David Ruddy

Developing a Metadata Exchange Format for Mathematical Literature

In: Petr Sojka (ed.): Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010. Masaryk University Press, Brno, Czech Republic, 2010. pp. 27--36.

Persistent URL: http://dml.cz/dmlcz/702570

Developing a Metadata Exchange Format for Mathematical Literature

David Ruddy

Project Euclid, Cornell University Library
107 D Olin Library, Ithaca, New York, 14853, USA
dwr4@cornell.edu

Abstract. This paper describes an effort to develop a metadata element set for the exchange of descriptive metadata about mathematical literature. The approach taken uses the Dublin Core Application Profile (DCAP) framework, based on the DC Abstract Model. A fully developed DCAP for mathematical literature would be valuable, as both a guide and constraint in the creation of metadata records suitable for harvesting via OAI or sharing through other means. Adhering to the DCAP model would also enhance global interoperability with other metadata schemes. The successful development of a DCAP for mathematical literature, however, will require broader DML community input to resolve open issues and gain acceptance.

Key words: metadata standards, metadata exchange, Dublin Core Application Profile

1 Introduction

In order for repositories to share rich metadata about mathematical publications, the Digital Mathematics Library (DML) community will need to reach agreement on a metadata exchange standard. Currently, the only exchange format used in common across multiple repositories is simple, unqualified Dublin Core (DC), the 15 descriptive metadata terms originally designed in 1995 [1]. One reason for this is that simple DC is the default, minimal record format required by OAI-PMH, a frequently used mechanism for sharing metadata [2].

Simple DC appears easy to use, and it is almost universally recognized. It has, however, a number of disadvantages, most of which are related to its perceived strength: simplicity. As a carrier of descriptive information, it is very constrained. The usefulness of aggregating even high-quality simple DC records is therefore debatable, as the range of functionality that can be supported is so limited. And yet a more pressing problem may be the difficulty of obtaining high-quality, conformant, and consistent metadata when harvesting simple DC records from numerous independent repositories. Since it was designed to be applicable to such a wide range of materials, it is not well-suited for most particular content types. Those using it for specific purposes are therefore

Petr Sojka (editor): DML 2010, Towards a Digital Mathematics Library, pp. 27–36.
 ISBN 978-80-210-5242-0

tempted to embed qualifications in ways that simple DC was not designed for, or to otherwise use elements in ways that strain the original element definitions. The resulting impact on metadata quality and consistency and the consequent challenges in building reliable services on top of such aggregated data have been described [3,4,5].

While obtaining quality metadata from independent repositories will always present challenges, we argue that a positive step forward would be the creation of an element set that was both richer and more rigorous than simple DC and that was designed specifically for mathematical literature. Such a metadata set would give content repositories a full set of well-defined elements, so that they would not need to overburden terms or guess where to put descriptive data. This would likely improve metadata quality and consistency. At the same time, a richer element set would support greater functionality once metadata was harvested and aggregated.

2 Dublin Core Application Profiles

The Dublin Core Metadata Initiative has provided a framework for the design and documentation of metadata applications. This effort recognizes both the unique needs of particular communities and the benefits of a shared approach. A Dublin Core Application Profile (DCAP) is a combination of precise element definitions and usage guidelines. A DCAP is not limited to DC elements—it can use terms defined in other namespaces. The major constraint on the design of a DCAP is that it adhere to the Dublin Core Abstract Model [6]. The semantics of this model are built on the Resource Description Framework (RDF). Among other things, this requires that referenced properties (terms, elements), syntax schemes, and vocabularies all be properly declared in an RDF schema and thus identifiable with URIs. While this is not an insignificant constraint, the potential benefits of using globally defined properties and vocabularies are precision and semantic interoperability. This has important consequences for the usefulness of metadata in Semantic Web or Linked Data applications [7].

If an acceptable DCAP can be developed, it would likely provide for the widest usefulness in a global context. But even if the DML community eventually decides to take another approach, a thorough exploration of the DCAP framework and its requirements will be valuable. The requirements of a DCAP are useful to consider for any community metadata effort. There are also other DCAP projects that can serve as models and provide useful properties and encoding schemes, such as the Scholarly Works Application Profile (formerly, Eprints Application Profile), and the DC Collections Application Profile [8,9]. Both of these profiles have been used in the present effort.

Compliance with the DCAP framework is well-defined [10,11,12]. The following section briefly describes the necessary components of a DCAP and proposes a response for how a Mathematical Literature Application Profile (MLAP) could meet these requirements. This work is built on an earlier effort to create recommendations for using simple DC for mathematical literature,

begun in 2005 [13]. Participants in that effort were Thierry Bouche (Institut Fourier & Cellule MathDoc), Thomas Fischer (Staats- und Universitätsbibliothek, Göttingen), Claude Goutorbe (Cellule MathDoc), and David Ruddy (Project Euclid). In particular, many of the usage recommendations concerning content values are derived from that work.

3 Mathematical Literature Application Profile

3.1 Functional Requirements

Metadata does not exist for its own sake but to support desired functionality. It is important, therefore, both initially and as the profile develops, to understand clearly how we intend to use the metadata governed by this application profile. Establishing use cases and functional requirements is a process of community negotiation and agreement. It is through these discussions that a shared sense of functional scope is established, which will then provide rationale and guidance for the design of particular metadata constructs. For these reasons, the DCMI requires that a DCAP include functional requirements.

Two broad functional objectives of the proposed MLAP can be described. One is to provide a mechanism for the exchange of richer and more consistent metadata among repositories of mathematical literature than is currently possible with simple Dublin Core. This will contribute to the development of a "world digital mathematics library" by providing the means by which repositories can share more complete and uniform metadata about their holdings, and service providers can build more reliable services on top of that aggregated metadata. Another objective, achieved by using the DCAP approach, is to position MLAP metadata so that it can participate in a global semantic environment, envisioned by the Semantic Web and Linked Data movements.

More specific functionality that the MLAP should support includes:

- the discovery of publications:
  - ◇ by means of fielded searching on various attributes, including titles, author names, subjects, and abstracts.
  - ◇ by means of browsing, beginning at a journal, book, or other high-level publication title.
  - ◇ by means of filtering search and/or browse results based on attributes such as publication type, date of publication, language, access restrictions, parent publication, etc.
- the identification of publications of interest, from among many, by allowing for the collection and display of identifying attributes such as a DOI or other unique identifier, title, author, and publication details (date of publication, publication name, publisher, etc.).
- the selection of publications of interest, from among many, by allowing for the collection and display of attributes such as subject, format, publication type, language, and restrictions on access.

- the acquisition of a copy of the publication by providing a DOI or other network resolvable URI, together with information about access restrictions.
- the capture, display, and indexing of titles and abstracts in multiple languages or transliterations.
- potential additional capabilities or services, such as links to name authority resources, citation analysis, OpenURL linking, and rich subject analysis.

As currently proposed, the MLAP is for the description of network-accessible, published literature in mathematics and statistics. Although this profile could be used for author copies and pre-prints, it is optimized for formally published literature. Adequate description of pre-prints would require additional properties to describe document versions and to record a greater range of date attributes. Other functionality that is currently out of scope includes:

- the description of publications that do not have copies available online.
- the identification and description of distinct FRBR entities [14] (such as handled by the SWAP application profile [8].)
- the capture of structured author and contributor descriptions, so as to include role, affiliation, email address, etc.
- the capture of machine-processable descriptions of access embargo periods.

3.2 Domain Model

A DCAP domain model is a representation of the distinct entities that will be described by the metadata application and the relationships among those entities. It defines the overall scope of the application profile. Either graphic depictions or text descriptions can be used. The entity model for the proposed MLAP is relatively simple. There are only two entities: **publication** and **publicationContainer**, with a single relationship:

publication *may be part of a single* **publicationContainer**

Defining a **publicationContainer** allows us to capture an unambiguous and easily accessible description of the parent publication, such as a journal issue or monograph. A potential additional entity is "author" or "creator," but we feel that the MLAP is not the appropriate place to maintain rich author descriptions, such as affiliation, email address, etc. The MLAP allows for the use of a URI in the creator property, and we anticipate linking to more detailed author descriptions (or better, using an authoritative name identifier), rather than capturing that information internally.

3.3 Description Set Profile

The DCAP Description Set Profile (DSP) provides a detailed definition of the application's metadata record. The DSP is based on the DCMI Description Set Model, which is part of the DC Abstract Model. The DSP is expressed

by means of *templates* and *constraints*, the use of which is defined by a DSP constraint language [15]. The repeatability of properties and the restrictions on allowed property values are all explicitly defined by the DSP. Adherence to the constraints defined in the DSP determines the validity of all metadata records of a particular application profile. In essence, the DSP is the definition of the DCAP.

An XML expression of the complete DSP for a proposed MLAP is maintained online [16]. The root level DescriptionSetTemplate contains two child DescriptionTemplate elements (**publication** and **publicationContainer**), which represent the two entities of the domain model. Each of these in turn contain a number of StatementTemplate elements, which make property=value assertions about the entities. The various constraints upon the value of a particular property are expressed within the StatementTemplate.

A much simplified presentation of information contained in the DSP is provided in tabular form in the Appendix. The namespaces used for properties and encoding schemes in the MPAP are found in Table 1.

Table 1. MLAP Namespaces and Namespace URIs

Properties

DCMI Metadata Terms	`http://purl.org/dc/terms/`
DC Collections Metadata Terms	`http://purl.org/cld/terms/`
PRISM: Publishing Requirements for Industry Standard Metadata	`http://prismstandard.org/namespaces/basic/2.0/`

Syntax encoding schemes

DCMI Metadata Terms	`http://purl.org/dc/terms/`
NISO OpenURL Framework Registry	`info:ofi/`

Vocabulary encoding schemes

DCMI Metadata Terms	`http://purl.org/dc/terms/`
Eprints Terms	`http://purl.org/eprint/terms/`

3.4 Usage Guidelines

While usage guidelines are not explicitly required by the DCAP framework, they are rather critical for the successful use of the application profile. Guidelines translate the DSP into a human-readable format, as well as provide rules that apply to content values. For example, guidelines would include an original property definition (e.g., "An entity primarily responsible for making the resource"), any local use that refines that definition (e.g., "An author of the

publication"), whether the element is optional and repeatable, whether and how the element values are restricted or datatyped, and any "cataloging rules" that should be applied to value strings (e.g., "family name, followed by comma, then space, followed by given name"). Other less prescriptive recommendations may also be included.

A complete usage guideline for the currently proposed MLAP is maintained online [17].

3.5 Syntax Guidelines

The DCAP framework is neutral regarding the encoding syntax used to express and transmit metadata records. DCAP conformant metadata will by definition adhere to the DC Description Set Model defined in the DC Abstract Model. DCMI has provided several publications that specify how to serialize a DC metadata *description set* in plain text, XML, RDF, and HTML/XHTML [18,19,20,21].

At this point, no recommendations are made regarding the syntactic expression of MLAP metadata. Examples found within the usage guidelines are expressed in plain text. In time, these will be linked to other potential serializations, which can serve as encoding models.

4 Design Considerations, Open Issues, Next Steps

Developing a metadata scheme requires balancing richness and complexity against simplicity and ease of application. If it is too simple, the resulting description may not support desired functionality, but if it is too complex, few will apply it accurately or use it at all. Attempting to achieve an optimal balance has influenced several design considerations in the present effort. For example, the proposed MLAP has relatively few required elements. Valid metadata records can include only a title, a publication date, a bibliographic citation, and a URL to the online resource. Of these, only the publication date has an enforced encoding syntax. At one extreme, therefore, the profile provides a relatively low-barrier means of sharing metadata. At the other end of the spectrum, data providers can construct very rich metadata records by including multilingual values, MathML in titles and abstracts, complete reference lists, and OpenURL Context Objects containing machine-processable bibliographic data (describing the primary resource as well as references).

Another design choice was to use several distinct and dedicated identifier properties rather than a single multi-use one. The DC identifier element could have been used to capture the identifiers prism:url, prism:issn, prism:eIssn, and prism:isbn. (It could also be used to capture an HTTP addressable version of prism:doi.) It was felt, however, that providing dedicated elements will reduce uncertainty and ambiguity in the preparation and interpretation of MLAP metadata. Following the same reasoning, it seemed advantageous to create a distinct and easily interpretable entity, publicationContainer, to hold a description of

the parent publication. The same descriptive data could be packaged within an OpenURL Context Object in the dcterms:bibliographicCitation element. Such a construct, however, is fairly complex, and we believe that to require this approach would place an unnecessary and in some cases insurmountable burden on data providers and harvesters.

An acknowledged weakness of the proposed application profile is the handling of publications at the monographic level. At several points, the MLAP is currently optimized for serial literature. There are a number of solutions to this problem, if in fact it is perceived as a problem, but they are all in the direction of increased complexity. For example, as constructed, it is not possible to capture a role attribute with the contributor element (such as "editor," or "translator"). Allowing for this would require that *contributor* become a distinct entity in the data model so that properties could be associated directly with it. This leads to an additional level of complexity, and whether it is desirable to go in this direction is an open question.

There are a number of other open issues in the proposed MLAP. These are noted in the complete usage guidelines. Next steps include obtaining input and discussion from the broader DML community regarding proposals made here. We hope that such feedback will help resolve open issues and allow for refinement of the MLAP. Once an acceptable profile can be agreed upon, working implementations can test the MLAP further.

References

1. Dublin Core Metadata Element Set, Version 1.1. http://www.dublincore.org/documents/dces/
2. Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). http://www.openarchives.org/pmh
3. Arms, W., Dushay, N., Fulker, D., Lagoze, C.: A case study in metadata harvesting: the NSDL. Library Hi Tech 21, no. 2, 228–237 (2003). doi:10.1108/07378830310479866
4. Lagoze, C., Krafft, D., Cornwell, T., Dushay, N., Eckstrom, D., Saylor, J.: Metadata aggregation and "automated digital libraries": a retrospective on the NSDL experience. Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries 230–239 (2006). doi:10.1145/1141753.1141804
5. Bruce, T., Hillmann, D.; The Continuum of Metadata Quality: Defining, Expressing, Exploiting. In: Hillmann, D., Westbrooks, E., (eds.), Metadata in Practice, pp. 238–256. ALA, Chicago (2004).
6. Dublin Core Metadata Initiative Abstract Model. http://www.dublincore.org/documents/abstract-model/
7. W3C Semantic Web. http://www.w3.org/standards/semanticweb/
8. SWAP: Scholarly Works Application Profile. http://www.ukoln.ac.uk/repositories/digirep/index/SWAP
9. Dublin Core Collections Application Profile. http://dublincore.org/groups/collections/collection-application-profile/
10. The Singapore Framework for Dublin Core Application Profiles. http://www.dublincore.org/documents/singapore-framework/

11. Guidelines for Dublin Core Application Profiles. http://www.dublincore.org/documents/profile-guidelines/
12. Criteria for the Review of Application Profiles. http://www.dublincore.org/documents/profile-review-criteria/
13. Digital Math Library Dublin Core (dml_dc): A Recommended Best Practice for Unqualified Dublin Core Metadata Records. http://projecteuclid.org/documents/metadata/dml_dc/
14. Functional Requirements for Bibliographic Records: Final Report. IFLA Study Group on the Functional Requirements for Bibliographic Records. (UBCIM Publications, New Series; v. 19). München: K.G. Saur, (1998). http://www.ifla.org/VII/s13/frbr/frbr.htm
15. Description Set Profiles: A constraint language for Dublin Core Application Profiles (currently a Working Draft). http://www.dublincore.org/documents/dc-dsp/
16. Mathematical Literature Application Profile: Description Set Profile. http://projecteuclid.org/documents/metadata/mlap/mlap_dsp.xml
17. Mathematical Literature Application Profile: Property Definitions and Guidelines. http://projecteuclid.org/documents/metadata/mlap/
18. Expressing Dublin Core metadata using the DC-Text format. http://www.dublincore.org/documents/dc-text/
19. Expressing Dublin Core Description Sets using XML (DC-DS-XML). http://www.dublincore.org/documents/dc-ds-xml/
20. Expressing Dublin Core metadata using the Resource Description Framework (RDF). http://www.dublincore.org/documents/dc-rdf/
21. Expressing Dublin Core metadata using HTML/XHTML meta and link elements. http://www.dublincore.org/documents/dc-html/

Appendix: Properties of the MLAP

The following table lists the properties of the proposed Mathematical Literature Application Profile (MLAP). A complete specification for the MLAP is provided online in an XML expression of the Description Set Profile (DSP) [16].

URIs for the namespace abbreviations included in the table are as follows (see Table 1 on page 31for more information):

dcterms	http://purl.org/dc/terms/
cld	http://purl.org/cld/terms/
prism	http://prismstandard.org/namespaces/basic/2.0/

publication properties				
Property	Namespace	Min	Max	Value Constraints
type	dcterms	0	1	Value must be a URI; recommended practice is to use a value from the Eprints Type Vocabulary Encoding Scheme.

Property	Namespace	Min	Max	Value Constraints
title	dcterms	1	1	A single, primary title is required. Language attribute may be provided. Value may include XML content.
alternative	dcterms	0	1	Additional titles for the same publication (variants, translations, transliterations). Multiple value strings (titles) may be included; language attributes are required on each. Value strings may include XML content.
creator	dcterms	0	∞	If used, a value string is required; a value URI may be provided.
contributor	dcterms	0	∞	If used, a value string is required; a value URI may be provided.
abstract	dcterms	0	1	Multiple value strings (abstracts) may be included; language attributes are required on each. Value strings may include XML content.
subject	dcterms	0	∞	If used, a value string is required; it may be from a controlled vocabulary. A value URI may also be provided.
issued	dcterms	1	1	Date of publication is required; value must adhere to W3CDTF syntax.
language	dcterms	0	∞	Language or languages of the publication; values must be taken from RFC4646.
format	dcterms	0	∞	Format (Internet media type) of electronic file; values must be from the IMT vocabulary.
bibliographic-Citation	dcterms	1	1	A description of the bibliographic source of the publication is required. Value may be a text string, an OpenURL Context Object, or both.
startingPage	prism	0	1	The first page of the publication.
endingPage	prism	0	1	The last page of the publication.
doi	prism	0	1	A DOI for the publication.
url	prism	1	1	A URI that resolves to a publication record page is required.
identifier	dcterms	0	∞	Additional identifiers for the publication may be provided; all identifiers must be URIs.

Property	Namespace	Min	Max	Value Constraints
publisher	dcterms	0	1	The publisher of the publication.
rights	dcterms	0	∞	Zero or more statements, or value URIs, concerning copyright ownership or the permitted uses of the publication.
accessRights	dcterms	0	1	Must be one of two possible values: *restricted* or *unrestricted*.
references	dcterms	0	∞	A work referenced by the publication. Each element value may be a text string, an OpenURL Context Object, or both.
isAccessedVia	cld	0	1	The service that provides access to the publication. A value URI is required. A string value may also be provided.
isPartOf	dcterms	0	1	The value of this property is the publicationContainer description.

publicationContainer properties

Property	Namespace	Min	Max	Value
publication-Name	prism	0	1	The title of the parent publication; for example, a journal, book, or proceedings title.
contributor	dcterms	0	∞	A contributor to the parent publication, such as an editor of a book or proceedings. If used, a value string is required; a value URI may be provided.
issn	prism	0	1	A journal ISSN number.
eIssn	prism	0	1	A journal e-ISSN number.
isbn	prism	0	1	A book ISBN number.
doi	prism	0	1	A DOI for the parent publication.
identifier	dcterms	0	1	Additional identifiers for the parent publication may be provided; all identifiers must be URIs.
volume	prism	0	1	A journal volume number or other alphanumeric volume identifier.
number	prism	0	1	A journal issue number or other alphanumeric issue identifier.