the set include the optional/recommended dc:identifier (these are the same subset as those that include dcterms:abstract)
5. Only 200 of the records contain dcterms:accessRights
6. Only 120 of the records contain either dc:creator or marcrel:OWN
7. The same 120 records also contain cld:isLocatedAt
8. A total of 15 records contain dc:publisher, not valid in this AP (no overlap with cld:isLocatedAt)
9. Only 45 records contain cld:isAccessedVia, there is no overlap with cld:isLocatedAt.

A data profile is created for this set, recording the data as it is harvested. The evaluation of the data confirms that this group of records contains a number of issues in terms of its conformance

with the stated Application Profile. It should be possible to proceed from the analysis to determine a rating for each record based on:

a) Presence or absence of mandatory properties
b) Use of text strings as a value for dc:type instead of a URI
c) Absence of recommended values (dc:identifier)
d) Presence of properties not valid in this AP (dc:publisher)
e) Completeness of records in terms of supplying sufficient information for a user to determine how to access the collection.

How such a rating is determined and expressed when the records are redistributed is an open issue, as is the question of whether and how downstream users will use the ratings. Ratings may potentially be used to present some records lower in rankings, or to flag them for specific services (improvement or enhancement, either machine- or human-based).

8. Conclusion: Moving Towards Full Machine Validation, Evaluation and Reporting

As Heery and Patel noted in 2000, implementation and experience are the teachers that best move metadata management techniques forward. To a great extent, the most important initial value of APs for implementers has been as a focus for community consensus and as a spur to discussion of metadata quality. But a machine-assisted way forward requires better rates of registration of the component parts of Application Profiles (metadata schemas and controlled vocabularies) as well as registration and change management for APs themselves. How this infrastructure will be built, sustained and extended is perhaps the most pressing question for implementers, and the lack of good answers the biggest impediment to true progress.

References

Baker, Thomas, Makx Dekkers, Thomas Fischer, and Rachel Heery. (2005). *Dublin Core application profile guidelines*. Retrieved April 21, 2007, from http://dublincore.org/usage/documents/profile-guidelines/.

Bird, Steven, and Gary Simons. (2004). Building an Open Language Archives Community on the DC foundation. In D. I. Hillmann and E.L. Westbrooks (Eds.), *Metadata in practice* (pp. 203-222). Chicago: ALA Editions.

Brickley, Dan. (2005). *CheckRDFSyntax and Schemarama revisited*. Retrieved April 18, 2007, from http://danbri.org/words/2005/07/30/114.

Bruce, Thomas R., and Diane I. Hillmann. (2004). The continuum of metadata quality: Defining, expressing, exploiting. In D. I. Hillmann and E.L. Westbrooks (Eds.), *Metadata in practice* (pp. 238-256). Chicago: ALA Editions.

CEN - European Committee for Standardization. (2003). *ECWA14855 - Dublin Core application profile guidelines*. Retrieved April 21, 2007, from http://www.cenorm.be/cenorm/businessdomains/businessdomains/isss/cwa/cwa14855.asp.

CEN - European Committee for Standardization. (2005). *CWA 15249 - Guidance for naming, versioning, evolution and maintenance of element declarations and application profiles*. Retrieved April 21, 2007, from http://www.cenorm.be/cenorm/businessdomains/businessdomains/isss/cwa/cwa15249.asp.

DCMI Collection Description Community. (2007). *Dublin Core Collections Application Profile summary*. Retrieved April 21, 2007, from: http://dublincore.org/groups/collections/collection-ap-summary/2007-03-09/.

DCMI Libraries Community. (2004). *Library Application Profile*. Retrieved April 21, 2007 from http://dublincore.org/documents/2004/09/10/library-application-profile.

DCMI. (2000). *Dublin Core Qualifiers*. Retrieved April 21, 2007 from http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/.

Dobbs, Lee. (2001). *Schemarama*. *XML.Com*. Retrieved April 20, 2007, from http://www.xml.com/lpt/a/727.

Dushay, Naomi, and Diane I. Hillmann. (2003). Analyzing metadata for effective use and re-use. *Proceedings of the International Conference on Dublin Core and Metadata Applications, Seattle*. Retrieved April 21, 2007, from http://www.siderean.com/dc2003/501_Paper24.pdf.

Heery, Rachel, and Manjula Patel. (2000). Application profiles: Mixing and matching metadata schemas. *Ariadne, 25*. Retrieved April 21, 2007, from http://www.ariadne.ac.uk/issue25/app-profiles/.

Open Archives Forum. (n.d.). *OAI for beginners, the Open Archives Forum online tutorial. 2. History and development of OAI-PMH*. Retrieved April 21, 2007, from http://www.oaforum.org/tutorial/english/page2.htm.

Phipps, Jon, Diane I. Hillmann, and Gordon Paynter. (2005). Orchestrating metadata enhancement services: Introducing Lenny. *Proceedings of the International Conference on Dublin Core and Metadata Applications, Spain*. Retrieved April 21, 2005, from http://arxiv.org/abs/cs.DL/0501083.

Powell, Andy. (2007). *Domains and ranges for DCMI Properties*. Retrieved April 21, 2007, from http://dublincore.org/documents/2007/02/05/domain-range/.

Schemarama 2: Testing Framework for RDF. (2005). Retrieved April 20, 2007, from http://isegserv.itd.rl.ac.uk/schemarama/.

Schematron: An XML structure validation language using patterns in trees. (2007). Retrieved July 4, 2007 from http://xml.coverpages.org/schematron.html.

Spotfire, Inc. (n.d.) *DecisionSite product suite*. Retrieved July 4, 2007, from http://www.spotfire.com/products/decisionsite.cfm.

W3C Semantic Web Best Practices and Deployment Working Group. (2004). FAQ: Using RDFS or OWL as a schema language for validating RDF. In *SWAD-Europe Weblog*. Retrieved April 20, 2007, from http://esw.w3.org/mt/esw/archives/000048.html.