Dublin Core Metadata Initiative

WORKING DRAFT

Dublin Core Reference Model

DC Working Draft - 08 December 1999

This Version: <http://archive.dstc.edu.au/RDU/DCDM/wd-dm-dcrm.html>

Latest Version: <http://archive.dstc.edu.au/RDU/DCDM/wd-dm-dcrm.html>

Editors: Renato Iannella <renato@dstc.edu.au>, Eric Miller <emiller@oclc.org >, Paul Miller <lispm@ukoln.ac.uk >

Status of this document

This document is a Working Draft of the Dublin Core Metadata Initiative [DCMI], and is intended to reflect consensus reached within the DataModel Working Group [DCDM].

Comments and feedback should be sent to the working group mailing list, dc-datamodel@mailbase.ac.uk, the archives for which may be browsed at <http://www.mailbase.ac.uk/lists/dc-datamodel/> (NOTE, you must be a member of the WG to post messages to the WG) or, alternatively, send your feedback to the Editors of this Working Draft.

Table of Contents

---

1. Introduction

This document expresses the modelling requirements for the Dublin Core Element Set (DCES). The purpose of which is to establish a common framework from which implementations of DCES can refer and conform. The Dublin Core Reference Model establishes the semantics for interoperability for instances of the Dublin Core Element Set.

The Dublin Core Reference Model provides a consistent view of the functional requirements of the Dublin Core Element Set. Conformance to the functional requirements will support interoperability within and across the Dublin Core and other metadata communities.

The Dublin Core Reference Model covers two versions of the element set:

- Basic Dublin Core - which supports the current 15 defined elements of the DCES; and
- Qualified Dublin Core - which supports the addition of qualified extensions of the DCES for greater expressibility.

The primary focus of the majority of the Dublin Core Metadata Initiative working groups has been in semantic clarification of the DCES and the identification of common cross-domain qualifiers to support richer descriptive requirements. While this focus has allowed rapid conceptual development free of the constraints imposed by specific syntactic or structural applications, this has led to a certain lack of clarity at times, especially in relation to development and deployment of 'qualification' mechanisms which enrich descriptions in the Dublin Core. It has also made interoperable implementation difficult, as individual implementors have typically developed their own internal mechanisms for actually encoding Dublin Core; mechanisms which are not always compatible with those of their potential collaborators elsewhere.

A formal defintion and systematic representation of 'qualifiers' are a neccessary component for suporting interoperability accross implementors. To faciliate this, the identification of both implicit and explict requirements are helpful in both designing and validating any formal model. The following are a basis of these functional requirements:

1. **Element Identity.** It is important to clearly identify elements defined by particular resource description communities. This is useful both in understanding a particular context ('Title' in the context of Dublin Core Element Set) and to eliminate potential name-collision (eg 'Title' in the Dublin Core Element Set is different than 'Title in the vCard specification).
2. **Internationalization.** The Dublin Core Metadata Initiative has identified the requirement for defining the same descriptive concepts in many different languages.
3. **Modularization/Extensibility.** The Dublin Core Metadata Initiative recognized early on that there would never be one "true" declarative set of semantics to represent all resource descriptive requirements. As such, the ability to mix semantics defined in various different communities may be necessary for supporting general descriptive practices.
4. **Semantic Refinement.** Many of the Dublin Core elements encompass broad concepts. It is useful to provides the means for semantically refining such elements in order to say something much more specific. (eg the concept 'Illustrator' in the context of CIMI is a refinement of the concept 'Contributor' in the context of the DCES).
5. **Identification of Encoding Schemes.** The ability to specify a particular encoding scheme is critical to the interpretation of some types of metadata. Dates, for example are written in different formats in different places. Ambiguity can be resolved by identifying the format in which the date is encoded via recommendations for encoding schemes, but a given application could specify an alternate encoding for local purposes.
6. **Specification of Controlled Vocabularies.** Many disciplines use classification schemes, controlled vocabularies or thesauri to describe the content of resources. Creators of metadata need to be able to specify such vocabularies so as to leverage this important class of tools. The Dewey Decimal System, Medical Subject Headings, the Art and Architecture Thesaurus are examples, but there are many others.
7. **Identification of Structured Compound Values.** The use of compound structures as values of metadata is important to many communities, including the use of authority records for authorized versions of the names of subjects, places, corporate entities, and people.

It is important to emphasize that communities having existing qualification vocabularies and encoding schemes will be able to invoke these , enabling such communities to tailor their metadata to the needs of their constituents while remaining within the generalized metadata architecture of the DCMI. Existing capabilities should not be invented anew, but rather the new technology must be able to integrate these important legacy assets and thereby promote the broader acceptance of these valuable tools.

1.1 About this Specification

This document, along with its normative references, includes all the specification necessary for the implementation of interoperable Dublin Core applications.

The following key words are used throughout the document and should be read as interoperability requirements. This specification uses words as defined in [RFC2119] for defining the significance of each particular requirement. These words are:

MUST or MUST NOT
: This word or the adjective "required" means that the item is an absolute requirement of the specification.

SHOULD or SHOULD NOT
: This word or the adjective "recommended" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

MAY
: This word or the adjective "optional" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

Examples of Qualifiers used in this document are for demonstration purposes only and should not be interpreted as official qualifiers.

1.2 Deprecated Terms

A number of terms have been used previously to refer to aspects of Dublin Core metadata. These terms are now no longer recommended as they have been superseeded by the vocabulary of the Reference Model document. These terms include, with the new terms in brackets:

- Simple Dublin Core (Basic Dublin Core)
- Unqualified Dublin Core (Basic Dublin Core)
- Complex Dublin Core (Qualified Dublin Core)
- Subelements (Qualifiers)
- Scheme (Vocabulary or Encoding Value Qualifier)

---

2. Basic Dublin Core Reference Model

The Basic Dublin Core Reference Model is a core model consisting of:

Resource
: A Resource is any object that has identity.

Element
: An element is a specific characteristic or property of the Resource

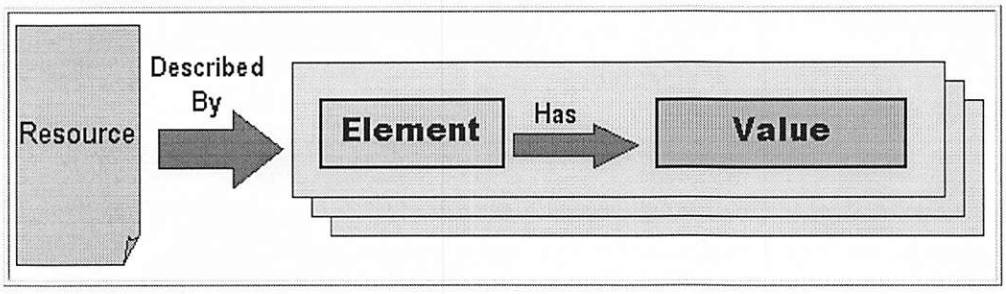
Value
: The Value is a literal text string corresponding to the Element

The Resource is described by a collection of Elements and their Values. This collection is sometimes referred to as a record.

Within the DCES context, a Resource will typically be an information or service resource, but may be applied more broadly.

The semantics of the Elements have been defined in the Version 1.1 the Dublin Core Element Set [DCES]. The Version 1.1 semantics form a normative part of Dublin Core Reference Model.

The figure below graphically represents the Basic Dublin Core Reference Model. The Resource is being being described by a collection of Elements and their Values. This collection is the instance of DCES metadata.



Basic Dublin Core Reference Model

2.1. Example

Below is an example of an instance of Basic Dublin Core metadata for the Resource with identity of <http://dublincore.org/doc/REC-DCRM.html>.

Element	Value
Title	Dublin Core Reference Model
Creator	Renato Iannella
Creator	Eric Miller
Creator	Paul Miller
Date	1999-12-01
Subject	Metadata, Dublin Core, Reference Model

2.2. Basic Conformance

The Basic Dublin Core Reference Model conformance requirements for interoperable metadata include:

1. Must support the semantics for all Version 1 releases of the Dublin Core Element Set.
2. Must support a mechanism to identify the Resource being described.
3. All Elements must be repeatable.
4. All Elements may be optional.
5. Limits to the length of text strings for Values may be set and clearly indicated.

---

3. Qualified Dublin Core Reference Model

Qualified Dublin Core Reference Model builds on the Basic Reference Model and provides greater functionality. The Qualified Reference Model is a more complex model, yet is backward-compatible by preserving the Basic Dublin Core Reference Model structure and semantics.

The richer semantics of Qualified Dublin Core are provided by the following *qualification* mechanisms:

Element Qualifier
: An Element Qualifier is a specialisation of its parent Element. An Element Qualifier supports all of the characteristics of parent Element, but refines some or all of these characteristics. Conversely, an Element is a generalisation of its Element Qualifier. For example, the Creation Date (specialisation) is an Element Qualifier for the Date Element (generalisation) since all the characteristics of Creation Date are present in Date.

Value Qualifiers
: Value Qualifiers are additional attributes or facts that further specify the element Value. There are four types of Value Qualifiers:

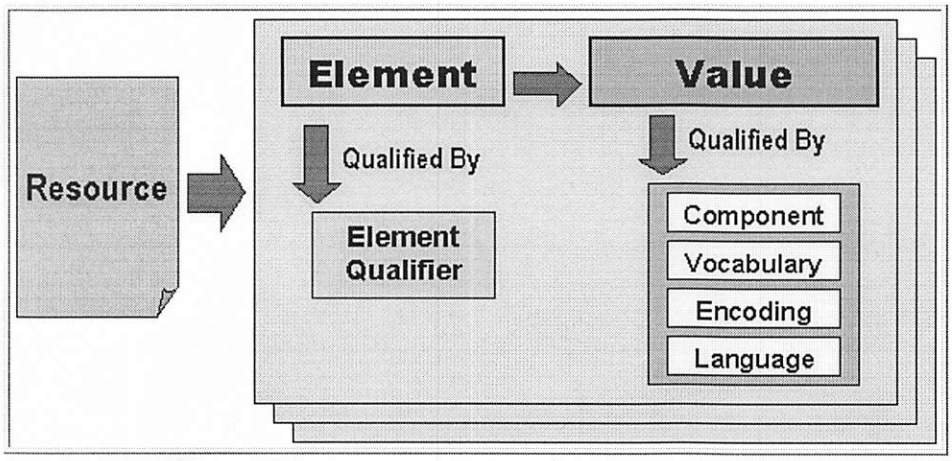
Value Component
: A Value Component represents a whole/part relationship with the Element Value. A Value Component is part of the whole Element Value and represents an attribute of the full value. For example, Firstname and Lastname (parts) are considered two Value Components for the Creator (whole) element value since you can divide a Creators name value into these two parts.

Value Vocabulary
: A Value Vocabulary represents a known vocabulary (or taxonomy, thesaurus, ontology) from where the Element Value was sourced. For example, the Library of Congress Subject Headings (LCSH) is a Value Vocabulary. Values used from Value Vocabularies would be identified as such with the Element Value.

Value Encoding
: A Value Encoding represents the formating rules used for the Element Value. For example, the W3C Datetime NOTE specifies the rules to encode date values to syntactically show the year, month, day etc separated with hyphen characters.. Values that follow particular Value Encodings would be identified as such with the Element Value.

Value Language
: The Value Language represents the written language of the Element Value. For example, "en" indicates that the Value is written in the English language. The recommended vocabulary for Value Language values is defined by [RFC1766]. Note that Value Language is a specialisation of Value Vocabulary.

The figure below graphically represents the Qualified Dublin Core Reference Model. The Resource is being being described by a collection of Elements and their Values. Each Element could be further qualified with an Element Qualifier. Each Value could be further qualified with one of four Value Qualifiers; Component, Vocabulary, Encoding, and Language.



Qualified Dublin Core Reference Model

3.1. Example

Below is an example of an instance of Qualified Dublin Core metadata for the Resource with identity of <http://dublincore.org/doc/REC-DCRM.html>.

Element	Element Qualifier	Value	Value Component	Value Vocabulary	Value Encoding	Value Language
Title		Dublin Core Reference Model				en
		Modello Di Riferimento Di Dublin Core				it
Creator		Renato Iannella	Name			en
		DSTC	Affiliation			en
Creator		Eric Miller	Name			en
		OCLC	Affiliation			en
Creator		Paul Miller	Name			en
		UKOLN	Affiliation			en
Date	Created	1999-12-01			W3CDT	
	Modified	1999-12-10			W3CDT	
Subject		Metadata, Dublin Core, Reference Model	Keywords			en
		Metadata		LCSH		en

3.2. Qualified Conformance

The Qualified Dublin Core Reference Model conformance requirements for interoperable metadata include:

1. The Basic Dublin Core Reference Model conformance requirements must be meet.
2. Must support the semantics for all Version 1 releases of the Dublin Core Qualifiers.
3. Mechanisms to support and identify Element Qualfiers must be addressed.
4. Mechanisms to support and identify the four types of Value Qualifiers must be addressed.
5. Support for RFC1766 for the Value Language qualifier must be supported.
6. Mechanisms to associate Qualifiers with the relevant Element must be supported.
7. Mechanisms to allow Elements (including Qualifiers) to be grouped together must be supported.
8. Mechanisms to allow the ordering of Elements (including Qualifiers) should be addressed.

---

4. Namespace Identifier

The semantics of both the Dublin Core Elements and the Qualfiers must be be cleared expressed in all instances of Dublin Core metadata. A unique identifier will be used for this purpose. The identifier will take the form of a [URI] but no assumptions should be made about this identifier. It should be treated as an atomic identifier.

The Namespace identifier for Version 1.1 of DCES will be:

http://dublincore.org/dces/1.1

and will be updated as new versions of the semantics is released.

The Namespace identifier for Version 1.0 of Dublin Core Qualifiers will be:

http://dublincore.org/dcq/1.0

and will be updated as new versions of the semantics is released.

Additionally, a Namespace Identifier can be used at a greater level of granularity to enable the unique identification of particular elements and qualifiers within the above two sets of semantics.

4.1. Namespace Conformance

The Namespace identifier conformance requirements for interoperable metadata include:

1. The Namespace Identifier referring to the appropriate Element and Qualifier semantics must appear.
2. The Namespace Identifier may be used to uniquely identify and refer to particular Elements and Qualifiers..

---

5. Extensibility

The DCES is built on the principle of extensibility. Other communities can add new Elements and additional Qualifiers that meet their particular needs. In such cases, a new Metadata Element Set is produced that is based on the DCES. The additional semantics (Elements and Qualifiers) should follow the general classifiation of Qualifiers presented for the Qualified Reference Model and should be clearly identified with a new Namespace Identifier.

Additionally, the new semantics must follow the principles outlined in this Reference Model. That is, new Element Qualfiers must specialise the Elements, and new Value Components must form parts of the Element Value. The source of additional Value Vocabularies and Value Encodings must be clearly identified. Additional Vocabulaires for the Value Language qualifier (apart from RFC1766) are permissable as long as the source Vocabulary is identified.

5.1. Extensibility Conformance

The Extensibility conformance requirements for interoperable metadata include:

1. The Namespace Identifier referring to the new Element and Qualifier semantics must appear.
2. The new Element and Qualifiers must be associated with the new Namespace Identifier.
3. New Qualifiers (for DCES Elements) must meet the same semantic principles as for the current DCES qualifiers.

---

6. Conversion Between Basic and Qualified

In a heterogeneous environment such as that in which the Dublin Core operates, it is necessary to offer a means by which perceived over-complexity may be handled. Such a need is manifested in cases where users or software expecting Basic Dublin Core encounter Qualified Dublin Core, as well as in situations where users or software able to handle Qualified Dublin Core encounter terminology from additional Namespaces with which they are unfamiliar, and which they do not wish to resolve through reference to the Namespace declaration itself.

In both cases, a clear and unambiguous mechanism is required by which the acceptable values (literal strings in the first example, and values only from understood Namespaces in the second) may be extracted from the description in such a way that intelligible results are obtained without significant loss of value.

The core principle for conversion is that when all qualifiers, including extensions, are removed from a Qualified Dublin Core metadata record, the resultant Basic Dublin Core metadata record will still makes semantic sense. That is, the information is not misleading. Also, every validly expressed Basic Dublin Core record is considered to be a valid description in Qualified Dublin Core.

6.1. Conversion Conformance

The Conversion conformance requirements for interoperable metadata include:

1. Qualified Dublin Core must be able to be converted to Basic Dublin Core without any negative semantic side-effects.
2. Addtional information must not be added to Basic Dublin Core when being converted to Qualified Dublin Core.
3. Basic and Qualified Dublin Core must be able to ingore extensions, such as additional Elements and Qualifiers, and result in valid Dublin Core records.

---

7. Conformance Requirements

Proposals for syntactical implementations of the Dublin Core Element Set must address the Dublin Core Reference Model functional requirements presented in this document. The proposals must include clear and unambiguous instructions on how the conformance requirements have been addressed. The proposal must address the conformance points outlined in:

- 2.2 Basic Conformance

6 pages

Abstracts of 1500 words — papers + tutorials + workshops

500 — posters + demos 2 pages