Towards an Interoperability Framework for Metadata Standards

Mikael Nilsson
KMR Group, NADA, Royal Institute of Technology, Stockholm, Sweden
mini@nada.kth.se

Pete Johnston
Eduserv Foundation, United Kingdom
pete.johnston@eduserv.org.uk

Ambjörn Naeve
KMR Group, NADA, Royal Institute of Technology, Stockholm, Sweden
amb@nada.kth.se

Andy Powell
Eduserv Foundation, United Kingdom
andy.powell@eduserv.org.uk

1. Abstract

This paper presents a conceptual metadata framework for Dublin Core metadata, intended to support the development of interoperable metadata standards and applications. The model rests on the fundamental concept of an "abstract model" for metadata, as exemplified by the DCMI Abstract Model, and is based on concepts and ideas that have developed over the years within the Dublin Core Metadata Initiative.

The model thus incorporates the concepts of metadata vocabularies, schemas, formats and application profiles into a single framework that can be used to analyse and compare metadata standards, and aid in the process of harmonization of metadata standards. IThe model is used to briefly compare the structures of the Dublin Core metadata specifications and the IEEE LOM standard. Some fundamental differences between the two standards are discussed briefly, and important gaps in the current set of Dublin Core metadata specifications are noted.

Keywords:

Dublin Core, abstract model, semantic interoperability.

2. Background

The publication of the DCMI Abstract model (DCAM) (Powell et al, 2005) in March 2005 marked a major milestone for the Dublin Core community and the DCMI. In developing the DCAM, the DCMI has shown its intention to gradually move away from dealing primarily with the “core” set of terms, moving instead to dealing primarily with community-specific application profiles, each defined within a common framework (Baker, 2005). Within such a framework, metadata terms from different and independent communities can co-exist, allowing for a controlled mix-and-match of community- and application-specific metadata constructs.

Although the framework used by the the Dublin Core community is still not formalized by the DCMI, considerable experience and documentation regarding the necessary components of such a framework have been collected over the years. It is the intention of this paper to introduce an over-arching model to describe the components of this framework, to serve as a possible basis for further formalization, and to highlight the strong and weak points of the current situation.

The model proposed in this paper is also intended to serve as a guide to understanding the conceptual relationships between the structures of the many different metadata standards currently in use. We will demonstrate this by using the model as a tool to compare the structure of the Dublin Core metadata framework with the IEEE LOM standard. Although the model has its origins in the Dublin Core metadata framework, we believe the model has a substantially more general applicability.

This attempt at designing a framework for Dublin Core metadata shares some features with the Warwick Framework (Lagoze, 1996), although that framework focused more on the packaging of metadata descriptions than on the nature of those metadata descriptions and the interoperability of the standards and specifications on which those metadata descriptions were based. The RDF suite of specifications, however, follow a more similar pattern to the framework presented here.

In other, related contexts, many similar kinds of frameworks have been designed over the years.

- The UML 2.0 specifications in general and the UML Meta Object Facility in particular, share some basic modeling principles with the framework presented here, albeit with a markedly higher level of complexity, and a primary focus on model-driven design.
- The MPEG-7 multimedia metadata framework also contains a complete framework for metadata vocabulary management, but with little emphasis on use in other contexts than multimedia.
- The ISO 11179 framework is of particular significance for describing metadata and metadata models, but is not concrete enough without further specialization to cater for the needs of real-world metadata interoperability.

A fuller analysis would require a much more thorough discussion. However, it can still be concluded that in comparison with these and other related frameworks, the most important distinguishing features of the Dublin Core metadata framework presented here is its relative simplicity, straightforwardness and cross-domain applicability.

3. The DCMI Metadata Framework and its Components

In this section, the metadata framework used by the Dublin Core community is examined and a set of components of that framework for Dublin Core metadata are identified: the *abstract model*, *metadata formats*, *metadata vocabularies*, the *vocabulary model*, *application profiles* and the *profile model*. Some of these components correspond to concepts that have been formalized by DCMI (as DCMI recommendations or other documents); other components represent abstractions based on current usage of Dublin Core metadata and on current directions in metadata interoperability.

3.1 The Abstract Model

The abstract model specifies the concepts used in the framework, the nature of terms and how they combine to form an information structure. An early effort to produce such framework for Dublin Core was presented in Bearman, Miller, Rust, Trant and Weibel (1999).

Subsequently the DCMI Usage Board developed the "DCMI Grammatical Principles" (DCMI Usage Board, 2003), as a summary expression of the key concepts underpinning the vocabularies developed by the DCMI. The DCMI Abstract Model, published in March 2005, was a substantial reformulation and clarification of these principles.

The DCMI Abstract Model defines the *description set* as the principal information structure used in Dublin Core metadata. It describes the nature of the components that make up that information structure and it also describes how that composite information structure is to be interpreted.

In summary, a *description set* is described as follows:

- a *description set* is made up of one or more *descriptions*
- a *description* is made up of
  - zero or one *resource URI* and
  - one or more *statements*
- a *statement* is made up of
  - exactly one *property URI* and
  - zero or one reference to a value in the form of a *value URI*
  - zero or more representations of a value, each in the form of a *value representation*
  - zero or one *vocabulary encoding scheme URI*
- a *value representation* is either
  - a *value string* or
  - a *rich representation*
- a value string may have an associated *value string language*
- a value string may have an associated *syntax encoding scheme URI*
- each value may be the subject of a *related description*

A DC metadata description set is to be interpreted as a set of assertions about the resources identified by those URIs, principally about the relationships between the two resources identified by the *resource URI* and the *value URI*.

The abstract model is the key used by a metadata application to unlock the secrets of a metadata expression given in a specific format, thus making it possible for a single standard, though expressed in several different formats, to still be understood in a uniform way by users and applications.

3.2 Metadata Formats

The abstract model describes an abstract information structure. Metadata applications construct and exchange instances of that abstract information structure, and they do so by representing the information structure as a digital object, using the rules specified by one of several *metadata formats* or *bindings*. In the case of Dublin Core, DCMI has published a set of “encoding guidelines” specifications which provide bindings for DC metadata.

A binding is constructed by specifying how each kind of concept in the abstract model is to be encoded in a particular format. Conversely, the binding also specifies how to interpret data given in a specific format in terms of the abstract model. For example, when interpreting a metadata record that uses the Dublin Core XML binding, an XML element called “dcterms:modified” used in a particular place in the XML document represents a property, and the value “dcterms:W3CDTF” of a particular XML attribute represents a syntax encoding scheme for the value string “2001-07-18” occurring as XML content in a particular position.

This fundamental process of *encoding/interpretation* is described in Figure 1. Application A uses the DCMI Abstract Model to represent some metadata about a resource. This

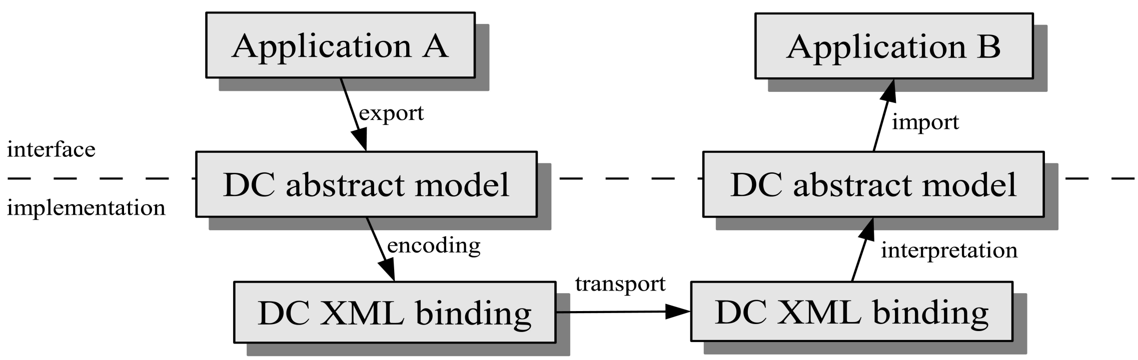

Figure 1. The process of encoding/interpretation of metadata within the framework of an abstract model.

metadata is encoded using the Dublin Core XML binding, and transferred to another application. Application B will use the rules of the Dublin Core XML binding to interpret the XML data in terms of the DCMI Abstract Model. This representation of the metadata can then be used in the application.

When two applications want to exchange Dublin Core metadata, they understand metadata through the lens of the abstract model. The abstract model functions as an opaque interface, an API, to the metadata. In practice, the exchange is realized using one of the Dublin Core bindings, but the details of the formats are of no interest to the applications, which instead analyse the metadata in terms of the interface given by the abstract model.

Note that it is possible to produce applications that process metadata without regard to the abstract model. Such *ad-hoc processing* of metadata records requires that the precise content of the records is well-known in advance, which is the case in many systems where extensibility, modularity and refinements are not design requirements. In contrast, the kind of *interoperable processing* based on the abstract model described above is necessary when an application needs to be prepared for metadata constructs that do not fall within the limits of such a precise, pre-conceived description. Thus, it should be clear that interoperable processing is a basic prerequisite for metadata interoperability.

3.3 Metadata Vocabularies

Although the abstract model specifies the nature of the terms that are used in a DC metadata description set, it does not list any fixed set of terms to be used. On the contrary, the Dublin Core metadata framework is based on the notion that the *vocabularies* used in DC metadata description sets are created and maintained separately from the abstract model.

Although the initial focus of the DCMI was on building consensus around the use of a small set of metadata terms that could be used to create fairly simple descriptions of a wide range of resources – the fifteen properties (or "elements") of the Dublin Core Metadata Element Set – the experience of implementing DC metadata highlighted that in practice these terms were supplemented with other terms to meet the requirements of some particular community or application context.

In Dublin Core metadata, a vocabulary can be one of two things:

1. A *value vocabulary*, consisting of concepts from a controlled set as specified by a vocabulary encoding scheme. For example, the "dcterms:LCSH" vocabulary encoding scheme refers to the vocabulary formed by the set of Library of Congress subject headings.
2. An *element vocabulary*, consisting of a set of metadata properties together with their definitions. For example, the Dublin Core Element Set, consisting of the 15 original Dublin Core elements (*dc:title*, *dc:subject*, etc.), is such a vocabulary.

Element vocabularies and value vocabularies have fundamentally different characteristics. While value vocabularies are used to construct taxonomies and thesauri that describe relationships between concepts in terms of broader/narrower, containment etc, element vocabularies are used to construct application profiles, schemas and ontologies that describe how metadata instances are to be constructed.

3.4 Vocabulary Model

As the Dublin Core community embraced the notion that DC metadata might utilize multiple metadata vocabularies, they also recognized that specific types of relationship could exist between the metadata terms referenced in DC metadata – both between terms within a single vocabulary and between terms in different vocabularies. An example of such a relationship between terms is that of "element refinement" where one property is described as a specialization of another property.

Consensus on the nature of these relationship types is the basis of an implicit *vocabulary model*. Clearly that vocabulary model is closely related to the DCMI Abstract Model since it is concerned specifically with the types of terms described by the abstract model, and the relationships between terms of those types.

If applications are to be able to act on information about such relationships between metadata terms, then those terms and the relationships between them must be described in a machine-processable form, i.e. a language for describing metadata vocabularies is necessary. Such a vocabulary language enables the description of element and value vocabularies in a form which enables applications to access information about the nature of the terms and their relationships with other terms in the same or in different vocabularies.

The Dublin Core vocabulary model has not yet been formalized, but embryos such as Baker (2003) exist. DCMI has a history of using RDF Schema (Brickley *et al* 2004) as a basis for its machine-readable term declarations. RDF Schema is useful for describing both element and value vocabularies.

3.5 Application Profiles

The Dublin Core metadata standard emerged from an interest in developing a resource description standard that could be applied across a broad range of communities and domains. Since its inception, the DC community had the expectation that Dublin Core would be deployed alongside other metadata standards. They also learned from experience that implementers tailored the standard to fit the requirements of their own context.

More recently, these two trends have converged in the notion of the DC application profile, and the principle that implementers of metadata standards should be able to assemble the components that they require for some particular set of functions - and if that means drawing on components that are specified within different metadata standards, that should be possible.

Duval et al (2002) employ the metaphor of the Lego set to describe this process: an application designer should be able to “snap together” selected “building blocks” drawn from the “kits” provided by different metadata standards to build the construction that meets their requirements, even if the kits that provide those blocks were created quite independently.

Heery and Patel (2000) present a compelling vision of metadata implementers “mixing and matching” “data elements”, constructing application profiles by selecting from the sets of “data elements” provided by metadata standards and by other implementers. Such application profiles are fundamental to a modern metadata framework.

Just as the description set construct defined by the DCMI Abstract Model embraces the description of a number of related resources, so too a DC application profile may specify the construction of the related descriptions of several kinds of related resources, such as a collection, the items it consists of and the associated contributors. Thus, such a specification is a multi-layered structure of some complexity, that can not, in general, be captured by a flat list of properties.

3.6 Profile Model

Although the concept of the DC application profile has gained general acceptance within DCMI and the DC implementer community, it has not yet been formalized by DCMI in the form of a model for a DC application profile.

Like the vocabulary model, the profile model is closely related to the abstract model, because it is concerned with specifying the creation of the particular information structures described by the abstract model – in the case of Dublin Core, description sets, as defined by the DCMI Abstract Model.

Any such model must not be tied to a specific metadata format, but must operate at the level of the abstract model, so that the application profile can be applied independently of the metadata format in which metadata instances are encoded.

Promising work on machine-processable DC application profiles can be seen in, e.g.,“Guidelines” (2005)

3.7 The DCMI Metadata Framework

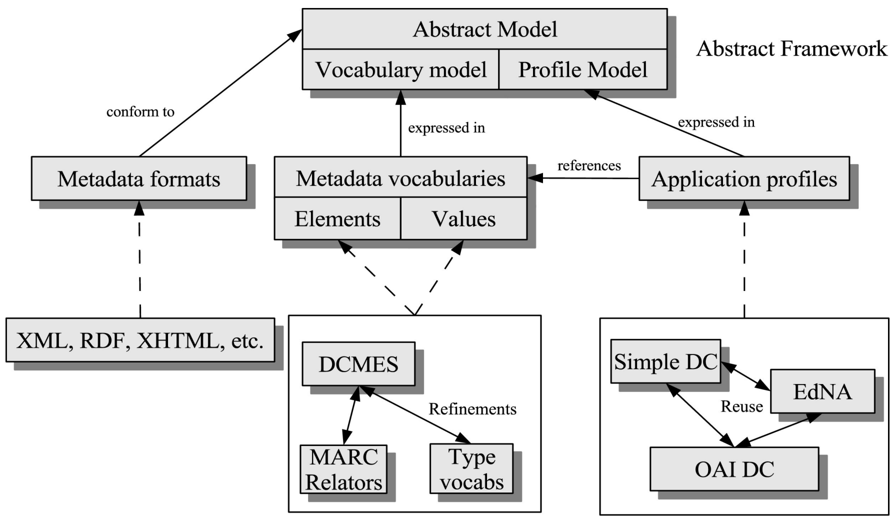

Figure 2. A model of the Dublin Core metadata framework.

This brief survey of DCMI specifications and DC metadata usage highlights the existence of a number of inter-related and inter-dependent features, which when viewed together can be seen, implicitly at least, as components of a larger framework. The relationships between these component parts of the Dublin Core metadata framework are depicted in Figure 2.

The diagram highlights the close relationship between the DCMI Abstract Model and the DC vocabulary model and DC application profile model.

4. A comparative view: applying the framework model to LOM, Dublin Core and the Semantic Web

This section seeks to generalize this model of a metadata framework and to apply it to the analysis of two other metadata standards, and to identify the corresponding components within the frameworks deployed by those standards.

The following table presents a summary view of the framework components as they are present within the IEEE LOM standard and within the Semantic Web suite of specifications, and indicates the extent to which each component is formally distinguished from other components within the framework. Note that by “Dublin Core framework” we refer to the complete set of DCMI specifications, and similarly for LOM.

Framework concept	*Dublin Core framework*	*LOM framework*	*Semantic Web framework*
Abstract Model	DCMI Abstract Model	Implicit in LOM Data Model	RDF Concepts and Abstract Syntax
Metadata Formats	XML, RDF and HTML bindings	XML binding	RDF/XML syntax, N-triples, etc.
Metadata Element Vocabularies	DCMES, large set of external properties and encoding schemes	LOM Data Model includes element vocabulary, various extensions to LOM	Many external element vocabularies
Metadata Value Vocabularies	DCMIType vocabulary. Many external value vocabularies	LOM Data Model includes several basic value vocabularies, many external vocabularies	Many external value vocabularies
Vocabulary Model	Not formalized, but see Baker (2003)	Not formalized	RDF Vocabulary Description Language
Application Profiles	Some published by DCMI, many external application profiles	LOM Data Model includes basic application profile, many external application profiles.	Many in the form of ontologies
Profile Model	Not formalized, but cf "Guidelines".	Not formalized	Possibly OWL, the Web Ontology Language

A few comments on this table:

- Not all parts are formalized. The DCMI is slowly progressing towards formalizing the complete abstract framework, including abstract model, vocabulary model and profile model. Similar efforts are not under way in LOM.
- The most mature parts are certainly value vocabularies, where many external sources exist. Dublin Core metadata element vocabularies are also relatively mature. To some extent, and to some extent application profiles have some maturity, even though there is still a certain amount of confusion in the community regarding the precise nature of an application profile.
- In spite of the existence of many application profiles and metadata vocabularies, no formal model is usually followed in their design.
- LOM has a relatively weak notion of element vocabularies, as noted in Nilsson et al (2006), that does not support URI identification of elements.
- The LOM Data Model defines, in a single standard, both an abstract model (implicitly, at least), a metadata element vocabulary, a set of metadata value vocabularies, and a basic application profile. This is one way of expressing the well known "monolithic" nature of the LOM standard.
- Further comparison with e.g. MODS, MPEG-7 etc. remains the subject of a future article.

In short, the above table can be used to analyse and compare metadata standards, and understand how they relate to different aspects of the Dublin Core universe.

5. Interoperability across metadata frameworks

Although the use of the model has enabled us identify the corresponding components within the frameworks of the different standards, significant differences may still exist between the corresponding components in the different frameworks. For example, although both the Dublin Core metadata standard and the LOM metadata standard incorporate the notion of an abstract model (either explicitly or implicitly), those two abstract models are quite different: the conceptual information structures that they describe, and the nature of the terms used in those conceptual information structures, are quite different – and those differences carry over into the corresponding vocabulary models and the profile models. In the cases of Dublin Core and the Semantic Web specifications, again there are differences between the two abstract models, but they are more similar than in the case of Dublin Core and LOM. In the case of Dublin Core and the Semantic Web specifications, the two abstract models are broadly compatible, and this is reflected in DCMI's use of the RDF Vocabulary Description Language to describe its vocabularies.

Such differences become critical when we begin to consider interoperability across different metadata standards constructed within their own metadata frameworks. A significant part of the motivation for the development of the profile models within both the Dublin Core and LOM frameworks was precisely to facilitate the (re)use of metadata vocabularies across the boundaries of the two corresponding frameworks. While those models have certainly increased interoperability *within* the respective frameworks, interoperability between the different frameworks remains a difficult problem.

With a similar aim in mind, the CORES Resolution (Baker and Dekkers, 2002), which has been signed by both the IEEE LTSC and the Dublin Core Metadata Initiative, encouraged the owners of metadata standards to assign URI references to their "elements", the "units of meaning comparable and mappable to elements of other standards". The assignment of a URI to an "element" means that it can be unambiguously cited in a global context, and this is a necessary condition for the sort of mixing and matching foreseen by Heery and Patel. However the assignment of a URI to an "element" does not change the nature of that "element": and it does not make it meaningful to use the URI of a LOM data element as, e.g., a property URI in a Dublin Core metadata description. Similar incompatibilities have been noted between, e.g., RDF and MPEG-7 (van Ossenbruggen, Nack and Hardman, 2004 and Nack, van Ossenbruggen and Hardman, 2005).

The analysis in Nilsson et al (2006) shows that we must not confuse the components used in a metadata format and the constructs in the abstract model. The components in a metadata format, such as "element URIs" may seem to be similar and compatible, but in reality they belong to completely different frameworks that might not be compatible. Thus, according to the analysis in Nilsson et al (2006), the notion of reusing "elements" between metadata standards and formats using incompatible frameworks is fundamentally flawed. While assigning URIs for the component parts of a metadata standard is clearly a worthwhile effort in other ways, this does not really address the fundamental issue when creating interoperable metadata standards, namely the compatibility of their respective frameworks, and in particular, their abstract models.

Basing metadata on a compatible abstract models carries a number of important benefits

- Clear guidelines on how to create and maintain customized metadata vocabularies. There is currently some confusion on how to best produce vocabularies, much due to the differing fundamental principles for vocabularies in the different metadata standards.
- Fine-grained control over relationships between terms from different standards, including refinement and partial mappings. Automation of interoperable metadata management systems will be greatly improved, and metadata vocabularies will be

able to build upon each other.

- A single set of format bindings. Contrast this with the current situation, which requires every metadata standard to have its own set of format bindings. This will make life easier not only for metadata standardization bodies, but also for applications that will only need to support one format.
- A single framework for extending and combining metadata from different standards. This will enable standardized principles for the construction of interoperable application profiles.
- A single storage and query model for very different types of data and vocabularies. For example, storing metadata from different specifications in the same database will become more straightforward. Implementing searching that includes dependencies between metadata expressed in different schemas will be simplified.

6. The word "Metadata Standard"

In light of the model presented here, it seems clear that the current use of the term "metadata standard" or "metadata schema" will need refinement. These terms are often used interchangeably to describe one of the following:

- The over-arching abstract model standard. This will also include a specification for how to express the semantics of vocabularies adhering to the abstract model (the vocabulary model) as well as a specification for how to express application profiles in a machine-processable way (the profile model).
- Metadata format specifications. These will include bindings of the abstract model to a set of formats and systems, including XML, database layouts, programming languages, etc., as well as translations or mappings to other knowledge representation systems such as RDF. Such specification are closely tied to the abstract model.
- Metadata vocabularies. These will include metadata terms from different communities. The Dublin Core terms, the LOM elements and so on are examples of metadata element vocabularies, and a large set of value vocabularies also fit into this category.
- Application profiles. These will specify usages of metadata vocabularies in complex combinations.

Clarification of the underlying framework can hopefully contribute to better terminology in this domain.

7. Looking forward

We have presented an overarching framework for Dublin Core metadata, based on the implicit structure of current Dublin Core metadata standardization and practise.

The authors believe that the Dublin Core Metadata Initiative would be greatly helped by applying this understanding to improve its documentation and vision of metadata interoperability. In particular, a high-level framework for Dublin Core metadata has not been proposed since the Warwick framework, and it is now time to revisit the overall structure of metadata standardization. Luckily, as the analysis shows, there is some coherency in the current set of DCMI specifications, though much of it remains implicit. Making the overall structure explicit has the advantages of increasing coherency of terminology, making it easier to communicate the relative significance of each specification, simplifying for users to understand how metadata constructs may be used and reused, and more.

Another issue is that of interoperability with other metadata standards. By reinterpreting the framework in terms of LOM and the Semantic Web, we learn about differences between the metadata standards and deficiencies in their respective frameworks. The authors have little hope that deep integration between metadata standards can be made a reality unless they adhere to a single common framework. Unfortunately, a thorough analysis shows (Nilsson *et al,* 2006) that there are fundamental incompatibilities between frameworks such as the LOM framework and that of Dublin Core. On the other hand, the framework of RDF and the semantic web share many features with Dublin Core, and advanced interoperability between those frameworks has already been demonstrated.

The authors therefore argue that the long-term solution is to proceed towards a *shared* metadata framework. Having all metadata standards expressed using a common abstract model, or at least using compatible abstract models, would greatly increase interoperability in several ways. It would also create a natural separation between the specification of the structure of metadata descriptions and the declaration of metadata terms used within that structure, so that both LOM vocabularies and Dublin Core vocabularies would appear as metadata vocabularies within that one structure. Great care must be taken to ensure that such an abstract model does not conflict with the emerging metadata format for the Web: RDF.

There are already initiatives to develop a common abstract model that covers both LOM and Dublin Core, but unfortunately it seems to be impossible to arrive at such a model without re-engineering at least one standard to retrofit it to the new abstract model, which naturally is a major undertaking. An alternative approach is to produce "compatibility layers" that allow one metadata standard to be described and used in a different framework based on a common abstract model. An example of this is the development of a mapping of LOM to the DCMI Abstract Model (See "Joint DCMI/IEEE LTSC Task Force"). Reaching out to embrace the other important metadata standards, such as MODS, MPEG-7 and the IMS set of standards is then the logical next step.

The basis of the envisioned metadata standardization framework is the abstract model. The incompatibilities of abstract models are the most significant stumbling blocks for metadata interoperability. The development of a common abstract model for metadata is therefore of central importance if we are ever going to experience true metadata interoperability.

8. References

Baker, T. (2003), DCMI Usage Board Review of Application Profiles. Retrieved July 1 2005, from http://dublincore.org/usage/documents/profiles/

Baker, T. & Dekkers, M., (2002), CORES Standards Interoperability Forum Resolution on Metadata Element Identifiers. Retrieved July 1, 2005, from http://www.cores-eu.net/interoperability/cores-resolution/

Baker, T. (2003), Usage Board Application Profile (draft). Retrieved April 30, 2006, from http://dublincore.org/usage/meetings/2003/06/Usage-ap.html

Baker, T. (2005), Diverse Vocabularies in a Common Model: DCMI at ten years, Keynote speech, DC-2005, Madrid, Spain. Retrieved April 30, 2006, from http://dc2005.uc3m.es/program/presentations/2005-09-12.plenary.baker-keynote.ppt

Bearman, D., Miller, E., Rust, G., Trant, J. & Weibel, S. (1999), A Common Model to Support Interoperable Metadata, *D-Lib Magazine*, January 1999. Retrieved July 1, 2005, from http://www.dlib.org/dlib/january99/bearman/01bearman.html

Brickley, D. & Guha, R. V. (2004), RDF Vocabulary Description Language 1.0: RDF Schema, *W3C Recommendation 10 February 2004*. Retrieved July 1, 2005, from http://www.w3.org/TR/rdf-schema/

DCMI Usage Board (2003), DCMI Grammatical Principles. Retrieved April 30, 2006 from http://dublincore.org/usage/documents/principles/

Dublin Core Application Profile Guidelines (2003), CEN Workshop Agreement CWA 14855. Retrieved July 1, 2005, from ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/MMI-DC/cwa14855-00-2003-Nov.pdf

Duval, E., Hodgins, W., Sutton, S. & Weibel, S. L. (2002), Metadata Principles and Practicalities, *D-Lib Magazine*, April 2002. Retrieved July 1, 2005, from http://www.dlib.org/dlib/april02/weibel/04weibel.html

Friesen, N., Mason, J. & Ward, N. (2002), Building Educational Metadata Application Profiles, *Dublin Core - 2002 Proceedings: Metadata for e-Communities*: *Supporting Diversity and Convergence.* Retrieved July 1, 2005, from http://www.bncf.net/dc2002/program/ft/paper7.pdf

Godby, C. J., Smith, D. & Childress, E. (2003), Two Paths to Interoperable Metadata, *Proceedings of DC-2003: Supporting Communities of Discourse and Practice – Metadata Research & Applications,* Seattle, Washington (USA). Retrieved July 1, 2005, from http://www.siderean.com/dc2003/103_paper-22.pdf

Guidelines for machine-processable representation of Dublin Core Application Profiles (2005), CEN Workshop Agreement CWA 15248. Retrieved July 1, 2005, from ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/MMI-DC/cwa15248-00-2005-Apr.pdf

Heery, R. & Patel, M. (2000), Application Profiles: mixing and matching metadata schemas, *Ariadne Issue 25*, September 2000. Retrieved July 1, 2005, from http://www.ariadne.ac.uk/issue25/app-profiles/

Heflin, J. (2004), OWL Web Ontology Language – Use Cases and Requirements, *W3C Recommendation 10 February 2004*. Retrieved July 1, 2005, from http://www.w3.org/TR/webont-req/

IMS Global Learning Consortium (2004), IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata. Retrieved July 1, 2005, from http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

Johnston, P., (2005a), XML, RDF, and DCAPs. Retrieved July 1, 2005, from http://www.ukoln.ac.uk/metadata/dcmi/dc-elem-prop/

Johnston, P., (2005b), Element Refinement in Dublin Core Metadata. Retrieved July 1, 2005, from http://dublincore.org/documents/dc-elem-refine/

Joint DCMI/IEEE LTSC Task Force. http://dublincore.org/educationwiki/DCMIIEEELTSCTaskforce

Klyne, G. & Carroll, J. J. (2004), Resource Description Framework (RDF): Concepts and Abstract Syntax, *W3C Recommendation 10 February 2004*. Retrieved July 1, 2005, from http://www.w3.org/TR/rdf-concepts/

Lagoze, C. (1996), The Warwick Framework – A Container Architecture for Diverse Sets of Metadata, D-Lib Magazine, July/August 1996. Retrieved April 30, 2006 from http://www.dlib.org/dlib/july96/lagoze/07lagoze.html

Manola, F. & Miller, E. (2004), RDF Primer, *W3C Recommendation 10 February 2004*. Retrieved July 1, 2005, from http://www.w3.org/TR/rdf-primer/

Miles, A. J. & Brickley, D. (2005), SKOS Core Guide, *W3C Working Draft 10 May 2005*. Retrieved July 1, 2005, from http://www.w3.org/TR/swbp-skos-core-guide

Memorandum of Understanding between the Dublin Core Metadata Initiative and the IEEE Learning Technology Standards Committee (2000). Retrieved July 1, 2005, from http://dublincore.org/documents/2000/12/06/dcmi-ieee-mou/

Nack, F., van Ossenbruggen, J. & Hardman, L. (2005), That obscure object of desire: multimedia metadata on the Web, part 2, *IEEE Multimedia 12* (1) 54-63. Retrieved July 1, 2005, from http://ieeexplore.ieee.org/iel5/93/30053/01377102.pdf?arnumber=1377102

Nilsson, M. (2001), The Semantic Web: How RDF will change learning technology standards, Feature article, Centre for Educational Technology Interoperability Standards (CETIS). Retrieved July 1, 2005, from http://www.cetis.ac.uk/content/20010927172953/viewArticle

Nilsson, M., Palmér, M. & Naeve, A. (2002), Semantic Web Metadata for e-Learning - Some Architectural Guidelines, *Proceedings of the 11th World Wide Web Conference (WWW2002)*, Hawaii, USA. Retrieved July 1, 2005, from http://kmr.nada.kth.se/papers/SemanticWeb/p744-nilsson.pdf

Nilsson, M., Palmér, M. & Brase, J. (2003), The LOM RDF Binding - Principles and Implementation, *Proceedings of the Third Annual ARIADNE conference*. Retrieved July 1, 2005, from http://kmr.nada.kth.se/papers/SemanticWeb/LOMRDFBinding-ARIADNE.pdf

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*, in press.

Powell, A., Nilsson, M., Naeve, A., Johnston, P. (2005), DCMI Abstract Model, *DCMI Recommendation.* Retrieved July 1, 2005, from http://dublincore.org/documents/abstract-model/

Uschold, M. & Gruninger, M. (2002), Creating Semantically Integrated Communities on the World Wide Web, Invited Talk, Semantic Web Workshop, Co-located with WWW 2002, Honolulu, HI, May 7 2002. Retrieved July 1, 2005, from http://semanticweb2002.aifb.uni-karlsruhe.de/USCHOLD-Hawaii-InvitedTalk2002.pdf

van Ossenbruggen, J., Nack, F. & Hardman, L. (2004), That obscure object of desire: multimedia metadata on the Web, part 1, *IEEE Multimedia 11* (4) 38-48. Retrieved July 1, 2005, from http://ieeexplore.ieee.org/iel5/93/29587/01343828.pdf?arnumber=1343828