Requirements for ISO MLR interoperability

This document presents a path for making the MLR standard harmonize with existing metadata standards in the field of education and related fields. We outline a set of requirements that will, if followed, enable MLR to interoperate maximally with Dublin Core, the Semantic Web and Web Architecture, and future library metadata standards, while retaining a level of interoperability with IEEE LOM.

Introduction

The MLR standard is a new metadata specification that will enter a domain already populated by several widely-used metadata specifications, notably IEEE LOM and Dublin Core.
It is a stated goal of the MLR standard that interoperability with LOM and Dublin Core should be achieved.

It has been demonstrated that complete interoperability between IEEE LOM and Dublin Core cannot be reached (see, for example. Nilsson et al. 2006). In order to not create additional disruption in the metadata community, it is absolutely essential that MLR does not become a third incompatible specification.

In Nilsson et al, (2008a), a roadmap for harmonization of metadata specifications was presented, with special focus on IEEE LOM and Dublin Core, but also discussing library metadata (MODS, METS, MARC and RDA) as well as MPEG-7 and W3C's RDF metadata framework. The analysis concludes that interoperable metadata requires a level of semantic harmonization that is best achieved by aligning the underlying abstract models of metadata specifications with that of the Resource Description Framework (RDF) and Dublin Core. Work is already progressing within the respective communities towards aligning IEEE LOM (see The Joint DCMI / iEEE LTSC Taskforce) and library metadata (see DCMI/RDA Task Group) with that model.

It is the belief of the signatories of this document that MLR should follow this path, and produce a standard that fully interoperates with the abstract models of Dublin Core and RDF, while retaining a level of backwards compatibility with LOM ensured by the work of the Joint DCMI/IEEE LTSC Taskforce.

This document outlines a number of steps that will make MLR compatible with the RDF model, while at the same time retaining the level of expressiveness and detail required by MLR. The proposal outlined here builds on the fundamental principle that details of data structures and encodings are best delegated to application profiles, with the central parts of the standard concentrated on semantic definitions of the terms used.

This principle also follows the requirements of the Singapore Framework for Dublin Core Application Profiles (Nilsson et al, 2008b).

Use Cases

The basic benchmark for metadata harmonization, as outlined in Nilsson et al, (2008a), is whether "elements" from one specification can be reused in the context of a different specification. We can see a few important use cases that MLR should support:

1. An application takes an MLR element, and reuses that in a domain-specific Dublin Core application profile. Results:
    - An MLR tool can process the metadata and extract the MLR element *without implementing manual translation tools*
    - A Dublin Core tool can process the metadata without issues.
2. Conversely, an application takes the MLR application profile, and adds a Dublin Core property. Results:
    - An MLR tool can process the metadata without issues
    - A Dublin Core tool can process the metadata and extract the Dublin Core property *without implementing manual translation tools*
3. A combined application profile constructed from IEEE LOM, MLR, Dublin Core, and FOAF (Friend-of-a-Friend) is designed. Results:
    - Tools supporting either IEEE LOM, MLR, Dublin Core or FOAF will be able to reuse the data they understand.
4. An application profile based on FOAF is produced, centering around a single person, linking to his/her teaching materials. MLR is used for describing aspects of the teaching material. Results:
    - Even though the application profile is centered around a person rather than a learning resource, MLR tools can process the MLR elements used
    - As above, FOAF tools can process the metadata
5. An application profile based on MLR is produced, which replaces the structures defined for persons with a simple reference to a person by identifier. Results:
    - An MLR application will still understand the relation between the learning resource and the person, though the data about the person is maintained in other metadata records.
6. An application profile based on MLR is produced, which replaces the the data element used to name the organization of a person, with a more detailed description to the organization. Results:
    - An MLR application will still understand the relation between the person and the organization, but may not understand all properties used to describe the organization.

In all cases, semantic web tools will be able to consume, index, query and otherwise analyze and manipulate all the metadata.

The fundamental requirements for enabling the above use cases are:

1. MLR elements need to be defined as properties in the sense defined by Dublin Core and RDF. This is the core argument in this document, and a core requirement for harmonization to be achieved.

2. MLR elements must not be limited to application profiles centered around a learning object. There will be many forms of application profiles, metadata fragments of different granularity will co-exist, and a monolithic standard will not cope with this requirement.
3. An individual MLR element must be reusable independently of any of the other elements. This way, the meaning of an MLR element is the same regardless of what context it is used in. This enables maximal return of investment for the implementation of MLR in diverse systems.
4. MLR elements must not make assumptions on what information regarding an entity that should be placed in an application profile. The amount of specificity is defined in the application profiles, not in the element definition.

Requirements for the definitions of MLR elements

In order to support the use cases outlined above, MLR elements need to be declared in a way compatible with the DCMI Abstract Model, and therefore also compatible with the RDF model. There are a few simple requirements needed to fulfill this goal:

- Each MLR element needs to be defined as a relation between two entities. for example, a creator element is a relation between a Resource and a Person (or Agent, generally).
- Each MLR elements must specify to which class of resources the element applies (the Domain of the property), and which class of resources the element can have as values (the Range of the property).
- Each MLR element must specify whether the value is a string, or whether it is a thing/entity (Person, Concept, Status, etc). Things/entities can be further described (in the same metadata record or elsewhere), while for string literals, no further description of the value is necessary or even possible.

Following these three guidelines goes a long way towards full interoperability with RDF, Dublin Core and the semantic web as outlined above. The document "Domains and Ranges for DCMI properties" contains a long discussion of the issue (see references).

Requirements for Vocabulary Definitions

As a consequence of the above, vocabularies come in two kinds. One kind is a set of string literals that are to be used as literal string values for properties. This is referred to in RDF terminology as a Datatype. Integer values are a typical example of this.

The other form is when the items in the vocabulary are entities of a certain type, that can in turn have descriptions. Among the Dublin Core elements, Languages are examples of this. A language is an entity, it may have labels in different languages, it may have an ISO code, etc. There are pre-defined vocabularies of languages.

Choosing between one form or the other is a modeling decision. If there are no cases where the value of a property might be further described somewhere, a literal value is often appropriate. In other cases, modeling them as entities is appropriate.

As discussed in Nilsson et al. (2008a), in the case of a vocabulary as entity, assigning URIs to each vocabulary item is a strong enabler of interoperability. This allows a simple method of disambiguation and linking of a vocabulary term with its description.

The role of Compound elements

The MLR as currently formulated relies heavily on the use of "compound" elements, much like IEEE LOM. The signatories believe this to be a construct that unnecessarily hinders semantic interoperability. This can be seen in use case 4-6 above, where the current assumptions in MLR about component elements break down. In use case 4, the top level entity is a person, not a learning resource. In use case 5, a compound element is collapsed to a single value, and in use case 6, a single value is expanded into a compound. Thus, it should be clear that in order to support these use cases, the notion of a compound element does not belong with the element definitions, but is an artifact of the choices made when designing an application profile.

What matters for interoperability is that the relationships between the described entities are well-defined. We refer to this as *semantic interoperability*, i.e. the independence of the meaning of resource relations of any particular encoding format or choice of granularity of description.

Thus the role of application profiles, as described in the Singapore Framework for Dublin Core Application Profiles, is to choose a particular set of entities and their relations, and thus define a particular data structure that lends itself well to implementation in the relevant systems. It should be clear, for example, that an implementation of use case 1 and use case 4 will be structured very differently. It is therefore a mistake to encode the structure of metadata in the definition of the elements - compounding belongs to application profiles.

Requirements on MLR-based Domain Models and Application profiles

The MLR standard may or may not choose to include one or several basic application profiles. Following the guidelines in the Singapore Framework, the definition of such an application profile encompasses two main steps: outlining the domain model and defining the details of the profile

Defining the Domain model

In order to implement the above requirements, the MLR application profile needs to be defined with what the Singapore Framework calls a Domain Model - a description of the basic entities that relate to the MLR elements, and their fundamental relations. The entities come in two kinds:

1. Primary entities - these are the entities that are themselves described by MLR. Examples include
    1. A learning resource
    2. Diverse agents (persons and organizations)
    3. Contributions (an entity that define a contribution to a resource by someone, on some role at some point in time)
    4. Annotations
    5. Classifications
    6. etc.
2. Secondary entities - these are entities that are only referenced (by some form of identifier) by the MLR. Typical examples are
    1. Vocabulary terms. These are usually described in different forms of vocabulary descriptions.

2. Independent resources - be they images, other learning objects, etc
3. etc.

It's important to realize that different application profiles make different choices regarding what constitute primary and secondary entities. For example, use case 5 makes persons secondary entities, while use case 6 makes organizations primary entities. This is one of the central reasons why the elements need to be defined independently of any choice an application profile may make.

Thus, by defining a domain model for the MLR application profile, we choose what elements are to be encoded as compounds, and what elements are not.

Defining the Application Profile

An application profile based on the MLR elements then must document, for each primary entity, what properties/elements are used, and if the value of the property is a primary entity further described in the same metadata record, a secondary entity only referenced in the metadata record, or a string literal. Statements can then be made regarding what properties are mandatory, etc.

It is worth noting that the Dublin Core community is working on a model for machine-processable definitions of application profiles based on the Singapore Framework, see Nilsson (2007). The model supports formalizing complex application profiles such as the FRBR-based ePrints application profile (see http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_Application_Profile), as well as simple, flat application profiles like the traditional 15-element Simple Dublin Core model.

This highlights the fact that a separation of semantics and structure does not mean a loss in detail and complexity. Quite the opposite - it allows for arbitrarily complex and detailed application profiles without losing interoperability.

Consequences for current MLR texts

Part 1

1. Each element needs to be defined using the additional Domain and Range specifications.
2. In addition, NO definition of compounds can be part of the element definitions.
3. Therefore, data types can only be given for elements with literal values.

A simple and sufficient model is the DCMI terms documentation (see references).

Part 2 etc.

1. The list of elements needs to be made flat - compounds are no longer allowed
2. Each element must be given a Domain and range.
3. Vocabulary terms that are entities should be given URIs, and their type declared.
4. vocabularies that are data types need to have their syntax and value spaces declared.

A change in how to express conformance

When moving away from a metadata standard based on a pre-defined structure, it becomes more difficult to state what, exactly, conformance means. A straightforward proposal would require an application to, as baseline criteria:

- Be able to process all MLR properties as defined in this standard
- Be able to process all MLR vocabularies as defined in this standard
- Adhere to the specifications of domains and ranges when using MLR elements

The MLR application profile may contain further criteria:

- Be able to generate metadata conforming to the application profile
- Be able to process all metadata in the MLR application profile

Conclusions

We believe the above steps are necessary to ensure the viability and interoperability of the MLR. We look forward to contributing to the implementation of these requirements in the MLR drafts!

Signatories:

Mikael Nilsson (Knowledge Management Research Group, Royal Institute of Technology, Stockholm)

Thomas Baker (Göttingen State University Library)

Ambjörn Naeve (Knowledge Management Research Group, Royal Institute of Technology, Stockholm)

Fredrik Paulsson (Umeå University)

Tore Hoel (Vice Chair, CEN/ISSS Workshop on Learning Technologies; Norwegian national expert to ISO/IEC JTC1 SC36)

Erlend Øverby (Norwegian national expert to ISO/IEC JTC1 SC36, Head of Standard Norway's national committee on learning technologies)

Peter Karlberg (Swedish national expert to ISO/IEC JTC1 SC36)

Jon Mason (Australian national expert to ISO/IEC JTC1 SC36)

Liddy Neville, (La Trobe University, DCMI Advisory Board, Australian national expert to ISO/IEC JTC1 SC36)

References:

Nilsson, M., Johnston, P., Naeve, A., Powell, A. (2006), *The Future of Learning Object Metadata Interoperability*, in Koohang A. (ed.) *Learning Objects: Standards, Metadata, Repositories, and LCMS*. http://kmr.nada.kth.se/papers/SemanticWeb/FutureOfLOMI.pdf

Nilsson, M. (2007), DCMI Description Set Profiles, DCMI Working Draft.
http://dublincore.org/architecturewiki/DescriptionSetProfile

Nilsson, M., Naeve, A., Duval, E., Johnston, P., Massart, D. (2008a), *Harmonization of Metadata Standards*, ProLEARN Deliverable D4.7.
http://ariadne.cs.kuleuven.be/lomi/images/5/52/D4.7-prolearn.pdf

Nilsson, M., Baker, T., Johnston, P. (2008b), *The Singapore Framework for Dublin Core Application Profiles*, Dublin Core Metadata Initiative.
http://dublincore.org/documents/singapore-framework/

Powell, A., Johnston, P., Baker, T. (2008), *Domains and Ranges for DCMI Properties*,
http://dublincore.org/documents/domain-range/

DCMI Usage Board (2008), *DCMI Metadata Terms*, http://dublincore.org/documents/dcmi-terms/

The Joint DCMI / IEEE LTSC Taskforce.
http://dublincore.org/educationwiki/DCMIIEEELTSCTaskforce