Title: Usage Board meeting in Seoul - Agenda
Identifier: http://dublincore.org:8080/usage/meetings/2009/10/seoul/.index.html
Created: 2009-09-06

Expected
Seoul: Tom Baker, Andrew Wilson, Stefanie Ruehle, Akira Miyazawa, Joe Tennis
Remote: Julie Allinson, Pete Johnston

Administrative Components (Andrew)
http://dublincore.org:8080/usage/meetings/2009/10/seoul/acore.pdf
-- www.bs.dk/standards/AdministrativeComponents.htm
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/acore-email-digest.html

-- Task: Prepare a short review of ACore for discussion in Seoul. What can or should the Usage Board say about ACore? The review should discuss the possibility of defining ACore as an application profile.

Errata and other changes to DCMI terms documentation (Akira)
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/errata.html
This document includes some proposals to be decided formally at the meeting (and documents some decisions already and in the queue for publication). After the Seoul meeting, this will be turned into a single decision document.

http://dublincore.org/usage/minutes/2008/2008-09-21.berlin-5Etc.html
The meeting notes for Berlin record the discussion of the literal range for Title and Alternative title.

-- Task: Carefully check all of the proposed changes. Write a very short summary of the rationale for assigning a literal range to Title and Alternative title (using the meeting notes above.

Usage issues (Pete)
-- Creator and Maker
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/dccreator.html
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/foaf-maker.html

-- Candidate issues for further discussion
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/dcidentifier.html
http://dublincore.org/documents/usageguide/appendix_roles.shtml
"Using Dublin Core Part 6: Using Agent Roles"

Simple Dublin Core
http://dublincore.org:8080/usage/meetings/2009/10/seoul/simpledc-guidelines.pdf
-- www.intute.ac.uk/publications/eprints-uk/simpledc-guidelines.html
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/simpledc.html

-- Task: Follow dc-architecture discussion regarding DC creator and FOAF maker and summarize discussion to dc-usage by Friday, 2 October. Prepare bullet points for discussing other issues (see above) in order to identify topics for further discussion in the Usage Board. Propose a one-paragraph glossary entry for "Simple Dublin Core" for discussion. Could Agent Roles be discussed in a short glossary entry?

Using Dublin Core - Elements and Qualifiers (Stefanie)
http://dublincore.org/documents/usageguide/elements.shtml
"Using Dublin Core Part 4: The Elements"
http://dublincore.org/documents/usageguide/qualifiers.shtml
"Using Dublin Core Part 5: The Qualifiers"
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/usingdc.html

-- Task: In Seoul, we will not have time to discuss all of the examples for elements and qualifiers in Using Dublin Core. Rather, the task here is to prepare bullet points on general issues related to the examples, for example the issue of literal versus non-literal values. Prepare to discuss possible ways forward for Using Dublin Core: what would need to be done in order to bring the document up to date? Is a document at this level needed, and what is its audience?

Glossary (Joe)

http://dublincore.org/documents/usageguide/glossary.shtml

"Using Dublin Core - Part 7: DCMI Glossary"

http://dublincore.org:8080/beta/glossary/

Tom proposes that the glossary be written as a relatively short document covering only major concepts that are characteristic of Dublin Core metadata. The glossary is potentially a good place to explain legacy issues, such as "dumb down" and "document-like object".

Candidate issues for inclusion (or summary) in a glossary:

http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/dumbdown.html

http://dublincore.org:8080/usage/meetings/2009/10/seoul/DumbDownNotes.htm

http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/dlo.html

http://dublincore.org:8080/usage/meetings/2009/10/seoul/IssuesWithCoverage.htm

-- Task: Identify which of the terms in the old glossary are most important for describing the "Dublin Core style" of metadata. For those most important terms, suggest bullet points for what a modern glossary entry should say. Flag any obvious gaps -- DC terminology that would need to be covered in a glossary, such as the "candidate issues" above (and suggest bullet points).

Frequently Asked Questions (Julie)

-- Legacy FAQ
http://dublincore.org/resources/faq/

-- DCMI Mixing and Matching FAQ (Andy Powell, 2005)
http://www.ukoln.ac.uk/metadata/dcmi/mixing-matching-faq/

-- Candidate FAQ
http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/range.html

-- Task: Identify which questions are still really frequently asked. For those questions, prepare bullet points proposing current answers. Propose an answer to the frequently asked question about the difference between dc:creator and dcterms:creator.

Introduction to metadata (Tom?)

http://dublincore.org:8080/usage/meetings/2009/10/seoul/dublincore-org-beta-pages.pdf

dublincore.org/documents/usageguide/index.shtml
"Using Dublin Core" Parts 1-3: Introduction to metadata

dublincore.org:8080/beta/metadata-basics/

dublincore.org:8080/beta/specifications/
Tom's proposed homepage text covering some of these issues.

dublincore.org:8080/usage/meetings/2009/10/seoul/.html/general.html
Karen Coyle comments.

dublincore.org:8080/usage/meetings/2009/10/seoul/.html/levels.html
Comments on Singapore Framework and interoperability level issues.

dublincore.org/documents/interoperability-levels/
Interoperability Levels

-- Task: Tom would like to lead a discussion on various texts he has drafted about DCMI basic concepts.

Old actions

http://dublincore.org:8080/usage/meetings/2009/10/seoul/.html/oldactions.html

AC - Administrative Components

Dublin Core DCMI Administrative Metadata

Creators:
Jytte Hansen, Danish Bibliographic Centre - E-mail: jyh@dbc.dk
Leif Andresen, Danish National Library Authority - E-mail: lea@bs.dk

Dublin Core Metadata Initiative Administrative Metadata Working Group - final version 28 October 2003.

Introduction

The goal of AC - Administrative Components is to provide a practical tool for users of metadata to manage metadata with special focus on interoperability between different systems. This indicates the limitations of this metadata set: it is for administration of metadata - not for administration of resources.

AC - Administrative Components is developed within the framework of Dublin Core Metadata Initiative Administrative Metadata Working Group 2001-2003.

Relation to other initiatives:

The focus for AC - Administrative Components metadata set is interoperability between systems with content metadata. Other initiatives have others focuses. The word "administrative" is in METS (Metadata Encoding & Transmission Standard) used for management of digital library objects. Other related functions are record keeping, resource management and preservation metadata.

OAI-PMH (The Open Archives Initiative Protocol for Metadata Harvesting) includes some metadata elements for administration. For users of OAI-PMH information such as identifier and date will be handled by OAI-PMH and not AC.

None of these initiatives meet all the requirements in relation to control interchange of metadata records between systems. This is the reason for the present specification.

This document sets out a three-part proposal for a standard for administrative metadata:

- Metadata for the entire record
- Metadata for update and change
- Metadata for batch interchange of records

One informative annex complements it.

January 2006 updated with new element “Source” and a link to an XML schema added in the end.

AC - Administrative Components

Dublin Core DCMI Administrative Metadata

Final version JULY 2003

Specification updated 27 January 2006.

General rules

The name of this metadata element set is Administrative Components.

The DCMI namespace URI is http://purl.org/ac/

All metadata elements are optional and repeatable. A specific implementation can define additional rules including that some elements are mandatory.

None of the elements are specified as mandatory for all kinds of use of AC. It is up to the individual project, organisation, web-site etc. to decide which elements are to be mandatory. A tool for that can be an Application Profile to specify instructions for use of Dublin Core, domain specific metadata element set(s) and metadata about management of the content metadata - including e.g. mandatory AC elements for specific use.
The idea is that the different projects, organisations, institutions etc. shall pick up the elements they can use.

Definitions

Metadata for the entire record

Name: **identifier**
Label: Identifier
Definition: A string or a number, which identifies the metadata record
Comment: Can be the internal number in a database.

Name: **source**
Label: Source
Definition: A string or a number, which identifies recording entity
Comment: Can be a library code or an acronym for museum, archive etc..

Name: **scope**
Label: Scope
Definition: Declaration of the scope of application
Comment: Will often be declared by means of a separate form. This element can be used either as unstructured text just with an informal declaration like "national bibliography" or by using a SCHEME for a formal declaration like "catalogue code".

Name: **comment**
Label: Comment

Definition: Comment on the Administrative Component metadata
Comment: E.g. comments pointing at special circumstances in connection with the transmitted metadata

Name: **location**
Label: Metadata Location
Definition: An unambiguous reference to the content metadata within a given context
Comment: This element is only used if the content metadata and administrative metadata are not in the same location.
Recommended best practice is to identify the content metadata by means of a string or number conforming to a formal identification system. Examples of formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)) and the Digital Object Identifier (DOI). Other identifiers, such as local repository/database keys, may be used.

Name: **language**
Label: Language
Definition: Language of metadata
Comment: Encoded ISO 8601, ISO 639-2.

Name: **rights**
Label: Rights Ownership
Definition: Information about rights held in and over the content metadata
Comment: Typically, the Rights element will contain a rights management statement for the content metadata, or refer to a service providing such information. For at more detailed control specific projects can use SCHEMEs for administration of rights for the specific project.

Name: **dateRange**
Label: Valid Date Range
Definition: The start and/or end date of the validity of the content metadata
Comment: Content metadata accessed outside the date ranges should be considered to be invalid.
Encoded to the W3C Profile of ISO 8601 including the use of the "/" to indicate the range scope. For example, "/1999-12-31" indicates validity up to 31 December 1999, "1999-01-01/" indicates validity from 1 January 1999 onwards, and "1999-01-01/1999-12-31 indicates validity between the two specified dates.

Name: **handling**
Label: Handling specification
Definition: Instructions for handling the administrative metadata and the metadata record in full. To this element is attached a SCHEME with the values:
· Harvest: the record shall be included in a harvesting
· Public: the content metadata must be shown to the public
· Manual: the metadata record must be checked automatically
· Keep: when adding administrative metadata, shall old versions of same element be kept
· Mail: Mail to be sent
Comment: This element defines instructions of future actions. (See also the element: Action)

Metadata for update and change

Name: **activity**
Label: Activity

Definition: This element reflects an action performed on the content metadata
Comment: The element functions as a container, which connects an action (see below) with the one responsible for its accomplishment, the date on wich the activity took place, etc.

Refinements

Name: **action**
Label: Action
Definition: The action performed on the content metadata by the responsible entity
Comment: The actions are taken from a non-exhaustive list including:
created, submitted, modified, checked, link collected, resource harvested, expired, mail sent and three codes for deleted: delete_error_record, delete_disappearance and delete_out_of_scope
This list shows the history of actions. (See also the element: Handling).
Other sources may be used for the action values such as codes from the USMARC Relator List.
To this element is attached a SCHEME *TypeOfActivity* with the values:
· created
· submitted
· modified
· checked
· link collected
· resource harvested
· expired
· mail sent
· delete_error_record
· delete_disappearance
· delete_out_of_scope

Name: **name**
Label: Name
Definition: The name of the entity responsible for undertaking a defined action on the content metadata
Comment: Examples of Name include a person, an organisation, or a service.
Where the person has an affiliation with an organisation, this information may be included.
The name of a person should be provided in reverse order, that is, last name before first name, with a comma separator.

Name: **email**
Label: Email Address
Definition: Electronic Mail address for the responsible entity
Comment: The email address must be encoded to be consistent with Internet Address standard RFC822.

Name: **contact**
Label: Contact Information
Definition: Information on how to contact the responsible entity
Comment: The information should be one or more of: a street or postal address, a telephone number, a facsimile number, an Internet address, or other forms of physical or electronic contact information.
Links to full descriptions of the responsible entity may also be included, such as name registries.

Name: **date**
Label: Date
Definition: The date on which the activity took place

Comment: Encoded to the W3C Profile of ISO 8601.
This unspecified date must be used in connection with an action, e.g. "submitted"

Name: **affiliation**
Label: Affiliation
Definition: The organization with which the named person was associated when involved with the resource
Comment: Often the "affiliation institution" will be the formally responsible entity

Metadata for batch interchange of records

A number of elements relevant in connection with data exchange via batch files

Name: **database**
Label: Database
Definition: Code identifying a database
Comment: The code is used to identify the database to which a batch file is sent.
Is related to Metadata Location.

Name: **transmitter**
Label: Transmitter
Definition: Name or code for transmitter
Comment: The name/code (e.g. a library number) will be used to identify an organization with which formal routines of data exchange are established.
A code may include the type of transmitter (e.g. public library, research library, publisher)

Name: **filename**
Label: Filename
Definition: Name of a batch file
Comment: Name of the individual batch file. It may be combined with transmitter name.

Name: **technicalFormat**
Label: Technical format
Definition: Technical data exchange format
Comment: The format is taken from a non-exhaustive list including:
ISO2709, XML, HTML

Name: **characterSet**
Label: Character set
Definition: Name of character set used
Comment: The character sets must refer to relevant standards

Name: **bibliographicFormat**
Label: Bibliographic format
Definition: Bibliographic format for data exchange
Comment: The format are taken from a non-exhaustive list including:
MARC21, danMARC2, DC

Name: **resultFile**
Label: Address of result file
Definition: Localization of result file

2009-01-12 From Martin Leese < DELETED > to dc-general

Sorry if this has been discussed before, but I couldn't
find it in the archives. What happened to the
Administrative Dublin Core (A-Core)?

I have a personal website, and implemented DC on it
in 2001. Revamping the metadata has finally reached
the top of my to-do list. Looking on the DC website,
in the intervening seven years there have been a
number of changes. The biggest, for my simple use,
is the absence of A-Core. Can I still use this?

I have been using the following elements from it:=20
AC.activity (created)
AC.name
AC.email

The other big change, from my point of view, is
Qualified DC. I can see that this will be most useful.

2009-01-12 From David Bromage <david.bromage@NAA.GOV.AU>

A-Core never really took off and in comparison with other standards is
of limited use. AGLS looked at this issue last year and decided to adapt
the administrative metadata in ISO 19115 (Geographic information -
Metadata). The terms are being implemented in a namespace called
adminterms. I can post a reference description if there is any interest.

David Bromage
Policy and Strategic Projects
Government Information Management Branch

National Archives of Australia
PO Box 7425
Canberra Business Centre ACT 2610

2009-01-12 From Douglas Campbell <Douglas.Campbell@NATLIB.GOVT.NZ>

That would be great.

I don't have ISO 19115 to hand, but from what I can see it has:
- file id for the metadata record
- language used in metadata record
- character set used in metadata record
- parent metadata record id
- contact details for creator of metadata record
- date metadata record was last updated
- schema (and version) used in the metadata record (eg. MARC, ISO 19115)
All the other elements in ISO 19115 seem to refer to the described =
resource rather than the metadata record itself.

Features from Admin Core [1] missing include:
- location (eg. URI) of metadata record
- rights ownership over the metadata record
- handling specification
- activities performed on the metadata record
- source database

We still use Admin Core, especially ac:rights, under the recommended =
namespace http://purl.org/ac/ [2]

Douglas Campbell
National Library of New Zealand

[1] http://dublincore.org/groups/admin/
[2] http://www.bs.dk/standards/AdministrativeComponents.htm

2009-01-12 From David Bromage <david.bromage@NAA.GOV.AU> to DC-GENERAL@JISCMAIL.AC.UK

This is the adminterms reference description. The schema has not yet been written so the property URIs are not yet active. RFC 2119 keyword definitions apply.

Term Name: fileIdentifier
URI: http://www.agls.gov.au/agls/adminterms/fileIdentifier
Label: Metadata File Identifier
Definition: Unique identifier for the metadata record.
Comment: The fileIdentifier for a metadata record must never change, irrespective of where that metadata record is stored. This property should be system generated. The metadata content creator should not be required to record any information against this property. Metadata creation systems must assign a unique identifier, expressed as a UUID and encoded as a URI, as the value.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
Has Range: http://www.w3.org/2000/01/rdf-schema#Literal

Term Name: metadataLanguage
URI: http://www.agls.gov.au/agls/adminterms/metadataLanguage
Label: Metadata Language
Definition: The written language used for completing the metadata record. This property does not describe the language used within the resource itself.
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

Term Name: metadataCharacterSet
URI: http://www.agls.gov.au/agls/adminterms/metadataCharacterSet
Label: Metadata Character Set
Definition: The code for the character set used in the metadata record. This property does not describe the character set used within the resource itself.
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

Term Name: metadataContact
URI: http://www.agls.gov.au/agls/adminterms/metadataContact
Label: Metadata Contact
Definition: Details about the individual, organisation and/or position associated with the metadata information. This property does not convey details about the individual, organisation and/or position associated with the resource itself
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
Has Range: http://www.w3.org/2000/01/rdf-schema#Literal

Term Name: dateStamp
URI: http://www.agls.gov.au/agls/adminterms/dateStamp
Label: Metadata Date Stamp
Definition: The date (and optionally time) that the metadata record was created, and not the date that the resource was created. It is not the date the metadata was last updated.
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

Term Name: metadataUpdateDate
URI:
http://www.agls.gov.au/agls/adminterms/metadataUpdateDate
Label: Metadata Update Date
Definition: The date (and optionally time) that the metadata was last updated or modified.
Comment: This property should be completed automatically by

metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

Term Name: metadataStandardName
URI: http://www.agls.gov.au/agls/adminterms/metadataStandardName
Label: Metadata Standard Name
Definition: The metadata standard followed for creation of the metadata.
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

Term Name: metadataStandardVersion
URI: http://www.agls.gov.au/agls/adminterms/metadataStandardVersion
Label: Metadata Standard Version
Definition: The version of the metadata standard followed for creation of the metadata.
Comment: This property should be completed automatically by metadata creation systems. The metadata content creator is not required to record any information against this property.
Type of Term: http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

David Bromage
Policy and Strategic Projects
Government Information Management Branch

National Archives of Australia
PO Box 7425
Canberra Business Centre ACT 2610

2009-01-13

They are basically describing the metadata record as a resource in itself, so they could very well use dcterms to do that.

fileIdentifier = dcterms:identifier
metadataLanguage = dcterms:language (but really unnecessary if xml:lang is used with string values)
metadataContact = dcterms:contributor
dateStamp = dcterms:created
metadataUpdateDate = dcterms:modified
metadataStandardName = dcterms:conformsTo
metadataCharacterSet = ?

The others are just duplications of dcterms properties /about/ a metadata record.

Issue: When one does not want to give an identifier to a metadata and wants to ship metametadata and metadata in the same XML record.

2009-02-05 From Mikael Nilsson <mikael@NILSSON.NAME>

A similar situation arises in LOM-DCAM, and an approach very like the one you propose is used there now. The same properties are used in distinct descriptions, and all is well.

> fileIdentifier = dcterms:identifier
> metadataLanguage = dcterms:language (but really unnecessary if
> xml:lang is used with string values)
> metadataContact = dcterms:contributor

or even publisher?

> dateStamp = dcterms:created
> metadataUpdateDate = dcterms:modified
> metadataStandardName = dcterms:conformsTo
> metadataCharacterSet = ?

most containers (such as XML) will solve that problem anyway - so mostly unnecessary.

> not wanting to ship metametadata and metadata in the same XML record.

I don't think there's an issue - if it's an XML language they design themselves:

1. There's no need to give an identifier for the metadata record in order to separate the two descriptions. One can just do

```
<metametadata>
  <dcterms:whatever>....
</metametadata>
<metadata>
  <dcterms:whatever>...
<metadata>
```

2. If they don't even want to create two separate containers, they can just use the dcterms definitions, but come up with their own XML element names and separate into records based on XML names. GRRDL can do that.

```
<metadata>
  <creator> dfgsdfg <creator> <!-- creator of the resource -->
  <meta-creator> sdfsdf </meta-creator> <!-- creator of the metadata -->
</metadata>

---> GRDDL --->

_:resource <dcterms:creator> _:creator .
_:creator <rdf:value> "dfgsdfg" .


_:record <dcterms:creator> _:mcreator .
_:mcreator <rdf:value> "sdfsdf" .
```

This still results in level 2 interop, so all is well. There are many other ways to accomplish this.

Errata and corrections

Creator: Tom Baker
Date: 2009-09-06
Description: This document describes various changes to DCMI terms documentation. Some of the changes described have been decided but not yet implemented because it takes a significant amount of careful work to prepare a new version of the term documentation. In Seoul, we should take a formal decision on all of these changes. Tom will then turn this into a decision document to which each of the changed term descriptions can point.

==========
DCMI Metadata Terms

2009-09-06 Proposal to be formally decided in Seoul

ACTION 2009-04-22: Tom to turn Pete's proposal at
https://www.jiscmail.ac.uk/cgi-bin/webadmin?A2=ind0903&L=DC-USAGE&P=982
into a decision document for finalization on next telecon.

Proposal:
=========

(i) For the property dcterms:contributor, delete the following sentence from the "Comment"

Typically, the name of a Contributor should be used to indicate the entity.

(ii) For the property dcterms:creator, delete the following sentence from the "Comment"

Typically, the name of a Creator should be used to indicate the entity.

(iii) For the property dcterms:publisher, delete the following sentence from the "Comment"

Typically, the name of a Publisher should be used to indicate the entity.

Note that no change is proposed for the descriptions of the properties dc:contributor, dc:creator, dc:publisher

Rationale:
==========

Issue 1: Range of Properties

The three properties above were defined with a range of the class dcterms:Agent - this is the characteristic which distinguishes the properties from their dc:* counterparts.

i.e. the intent is that, from the perspective of the DCAM, they are used with non-literal values, rather than literal values, or, from the perspective of the RDF model, when one of these properties is referred to in an RDF predicate the object should be a URI or blank node, but not a literal. i.e. the intent is that the properties are to be used in constructs of the form

(DCAM-1)

```
Description (
  Statement (
    PropertyURI (dcterms:creator )
    ValueString ( "Jim Jarmusch" )
  )
)
```

(RDF-Turtle-1)

```
_:filmF dcterms:creator _:personP .
_:personP rdf:value "Jim Jarmusch" .
```

However, the presence of the comment above introduces a contradiction, or at least an element of ambiguity, because the phrase "the name ... should be used to indicate the entity" might be read as encouraging the use of the forms:

(DCAM-2)

```
Description (
  Statement (
    PropertyURI (dcterms:creator )
    LiteralValueString ( "Jim Jarmusch" )
  )
)
```

(RDF-Turtle-2)

```
_:filmF dcterms:creator "Jim Jarmusch" .
```

These two constructs (DCAM-2, RDF-Turtle-2) are inconsistent with the range of the property. One of the primary motivations for creating these three dcterms:* properties - apart from the "why can't we just use one namespace?" issue - was to try to achieve a consistent usage of these properties, i.e. to declare/describe the properties in such a way that implementers would use the forms DCAM-1 and RDF-Turtle-1, and the forms DCAM-2 and RDF-Turtle-2 were not used.

In the DCAM case, it might be argued that a DCAM implementer should read the phrase "the name ... should be used to indicate the entity" together with the assertion of a non-literal range as implying that the usage (DCAM-1), rather than (DCAM-2), is required, and on this basis, it might be argued that there is no contradiction.

But in the RDF case, to an RDF implementer, the phrase "the name ... should be used to indicate the entity" strongly implies the use of a literal object. It is not clear how an RDF implementer can reconcile this with the non-literal range, and the contradiction is rather more stark. See [1] for an example of this confusion.

Issue 2: A Name is not Required

It is not a requirement that a name is provided when using these properties: it is quite acceptable when using these properties to provide only a "value URI"/URI as object, or to describe the value/use a blank node, and provide various properties of the agent (email address, home page, whatever), without providing a name of the agent as a value string.

Issue 3: Consistency with other DCMI term descriptions

The form of words "the name ... should be used to indicate the entity" is not used with the dcterms:rightsHolder property, which was defined after the original dc:* properties and before the three dcterms:* properties discussed here.

In conclusion, deleting the sentence would

(i) take a step towards removing the ambiguity; it doesn't stop people ignoring the range and using literal objects, but it avoids making the implication that this is an option;
(ii) remove the requirement that a name "should" be provided;
(iii) make the definitions of these three properties consistent with that of dcterms:rightsHolder
(iv) have no effect on the definitions of the dc:* properties

---

2009-09-21 Range for dcterms:title (already approved!)

http://dublincore.org/usage/minutes/2008/2008-12-08.berlin-summary-of-actions-posted.html

APPROVED: that the DCTERMS properties Title and Alternative be assiged a range of RDFS literal.

2009-04-21

Two places that link to ISO 15836:

-- http://dublincore.org/documents/dces/
reference to 15836-2003
http://dublincore.org/documents/dces/#ISO15836 is a search
in the ISO catalogue that brings up the 2009 version.

-- http://dublincore.org/documents/dcmi-terms/
reference to 15836-21003 (sic!) that also
points to the 2009 version in the ISO catalogue.

Note: "ISO 15836:2009" (now the correct way for indicating
the year, with a semicolon).

2009-05-12 Simon Grant <asimong@GMAIL.COM> to dc-general
Subject: is this a small error in documentation?

Take a look at

http://dublincore.org/usage/terms/history/#replaces-002

it says

"Is Replaced By: Is-Replaced-By-003"

perhaps it should read

"Is Replaced By: replaces-003" ?

(to be sure, an easy mistake to make, getting confused by
what is replacing what in the definitions of replacing...
:-))

Tom replies:

Yes indeed. That file is generated by a script, and the mistake
was in the source, which I have now corrected. When the next
snapshot of DCMI Metadata Terms is published, this will appear
correctly.

2009-05-19 Pete

Not so long ago we added an "index" to the DCMI Metadata Terms document

http://dublincore.org/documents/dcmi-terms/

i.e. so that there were "ready reference" lists of hyperlinks at the
head of the doc to each individual term description.

I just noticed today that we do it for

Properties in the /terms/ namespace -> entries in Section 2
Properties in the legacy /elements/1.1/ namespace -> entries in Section 3
Vocabulary Encoding Schemes -> entries in Section 4
Syntax Encoding Schemes -> entries in Section 5
Classes -> entries in Section 6

But we don't have an index for the DCMI Type Vocabulary
classes or the Terms Related to the DCMI Abstract Model,
which seems a bit inconsistent.

I think we should probably add these to the index?

(If it involves changing the XSLT, I can probably do that if necessary)

2009-08-24 Jacco van Ossenbruggen to dc-general
Subject: missing definition of dcterms:Collection?

I noticed that in

http://dublincore.org/documents/dcmi-terms/#terms-accrualMethod
accrualMethod and other properties have a domain defined to be

http://purl.org/dc/terms/Collection

However, this URI seems not to be defined anywhere, not in the RDF Schema nor in the English text.

In contrast,

http://purl.org/dc/dcmitype/Collection

has been properly defined, but is in a different namespace.

Is this an oversight or am I missing something?

Tom to Jacco:

Thank you for pointing this out. That is indeed an error and we will fix it in the next build of the documentation. The correct URI is http://purl.org/dc/dcmitype/Collection.

2008-10-15 Note received by Website feedback

Please note this comment. I think this should point to http://www.iso.org/iso/country_codes/iso_3166_code_lists/english_country_names_and_code_elements.htm instead.

> I was recently using your website as a reference for a record when I noticed that one of the links is incorrect. I've included a copy of the section below for you to verify. The section regards the Element Refinement, Spatial, for the Coverage section. It appears that the reference link to the ISO 3166 standard directs us to a DIN standards site instead.

Section in question-

ISO 3166

Name: ISO3166
Label: ISO 3166
Definition: ISO 3166 Codes for the representation of names of countries
See also:
http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1

==========
RDF schemas

2008-09-21 Usage Board action for Tom to correct...
ACTION 2008-09-21: Tom to correct RDF schemas of DCMI Metadata Terms to use blank node with publisher.

2009-07-23 Pete suggestion to add to RDF schema for http://purl.org/dc/terms/

In the description of the DCMI Type Vocabulary VES in the "namespace document", we provide a rdfs:seeAlso link to

http://dublincore.org/documents/dcmi-type-vocabulary

which is a human-readable document listing the individual members of the VES.

But we don't provide a link to

http://purl.org/dc/dcmitype

which is the machine-readable equivalent.

I think it might be useful to do so?

==========

Guidelines for Dublin Core Application Profiles
http://dublincore.org/documents/profile-guidelines/

2009-07-23 Pete
Subject: Re: ISO639-2 in SKOS (Was: More on lists)
To: DC-USAGE@JISCMAIL.AC.UK

On a related note, I see that in the "Guidelines for Dublin Core Application Profiles"

http://dublincore.org/documents/2009/05/18/profile-guidelines/

In the text in section 5, dcterms:ISO639-2 is (correctly) referred to as a SES, but in the XML version of the DSP in Appendix B, firstly the reference is to dcterms:ISO639-3, and secondly it is referred to as a VES.

Also I think in that example, each Description Template should have a Resource Class constraint, for the classes Book and Person respectively. As it stands at the moment (without those constraints), I think a description of a resource of any type will be bound to both templates, which fails the requirement in DSP section 3 that a description should be bound to exactly one DT. We should make this clearer in section 5 too, I think.

Anyway, just another couple of things we should fix when cleaning up the documentation! :-)

Formal definition of dcterms:title should now read "rdfs:range rdfs:Literal".
-- roll this into one decision document

Identifier: http://dublincore.org/usage/minutes/2008/2008-09-21.berlin-5Etc.html
Description: This is one part of the minutes for the Usage Board meeting of 20-21 September 2009. See http://dublincore.org/usage/minutes/.

Unfinished Business: Range for DCTERMS Title

Range of dcterms:title (Tom) - formal vote to complete unfinished business -- http://dublincore.org/usage/meetings/2008/09/berlin/terms-titlerange/

Tom proposes that we assign a range RDFS literal to DCTERMS title, and ask for a vote.

Pete: the proposal is option three in the meeting packet (page 77).

Stefanie: has a problem of word sequences (or something like that), we do not solve the problem when we only change DCTERMS title.

Andrew: why not a non-literal?

Tom: Because the implementer community sees it as having a literal range.

Andrew: How do I describe series using DC? If it's a literal I can't link that series title because of contextual information.

Pete: It's the thing that has the title, not the title that's a thing.

Andrew: But i want to link that title .

Diane: Model it differently.

Stefanie: We have a problem with uniform title.

Diane: Do it through relation, don't do it through title.

Andrew: Relation doesn't work. Why can't DC let me link to title?

Diane: You're having one-to-one problem.

Andrew: A series is an organic whole, and it has a title, and we want to expose that, we want to link from DC title to series title. If I want to put a description of series on the web page, why can't I link?

Pete: What is being described? We have a series that has a title, and?

Andrew: We have a database that has information about items and series and creating agencies, and I want to link ...

Pete: You're confusing metadata record and hyperlinks.

Andrew: I am, you're right!

Stefanie: We want a record of the title, we want a transliteration of a title.

Diane: I think you're looking for a richer schema, which DC isn't. It doesn't make sense in a DC context.

Stefanie: It doesn't make sense in a DC context to say that we have a solution for title and nothing else.

Diane: You want to create an authority record for the title. You should look at RDA, because that's the mindset they're working on it.

Stefanie: Akira wants a solution, and does this solve the problem?

Diane: Tom has established the appropriate boundary (in my opinion).

Akira: One question. The issue is only for the title? Yes. The description is a separate issue. My colleague says that title or name of something is still not literal (is my opinion), but as an approximation of modeling to see a name as a literal makes things very simple.

Tom: There's nothing precluding creating a property of "name" with range non-literal.

Akira: name is like URI in natural language, so it should be unique, and thus it should be literal, but historically and for many reasons it is not really unique, but still many people believe a name is unique, thus, the approximation of literal works in most cases, and simplicity we gain from modeling it this way is great.

VOTE that the property DCTERMS Title be assiged a range of RDFS literal (ramification: we need to change Alternative Title as well).

    In favor: 8
    Opposed: 0
    Abstain: 0

Vote that the property DCTERMS Alternative Title be assigned a range of RDFS literal:

    In favor: 8
    Opposed: 0
    Abstain: 0

After lunch, write a description/definition of literal non-literal.

Other Actions
==========

Tom: there are little errors and fixes in documentations (pg 79 of meeting packet). We need to take a look and see if these still need to be done.

1. Action from last november - Joe and Andrew for discussion at a future teleconference. Is this something we want to keep on our active list? Yes, keep working on it.

ACTION: Joe and Andrew to continue work on Coverage

2. Action on Mikael and Tom to create AP making everything literals, documenting...

ACTION: Tom (and Mikael) to continue work on this

3. functional requirements of agents against FOAF

Was on a critical path for hosting FOAF, so the question is whether we want to take this on. Wo questions: should this be done and should it be a Usage Board issue?

Andrew: we thought that putting it on the UB agenda it would get done, but may not be appropriate.

Diane: maybe it's ok.

Andrew: there is still an Agents Community that can pursue this.

Diane: there is some important stuff that needs to be done before DC can decide whether we can do it.

Tom: we are not going to take it on. Dan and colleagues are still working on FOAF, still tweaking in various ways. In the end, it is owned by two private individuals. They are looking for trusted context and long-term preservation of the namespace. How can they shore up the long-term credibility?

2009-09-06

Hi Dan,

On June 19, Tom wrote to dc-usage:
> By popular request, Dan wants FOAF to say that foaf:maker is a
> sub-property of dcterms:creator and asks whether DCMI could make
> a reciprocal claim. I invited him to submit a short proposal
> describing how the properties are defined, with a proposed
> mapping claim. We could discuss this and decide at the meeting
> in Seoul.

I am writing to progress the idea of reciprocal mapping claims relating foaf:maker to dcterms:creator (see email digest below).

We have an opportunity to discuss this at the Usage Board meeting in Seoul on Friday, 16 October. I would need a few sentences proposing the mapping claim and any other proposed changes to DCMI term documentation and schemas, such as a seeAlso by a week from now -- Monday, 14 September -- at the latest. An email message is enough. Our point of reference will be the two definitions cited below.

In Seoul, we will also consider a proposal to DROP the second sentence of the usage comment for dcterms:creator ("Typically, the name of a Creator should be used to indicate the entity."), leaving the comment in place for dc:creator.

This might also be a good time to discuss the recommendation made in the FOAF specification, that "FOAF descriptions are encouraged to use dc:creator only for simple textual names". Is this still what we want to encourage?

I am posting this on dc-architecture in order to open the discussion beyond dc-usage. I believe it was Bernard Vatant, who recently joined dc-architecture, that originally raised this topic on public-lod@w3.org.

Tom

---

http://dublincore.org/documents/dcmi-terms/#terms-creator

Definition: An entity primarily responsible for making the resource.
Comment: Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.

http://xmlns.com/foaf/spec/#term_maker

maker - An agent that made this thing.
Status: stable
Domain: http://www.w3.org/2002/07/owl#Thing
Range: foaf:Agent

The foaf:maker property relates something to a foaf:Agent that foaf:made it. As such it is an inverse of the foaf:made property.

The foaf:name (or other rdfs:label) of the foaf:maker of something can be described as the dc:creator of that thing.

For example, if the thing named by the URI http://rdfweb.org/people/danbri/ has a foaf:maker that is a foaf:Person whose foaf:name is 'Dan Brickley', we can conclude that http://rdfweb.org/people/danbri/ has a dc:creator of 'Dan Brickley'.

FOAF descriptions are encouraged to use dc:creator only for simple textual names, and to use foaf:maker to indicate creators, rather than risk confusing creators with their names. This follows most Dublin Core usage. See

UsingDublinCoreCreator for details.

==========

Digest of related email

2009-06-18 From: Dan Brickley <danbri@danbri.org>
To: Bernard Vatant <bernard.vatant@mondeca.com>
CC: public-lod@w3.org, Thomas Baker <tbaker@tbaker.de>
Subject: Re: Common Tag, FOAF and Dublin Core Re: Common Tag - semantic tagging convention

On 18/6/09 13:31, Bernard Vatant wrote:
>>>> ... why not use simply dc:creator and dc:date to this effect?
>>>
>>> Right. dc:date would seem a good choice, though I reckon foaf:maker
>>> might be a better option than dc:creator as the object is a resource
>>> (a foaf:Agent) rather than a literal. While it's likely to mean an
>>> extra node in many current scenarios, it offers significantly more
>>> prospect for linking data (and less ambiguity).
>>
>> dcterms:creator would also allow for use of a resource. Bibliontology
>> uses dcterms over dc.
> Well I actually meant dcterms:creator when I wrote dc:creator, sorry. So
> you can link your personal tags to your foaf profile, for example.
> And it's consistent even for tag:AutoTag, since the range of
> dcterms:creator is dcterms:Agent, including person, organisation and
> software agent as well.
> Unless I miss some sublte distinguo dcterms:Agent is equivalent to
> foaf:Agent, and dcterms:creator equivalent to foaf:maker. BTW, with due
> respect to danbri, I wish FOAF would be revised to align whenever
> possible on dcterms vocabulary, now that it has clean declarations of
> classes, domains and ranges ...
> http://dublincore.org/documents/dcmi-terms is worth (re)visiting :-)

Completely agree. I'm very happy with the direction of DC terms. The
foaf:maker property was essential for a while, until DC was cleaned up.
I'll mark it as a sub-property of dcterms:creator. I hope we'll get
reciprocal claims into the Dublin Core RDF files some day too...

Copying Tom Baker here. Tom - what would the best process be for adding
in mapping claims to the DC Terms RDF? Maybe we could draft some RDF,
put it onto dublincore.org elsewhere, and for now add a seeAlso from the
namespace RDF?

2009-06-18 From: Danny Ayers <danny.ayers@gmail.com>

+1
(I keep forgetting the excellent DC makeover)

2009-06-18 from Tom
Cc: Bernard Vatant <bernard.vatant@mondeca.com>, public-lod@w3.org

If you could write up a short proposal -- how the properties are
defined, with a proposed mapping claim -- we could discuss this
in the DCMI Usage Board and take a decision. We associate
changes in the namespace RDF (and related namespace
documentation) with formal decisions so would need to follow a
process.

2009-06-18 From: Dan Brickley <danbri@danbri.org>
Sounds like a plan! Thanks. I'll take it to DC lists and report back
here as things progress.

2009-06-19 to dc-usage
FYI - an exchange with Dan Brickley on the Linked Open Data
mailing list.

By popular request, Dan wants FOAF to say that foaf:maker is a
sub-property of dcterms:creator and asks whether DCMI could make
a reciprocal claim. I invited him to submit a short proposal
describing how the properties are defined, with a proposed

mapping claim. We could discuss this and decide at the meeting in Seoul.

==========

From tbaker@tbaker.de Sun Sep 6 21:19:42 2009
Date: Sun, 6 Sep 2009 21:19:42 -0400
From: Thomas Baker <tbaker@tbaker.de>
To: DCMI Architecture <dc-architecture@jiscmail.ac.uk>
Cc: Dan Brickley <danbrickley@gmail.com>
Subject: Best practice for dc/dcterms:creator, foaf:maker, foaf:name

Hi Dan,

On June 19, Tom wrote to dc-usage:
> By popular request, Dan wants FOAF to say that foaf:maker is a
> sub-property of dcterms:creator and asks whether DCMI could make
> a reciprocal claim. I invited him to submit a short proposal
> describing how the properties are defined, with a proposed
> mapping claim. We could discuss this and decide at the meeting
> in Seoul.

I am writing to progress the idea of reciprocal mapping claims relating foaf:maker to dcterms:creator (see email digest below).

We have an opportunity to discuss this at the Usage Board meeting in Seoul on Friday, 16 October. I would need a few sentences proposing the mapping claim and any other proposed changes to DCMI term documentation and schemas, such as a seeAlso by a week from now -- Monday, 14 September -- at the latest. An email message is enough. Our point of reference will be the two definitions cited below.

In Seoul, we will also consider a proposal to DROP the second sentence of the usage comment for dcterms:creator ("Typically, the name of a Creator should be used to indicate the entity."), leaving the comment in place for dc:creator.

This might also be a good time to discuss the recommendation made in the FOAF specification, that "FOAF descriptions are encouraged to use dc:creator only for simple textual names". Is this still what we want to encourage?

I am posting this on dc-architecture in order to open the discussion beyond dc-usage. I believe it was Bernard Vatant, who recently joined dc-architecture, that originally raised this topic on public-lod@w3.org.

Tom

---

http://dublincore.org/documents/dcmi-terms/#terms-creator

Definition: An entity primarily responsible for making the resource.
Comment: Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.

http://xmlns.com/foaf/spec/#term_maker

maker - An agent that made this thing.
Status: stable
Domain: http://www.w3.org/2002/07/owl#Thing
Range: foaf:Agent

The foaf:maker property relates something to a foaf:Agent that foaf:made it. As such it is an inverse of the foaf:made property.

The foaf:name (or other rdfs:label) of the foaf:maker of something can be described as the dc:creator of that thing.

For example, if the thing named by the URI http://rdfweb.org/people/danbri/ has a foaf:maker that is a foaf:Person whose foaf:name is 'Dan Brickley', we can conclude that http://rdfweb.org/people/danbri/ has a dc:creator of 'Dan Brickley'.

FOAF descriptions are encouraged to use dc:creator only for simple textual names, and to use foaf:maker to indicate creators, rather than risk confusing creators with their names. This follows most Dublin Core usage. See UsingDublinCoreCreator for details.

==========

Digest of related email

2009-06-18 From: Dan Brickley <danbri@danbri.org>
To: Bernard Vatant <bernard.vatant@mondeca.com>
CC: public-lod@w3.org, Thomas Baker <tbaker@tbaker.de>
Subject: Re: Common Tag, FOAF and Dublin Core Re: Common Tag - semantic tagging convention

On 18/6/09 13:31, Bernard Vatant wrote:
>>>> ... why not use simply dc:creator and dc:date to this effect?
>>>
>>> Right. dc:date would seem a good choice, though I reckon foaf:maker
>>> might be a better option than dc:creator as the object is a resource
>>> (a foaf:Agent) rather than a literal. While it's likely to mean an
>>> extra node in many current scenarios, it offers significantly more
>>> prospect for linking data (and less ambiguity).
>>
>> dcterms:creator would also allow for use of a resource. Bibliontology
>> uses dcterms over dc.
> Well I actually meant dcterms:creator when I wrote dc:creator, sorry. So
> you can link your personal tags to your foaf profile, for example.
> And it's consistent even for tag:AutoTag, since the range of
> dcterms:creator is dcterms:Agent, including person, organisation and
> software agent as well.
> Unless I miss some sublte distinguo dcterms:Agent is equivalent to
> foaf:Agent, and dcterms:creator equivalent to foaf:maker. BTW, with due
> respect to danbri, I wish FOAF would be revised to align whenever
> possible on dcterms vocabulary, now that it has clean declarations of
> classes, domains and ranges ...
> http://dublincore.org/documents/dcmi-terms is worth (re)visiting :-)

Completely agree. I'm very happy with the direction of DC terms. The foaf:maker property was essential for a while, until DC was cleaned up. I'll mark it as a sub-property of dcterms:creator. I hope we'll get reciprocal claims into the Dublin Core RDF files some day too...

Copying Tom Baker here. Tom - what would the best process be for adding in mapping claims to the DC Terms RDF? Maybe we could draft some RDF, put it onto dublincore.org elsewhere, and for now add a seeAlso from the namespace RDF?

2009-06-18 From: Danny Ayers <danny.ayers@gmail.com>

+1
(I keep forgetting the excellent DC makeover)

2009-06-18 from Tom
Cc: Bernard Vatant <bernard.vatant@mondeca.com>, public-lod@w3.org

If you could write up a short proposal -- how the properties are defined, with a proposed mapping claim -- we could discuss this in the DCMI Usage Board and take a decision. We associate changes in the namespace RDF (and related namespace documentation) with formal decisions so would need to follow a process.

2009-06-18 From: Dan Brickley <danbri@danbri.org>
Sounds like a plan! Thanks. I'll take it to DC lists and report back here as things progress.

2009-06-19 to dc-usage
FYI - an exchange with Dan Brickley on the Linked Open Data mailing list.

> By popular request, Dan wants FOAF to say that foaf:maker is a sub-property of dcterms:creator and asks whether DCMI could make a reciprocal claim. I invited him to submit a short proposal describing how the properties are defined, with a proposed mapping claim. We could discuss this and decide at the meeting in Seoul.

--
Thomas Baker <tbaker@tbaker.de>

2008-11-14 Question posed off-list

I tried my best to generate a xml that will have a correct format but is not working very good. it can be that my knowledge in xml is not as good as it should be. is this the correct way to introduce a ISBN in DC?

<epdcx:statement
epdcx:propertyURI="http://purl.org/dc/elements/1.1/identifier"
epdcx:valueURI="URI:ISBN:0552996009" />

If not, have you got any examples that I could follow?

One draft answer

1. Any identifier can be used with dc:identifier and dcterms:identifier

2. If the identifier is defined as a URI, including URLs and URNs, the syntax encoding scheme dcterms:URI can be specified. Identifiers that have been registered as <http://www.iana.org/assignments/urn-namespaces/>; URN namespaces (e.g. ISBN, ISSN, UUID) are distinguished by that namespace (e.g. URN:ISSN:0-395-36341-1). HTTP URIs cannot be easily distinguished (e.g. it cannot be determined automatically that <blocked::http://dx.doi.org/10.1000/186>; http://dx.doi.org/10.1000/186 is a DOI and not just a URL).

3. If the identifier is not a URI, a syntax encoding scheme should be specified, e.g. somenamespace:GOVDOC, to indicate the context that the identifier is related to.

2008-10-21 Question posed

I need to use your schema to generate a XML to upload it to SWORD but I need to difference between diferent types of Identifiers: DOI, ISBN, URI, EISSN, GOVDOC.....

http://dublincore.org/2008/01/14/dcelements.rdf#identifier

Could you please tell me how to do it. Does your schema allow that?

2008-11-14 Pete draft answer

I think your example is using the eprints DC XML format [1]. This isn't a format owned/maintained by DCMI, but more on that below.

Using that XML format, in your example

> <epdcx:statement
> epdcx:propertyURI="http://purl.org/dc/elements/1.1/identifier"
> epdcx:valueURI="urn:ISBN:0552996009" />

your statement is "saying":

"The described resource is-identified-by a second resource (the value) which in turn is identified by the URI urn:ISBN:0552996009 (note: I think the URI scheme should be "urn", not "uri")

It's perhaps easier to discuss this using the DC-Text format [2], rather than an XML format. Using DC-Text this would be represented

```
@prefix dc: <http://purl.org/dc/elements/1.1/>; .

DescriptionSet (
  Description (
    Statement (
      PropertyURI ( dc:identifier )
      ValueURI ( <urn:ISBN:0552996009> )
```

```
      )
    )
  )
```

What I think you intend to "say" is

"The described resource is-identified by the URI urn:ISBN:0552996009

i.e. rather than using a second "thing" as value, you want to use the URI-as-literal as value. So, again using the DC-Text syntax, this would be represented

```
@prefix dc: <http://purl.org/dc/elements/1.1/>; .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>; .

DescriptionSet (
  Description (
    Statement (
      PropertyURI ( dc:identifier )
      LiteralValueString ( "urn:ISBN:0552996009"
        SyntaxEncodingSchemeURI (xsd:anyURI )
      )
    )
  )
)
```

And using the eprints DC XML format you would represent this as

```
<epdcx:statement
 epdcx:propertyURI="http://purl.org/dc/elements/1.1/identifier">
  <epdcx:valueString
      epdcx:sesURI="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:ISBN:0552996009</epdcx:valueString>
</epdcx:statement>
```

As I say above, the eprints DC XML format is not a DCMI-owned/maintained XML format. It was developed by the owners of the Scholarly Works Application Profile, because at the time SWAP was developed, DCMI did not have an XML format available for representing DC description sets (disclaimer: I wrote the eprints DC-XML spec, though I'm no longer directly involved in the maintainance of SWAP). There is a slight problem with eprints DC XML (which makes discussions like this one rather more complicated than they should be!), because eprints DC-XML is actually based on a version of the DCMI Abstract Model which has been superceded, and that version of the DCAM did not adequately distinguish literal from non-literal values.

DCMI has recently published a proposed recommendation for a new XML format, called DC DS XML [3], which _is_ based on the current version of the DCAM, and it may be helpful to look at that format rather than at eprints DC XML, though I should emphasise that it is likely to change in the near future in the light of comments received on the current version.

[1] http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_DC_XML
[2] http://dublincore.org/documents/2007/12/03/dc-text/
[3] http://dublincore.org/documents/2008/09/01/dc-ds-xml/

2008-05-22 Re: Identifier from Douglas Campbell To: dc-identifiers@jiscmail.ac.uk
Subject: Re: DC, OAI and identifying the digital object

I'd like to cover four issues, though probably not provide many answers... :-(

1. Resolvability of identifiers

Getting theoretical for a bit...

dc:identifier is a place to capture any, and all, identifiers for the resource - "An unambiguous reference to the resource within a given context". The question is, what resource is being identified?

Typically we would consider the following to be all identifiers for a book in our library collection:
* ISBN 1234567890
* Digitised books database record 333
* http://digitised.com/did/333
* OPAC catalogue record 888 (about the book)
* http://opac.com/bibid/888
* OPAC catalogue record 999 (about the digitised book)
* http://opac.com/bibid/999
* http://handle.net/1/123

On one level, they ARE all identifiers for the same thing, so could all go in dc:identifier. But following a lot of deeper analysis within the Semantic Web community, we are realising there are four independent (but related) resources here:
- a physical book
- a digitised version of the book
- a catalogue record (describing our holding of the physical book)
- a catalogue record (describing our holding of the digitised version of the book)

I believe this is the point-of-view Mikael has when he suggests the confusion of mixing up object and record identifiers.

Section 3 of the "Architecture of the World Wide Web, Volume One" 2004 [1] states:
a) Agents may use a URI to access the referenced resource; this is called dereferencing the URI. Access may take many forms, including retrieving a representation of the resource, adding or modifying a representation of the resource, and deleting some or all representations of the resource.
b) A representation is data that encodes information about resource state. Representations do not necessarily describe the resource, or portray a likeness of the resource, or represent the resource in other senses of the word "represent".

But this is fairly silent on what is acceptable as a representation, especially for an offline resource. So it would seem acceptable to me if, say, http://handle.net/1/123 resolved to a catalogue record (which contains links to the digitised version).

However, there has been some thinking subsequently which DOES consider what happens when a URI points to offline things/concepts [2]. But this is within the Semantic Web context and doesn't seem to have reached a conclusion yet. Though it does suggest you should have different URIs for the basic identifier, the page describing it, the metadata describing it, etc. And to be careful that dereferencing a URI does actually return a representation of the thing that URI identifies (eg. which of the above four book resources). One option it suggests is to redirect using an HTTP 303 code - I note the http://hdl.handle.net/ resolver currently uses the HTTP 302 (moved temporarily) code.

To get back to the matter in hand...

I guess the question is whether it is acceptable at all to include in dc:identifier a URI (eg. handle) that resolves to a metadata record/description page, not to the object? In the case of an OAI-type repository, the handle IS an identifier for the digital version of the paper, the problem is dereferencing the handle doesn't return a representation of the paper, it typically provides a description page that contains a link to the representation of the paper.

I don't have an answer to my question. The above OAI repository behaviour seems intuitive for us humans so we'd tend to say yes, but we need to be aware this is incompatible with the Semantic Web direction.

2. The 'main' URI

You are interested in which of multiple identifiers is resolvable in an online context, in particular the 'best' resolvable one. This is quite a specialist requirement because you happen to be online, e.g. would we also need a "best" identifier for offline objects?

The refinement/subproperty seems to be the most appropriate direction,

such as Mikael suggest s. At the risk of saying "here's one I prepared earlier", we defined a set similar to Mikael's list back in 2002 which included pid, digital object [location], and local identifier [3], though thus far we've really only used "local". The Collection Description Application Profile also has something in a similar area - cld:isLocatedAt and cld:isAccessedVia [4], though interestingly these refine dc:relation, not dc:identifier.

Defining some refinements to dc:identifier useful to multiple domains is certainly something the DCMI Identifiers Community may want to consider further. Any support from others on the list for pursuing this?

3. Simple DC is here to stay

I sympathise with Mikael's desire for developers to move away from the restrictions of DCMES, but the reality is were stuck with it, on a large scale. I am coming to realise that the things that establish a huge support are the extremely simple things, that usually have all sorts of failings, eg:
- hyperlinks (<a href>) - there is no typing (XLink solves this but hasn't really taken off)
- tagging - inconsistency and untyped (but much more popular than controlled vocabularies)
- wikis - unstructured and untyped information
- DCMES - ambiguity of types of values, eg. dates, identifiers

No one is trying to stop people using hyperlinks or tagging or non-semantic wikis [OK, there are some people, but it looks futile to me], so we shouldn't discourage people from using DCMES. Useful, specialist extensions have been built on all these (XLink, Pete's DC tagging, semantic wikis, DC qualifiers), but they will always remain niche due to the extra effort involved.

We need a parallel stream to improve the workability of DCMES implementations. Self-descibing values may be a first step...

4. Self-describing values

I have been toying with this idea for a while. Potential alternative names for this concept: self-describing values, self-encoding values, decodable values?

Basically it is a principle that instead of defining the encoding scheme separate to the value, wherever possible it is embedded into the value itself as a namespace prefix, preferably using an officially registered URI namespace (not a made up one) so it is globally unique (eg. "nlnz:334" doesn't mean much to any system outside the National Library of NZ). In DCMI Abstract Model [5] terminology, a non-literal surrogate with separate vocabulary encoding scheme URI is replaced with a value URI wherever possible.

So instead of:

```
<dc:type xsi:type="dcterms:DCMIType">StillImage</dc:format>
<dc:format xsi:type="dcterms:IMT">image/jpeg</dc:format>
<dc:subject>
   <dcam:memberOf rdf:resource="http://lcsh.info/"/>
   <rdf:value>Science</rdf:value>
</dc:subject>
<dc:identifier>
   <dcam:memberOf rdf:resource="http://www.isbn.org/"/>
   <rdf:value>1234567890</rdf:value>
</dc:identifier>
<dc:identifier>
   <dcam:memberOf rdf:resource="http://www.natlib.govt.nz/"/>
   <rdf:value>EP/1994/2454/10-F</rdf:value>
</dc:identifier>
```

use

```
<dc:type>http://purl.org/dc/dcmitype/StillImage<;/dc:format>
<dc:format>http://www.isi.edu/in-notes/iana/assignments/media-types/imag
e/jpeg</dc:format>
<dc:subject>http://lcsh.info/sh85118553#concept<;/dc:subject>
```

<dc:identifier>urn:isbn:1234567890</dc:identifier>
<dc:identifier>urn:nbn:nz:wtu:EP%2F1994%2F2454%2F10-F</dc:identifier>

It makes the data less readable to humans, but allows DCMES data to be more machine-friendly - if the data contains a colon (:), then parse it to see if it starts with the namespace for encoding schemes you know about.

This doesn't solve the question of which is the best URL to access the object, but it may help weed out ones that aren't.

Apologies for going slightly off topic here... Part of my motivation for pursuing self-encoding values is to make metadata records really simple, but still rich and decodable.

In Matapihi [6], we currently require partner organisations supply their metadata converted into a fairly complex RDF XML representation [yes, I know the RDF is incorrect, but it pre-dates the DCMI recommendation]. This complexity has been quite a hurdle for small organisations with no technical staff. I'd like to be able to offer an alternative that is more straight-forward.

It seems to me, people can grasp the concept of one XML element tag with one data value inside, but when you start adding attributes and embedded <rdf:value> elements, etc., etc. it all gets too complicated.
So I am aiming to collect encoding scheme information within the data values, as can be seen in the above before/after samples. It will be interesting to see if this improves the situation. One major key prerequisite will be to start defining more namespaces (many encoding schemes don't have any defined).

Thanx,
Douglas Campbell
National Library of New Zealand

[1] http://www.w3.org/TR/webarch/
[2] http://www.w3.org/TR/cooluris/
[3] http://www.natlib.govt.nz/dr/drterms.html
[4] http://dublincore.org/groups/collections/collection-application-profile/#colcldisAccessedVia
[5] http://dublincore.org/documents/abstract-model/
[6] http://matapihi.org.nz/

>>> Mikael Nilsson <mikael@NILSSON.NAME> 21/05/08 03:37 >>>
> Many repositories are now using multiple identifiers in each metadata
> record. This may include a handle for the metadata record, an ISBN, a
> local library catalogue record number and a DOI to the published
> version. All these are valid identifiers and are useful for the
> repository to store and use for its own purposes. (Individual handles
> for each object attached to the record generally use the DC:relation
> element).

Sounds like a reasonable scenario...

> "Consumers" of repository records for other services often use simple
> DC as the harvested format because it is the lowest common
> denominator. A key requirement of the "consumers" is to link back to
> the digital object or the record for the digital object. For example,
> the Australasian Digital Theses Program service (which I manage), uses
> OAI-PMH to harvest sets of simple DC metadata theses records from
> members` repositories. After loading, the central service uses the
> harvested identifier to provide a hypertext link to the local
> repository record so a user can click through to the PDF.
>
> However, simple DC does not allow consumer services to readily
> distinguish between DC:identifiers. In particular, which resolves to
> the local record. My preference goes slightly beyond this as I would
> like to know which resolves to a record that has the object attached,
> i.e. it identifies the record for the object not just a source or
> original metadata record that does not have an object attached.

It seems to me that the functional requirements you mention are in inescapable conflict - "simple DC" (=using only properties from the DCMES) does not support a feature that you need. Something needs to give here: either DCMES is not enough, or you need to make do with what you

have.

> Alternatively, identifiers could be self-describing such as
> urn:isbn:9781591583066, urn:doi:10.1108/00242530810865484. Could we
> add ooi for "original object identifier" (doi is taken)? With such
> self-description, consumer services could filter, configure,
> manipulate or resolve with safety and confidence for any repository.

There are multiple issues here.

1. It seems to me that you are seeing records using dc:identifier for both the object identifier and identifiers for the record. That may be one of the things creating confusion - you won't know what you get when you click, the object or just a description. That's not an ideal situation. The definition is pretty clear on this point: dc:identifier is intended to give identifiers for the object only.

This practice, in my opinion, should be strongly discouraged, as it severely undermines the possibility of using the data reliably in many machine-processing situations. In essence, it lowers the quality of the metadata significantly.

2. Whether or not a particular URI actually resolves to a digital object or not isn't really a feature of the URI itself (not even all http: URIs denote digital objects, see for example http://purl.org/dc/terms/creator ...), so I'd hesitate before encoding such information into a URN scheme.

3. There are numerous other problems with the idea of setting up and maintaining a URN scheme, but I will leave that discussion open.

To me, it seems like you are lacking features in the metadata model. I can see how there might be a need for refinements of the dc:identifier property. For example, I can imagine subproperties like

ex:resourceLocation (value type: URI), for web locations

ex:localIdentifier (value type: string), for identifiers that are not very useful outside the scope of a particular application.

ex:globallyUniqueIdentifier (value type: URI), for unique identifiers that don't resolve to the object, but are still global in scope

etc....

I would support if the DC-identifiers community decided to develop a set of such refinements to dc:identifier.

Also, I'd encourage the system developers to stop limiting themselves to the DCMES. Your example shows clearly that the use cases require more, so it's difficult to see why that restriction is still in place.

The alternative, as you describe, is to develop a carefully crafted set of heuristics. It seems clear to me that such a method will not scale very well.

Using Dublin Core - Appendix, Roles

Date Issued: 2005-12-07
Identifier: http://dublincore.org/documents/2005/12/07/usageguide/appendix_roles.shtml
Replaces: http://dublincore.org/documents/2005/11/07/usageguide/appendix_roles.shtml
Is Replaced By: Not Applicable
Is Part Of: http://dublincore.org/documents/2005/11/07/usageguide/
Latest version: http://dublincore.org/documents/usageguide/appendix_roles.shtml
Translations: http://dublincore.org/resources/translations/
Status of document: DCMI Recommended Resource
Description of document: This document describes the use of a subset of the MARC Relator list as a source of role properties for Qualified Dublin Core.

6. Using Agent Roles in Dublin Core

Introduction

MARC Relator terms are properties used to describe the relationship of an agent to a resource by specifying the particular nature of the relationship. They can be used to describe the various roles people and organizations play in the development and use of a resource. The property "Illustrator", for example, can be used for an agent which provided illustrations for the resource.

In Dublin Core, agent roles are expressed as properties (i.e., elements or element refinements). As explained below, most are refinements of the element dc:contributor. In order to identify a subset of MARC Relator Terms as refinements of dc:contributor, DCMI and the Library of Congress cooperated on the evaluation of all (circa) 150 MARC Relator Terms with regard to whether they represented "an entity responsible for making contributions to the content of the resource."

The MARC Relator List: What It Is and How It's Structured

The MARC Code List for Relators was developed for use in MARC 21 bibliographic records to express the relationship between a name and a work. The list includes both role terms and three-character codes representing those terms. The terms were only included on the list when the name and its associated role were considered important enough to include on a bibliographic record as an access point. The Library of Congress is the maintenance agency for this list and regularly adds new terms when a need is expressed and documented. The agreement between DCMI and the Library of Congress specifies that new terms submitted to LC will be referred to the DCMI Usage Board for endorsement of sub-property relationships asserted with regard to Dublin Core elements. This agreement is described in the Web document "MARC Relator Terms and Dublin Core".

The MARC Relator list includes three-character alphabetic codes to be used to identify roles. In addition the list provides definitions for the terms (and associated codes). In MARC records, the codes are synonyms for the term they represent. In DC metadata descriptions, properties are referred to using unique identifiers (URIs), and the codes were used to form unique identifiers for these properties. The list of MARC Relator Terms is maintained by the Library of Congress, so the terms have been assigned URIs on the basis of a namespace established by LC. Schemas or instance metadata will need to cite these URIs (or the MARC relator namespace) in order to use any of the MARC Relator properties, be they sub-properties of Dublin Core elements or not. See the document "Guidelines for Implementing Dublin Core in XML" for specific information on using non-DCMI namespaces.

In addition to providing Web documentation of the MARC Relator Terms, the Library of Congress provides a representation of the MARC Relator Terms in RDF/XML. Refinements of dc:contributor are, in the RDF/XML representation, asserted to be sub-properties of dc:contributor. In RDF/XML, this is done as follows:

```
<rdf:Description rdf:about="http://www.loc.gov/loc.terms/relators/ILL">
<rdfs:subPropertyOf
rdf:resource="http://purl.org/dc/elements/1.1/contributor" />
</rdf:Description>
```

In determining whether a sub-property relationship was to be asserted, LC and the Usage Board took a fairly narrow view. The relationship was asserted only if the contribution was judged to apply, by its nature, to the content of the resource. For example, whether or not "binder" is to be considered a sub-property of dc:contributor depends on the nature of the resource. Where the resource in question is valued as an art object, a binder may be construed as a "contributor" to its content; in other cases, the binder may not have this role.

Roles as refinements of Dublin Core elements

A subset of MARC Relator Terms have been identified as refinements of dc:contributor. The MARC Relator term marcrel:CRE (Creator) is asserted to be a sub-property of both dc:creator and dc:contributor. In a few cases, MARC Relator terms are considered to be refinements of Dublin Core elements other than dc:contributor. The MARC Relator terms marcrel:PBL (Publisher) and marcrel:DST (Distributor) are considered refinements of dc:publisher, as a publisher may or may not also be a "contributor" to the resource. The term marcrel:DPC (Depicted) is considered a sub-property of dc:subject.

Because roles are generally used with dc:contributor, appropriate "Dumb Down" of most agent refinements in the MARC Relator subset will be to dc:contributor, with exceptions noted above. Implementors may choose to describe "creators" using either marcrel:CRE (which will dumb down both to dc:creator and dc:contributor) or dc:creator (which will remain distinct from dc:contributor in Simple Dublin Core).

A document with further examples of refinement relationships and Dumb Down, along with examples of usage in XML, XHTML and RDF/XML, can be found at: http://www.ukoln.ac.uk/metadata/dcmi/marcrel-ex/.

Terms Not on the MARC Relators List

The MARC Relator list has been developed over many years to meet a wide variety of needs. New terms are added on the basis of need, and LC has expressed willingness to continue to expand the list upon request. Implementers also have the option to create and expose alternative vocabularies for the expression of other kinds of roles not reflected in the MARC Relator list.

For those implementations wishing to use terms from the MARC relators list that do not have a sub-property relationship to Dublin Core elements, it should be noted that an implementation may use such terms with no intrinsic harm to interoperability by using them directly, as elements, in their metadata. In the context of a Dublin Core record based on an application profile using MARC relator terms, roles not on the list as valid sub-properties endorsed by DCMI could be used in a Qualified DC expression, but not in a Simple DC expression.

Managing the Use of Role in an Implementation

The full MARC Relator list includes approximately 150 separate terms for various roles. A subset includes sub-property relationships with DC elements endorsed by DCMI. Even within this subset some of the relator terms on the list were created for specific domains and would be of little use in other communities. It might therefore be useful for implementations to declare a further subset of the relator vocabulary as relevant to their specific goals, preferably by way of a formal application profile.

The full list of MARC Relator Terms (including refinements of Dublin Core elements): http://www.loc.gov/loc.terms/relators/

The subset of MARC Relator Terms which refine Dublin Core terms: http://www.loc.gov/loc.terms/relators/dc-contributor.html

The RDF representation of MARC Relator Terms: http://www.loc.gov/loc.terms/relators/dc-relators.xml

For further information, see "MARC Relator Terms and Dublin Core" http://dublincore.org/usage/documents/relators/.

2005-01-16, Errata: broken link repaired.

Metadata associated with this resource: http://dublincore.org/documents/usageguide/appendix_roles.shtml.rdf

ePrints UK

Using simple Dublin Core to describe eprints

by Andy Powell, Michael Day and Peter Cliff
UKOLN, University of Bath
Version 1.2

Introduction

This document provides some recommendations for the use of simple Dublin Core metadata [1] to describe eprints in eprint archives.

These recommendations are not primarily targeted at the end-users of eprint archives. Rather, they are intended to guide 'best-practice' for the maintainers of eprint archives in order that such systems can be configured to maximise the benefits of a shared approach to metadata disclosure. The intention is to facilitate more consistent results when 'cross-searching' and browsing metadata records gathered from multiple eprint archives. Nonetheless, we would hope that these guidelines to have some impact on the cataloguing guidelines and help sub-systems offered to end-users of institutional eprint archives.

As can be seen from the recommendations for dc:type (below), this document uses a fairly wide working definition of 'eprint':

> *"an electronic copy of an academic paper"* [2].

These recommendations are specific to the use of simple (unqualified) Dublin Core. The intention is to develop a separate set of guidelines for the use of qualified Dublin Core (including the use of element refinements and encoding schemes) to describe eprints.

These recommendations draw on three documents:

- DCMI Elements and Element Refinements - a current list [3]
- Using Dublin Core [4]
- The RDN Cataloguing Guidelines [5]

General Recommendations

All eprint archives must support the 'oai_dc' record format (this is mandated by the OAI-PMH [6]).

All eprint archives are strongly recommended to support the following minimal set of DC elements within the 'oai_dc' record format.

- dc:title
- dc:creator

- dc:subject
- dc:description
- dc:publisher
- dc:date
- dc:type
- dc:identifier
- dc:language
- dc:relation

The values of these elements must **not** contain any HTML (or XML) markup. They may contain LaTeX [7] commands if desired but it is worth remembering that there is no mechanism for explicitly indicating that LaTeX is being used.

The Elements

This section lists each of the Dublin Core elements. For each element, an eprint-specific recommendation is provided followed by the authoritative definitions and comments from the Dublin Core Metadata Initiative.

(*) indicates that an element is part of the minimal set.

dc:title (*)

Eprint-specific Recommendation:	The title of the eprint.
	
	Preserve the original wording, order and spelling of the eprint title. Only capitalize proper nouns. Punctuation need not reflect the usage of the original. Subtitles should be separated from the title by a colon. For example:
	
	`<dc:title>Initial sequencing and analysis of the human genome</dc:title>`
	`<dc:title>The new nationalism and the old history: perspectives on the West German Historikerstreit</dc:title>`
	
	If necessary, repeat this element for multiple titles.

DC Label:	Title
DC Name:	title
DC Definition:	A name given to the resource.
DC Comment:	Typically, a Title will be a name by which the resource is formally known.

dc:creator (*)

Simple Dublin Core

Notes from Seattle, 2007:

"Simple DC" is a concept used in a number of ways in a number of places. Do we address it in reference to OAI?

-- "Simple DC is a description set with one description that describes a resource with 15 optional property usages"

-- If we say Simple DC is a conforming AP, then folks will be using it as a model.

-- What goes on a AP for Simple DC? It should include those properties of the DCMI Profile Model (Application Profile of Application Profiles).

-- ISSUE: OAI uses value string language. Do we put anything about value string language in this AP?

-- AGREED: Simple DC includes value string language, and this is optional.

-- AGREED: Need to include URI in Simple DC AP, not the QName.

-- AGREED: Do not cite our own XML schemas in the AP for Simple DC

-- ISSUE: Documentation mentioning "Simple Dublin Core" should be revised to point to the Simple DC AP.

Use of "Simple Dublin Core" in DCMI documentation

http://dublincore.org/documents/dcmes-xml/

> This document describes an encoding for the DCMES in XML subject to these restrictions:
>
> * The Dublin Core elements described in the DCMES V1.1 reference can be used
> * No other elements can be used
> * No element qualifiers can be used
> * The resulting RDF/XML cannot be embedded in web pages

http://dublincore.org/documents/usageguide/

> The Dublin Core standard includes two levels: Simple and Qualified. Simple Dublin Core comprises fifteen elements; Qualified Dublin Core includes three additional elements (Audience, Provenance and RightsHolder), as well as a group of element refinements (also called qualifiers) that refine the semantics of the elements in ways that may be useful in resource discovery.

http://dublincore.org/documents/usageguide/glossary.shtml

> Simple Dublin Core
>
> The fifteen Dublin Core elements used without qualifiers, that is without element refinement or encoding schemes. Sometimes referred to as Dublin Core simple.

http://dublincore.org/resources/faq/

> "Simple Dublin Core" is Dublin Core metadata that uses no qualifiers; only the main 15 elements of the Dublin Core Metadata Element Set are expressed as simple attribute-value pairs without any "qualifiers" (such as encoding schemes, enumerated lists of values, or other processing clues) to provide more detailed information about a resource.

http://dublincore.org/schemas/xmls/

Simple DC XML schema, version 2002-12-12

This schema defines terms for Simple Dublin Core, i.e. the 15 elements from the http://purl.org/dc/elements/1.1/ namespace, with no use of encoding schemes or element refinements.

2009-05-26 Pete writes about a misconception re: Simple DC

> 3. but, dc elements don't have refinements.

Ah, no, that's not true. (More below).

> (Simple Dublin Core: The fifteen Dublin Core elements used
> without qualifiers, that is without element refinement or
> encoding schemes. Sometimes referred to as Dublin Core
> simple. http://dublincore.org/documents/usageguide/glossary.shtml)
>
> How can dc:title be refined by dcterms:alternative but then
> have no refinements? Isn't that a contradiction?

No, there's no contradiction.

I think the important point here is to maintain the distinction between

(i) the Dublin Core Metadata Element Set i.e. the set of 15 properties

(ii) "Simple Dublin Core", as a "DC Application Profile" (DCAP), which defines one set of constraints - and only one of many possible sets of constraints - on the use of those 15 properties to construct a DC description set (containing a single description).

The former is just a set of properties, which may have relationships to other properties (through e.g. subproperty (element refinement) relationships), and to classes (through domain/range relationships).

Those relationships are specified by assertions made in the authoritative dsecriptions of those properties provided by DCMI:

http://dublincore.org/documents/dces/ (or in machine-readable form http://purl.org/dc/elements/1.1/

(In the case of the DCMES, no domain/range is specified, so in each case it is the class of all things (rdfs:Resource)).

And these are "global" characteristics of the properties: they apply wherever the properties are used. Regardless of what is said in the context of any particular DCAP, dcterms:alternative is a subproperty (element refinement of) dc:title.

Beyond the limits imposed by the domain/range considerations, those global descriptions of those properties provided by DCMI don't enforce any particular constraints on thse use of those properties in a description - they don't tell me whether a statement using dc:title is required or optional, or a statement using dc:subject should provide a VES or not, and so on.

"Simple Dublin Core" is a DC application profile which specifies one "pattern of use", one set of structural constraints on the use of, those 15 properties in a description set - a Description Set Profile. It says that in a description conforming to that DCAP, a statements are limited to using the 15 DCMES properties; the presence of a staement using each of the 15 DCMES properties is optional, and there may be multiple statements using a single property; only literal values may be used, so no VES URI and no value URI permitted; and SES are not permitted.

But N.B. that is just my suggested definition for Simple DC! I think it's consistent with historical usage of the term (and it is a reasonable fit as a retrospective interpretation of the oai_dc syntax usued in OAI-PMH - but it is still a retrospective interpretation!) Unfortunately DCMI doesn't have an authoritative description of Simple DC as a DCAP at this moment. Tom & Mikael worked on a draft of this some time ago - though I disagreed with some of it, IIRC! :-) - but that work has been on the back-burner for a while. I think having this would clarify some of these issues, but there just hasn't been enough effort available to take it forward.

So there's no contradiction between, on the one hand, the global assertion that dcterms:alternative is a refinement of dc:title and, on the other, some particular DCAP such as "Simple DC" saying that it supports the use of dc:title, but not dcterms:alternative, in description sets based on that DCAP.

But I readily admit that DCMI's documentation doesn't make this distinction clear - and some of the older documents are particularly muddled. There are several key documents which need to be finalised or rewritten to really address this, it seems to me.

> In discussing this with you, we had a couple of other questions.
>
> a. Does dcterms:alternative refine dc:title in order to
> achieve dumb-down from dcterms:alternative to dc:title (which
> can be achieve only by qualification, as I read the definition)?
>
> (Dumb-down Principle: The qualification of Dublin Core
> Elements is guided by a rule known colloquially as the
> Dumb-Down Principle. According to this rule, a client should
> be able to ignore any qualifier and use the value as if it
> were unqualified. While this may result in some loss of
> specificity, the remaining term value (minus the qualifier)
> must continue to be generally correct and useful for
> discovery. Qualification is therefore supposed only to
> refine, not extend the semantic scope of an Element.
> http://dublincore.org/documents/usageguide/glossary.shtml)

I think you have to remember that the Usage Guide is really a historical document which hasn't been aligned with recent developments. It really reflects a "view" in which the 15 properties of the DCMES were perceived as the "centre" to which everything else had to be related.

But that isn't the current "view" of the DC properties, and I think it's significant that "dumb-down" is no longer mentioned in the DCMI Abstract Model. Rather, the DCAM just specifies how the "historical" types of relationship mentioned in the Grammatical Principles correspond to the relationship types defined by RDF, RDF Schema or (for VES) by the DCAM itself.

And like Simple DC, I don't think "dumb-down" is really well-defined anywhere :-(

But I think it is usually taken to mean the generation of a "Simple DC" description set. And that is typically achieved by applying a combination of

(i) the formal inferencing rules defined by RDF Schema (more on this below), and
(ii) some additional rules about how to map the components which can occur in a DC description set in the general case to the subset of components

The second of these things isn't really well-defined: the text you cite above suggests I can "ignore qualifiers", so OK, the rule is discard SES URI and VES URI. But if we accept my definition of Simple DC above, Simple DC does not support the use of value URIs: what is the rule for treating a value URI? One option would be to say that a value URI is mapped to a "value string" - but I don't think that rule is written down anywhere! :-)

> b. Is it desirable to be able to dumb dcterms:alternative
> down to dc:title?
> An alternative

> relationship between dcterms:alternative and dc:title might
> be a combination of dumb-down and mapping, as follows:
>
> dumb down: dcterms:alternative refines dcterms:title
> mapping: dcterms:title to dc:title

I'm afraid I'm not sure I really grasp the distinction you are making here: to me, as I've said above, "dumb-down" is just a special case of "mapping", where the target of the mapping is (usually) "Simple DC".

Or to put it another way, both "dumb-down" and "mapping" (at least mapping between RDF vocabularies) involves defining and applying some set of inferencing rules, typically the basic rules defined by RDF Schema, maybe supplemented by other more complex rules.

DCMI's definition of dcterms:alternative says

dcterms:alternative rdfs:subPropertyOf dcterms:title
dcterms:alternative rdfs:subPropertyOf dc:title

Which means that, following the inferencing rules defined by RDF Schema, anytime I encounter a triple

doc:D dcterms:alternative "abc"

Then I can infer the following two triples

doc:D dcterms:title "abc"
doc:D dc:title "abc"

i.e. whenever I "say" the first statement, I imply the second two statements. And that applies for _any_ triple/statement where the property is used, regardless of the subject/described resource and the value.

Similarly for the case of dcterms:title, DCMI's definition says

dcterms:title rdfs:subPropertyOf dc:title

So any time I see

doc:D dcterms:title "abc"

I can infer

doc:D dc:title "abc"

> c. Can one map from dc:title to dcterms:title and vice-versa?

As above, if I find

doc:D dcterms:title "abc"

I can infer

doc:D dc:title "abc"

Or to put it another way, one can map from dcterms:title to dc:title.

But not vice versa: I can't map from dc:title to dcterms:title. There's no basis for doing so in the definitions of the properties. The two properties aren't equivalents. And the reason for this is that they have different ranges: for dcterms:title, the range is the class of literals, but for dc:title the range is the class of all resources. In theory at least, dc:title could be used with a non-literal value, which the range of dcterms:title does not support.

> d. I haven't looked into the question of to what extent these
> are issues for the other Simple Dublin Core elements.

I think essentially the same argument applies for all the properties.

I hope this sort of makes sense. Some of it is, as I say, necesaarily just my own conceptualisation (of things like Simple DC), which does not (yeet at least) have a formal DCMI endorsement (though I'd argue it's a

reasonable conceptualisation!)

The underlying problem, it seems to me, is that the DCMI Web site has such a mix of old documents with concepts described in rather loose, informal (and sometimes contradictory) terms, and newer documents based on a rather more formal framework. And not all of those "old world" concepts have been "recast" within the formal framework - though I'd like to think we are getting there, slowly.

2009-05-29 Response

I think our experience may be typical for what one finds in the community: we thought that before reading your explanation, we understood. But as in the case of the blind man and the elephant [1], we understood only as much as our experience allowed. We now have a much better understanding, but we both agree that it would be really good to revise those "legacy" (which remain absolutely essential) documents again so that the entire community can benefit. In fact, perhaps you would consider making explanations such as this available to the community, e.g. in the form of an unofficial FAQ or a blog or ...

> permitted; and SES are not permitted.

** OK. Here is a point where we both said that we were never quite sure where DCMES left off and a DCAP began, but I think that it is now clearer and I now see that there is no contradiction. So, more precise documentation on this point would be welcome.**

> (ii) some additional rules about how to map the components
> which can occur in a DC description set in the general case
> to the subset of components

It would also be good to formalize and define this process.

> I've said above, "dumb-down" is just a special case of
> "mapping", where the target of the mapping is (usually)
> "Simple DC".

This question is now moot. It was a good question (from my point of view :)) if one didn't understand that dumb down is formally a form of mapping.

> Or to put it another way, one can map from dcterms:title to dc:title.

Now it is clear! So a rdfs:subPropertyOf predicate defines a "mapping-to" relation.

> But not vice versa: I can't map from dc:title to
> dcterms:title. There's no basis for doing so in the
> definitions of the properties. The two properties aren't
> equivalents. And the reason for this is that they have
> different ranges: for dcterms:title, the range is the class
> of literals, but for dc:title the range is the class of all
> resources. In theory at least, dc:title could be used with
> a non-literal value, which the range of dcterms:title does
> not support.

Yes, now I see how this follows. If "mapping-to" may only occur in the case of subset relations or identity relations, then mapping from from dc:title to dcterms:title would not qualify. But then, "mapping" would also have to be formally defined.

> The underlying problem, it seems to me, is that the DCMI
> Web site has such a mix of old documents with concepts
> described in rather loose, informal (and sometimes
> contradictory) terms, and newer documents based on a rather
> more formal framework. And not all of those "old world"
> concepts have been "recast" within the formal framework -
> though I'd like to think we are getting there, slowly.

We agree. But in so doing, it would be important to not recast the old world documents in such a way that the implementations that are based on them are no longer well-defined.

[1] http://en.wikipedia.org/wiki/Blind_Men_and_an_Elephant

2007-08-24 Mikael

> Second (and this is my main point here) I guess I do wonder whether it's
> something of a "revision of history too far" to define Simple DC as a
> DSP which
>
> (i) defines Statement Templates which do _not_ reference the
> http://purl.org/elements/1.1/... properties
> (ii) uses both literal and non-literal value surrogates (for different
> Statement Templates referencing different properties)

As I'm sure you can tell, proposing that *was* a conscious decision. We'll need to think about the consequences, though.

I *do* think we want to do what we have done. I'd not call it revisionist, but rather an important update.

To revisit the definition of "Simple DC" in the light of the new dcterms properties, the new DCAM, the domains and ranges, and the DSP model, is, I think, a good way to highlight the fact that DCMI *has* done all that.

Keeping the old notion of Simple DC leads to the risk of sending a double message: yes, a number of important things have changed, but we're not really ready to take the consequences :-)

So I'd advocate using the "Simple DC" term, but of course acknowledge the existence of a "legacy" term.

Looking forward to other opinions on this matter!

2007-08-24

> As I'm sure you can tell, proposing that *was* a conscious decision.

Heheh ;-)

> We'll need to think about the consequences, though.
>
> I *do* think we want to do what we have done. I'd not call it
> revisionist, but rather an important update.
>
> To revisit the definition of "Simple DC" in the light of the
> new dcterms properties, the new DCAM, the domains and ranges,
> and the DSP model, is, I think, a good way to highlight the
> fact that DCMI *has* done all that.
>
> Keeping the old notion of Simple DC leads to the risk of
> sending a double message: yes, a number of important things
> have changed, but we're not really ready to take the consequences :-)

Though I think one way of taking the consequences would be to say that "Simple DC" is indeed a legacy term, but it is one which has an interpretation which is "out there", and so we can't overnight reuse it for something else, and this new DSP described here is called something other than "Simple DC".

> So I'd advocate using the "Simple DC" term, but of course
> acknowledge the existence of a "legacy" term.
>

OK... If we do take the route proposed, I think we (and I suppose I mean the OAI folk and DCMI between us, but largely us given that we ae making the changes!) do have to come up with a DSP which we can "retrofit" to the oai_dc XML format.

Which (I think) means either:

(i) saying oai_dc _is_ a format for this new "Simple DC" DSP, and the mapping from XML Infoset to DC description set has to be done on an element-by-element basis (as sketched in my previous message)

(ii) saying that oai_dc is a format for some other DSP (as sketched towards the end of my previous message) called "the OAI DC DSP" or whatever, which is different from what DCMI now calls "Simple DC", and the mapping from XML Infoset to DC description set is more "uniform" (i.e. doesn't vary with XML element name etc)

I really don't think we can say to the OAI people that the world has moved on, this is what "Simple DC" means now, and there is no "good fit" between a DSP and the oai_dc XML format (especially given that OAI-PMH repositories exposing oai_dc XML instances represent probably the single largest source of DC metadata on the Web).

Using Dublin Core - The Elements

Creator: Diane Hillmann

Date Issued: 2005-11-07
Identifier: http://dublincore .org/documents/2005/11/07/usageguide/elements.shtml
Replaces: http://dublincore .org/documents/2005/08/15/usageguide/elements.shtml
Is Replaced By: Not applicable
Is Part Of: http://dublincore.org/documents/2005/11/07/usageguide/
Latest Version: http://dublincore.org/ documents/usageguide/elements.shtml
Status of Document: DCMI Recommended Resource

Description of Document: This document is intended as an entry point for users of Dublin Core. For non-specialists, it will assist them in creating simple descriptive records for information resources (for example, electronic documents). Specialists may find the document a useful point of reference to the documentation of Dublin Core, as it changes and grows.

4. The Elements

This section lists each element by its full name and label. For each element there are guidelines to assist in creating metadata content, whether it is done "from scratch" or by converting an existing record in another format.

In the element descriptions below, a formal single-word label is specified to make the syntactic specification of elements simpler for encoding schemes. Although some environments, such as HTML, are not case-sensitive, it is recommended best practice always to adhere to the case conventions in the element names given below to avoid conflicts in the event that the metadata is subsequently converted to a case-sensitive environment, such as XML.

Some information may appear to belong in more than one metadata element. While there will normally be a clear preferred choice, there is potential semantic overlap between some elements. Consequently, there will occasionally be some judgment required from the person assigning the metadata.

4.1. Title

Label: Title

Element Description: The name given to the resource. Typically, a Title will be a name by which the resource is formally known.

Guidelines for creation of content:

If in doubt about what constitutes the title, repeat the Title element and include the variants in second and subsequent Title iterations. If the item is in HTML, view the source document and make sure that the title identified in the title header (if any) is also included as a Title.

Examples:

Title="A Pilot's Guide to Aircraft Insurance"

Title="The Sound of Music"
Title="Green on Greens"
Title="AOPA's Tips on Buying Used Aircraft"

4.2. Subject

Label: Subject and Keywords

Element Description: The topic of the content of the resource. Typically, a Subject will be expressed as keywords or key phrases or classification codes that describe the topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.

Guidelines for creation of content:

Select subject keywords from the Title or Description information, or from within a text resource. If the subject of the item is a person or an organization, use the same form of the name as you would if the person or organization were a Creator or Contributor.

In general, choose the most significant and unique words for keywords, avoiding those too general to describe a particular item. Subject might include classification data if it is available (for example, Library of Congress Classification Numbers or Dewey Decimal numbers) or controlled vocabularies (such as Medical Subject Headings or Art and Architecture Thesaurus descriptors) as well as keywords.

When including terms from multiple vocabularies, use separate element iterations. If multiple vocabulary terms or keywords are used, either separate terms with semi-colons or use separate iterations of the Subject element.

Examples:

Subject="Aircraft leasing and renting"
Subject="Dogs"
Subject="Olympic skiing"
Subject="Street, Picabo"

4.3. Description

Label: Description

Element Description: An account of the content of the resource. Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.

Guidelines for creation of content:

Since the Description field is a potentially rich source of indexable terms, care should be taken to provide this element when possible. Best practice recommendation for this element is to use full sentences, as description is often used to present information to users to assist in their selection of appropriate resources from a set of search results.

Descriptive information can be copied or automatically extracted from the item if there is no abstract or other structured description available. Although the source of the description may be a web page or other structured text with presentation tags, it is generally not good practice to include HTML or other structural tags within the Description element. Applications vary considerably in their ability to interpret such tags, and their inclusion may negatively affect the interoperability of the metadata.

Examples:

Description="Illustrated guide to airport markings and lighting signals, with particular reference to SMGCS (Surface Movement Guidance and Control System) for airports with low visibility conditions."

Description="Teachers Domain is a multimedia library for K-12 science educators, developed by WGBH through funding from the National Science Foundation as part of its National Science Digital Library initiative. The site offers a wealth of classroom-ready instructional resources, as well as online professional development materials and a set of tools which allows teachers to manage, annotate, and share the materials they use in classroom teaching."

4.4. Type

Label: Resource Type

Element Description: The nature or genre of the content of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DCMIType vocabulary). To describe the physical or digital manifestation of the resource, use the FORMAT element.

Guidelines for content creation:

If the resource is composed of multiple mixed types then multiple or repeated Type elements should be used to describe the main components.

Because different communities or domains are expected to use a variety of type vocabularies, best practice to ensure interoperability is to include at least one general type term from the DCMIType vocabulary in addition to the domain specific type term(s), in separate Type element iterations.

Examples:

Type="Image"
Type="Sound"
Type="Text"
Type="simulation"

Note: The first three values are taken from the DCMI Type Vocabulary, and follow the capitalization conventions for that vocabulary. The last value is a term from an unspecified source.

The item described is an *Electronic art exhibition catalog:*

Type="Image"
Type="Text"
Type="Exhibition catalog"

Note: The first two values are taken from the DCMI Type Vocabulary, and follow the capitalization conventions for that vocabulary. The last value is a term from an unspecified source.

The item described is a *Multimedia educational program with interactive assignments:*

Type="Image"
Type="Text"
Type="Software"
Type="InteractiveResource"

Note: All values in this example are taken from the DCMI Type Vocabulary, and follow the capitalization conventions for that vocabulary.

4.5. Source

Label: Source

Element Description: A Reference to a resource from which the present resource is derived. The present resource may be derived from the Source resource in whole or part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.

Guidelines for content creation:

In general, include in this area information about a resource that is related intellectually to the described resource but does not fit easily into a Relation element.

Examples:

Source="RC607.A26W574 1996" [where "RC607.A26W574 1996" is the call number of the print version of the resource, from which the present version was scanned]

Source="Image from page 54 of the 1922 edition of Romeo and Juliet"

4.6. Relation

Label: Relation

Element Description: A reference to a related resource. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.

Guidelines for content creation:

Relationships may be expressed reciprocally (if the resources on both ends of the relationship are being described) or in one direction only, even when there is a refinement available to allow reciprocity. If text strings are used instead of identifying numbers, the reference should be appropriately specific. For instance, a formal bibliographic citation might be used to point users to a particular resource.

Because the refined terms used with Relation provide significantly more information to a user than the unqualified use of Relation, implementers who are describing heavily interrelated resources might choose to use qualified Dublin Core.

Examples:

Title="Reading Turgenev"
Relation="Two Lives" [Resource is a collection of two novellas, one of which is "Reading Turgenev"]
*[Relationship described is **IsPartOf**.*

[Part/Whole relations are those in which one resource is a physical or logical part of another]

Title="Candle in the Wind"
Subject="Diana, Princess of Wales"
Date="1997"
Creator="John, Elton"
Type="sound"
Description="Tribute to a dead princess."
Relation="Elton John's 1976 song Candle in the Wind"
*[Relationship described is **IsVersionOf**.*

[Version relations are those in which one resource is an historical state or edition, of another resource by the same creator]

Title="Electronic AACR2"
Relation="Anglo-American Cataloging Rules, 2nd edition"
*[Relationship described is **IsFormatOf**]*

Title="Landsat TM dataset of Arnhemland, NT, Australia"
Relation="arnhem.gif"
[Relationship described is **HasFormat**]

[Format transformation relations are those in which one resource has been derived from another by a reproduction or reformatting technology which is not fundamentally an interpretation but intended to be a representation.]

Title="Morgan's Ancient Society"
Relation="Engels' Origin of the Family, Private Property and the State"
*[Relationship described is **IsReferencedBy**]*

Title="Nymphet Mania"
Relation="References Adrian Lyne's 'Lolita'"
*[Relationship described is **References**]*

[Reference relations are those in which the author of one resource cites, acknowledges, disputes or otherwise make claims about another resource.]

Title="Peter Carey's novel Oscar and Lucinda"
Relation="1998 movie Oscar and Lucinda"
*[Relationship described is **IsBasisFor**]*

Title="The movie My Fair Lady"
Relation="Shaw's play Pygmalion"
*[Relationship described is **IsBasedOn**]*

[Creative relations are those in which one resource is a performance, production, derivation, adaptation or interpretation of another resource.]

Title="Dead Ringer"
Relation="Gemstar e-book"
*[Relationship described is **Requires**]*

[Dependency relations are those in which one resource requires another resource for its functioning, delivery, or content and cannot be used without the related resource being present.]

4.7. Coverage

Label: Coverage

Element Description: The extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [Getty Thesaurus of Geographic Names, http://www. getty.edu/research/tools/vocabulary/tgn/]). Where appropriate, named places or time periods should be used in preference to numeric identifiers such as sets of co-ordinates or date ranges.

Guidelines for content creation:

Whether this element is used for spatial or temporal information, care should be taken to provide consistent information that can be interpreted by human users, particularly in order to provide interoperability in situations where sophisticated geographic or time-specific searching is not supported. For most simple applications, place names or coverage dates might be most useful. For more complex applications, consideration should be given to using an encoding scheme that supports appropriate specification of information, such as DCMI Period, DCMI Box or DCMI Point.

Examples:

Coverage="1995-1996"
Coverage="Boston, MA"
Coverage="17th century"
Coverage="Upstate New York"

4.8. Creator

Label: Creator

Element Description: An entity primarily responsible for making the content of the resource. Examples of a Creator include a person, an organization, or a service. Typically the name of the Creator should be used to indicate the entity.

Guidelines for creation of content:

Creators should be listed separately, preferably in the same order that they appear in the publication. Personal names should be listed surname or family name first, followed by forename or given name. When in doubt, give the name as it appears, and do not invert.

In the case of organizations where there is clearly a hierarchy present, list the parts of the hierarchy from largest to smallest, separated by full stops and a space. If it is not clear whether there is a hierarchy present, or unclear which is the larger or smaller portion of the body, give the name as it appears in the item.

If the Creator and Publisher are the same, do not repeat the name in the Publisher area. If the nature of the responsibility is ambiguous, the recommended practice is to use Publisher for organizations, and Creator for individuals. In cases of lesser or ambiguous responsibility, other than creation, use Contributor.

Examples:

Creator="Shakespeare, William"
Creator="Wen Lee"
Creator="Hubble Telescope"
Creator="Internal Revenue Service. Customer Complaints Unit"

4.9. Publisher

Label: Publisher

Element Description: The entity responsible for making the resource available. Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.

Guidelines for content creation:

The intent of specifying this field is to identify the entity that provides access to the resource. If the Creator and Publisher are the same, do not repeat the name in the Publisher area. If the nature of the responsibility is ambiguous, the recommended practice is to use Publisher for organizations, and Creator for individuals. In cases of ambiguous responsibility, use Contributor.

Examples:

Publisher="University of South Where"
Publisher="Funky Websites, Inc."
Publisher="Carmen Miranda"

4.10. Contributor

Label: Contributor

Element Description: An entity responsible for making contributions to the content of the resource. Examples of a Contributor include a person, an organization or a service. Typically, the name of a Contributor should be used to indicate the entity.

Guideline for content creation:

The same general guidelines for using names of persons or organizations as Creators apply here. Contributor is the most general of the elements used for "agents" responsible for the resource, so should be used when primary responsibility is unknown or irrelevant.

4.11. Rights

Label: Rights Management

Element Description: Information about rights held in and over the resource. Typically a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

Guidelines for content creation:

The Rights element may be used for either a textual statement or a URL pointing to a rights statement, or a combination, when a brief statement and a more lengthy one are available.

Examples:

Rights="Access limited to members"
Rights="http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Terms& quot;

4.12. Date

Label: Date

Element Description: A date associated with an event in the life cycle of the resource. Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [Date and Time Formats, W3C Note, http://www.w3.org/TR/NOTE- datetime] and follows the YYYY-MM-DD format.

Guidelines for content creation:

If the full date is unknown, month and year (YYYY-MM) or just year (YYYY) may be used. Many other schemes are possible, but if used, they may not be easily interpreted by users or software.

Examples:

Date="1998-02-16"
Date="1998-02"
Date="1998"

4.13. Format

Label: Format

Element Description: The physical or digital manifestation of the resource. Typically, Format may include the media-type or dimensions of the resource. Examples of dimensions include size and duration. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource.

Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [http:// www.iana.org/ assignments/media-types/] defining computer media formats).

Guidelines for content creation:

In addition to the specific physical or electronic media format, information concerning the size of a resource may be included in the content of the Format element if available. In resource discovery size, extent or medium of the resource might be used as a criterion to select resources of interest, since a user may need to evaluate whether they can make use of the resource within the infrastructure available to them.

When more than one category of format information is included in a single record, they should go in separate iterations of the element.

Examples:

Title="Dublin Core icon"
Identifier="http://purl.org/metadata/dublin_core/images/dc2.gif& quot;
Type="Image"
Format="image/gif"
Format="4 kB"

Subject="Saturn"
Type="Image"
Format="image/gif 6"
Format="40 x 512 pixels"
Identifier="http://www.not.iac.es/newwww/photos/images/satnot.gif "

Title="The Bronco Buster"
Creator="Frederic Remington"
Type="Physical object"
Format="bronze"
Format="22 in."

4.14. Identifier

Label: Resource Identifier

Element Description: An unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Examples of formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).

Guidelines for content creation:

This element can also be used for local identifiers (e.g. ID numbers or call numbers) assigned by the Creator of the resource to apply to a particular item. It should not be used for identification of the metadata record itself.

Examples:

Identifier="http://purl.oclc.org/metadata/dublin_core/& quot;
Identifier="ISBN:0385424728"
Identifier="H-A-X 5690B" [publisher number]

4.15. Language

Label: Language

Element Description: A language of the intellectual content of the resource. Recommended best practice for the values of the Language element is defined by RFC 3066 [RFC 3066, http://www.ietf.org/rfc/ rfc3066.txt] which, in conjunction with ISO 639 [ISO 639, http://www.oasis- open.org/cover/iso639a.html]), defines two- and three-letter primary language tags with optional subtags. Examples include "en" or "eng" for English, "akk" for Akkadian, and "en-GB" for English used in the United Kingdom.

Guidelines for content creation:

Either a coded value or text string can be represented here. If the content is in more than one language, the element may be repeated.

Examples:

Language="en"
Language="fr"
Language="Primarily English, with some abstracts also in French."
Language="en-US"

NOTE: Audience, Provenance and RightsHolder are elements, but not part of the Simple Dublin Core fifteen elements. Use Audience, Provenance and RightsHolder only when using Qualified Dublin Core.

4.16. Audience

Label: Audience

Element Description: A class of entity for whom the resource is intended or useful. A class of entity may be determined by the creator or the publisher or by a third party.

Guidelines for content creation:

Audience terms are best utilized in the context of formal or informal controlled vocabularies. None are presently recommended or registered by DCMI, but several communities of interest are engaged in setting up audience vocabularies. In the absence of recommended controlled vocabularies, implementors are encouraged to develop local lists of values, and to use them consistently.

Examples:

Audience="elementary school students"
Audience="ESL teachers"
Audience="deaf adults"

4.17. Provenance

Label: Provenance

Element Description: A statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity and interpretation. The statement may include a description of any changes successive custodians made to the resource.

Guidelines for content creation:

Examples:

Provenance="This copy once owned by Benjamin Spock."
Provenance="Estate of Hunter Thompson."
Provenance="Stolen in 1999; recovered by the Museum in 2003."

4.18. RightsHolder

Label: Rights Holder

Element Description: A person or organization owning or managing rights over the resource. Recommended best practice is to use the URI or name of the Rights Holder to indicate the entity.

Guidelines for content creation:

Since, for the most part, people and organizations are not typically assigned URIs, a person or organization holding rights over a resource would be named using a text string. People and organizations sometimes have websites, but URLs for these are not generally appropriate for use in this context, since they are not clearly identifying the person or organization, but rather the location of a website about them.

Examples:

RightsHolder="Stuart Weibel"
RightsHolder="University of Bath"

4.19. InstructionalMethod

Label: Instructional Method

Element Description: A process, used to engender knowledge, attitudes and skills, that the resource is designed to support. Instructional Method will typically include ways of presenting instructional materials or conducting instructional activities, patterns of learner-to-learner and learner-to-instructor interactions, and mechanisms by which group and individual levels of learning are measured. Instructional methods include all aspects of the instruction and learning processes from planning and implementation through evaluation and feedback.

Guidelines for content creation:

Best practice is to use terms from controlled vocabularies, whether developed for the use of a particular project or in general use in an educational context.

Examples:

InstructionalMethod="Experiential learning"
InstructionalMethod="Observation"
InstructionalMethod="Large Group instruction"

4.20. AccrualMethod

Label: Accrual Method

Element Description: The method by which items are added to a collection. Recommended best practice is to use a value from a controlled vocabulary.

Guidelines for content creation:

Terms from controlled vocabularies may be developed for the use of a particular project or in general use in a cultural materials context.

Examples:

AccrualMethod="Deposit"
AccrualMethod="Purchase"

4.21. AccrualPeriodicity

Label: Accrual Periodicity

Element Description: The frequency with which items are added to a collection. Recommended best practice is to use a value from a controlled vocabulary.

Guidelines for content creation:

Terms from controlled vocabularies may be developed for the use of a particular project or in general use in a cultural materials context.

Examples:

AccrualPeriodicity="Annual"
AccrualPeriodicity="Irregular"

4.22. AccrualPolicy

Label: Accrual Policy

Element Description: The policy governing the addition of items to a collection. Recommended best practice is to use a value from a controlled vocabulary.

Guidelines for content creation:

Terms from controlled vocabularies may be developed for the use of a particular project or in general use in a cultural materials context.

Examples:

AccrualPolicy="Active"
AccrualPolicy="Closed"

Errata

2006-02-07: Corrected spelling (s/Languge/Language/)
2006-08-28, Errata: 'Type="Interactive Resource"' changed to 'Type="InteractiveResource"' in Section 4.4.

Metadata associated with this resource: http://dublincore.org/documents/usageguide/elements.shtml.rdf

Using Dublin Core - Dublin Core Qualifiers

Date Issued: 2005-11-07
Identifier: http://dublincore.org/documents/2005/11/07/usageguide/qualifiers.shtml
Replaces: http://dublincore.org/documents/2005/08/15/usageguide/qualifiers.shtml
Is Replaced By: Not Applicable
Is Part Of: http://dublincore.org/documents/2005/11/07/usageguide/
Latest version: http://dublincore.org/documents/usageguide/qualifiers.shtml
Translations: http://dublincore.org/resources/translations/
Status of document: DCMI Recommended Resource
Description of document: This document describes the principles governing Dublin Core qualifiers, the two categories of qualifiers, and lists instances of qualifiers approved by the Dublin Core Usage Board as well as guidance for their use.

5. Dublin Core Qualifiers

This document presents in part the results of an ongoing process to develop exemplary terms extending or refining the original 15 elements of the Dublin Core Metadata Element Set (DCMES). The terms or "qualifiers" listed here were identified, generally in working groups of the Dublin Core Metadata Initiative, (DCMI) and judged by the DCMI Usage Board to be in conformance with principles of good practice for the qualification of Dublin Core metadata elements.

In determining the makeup of these qualifiers, preference was given to vocabularies, notations, and terms already maintained by established agencies. It should be emphasized that the list of externally-maintained vocabularies identified here is a preliminary list. There are many more controlled vocabularies or classification systems that are not on this list. Detail on currently recommended schemes are listed at: DCMI Encoding Schemes - a current list

Inevitably, there will be situations where an agent or client will encounter DCMES descriptions that use unfamiliar qualifiers developed by implementors for specialized local or domain-specific needs. The useful interpretation of such a DCMES description will depend on the ability of an application to ignore the unknown qualifiers and fall back on the broader meaning of the element in its unqualified form. The guiding principle for the qualification of Dublin Core elements, colloquially known as the "Dumb-Down Principle," is that a client should be able to ignore any qualifier and use the information as if it were unqualified. While this may result in some loss of specificity, the remaining element value (without the qualifier) should continue to be generally correct and useful for discovery.

It is expected that implementors will develop additional qualifiers for use within local applications or specific domains. Such qualifiers may not be understood by other applications. However, qualifiers that conform to the principles of qualification defined here are more likely to be reusable by other communities within the broader context of cross-domain discovery.

At the time of the ratification of this document, the DCMI recognizes two broad classes of qualifiers:

- **Element Refinement.** These qualifiers make the meaning of an element narrower or more specific. A refined element shares the meaning of the unqualified element, but with a more restricted scope. A client that does not understand a specific element refinement term should be able to ignore the qualifier and treat the metadata value as if it were an unqualified (broader) element. The definitions of element refinement terms for qualifiers must be publicly available.
- **Encoding Scheme.** These qualifiers identify schemes that aid in the interpretation of an element value. These schemes include controlled vocabularies and formal notations or parsing rules. A value expressed using an encoding scheme will thus be a token selected from a controlled vocabulary (e.g., a term from a classification system or set of subject headings) or a string formatted in accordance with a formal notation (e.g., "2000-01-01" as the standard expression of a

date). If an encoding scheme is not understood by a client or agent, the value may still be useful to a human reader. The definitive description of an encoding scheme for qualifiers must be clearly identified and available for public use.

All of the qualifiers listed in this document fall into one of these two categories. Specific guidance is given below for element refinements. If a particular encoding scheme is available for the element and or/element refinement, its application is generally described either in this document or in documentation available with the encoding scheme itself. Audience, Provenance and RightsHolder, which are at the element level but not one of the original 15 elements, are described along with the other elements.

Additional qualifier categories may evolve over time and with implementation experience. The qualifiers listed here do not constitute a closed set, designed to meet all of the descriptive needs of implementors. Rather, they form the foundation for a larger body of qualifiers that will evolve as additional qualifiers are developed by various communities, some of which may eventually be submitted to the DCMI Usage Board for review and approval. Implementors may deploy the qualifiers on these lists with confidence that they conform to the Dumb-Down Principle, and are encouraged to use these qualifiers as examples to guide development of local qualifiers for Dublin Core metadata elements.

Summary Refinement and Scheme Table

This summary of the element refinements and schemes is provided for the convenience of users. Terms in this summary may have the status of "recommended" or "conforming." The reference definitions and status indications may be found in DCMI Terms. Click on the term to go directly to the reference definition for that term.

DCMES Element	Element Refinement(s)	Element Encoding Scheme(s)
Title	Alternative	-
Creator	-	-
Subject	-	LCSH MeSH DDC LCC UDC
Description	Table Of Contents Abstract	-
Publisher	-	-
Contributor	-	-
Date	Created Valid Available Issued Modified Date Accepted Date Copyrighted Date Submitted	DCMI Period W3C-DTF
Type	-	DCMI Type Vocabulary
Format	-	IMT
	Extent	-
	Medium	-
Identifier	-	URI
	Bibliographic Citation	-
Source	-	URI
Language	-	ISO 639-2RFC 3066

Relation	Is Version Of Has Version Is Replaced By Replaces Is Required By Requires Is Part Of Has Part Is Referenced By References Is Format Of Has Format Conforms To	URI
Coverage	Spatial	DCMI Point ISO 3166 DCMI Box TGN
	Temporal	DCMI Period W3C-DTF
Rights	Access Rights	-
	License	URI
Audience	Mediator Education Level	-
Provenance	-	-
Rights Holder	-	-
Instructional Method	-	-
Accrual Method	-	-
Accrual Periodicity	-	-
Accrual Policy	-	-

Properties of Dublin Core Qualifiers

Dublin Core qualifiers have the following properties:

- **Name:** The unique token assigned to the qualifier.
- **Label:** The human-readable label assigned to the qualifier.
- **Definition:** A statement that represents the concept and essential nature of the qualifier.
- **Comment:** Additional information associated with the qualifier (if available).
- **See Also:** A link to more information about the qualifier (if available).

For the up-to-date specification of all metadata terms maintained by the Dublin Core Metadata Initiative, including elements, element refinements, encoding schemes, and vocabulary terms (the DCMI Type Vocabulary), see http://dublincore.org/documents/dcmi-terms/. In the listing below, the Name and Label attributes are the same as in the specification, but the Definition and Comment appear together as "Term Description", and guidance and examples are added.

Multiple Language Encodings of Dublin Core Entities

Dublin Core qualifiers will be expressed in languages other than English. A single invariant token assigned to each qualifier -- the Name property -- stands for a given qualifier concept irrespective of the language in which it is defined. This token can be incorporated into a URI to form a unique identifier for the qualifier. All other properties of a qualifier (Label, Definition, Comment, and aspects of See Also as appropriate) can be translated from English into any other language.

All other properties of Dublin Core entities (Label, Definition, Comment, and aspects of See Also as appropriate) will be expressed in the language and character set of the translation.

Element Refinements

These element refinement terms are extensions to the "Simple Dublin Core" 15 elements or to the additional element terms Audience, Provenance and RightsHolder.

Refinement(s) for element: Title

Alternative

Label: Alternative

Term description: Any form of the title used as a substitute or alternative to the formal title of the resource. This qualifier can include Title abbreviations as well as translations.

Guidelines for creation of content:

An alternative title can be used to provide access to secondary titles, but should only be used when a value is present in the Title element.

Examples:

Alternative="AMA newsletter" (Title="American Meteorological Association newsletter")
Alternative="Ocho semanas" (Title="Eight weeks")

Refinement(s) for element: Description

tableOfContents

Label: Table of Contents

Term description: A list of subunits of the content of the resource.

Guidelines for creation of content:

When a description of a resource consists of a list of the contents, whether from a menu or other mechanism, tableOfContents can be used to differentiate this list from descriptive text that is written in sentence form. This allows more options for display and indexing.

Examples:

tableOfContents="Introduction; Vertebrates; Invertebrates; Molluscs"

Abstract

Label: Abstract

Term description: A summary of the content of the resource.

Guidelines for creation of content:

Used when a description of a resource consists of a formal abstract. For implementations where formal abstracts are preferred, using the specific term allows the label to better reflect the level of the description.

Examples:

Abstract="This article describes the work of the IFB Chaos Committee, including a summary of its major findings."

Refinement(s) for element: Date

Date refinements are generally useful in situations where more than one date is needed, and the difference between the dates may be important to users. Note that the first five Date refinement terms were among the earlier ones approved by DCMI, and the naming convention of the time was not to include "date" as part of the refined term. The most recent ones reflect changes in the naming convention used, in which the name of the refined term expresses more clearly the relationship to the parent element. When using date refinements it can be unwise to insert a text string that repeats the distinction created by the refinement itself. For instance, the string "Valid 20010211" in a statement where the refinement "valid" is used might show up in a labelled display as: VALID: Valid 20010211.

Created

Label: Created

Term description: Date of creation of the resource.

Guidelines for creation of content:

If the date of creation of the resource is known, and that date is important to note specifically (e.g., there are other relevant dates to record), use the term Created for the creation date of the resource. Note that the "one-to-one" rule requires that the creation date be that of the resource being described, not any early version from which the current resource is derived.

Valid

Label: Valid

Term description: Date (often a range) of validity of a resource.

Guidelines for creation of content:

If the resource is only valid or relevant for a particular date or range of dates, the term Valid may be used to express those dates. This may be particularly important if the resource will be retained over time but its use is valid only during a particular period or until a particular date.

Available

Label: Available

Term description: Date (often a range) that the resource will become or did become available.

Guidelines for creation of content:

In general, the term Available should be used in the case of a resource for which the date of availability may be distinct from the date of creation, and the date of availability is relevant to the use of the resource.

Issued

Label: Issued

Term description: Date of formal issuance (e.g., publication) of the resource.

Guidelines for creation of content:

The term Issued should be applied when a formal date of issuance or publication is relevant to the resource, and is distinct from other dates that may be used with the resource.

Modified

Label: Modified

Term description: Date on which the resource was changed.

Guidelines for creation of content:

Modified dates may be used to record either all instances of modification or only the latest. When only one modified date is recorded, it is assumed to be the latest.

dateAccepted

Label: Date Accepted

Term description: Date of acceptance of the resource (e.g. of thesis by university department, of article by journal, etc.).

Guidelines for creation of content:

If, in the lifecycle of a resource, the date of acceptance by a formal body or entity is relevant to the use of the resource, dateAccepted may be used.

dateCopyrighted

Label: Date Copyrighted

Term description: Date of a statement of copyright.

Guidelines for creation of content:

If, in the lifecycle of a resource, the date of copyright is relevant to the use of the resource, dateCopyrighted may be used.

dateSubmitted

Label: Date Submitted

Term description: Date of submission of the resource (e.g. thesis, articles, etc.).

Guidelines for creation of content:

If, in the lifecycle of a resource, the date of submission to a body or entity is relevant to the use of the resource, dateSubmitted may be used.

Refinement(s) for element: Format

Extent

Label: Extent

Term description: The size or duration of the resource.

Guidelines for creation of content:

Because the refinement Extent is used in a variety of situations, it generally consists of both a numeric value and a caption that is needed to interpret the numeric value. Best practice is to separate the numeric value and the caption with a space, whether the caption appears before or after the value.

Examples:

Extent="folio"
Extent="899 Kb"
Extent="21 minutes"

Medium

Label: Medium

Term description: The material or physical carrier of the resource.

Guidelines for creation of content:

Medium is generally used when the resource is of a physical nature, for instance a painting or model, where the physical carrier or material used is relevant to the user. For instance, if the resource is a movie on DVD, and is only available as a physical object, it should be described as such. If it is available digitally, for download or presentation on a website, its format would be reflected in the Format element. Note that, because of the physical nature of materials described with this refinement, the encoding scheme "IMT" is not valid for use with Medium.

Examples:

Medium="cotton fabric with sequins"
Medium="bronze on wooden pedestal"
Medium="oil on wood"

Refinement(s) for element: Relation

Most of the refinements of Relation are expressed as "reciprocals" and may be used to link resources in two directions, though this is not required. Implementors need not describe both or all resources involved in a reciprocal relationship to express the relationship--in other words, they may describe a later version and relate it to the earlier without having the need or opportunity to describe the earlier, and vice versa. In some of the relationships below, maintaining reciprocality is more important. In others, one direction of the relationship is more relevant that the other. These differences will be mentioned in the guidelines for specific terms.

In All cases, either a string or a URI may be used as a value. If a URI is used, the scheme should be designated.

Examples for Relation refinements can be found with the Relation element. When using Relation refinements, do not use embedded text labels, as the examples illustrate.

isVersionOf

Label: Is Version Of

Term description: The described resource is a version, edition, or adaptation of the referenced resource. Changes in version imply substantive changes in content rather than differences in format.

Guidelines for creation of content:

Use only in cases where the relationship expressed is at the content level. Relationships need not be close for the relationship to be relevant. "West Side Story" is a version of "Romeo and Juliet" and that may be important enough in the context of the resource description to be expressed using isVersionOf. The Broadway Show and the movie of "West Side Story" also relate at a similar level, but the video and DVD of the movie are more usefully expressed at the level of format, the content being essentially the same.

See also isFormatOf.

hasVersion

Label: Has Version

Term description: The described resource is a version, edition, or adaptation of the referenced resource. Changes in version imply substantive changes in content rather than differences in format.

Guidelines for creation of content:

See isVersionOf for basic guidelines.

isReplacedBy

Label: Is Replaced By

Term description: The described resource is supplanted, displaced, or superseded by the referenced resource.

Guidelines for creation of content:

When establishing a chain of versions, where only one version is valid, the use of isReplacedBy and Replaces allows the relationship to be expressed and the user directed to the appropriate version. In this case, the reciprocal relationships are quite important.

Replaces

Label: Replaces

Term description: The described resource supplants, displaces, or supersedes the referenced resource.

Guidelines for creation of content:

See isReplacedBy for basic guidelines.

isRequiredBy

Label: Is Required By

Term description: The described resource is required by the referenced resource, either physically or logically.

Guidelines for creation of content:

In the case of IsRequiredBy and Requires, there is a clearer need to express the Requires relationship than the IsRequiredBy, though both can be useful. This relationship is most often seen in relationships between software and documents or applications and hardware and/or software requirements.

Requires

Label: Requires

Term description: The described resource requires the referenced resource to support its function, delivery, or coherence of content.

Guidelines for creation of content:

See isRequiredBy for basic guidelines.

isPartOf

Label: Is Part Of

Term description: The described resource is a physical or logical part of the referenced resource.

Guidelines for creation of content:

The isPartOf and hasPart relationships are essentially "parent/child" relationships--hierarchical in nature. With them can be expressed both one-to-one and one-to-many types of relationships.

hasPart

Label: Has Part

T*erm description:* The described resource includes the referenced resource either physically or logically.

Guidelines for creation of content:

See isPartOf for basic guidelines.

isReferencedBy

Label: Is Referenced By

Term description: The described resource is referenced, cited, or otherwise pointed to by the referenced resource.

Guidelines for creation of content:

The isReferencedBy and References refinements enable the expression of relationships that aid the user but are not necessary tied to the lifecycle or necessary for the intended use of the resource. This relationship might be used to link an article critical of a resource to that resource, a satire of a speech to the original speech, etc.

References

Label: References

Term description: The described resource references, cites, or otherwise points to the referenced resource.

Guidelines for creation of content:

See isReferencedBy for basic guidelines.

isFormatOf

Label: Is Format Of

Term description: The described resource is the same intellectual content of the referenced resource, but presented in another format.

Guidelines for creation of content:

This relationship is explicitly for the expression of relationships between resources for which format is the primary variable.

Because Dublin Core maintains the principle of "one-to-one," each resource is expected to have its own description.

See also isVersionOf.

hasFormat

Label: Has Format

Term description: The described resource pre-existed the referenced resource, which is essentially the same intellectual content presented in another format.

Guidelines for creation of content:

See isFormatOf for basic guidelines.

conformsTo

Label: Conforms To

Term description: A reference to an established standard to which the resource conforms.

Guidelines for creation of content:

The standards referenced might be educational standards, accessibility standards, or any other established standard that is relevant to the use of the resource.

Refinement(s) for element: Coverage

Spatial

Label: Spatial

Term description: Spatial characteristics of the intellectual content of the resource.

Guidelines for creation of content:

Spatial characteristics may include geographic names, latitude/longitude, or other established georeferenced values. Clearly, this refinement does not allow complex or sophisticated georeferencing, but attention to standard schemes and controlled vocabularies should provide useful results. Controlled vocabulary terms can be drawn from recommended vocabularies, or standard labelling within the value can provide useful assistance to users and applications. For additional information on encoding spatial information see the DCMI Box Encoding Scheme and the DCMI Point Encoding Scheme.

Examples:

Spatial="Chicago, Ill."
Spatial="Lat: 44 00 00 S Long: 068 00 00 W Name: Patagonia"
Spatial="Upstate New York"

Temporal

Label: Temporal

Term description: Temporal characteristics of the intellectual content of the resource.

Guidelines for creation of content:

Temporal characteristics include those aspects of time that relate to the intellectual content of a resource and not its lifecycle. Examples might include a resource describing some aspect of the 19th century but itself created this year. In that case, the Temporal Coverage would be the 19th century, and the Date (or Date Created) would be 2003. Values can be text strings or encoded values. Specific suggestions for encoding Temporal characteristics may be found in the DCMI Period Encoding Scheme.

Examples:

Temporal="Jurassic Period"
Temporal="1922-1978"
Temporal="Twentieth Century"

Refinement(s) for element: Audience

Mediator

Label: Mediator

Term description: A class of entity that mediates access to the resource and for whom the resource is intended or useful. The audiences for a resource are of two basic classes: (1) an ultimate beneficiary of the resource, and (2) frequently, an entity that mediates access to the resource. The mediator element refinement represents the second of these two classes.

Guidelines for creation of content:

In an educational setting, a teacher might be designated the Mediator for a resource intended for use by a teacher in a classroom of students of a particular level or sharing other similar characteristics. Resources intended to be used directly by those same students would not include a Mediator. Mediators may be expressed in more or less specific terms, depending on the needs of the implementation. Controlled vocabularies can be useful in distinguishing Mediators.

Examples:

Mediator="Reading specialist"
Mediator="ESL teachers"

educationLevel

Label: Education Level

Term description: A general statement describing the education or training context. Alternatively, a more specific statement of the location of the audience in terms of its progression through an education or training context.

Guidelines for creation of content:

Commonly, this term would be used for a grade level for materials intended for an educational setting. Although no specific controlled vocabulary has been recommended for use with educationLevel, consistent use of terminology or reliance on an available controlled vocabulary enables more consistent results.

Examples:

educationLevel="elementary school students"
educationLevel="4th-5th grade"
educationLevel="secondary science"

Refinement(s) for element: Rights

accessRights

Label: Access Rights

Term description: Information about who can access the resource or an indication of its security status. Access Rights may include information regarding access or restrictions based on privacy, security or other regulations.

Guidelines for creation of content:

Access rights is intended to allow the characterization of restrictions to view, search or use resources, based on attributes of the resource itself or the class or category of user. An example would be a resource that was restricted to users holding a particular security clearance, or one that required login or authentication at a particular website.

Examples:

accessRights="Available to subscribers only."
accessRights="Viewable by Medium security cleared staff only."

license

Label: License

Term description: A legal document giving official permission to do something with the resource. Recommended best practice is to identify the license using a URI. Examples of such licenses can be found at http://creativecommons.org/licenses/.

Guidelines for creation of content:

License is designed to allow the inclusion of specific licensed uses to be specified. An example would be a resource that was available to be used freely but not for reproduction within commercial applications.

Examples:

license="http://creativecommons.org/licenses/by-nc-nd/2.0/ legalcode"
license="Licensed for use under Creative Commons Attribution 2.0."

Refinement(s) for element: Identifier

bibliographicCitation

Label: Bibliographic Citation

Term description: A bibliographic reference for the resource. Recommended practice is to include sufficient bibliographic detail to identify the resource as unambiguously as possible, whether or not the citation is in a standard form.

Guidelines for creation of content:

Because this term is not describing a relationship to another resource, it should be limited to citations to the resource described in the remainder of the record. For instance, if the resource is an article for a journal, it is appropriate to include very specific information about the article, even page references, if such information is used to cite the article in a standard format for reference by other resources, *even if the article being described is in a digital format*.

Examples:

bibliographicCitation="ESOP, v.2, no. 1, Apr. 2003, p. 5-8"
bibliographicCitation="Nature, v.87, p. 200"

For additional guidance on using this refinement, see: Guidelines for Encoding Bibliographic Citation Information in Dublin Core.

Metadata associated with this resource: http://dublincore.org/documents/usageguide/qualifiers.shtml.rdf

---

2009-08-14 Pete on "Using Dublin Core"

> This message came in last week with a suggestion to make the
> 1.1 specification more user-friendly by including more explanation.

On his specific point, there _are_ "examples of values and detailed description of use for each element" in

[1] http://dublincore.org/documents/usageguide/elements.shtml

which pretty much parallels

[2] http://dublincore.org/documents/usageguide/qualifiers.shtml

Having said that, it seems to me many of those examples are thoroughly misleading. The Usage Guide in general makes no distinction between literals and non-literals, and [2] in particular gives the impression that literal values should be used for many of the properties that have been described for the last couple of years as having non-literal ranges. :-(

We have it as an item in the workplan to make a start on a comprehensive revision of the Usage Guide and related documentation. See first item in

http://dublincore.org:8080/usage/minutes/2009/2009-06-17.dcub-telecon-report.html

And I guess one thing to consider here would be whether any of the documentation currently in the Usage Guide should be added (or transferred) to

http://dublincore.org/documents/dcmi-terms/

and

http://dublincore.org/documents/dces/

We should also be thinking about serving that content (or appropriate subsets of it) as human-readable representations of the "namespace documents" (for which currently only RDF/XML is served)

2009-08-10
Subject: FW: Extended Date Time Format
To: DC-USAGE@JISCMAIL.AC.UK

This might be something else to bear in mind when we're looking at revising the bit of the Usage Guide dealing with dates...

> From: Pete Johnston
> Sent: 10 August 2009 17:17
> To: 'dc-date@jiscmail.ac.uk'
> Subject: Extended Date Time Format
>
> I'm not sure if this list is still active, but FWIW, the
> Extended Date Time Format datatype developed by the Library
> of Congress
>
> http://www.loc.gov/standards/datetime/
>
> looks as if it addresses several of the requireements that
> this group considered in the past.
>
> I can see how to refer to the datatype using an XML
> QName/expanded name, but from those docs, I'm not sure what
> its URI is for use in RDF (but I only skimmed quickly so I
> may have missed it).

2009-07-23

See message below from Rebecca to dc-architecture re LoC planning on

making ISO639-2 available as SKOS/RDF.

Currently, DCMI has defined a Syntax Encoding Scheme for ISO639-2, i.e. DCMI models it as a "set of strings".

If I understand the message below correctly (and we really need to wait and see), LoC will be modelling it as a "set of things", so we may need to think about documenting the distinctions, and illustrating how the two are used.

This is the sort of thing which it would be good to cover in detail in the revised Usage Guide, I think.

> Rebecca S Guenther
> To: DC-ARCHITECTURE@JISCMAIL.AC.UK
>
> I have not been following this whole thread, but I will report that the Library of Congress, which is the maintenance agency for ISO 639-2, will be making that standard available as RDF/XML in the future (using SKOS). Thus of course we will be providing URIs for entities which are languages and using altLabels in SKOS to specify the alternative codes. (If you're not familiar with it, ISO 639-2 has 20 languages that have alternative codes for historical reasons, called the bibliographic and terminology codes, but these are too be considered synonyms to represent the same language, e.g. "fra" and "fre" for French.) There will also be relationships asserted to the other language code lists (639-1, 639-3 and 639-5).
>
> It will be under http://id.loc.gov and hopefully available by November or so.

2009-08-30

As someone new to implementing Dublin Core, I have a suggestions to make it a bit easier to start using it. On the simplified DC page (dublincore.org/documents/dces/) it would be great if you listed some examples of values and detailed description of use for each element the same way you did on the qualifiers page (dublincore.org/documents/usageguide/qualifiers.html).

DCMI Glossary

Creator: Mary S. Woodley
Contributor: Gail Clement
Contributor: Pete Winn
Date Issued: 2005-11-07
Identifier: http://dublincore.org/documents/2005/11/07/usageguide/glossary.shtml
Replaces: http://dublincore.org/documents/2005/08/15/usageguide/glossary.shtml
Is Replaced By: Not applicable
Is Part Of: http://dublincore.org/documents/2005/11/07/usageguide/
Latest version: http://dublincore.org/documents/usageguide/glossary.shtml
Status of Document: DCMI Recommended Resource

The DCMI Glossary is a collaborative effort of the User Guide Committee with special thanks to Gail Clement & Pete Winn, whose original glossary was a basis for this version. Terms included in this glossary are based on DCMI documents, presentations at DC conferences, and discussions on the DC General listserv. We welcome comments and feedback regarding additions, deletions or changes to the terms and/or definitions found below.

The glossary was last updated on 23 April 2004.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Related Standards

1:1 principle
The principle whereby related but conceptually different entities, for example a painting and a digital image of the painting, are described by separate metadata records

A

AACR2
See Anglo-American Cataloguing Rules

administrative metadata
Metadata used in managing and administering information resources, e.g., location or donor information. Includes rights and access information, data on the creation and preservation of the digital object.

AGLS
See AGLS (Australian Government Locator Service)

Anglo-American Cataloguing Rules (AACR2)
The dominant bibliographic standard regulating cataloging in the English-speaking world. AACR2 represents a set of rules for the standard description of and access to all materials which a library holds or to which it has access.

American Standard Code for Information Interchange (ASCII)
A scheme that provides standard numeric values to represent letters, numbers, punctuation marks and other characters. The use of standard values allows computers and computer programs to exchange data.

application profile
In DCMI usage, an application profile is a declaration of the metadata terms an organization, information resource, application, or user community uses in its metadata. In a broader sense, it includes the set of metadata elements, policies, and guidelines defined for a particular application or implementation. The elements may be from one or more element sets, thus allowing a given

application to meet its functional requirements by using metadata elements from several element sets including locally defined sets. For example, a given application might choose a specific subset of the Dublin Core elements that meets its needs, or may include elements from the Dublin Core, another element set, and several locally defined elements, all combined in a single schema. An application profile is not considered complete without documentation that defines the policies and best practices appropriate to the application.

Appropriate values

Best practice for a particular Element or Qualifier may vary by context. Definitions may provide some guidance; other information may be found in *"Using Dublin Core"*.

ASCII

See American Standard Code for Information Interchange

Audience

Dublin Core element to record a class of entity for whom the resource is intended or useful. A class of entity may be determined by the creator or the publisher or by a third party. See also *"Using Dublin Core"*.

Author

See Creator

authority control

A set of rules or procedures that assist in the maintenance of consistent forms of names or terms within a database.

authority file

A collection of authority records.

authority record

A record that registers the preferred form of a personal or corporate name, geographic region or subject term. It may indicate variant forms of the established heading, biographical or cultural information associated with the heading, as well as related headings.

B

Basic Semantics Register

An ISO Standard ISO/TS 16668:2000 which identifies and defines semantic components for use in data exchange.

best practice

Guidance and documentation to describe and standardize the use of metadata elements that best support a community's needs.

BSR

See Basic Semantics Register

C

case-sensitive

Lower and upper case letters are not treated as if they were interchangeable; e.g. 'a' is not the same as 'A'.

CEN

European Committee for Standarization (Comité Européen de Normalisation; Europäisches Komitee für Normung) "CEN is contributing to the objectives of the European Union and European Economic Area with voluntary technical standards which promote free trade, the safety of workers and consumers, interoperability of networks, environmental protection, exploitation of research and development programmes, and public procurement." http://www.cenorm.be/cenorm/index.htm

classification

A logical scheme for arrangement of knowledge, usually by subject. Classification schema are alpha and/or numeric; for example, Library of Congress Classification, Dewey Classification, Universal Decimal Classification.

controlled vocabulary

A prescribed set of consistently used and carefully defined terms.

Contributor

The Dublin Core element used to designate the entity responsible for making contributions to the content of the resource. Examples of a Contributor include a person, an organization or a service. Typically, the name of a Contributor should be used to indicate the entity. See also *"Using Dublin Core"*.

Coverage

The Dublin Core element used to designate the extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary, and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of co-ordinates or date ranges. See also *"Using Dublin Core"*.

Creator

The Dublin Core element used to designate the entity primarily responsible for making the content of the resource. Examples of a Creator include a person, an organization, or a service. See also *"Using Dublin Core"*.

Crosswalk

A table that maps the relationships and equivalencies between two or more metadata schemes. Crosswalks or metadata mapping support the ability of search engines to search effectively across heterogeneous databases.

D

Date

The Dublin Core element used to designate the date associated with an event in the life cycle of the resource. Typically, Date will be associated with the creation or availability of the resource. See also *"Using Dublin Core"*.

DCAPS

Dublin Core Application Profile " is a declaration specifying which metadata terms an organization, information provider, or user community uses in its metadata. By definition, a DCAP identifies the source of metadata terms used - whether they have been defined in formally maintained standards such as Dublin Core, in less formally defined element sets and vocabularies, or by the creator of the DCAP itself for local use in an application. Optionally, a DCAP may provide additional documentation on how the terms are constrained, encoded, or interpreted for application-specific purposes." See http://www.ukoln.ac.uk/metadata/dcmi/dc-elem-prop/

DCMES

Dublin Core Metadata Element Set. See Dublin Core.

DCMI

See Dublin Core Metadata Initiative

DCMI recommendation

A DCMI recommendation is a human-readable document that may define one or more DCMI terms.

DCMI term

A DCMI term is a DCMI element, a DCMI qualifier or term from a DCMI-maintained controlled vocabulary. Each DCMI term is defined in a DCMI recommendation and is identified by a Uniform Resource Identifier (URI) within a DCMI namespace.

DCMI term declaration

A DCMI term declaration is the machine-processable representation of one or more DCMI terms, expressed in a schema language.

Description

The Dublin Core element used to designate a textual description of the content of the resource. See also *"Using Dublin Core"*.

DCSV

See Dublin Core Structured Value

descriptive metadata

Metadata that supports the discovery of an object.

digital tourist

An inexperienced searcher in the digital environment who does not possess knowledge of community- specific vocabularies. The Dublin Core provides a rudimentary vocabulary, or "pidgin language" for information discovery when exploring new digital territories. Coined by Ricky Erway at the Metadata Workshop on Metadata for Networked Images, September 24-25, 1996.

discovery software

A computer application designed to simplify, assist and expedite the process of finding information resources.

Digital Object Identifier

DOI was developed by the International DOI Foundation as a system for identifying and exchanging intellectual property in the digital environment. It provides a mechanism to link a searcher to digital content and facilitates copyright management.

Document Type Definition (DTD)

In SGML or XML, a formal description of the components of a specific document or class of documents. DTDs provide a formal grammar used for machine processing (parsing) of documents expressed in SGML or XML. A DTD description includes:

- The containers or elements that make up the document; e.g., paragraphs, headings, list items, figures, tables, etc.
- The logical structure of the document; e.g., chapters containing sections, etc.
- Additional information associated with elements (known as attributes); e.g., identifiers, date stamps, etc.

document-like object (DLO)

Originally defined as an entity that resembles a document from the standpoint that it is substantially text-based and shares other properties of a document; e.g., electronic mail messages or spreadsheets. The definition was expanded at the 3rd DC workshop to refer to any discrete information resource that are characterized by being fixed (i.e., having identical content for each user). Examples include text, images, movies, and performances.

DOI

see Digital Object Identifier

dot.syntax

A mechanism for refining the meaning of the element in HTML; for example, <META NAME="DC. Title.Alternative" CONTENT="*Title*">

DTD

See Document Type Definition

Dublin Core

The Dublin Core is a metadata element set. It includes all DCMI terms (that is, refinements, encoding schemes, and controlled vocabulary terms) intended to facilitate discovery of resources. The Dublin Core has been in development since 1995 through a series of focused invitational workshops that gather experts from the library world, the networking and digital library research communities, and a variety of content specialties. See the Dublin Core Web Site for additional information.

Dublin Core Metadata Initiative

The Dublin Core Metadata Initiative is the body responsible for the ongoing maintenance of Dublin Core. DCMI is currently hosted by the OCLC Online Computer Library Center, Inc., a not-for-profit

international library consortium. The work of DCMI is done by contributors from many institutions in many countries. DCMI is organized into Communities and Task Groups to address particular problems and tasks (see the DCMI Work structure page). Participation in DCMI is open to all interested parties. Instructions for joining can be found at the DCMI web site on the DCMI Contact information page.

Dublin Core Simple

See Simple Dublin Core

Dublin Core Structured Values

DCSV recognizes two types of substrings: labels and values. A label is the name of the type of a value, and a value is the data itself. A value that is comprised of components, i.e. a value which has its own label and value, is called a structured value. Punctuation supports the parsing of the DCSV.

Dublin Core Terms

See DCMI term

Dumb-down Principle

The qualification of Dublin Core Elements is guided by a rule known colloquially as the Dumb-Down Principle. According to this rule, a client should be able to ignore any qualifier and use the value as if it were unqualified. While this may result in some loss of specificity, the remaining term value (minus the qualifier) must continue to be generally correct and useful for discovery. Qualification is therefore supposed only to refine, not extend the semantic scope of an Element.

E

EAD

see EAD (Encoded Archival Description)

electronic information resource

An information resource that is maintained in electronic, or computerized format, and may be accessed, searched and retrieved via electronic networks or other electronic data processing technologies (e.g., CD-ROM)

element

An element is a property of a resource. As intended here, "properties" are attributes of resources -- characteristics of a resource, such as a Title, Publisher, or Subject. Elements are formally defined terms which are used to describe attributes and properties of a resource.

element refinement (qualifier)

Qualifiers make the meaning of an element narrower or more specific. An element refinement is a property of a resource which shares the meaning of a particular DCMI Element but with narrower semantics. In some application environments (notably HTML-based encodings), Element refinements are used together with elements in the manner of natural-language "qualifiers" (i.e., adjectives) . However, since element refinements are properties of a resource (like elements), element refinements can alternatively be used in metadata records independently of the properties they refine. In DCMI practice, an Element refinement refines just one parent DCMI property.

embedded metadata

Metadata that is maintained and stored within the object it describes; the opposite of stand-alone metadata.

Encoded Archival Description

An SGML DTD that represents a highly structured way to create digital finding aids for a grouping of archival or manuscript materials. The standard is maintained in the Network Development and MARC Standards Office of the Library of Congress (LC) in partnership with the Society of American Archivists. For more information see http://lcweb.loc.gov/ead/.

encoding scheme

An encoding scheme provides contextual information or parsing rules that aid in the interpretation of a term value. Such contextual information may take the form of controlled vocabularies, formal notations, or parsing rules. If an encoding scheme is not understood by a client or agent, the value may still be useful to a human reader. There are two types of encoding schemes: Vocabulary Encoding Schemes and Syntax Encoding Schemes

extensible

Having the potential to be expanded in scope, area or size. In the case of Dublin Core, the ability to extend a core set of metadata with additional elements.

Extensible Markup Language (XML)

A subset of Standard Generalized Markup Language (SGML), a widely used international text processing standard. XML is being designed to bring the power and flexibility of generic SGML to the Web, while maintaining interoperability with full SGML and HTML. For more information, see http://www.w3.org/XML/

F

Format

The Dublin Core element used to designate the physical or digital manifestation of the resource. See also *"Using Dublin Core"*.

FRBR

Functional Requirements for Bibliographic Records is a conceptual model to describe bibliographic entities, their relations and attributes within IFLA (International Federation of Library Associations). The complete work is available at http://www.ifla.org/VII/s13/frbr/frbr.htm and a presentation at http://www.loc.gov/catdir/cpso/frbreng.pdf

G

GIF
See Graphics Interchange Format

GILS
See GILS (Global Information Locator Service)

glossary
An alphabetized list of terms with definitions often created by an organization to reflect its needs. Normally lacks hierarchical arrangement or cross references. Also known as a term list.

Graphics Interchange Format (GIF)
The dominant graphics format on the Web, limited to 256 colors. GIFs provide sharper black & white images than JPEGs.

granularity
The level of detail at which an information object or resource is viewed or described.

H

Harvester
A harvester is a client application that issues OAI-PMH requests. A harvester is operated by a service provider as a means of collecting metadata from repositories. (http://www.openarchives.org/OAI/openarchivesprotocol.html#harvester)

HayStack
An MIT project to develop to personal management systems that would all individuals to organize their information objects, such as email, web pages, documents, images, calendars. The information can be categorize and create relationships that are meaningful for themselves. See http://haystack.lcs.mit.edu/

HTML
See Hypertext Markup Language

Hypertext Markup Language (HTML)
The standard text-formatting language for documents on the World Wide Web. HTML text files contain content that is rendered on a computer screen and markup, or tags, that can be used to tell the computer how to format that content. HTML tags can also be used to encode metadata and to tell the computer how to respond to certain user actions, such as a mouse click. For more information, see http://www.w3.org/MarkUp/.

I

Identifier
The Dublin Core element that is an unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. See also *"Using Dublin Core"*.

IETF
See Internet Engineering Task Force

IEEE LOM
See IEEE LOM (Institute of Electrical and Electronics Engineers. Learning Object Metadata)

IMS
See IMS Instructional Management Systems

IMT
See Internet Media Type

INDECS
See INDECS (Interoperability of Data in D-Commerce Systems)

indexing
The process of evaluating information entities and creating terms that aid in finding and accessing the entity. Index terms may be in natural language or controlled vocabulary or a classification notation.

indexing program
Computer software used to order things; frequently used to refer to software that alphabetizes some or all of the terms in one or more electronic documents.

information resource
Any entity, electronic or otherwise, capable of conveying or supporting intelligence or knowledge; e. g. a book, a letter, a picture, a sculpture, a database, a person. See also DLO

instantiation
An identifiable occurrence or occasion of something; in the case of Dublin Core, a specific occurrence of an information resource.

International Organization for Standardization
ISO was established in 1947 as a worldwide federation of national standards bodies from some 130 countries.

Internet Commons
The global Internet environment, collection of information-bearing repositories whose data can be accessed through the Internet.

Internet Engineering Task Force (IETF)
The IETF is responsible for solving short-term engineering needs of the Internet. It has over 40 Working Groups.

Internet Media Type (IMT)

A set of terms that describe types of resources on the Internet. Used as an encoding scheme for the Format element in Dublin Core. http://www.isi.edu/in-notes/iana/assignments/media-types/media-types

interoperability

The ability of different types of computers, networks, operating systems, and applications to work together effectively, without prior communication, in order to exchange information in a useful and meaningful manner. There are three aspects of interoperability: semantic, structural and syntactical.

Interoperability Qualifiers

Additional metadata used either to refine the semantics of a Dublin Core metadata element's value, or to provide more information about the encoding scheme used for the value.

ISO

See International Organization for Standardization

J

Joint Photographic Experts Group (JPEG)

A standard for compressing digital images. The advantage of JPEG is that it uses compression to make graphics files smaller, making them faster to transfer and view over the World Wide Web. More than 16 million color hues are available. Better than GIF for color photographs. The disadvantage is some loss of image quality due to data loss during compression. For more information see http://www.jpeg.org/

JPEGs

See Joint Photographic Experts Group

K

Keywords

See Subject

L

Language

The Dublin Core element used to designate the language of the intellectual content of the resource. Recommended best practice for the values of the Language element is defined by RFC 3066. See also *"Using Dublin Core"*.

literal

A literal or "appropriate literal" is the value of any given metadata entity that can be either a hyperlink or a string value (literal). A literal affords a great deal of flexibility and power, but increases complexity. Metadata should as well include an appropriate literal that reflects the base value of the metadata entity. For example, in these fragments: creator = "Public, John Q." creator = " http://authority.org/public-john-q-1234" the first has a value expressed as an appropriate literal whereas the second has a (hypothetical) link to an authority structure. It is not entirely clear what a person or application will find at the end of the link, so the metadata should contain an appropriate literal for simple discovery purposes.

M

mapping metadata

See crosswalk

MARC

See MARC (Machine-Readable Cataloging Record)

META tag

The HTML element used to demarcate metadata on a Web page. <META> </META>.

metadata

In general, "data about data;" functionally, "structured data about data." Metadata includes data associated with either an information system or an information object for purposes of description, administration, legal requirements, technical functionality, use and usage, and preservation. . In the case of Dublin Core, information that expresses the intellectual content, intellectual property and/or instantiation characteristics of an information resource. See Section 1.1 of this guide. For a history of the term See Caplan,pp. 1-3.

metadata record

A syntactically correct representation of the descriptive information (metadata) for an information resource. In the case of Dublin Core, a representation of the Dublin Core elements that has been defined for the resource. The majority of metadata records and record fragments in this document are presented in HTML syntax.

metadata schema registry

A publicly accessible system that records the semantics, structure and interchange formats of any type of metadata. A formal authority, or agency, maintains and manages the development and evolution of a metadata registry. The authority is responsible for policies pertaining to registry contents and operation. See also http://www.dlib.org/dlib/may02/wagner/05wagner.html

METS (Metadata Encoding & Transmission Standard
See METS (Metadata Encoding & Transmission Standard

MIME
See Multipurpose Internet Mail Extensions

MODS
See MODS (Metadata Object Description Schema)

Multipurpose Internet Mail Extensions
The standard for attaching files to Internet e-mail messages. Attached files may be text, graphics, spreadsheets, documents, sound files, etc.

N

National Information Standards Organization
NISO, accredited by ANSI, develops and promotes technical standards used in a wide variety of information services.

namespace
A DCMI namespace is a collection of DCMI terms. Each DCMI namespace is identified by a URI. An XML namespace [XML-NAMES] is a collection of names, identified by a URI reference [RFC2396], that are used in XML documents as element types and attribute names. The use of XML namespaces to uniquely identify metadata terms allows those terms to be unambiguously used across applications, promoting the possibility of shared semantics. DCMI adopts this mechanism for the identification of all DCMI terms. For example, the namespace for Dublin Core elements and qualifiers would be expressed respectively in XML as:

xmlns:dc = "http://purl.org/dc/elements/1.1/
xmlns:dcterms = "http://purl.org/dc/terms/

The use of namespaces allows the definition of an element to be unambiguously identified with a URI, even though the label "title" alone might occur in many metadata sets. In more general terms, one can think of any closed set of names as a namespace. Thus, a controlled vocabulary such as the Library of Congress Subject Headings, a set of metadata elements such as DC, or the set of all URLs in a given domain can be thought of as a namespace that is managed by the authority that is in charge of that particular set of terms.

networked resource
An object that is available electronically via a network.

NISO
See National Information Standards Organization

O

OAI
See Open Archives Initiative

OAI-PMH
See Open Archives Initiative Protocol for Metadata Harvesting

OCLC
See Online Computer Library Center

ONIX
See ONIX (ONline InformationeXchange)

Ontology
A hierarchical structure that formally defines the semantic relationship of a set of concepts. Used to create structured / controlled vocabularies for the discovery or exchange of information. A thesaurus, like the AAT is an example.

Online Computer Library Center (OCLC)
The major source of cataloging data for libraries around the world; located in Dublin, Ohio, US. For more information, see http://www.oclc.org/.

Open Archives Initiative
"Develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content. The Open Archives Initiative has its roots in an effort to enhance access to e-print archives as a means of increasing the availability of scholarly communication" For more information see http://www.openarchives.org/organization/index.html.

OpenURL
A method for describing resources and associated resources that are referenced in a network environment. It defines the mentods for transporting these descriptions between networked systems. NISO standard NISO Z39.88-2003 (still in draft). The Standard has been issued in two parts and it available for comment through California Institute of Technology http://library.caltech.edu/openurl/Public_Comments.htm

Open Archives Initiative Protocol for Metadata Harvesting
The Protocol "provides an application-independent interoperability framework based on metadata harvesting. There are two classes of participants in the OAI-PMH framework: Data Providers administer systems that support the OAI-PMH as a means of exposing metadata; and Service Providers use metadata harvested via the OAI-PMH as a basis for building value-added services. " For more information see http://www.openarchives.org/organization/index.html.

OWL

Web ontology language http://www.w3.org/TR/owl-features/. OWL is a language for describing ontologies and schema. It can specify concepts and their relationships. OWL/XDD (XML declaration description) allows a means to express complex rules and constraints.

P

parsing

Parsing may be divided into parts: lexical analysis and semantic parsing. Lexical analysis divides strings into components based on punctuation or tagging. Semantic parsing then attempts to determine the meaning of the string.

Persistent Uniform Resource Locator

An approach to the URL permanence problem proposed by OCLC. A PURL is a public alias for a document. A PURL remains stable, while the document's background URL will change as it is managed (e.g. moved) over time. A PURL is created by a Web administrator who is registered as a PURL "owner" and who maintains a mapping of the PURL to a current and functioning URL. A PURL is a form of URN.

POI (PURL-based Object Identifier)

A specification for resource identifiers that are described by metadata 'items' in OAI-compliant repositories. POI are based on the PURL system [POI]. "Because POIs conform to the URI specification, they can be used unmodified in DC metadata and LOM." See http://www.ukoln.ac.uk/metadata/dcmi-ieee/identifiers/ and http://www.ukoln.ac.uk/distributed-systems/poi/

Property

A property is a specific aspect, characteristic, attribute, or relation used to describe a resource. Dublin Core metadata elements are properties http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/

Provenance

Dublin Core element used for making statements of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity and interpretation. The statement may include a description of any changes successive custodians made to the resource. See also *"Using Dublin Core"*.

Publisher

The Dublin Core element used to designate the entity responsible for making the resource available. Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity. See also *"Using Dublin Core"*.

PURL

See Persistent Uniform Resource Locator

Q

qualifier

"Qualifiers" is the generic heading traditionally used for terms now usually referred to specifically as Element Refinements or Encoding Schemes. A qualifier must follow the Dumb-Down Principle. There are two broad categories of qualifiers: Encoding schema and Element refinement.

Qualified Dublin Core

Qualified Dublin Core includes an additional element, Audience, as well as a group of element refinements (also called qualifiers) that refine the semantics of the elements in ways that may be useful in resource discovery

R

RDF

See Resource Description Framework.

RDF Site Summary

RSS was created and popularized by Netscape for their personalized portal site. Rich Site Summary (RSS) is a lightweight XML application designed to exchange headline metadata between news content providers and portals.

record

A record is some structured metadata about a resource, comprising one or more properties and their associated values. http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/

registry

A system to provide management of metadata elements. See also metadata schema registry The DCMI Registry Working Group (WG) is the development of a metadata registry providing authoritative information regarding the DCMI vocabulary and the relationship between terms in that vocabulary.

Relation

The Dublin Core element used to designate A reference to a related resource. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system. See also *"Using Dublin Core"*.

Request for Comment (RFC)

A Request for Comment (RFC) is the process of establishing a standard on the Internet. Discussion of the proposed standard on the Internet is facilitated by the Internet Engineering Task Force (IETF). Once approved, the standard receives a unique number which identifies it; e.g., RFC See

http://www.isi.edu/rfc-editor/. and http://www.ietf.org/rfc.html

resource
A resource is anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources. http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/

Resource Description Framework (RDF)
The basic language for writing metadata; a foundation which provides a robust flexible architecture for processing metadata on the Internet. RDF will retain the capability to exchange metadata between application communities, while allowing each community to define and use the metadata that best serves their needs. For more information see http://www.w3.org/RDF/

resource discovery
The process through which one searches and retrieves an information resource.

Resource Type
See Type.

Resource Description
See Description.

Resource Identifier
See Identifier

RFC
See Request for Comment

Rights
The Dublin Core element used to provide a link to information about rights held in and over the resource. Typically a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource. See also *"Using Dublin Core"*.

RightsHolder
Dublin Core element used to record a person or organization owning or managing rights over the resource. Recommended best practice is to use the URI or name of the Rights Holder to indicate the entity. See also *"Using Dublin Core"*.

Rights Management
See Rights

ROADS
Resource Organisation And Discovery in Subject based services. A UK funded project whose aim is to develop discovery software for Internet resources.

RSS
See RDF Site Summary .

S

schema or scheme(plurals schemas or schemata; schemes)
In general terms, any organization, coding, outline or plan of concepts. In terms of metadata, a systematic, orderly combination of elements or terms. In terms of DCMI term declarations represented in XML or RDF schema language, schemas are machine-processable specifications which define the structure and syntax of metadata specifications in a formal schema language. In terms of an encoding scheme, is a set of rules for encoding information that supports a specific community of users. See also Encoding scheme.

scheme
See schema

SCORM
See SCORM (Sharable Content Object Reference Model)

search engine
A utility capable of returning references to relevant information resources in response to a query.

semantic interoperability
Ability to search for digital information across heterogeneous distributed databases whose metadata schemas have been mapped to one another. It is achieved through agreements about content description standards; for example, Dublin Core, Anglo-American Cataloging Rules.

Semantic Web
A term coined by Tim Berners-Lee which views the future Web as a web of data, like a global database. The infrastructure of the Semantic Web would allow machines as well as humans to make deductions and organize information. The architectural components include semantics (meaning of the elements), structure (organization of the elements), and syntax (communication). http://www.w3.org/DesignIssues/Semantic.html

semantics
Significance or meaning. In the case of Dublin Core, the significance or intended meaning of individual metadata elements and their components.

SGML
See Standard Generalized Markup Language

SICI
Serial Item and Contribution Identifier (ANSI/NISO Z39.56-1996 Vers. 2) A numeric notation to identify serial issues and articles uniquely regardless of their distribution medium (paper, electronic, microform).

Simple Dublin Core
The fifteen Dublin Core elements used without qualifiers, that is without element refinement or encoding schemes. Sometimes referred to as Dublin Core simple.

SOAP
A protocol that uses XML for the exchange of structured information, that is messages, in a distributed environment. See http://www.w3.org/TR/soap12-part1/

software agent
A computer program that carries out tasks on behalf of another entity. Frequently used to reference a program that searches the Internet for information meeting the specified requirements of an individual user.

Source
The Dublin Core element used to designate a reference to a resource from which the present resource is derived. The present resource may be derived from the Source resource in whole or part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system. See also *"Using Dublin Core"*.

spatial
An element refinement of Coverage. Spatial characteristics of the intellectual content of the resource.

Standard Generalized Markup Language (SGML)
A non-proprietary language/enabling technology for describing information. Information in SGML is structured like a database, supporting rendering in and conversion between different formats. Both XML and later versions of HTML are instances of SGML. For more information see http://www.w3.org/MarkUp/SGML/.

stand-alone metadata
Metadata that is created, maintained and stored independently of the object it describes. The opposite of embedded metadata.

structured value
See Dublin Core Structured Value

structural interoperability
Is achieved through data models for specifying semantic schemas in a way that they can be shared; for example, RDF.

structural metadata
Structural metadata defines the digital object's internal organization and is needed for display and navigation of that object.

sub-element
See element refinement

Subject
The Dublin Core element used to describe the content of the resource. The element may use controlled vocabularies or keywords or phrases that describe the subject or content of the resource. See also *"Using Dublin Core"*.

Subject Headings
An alphabetical list of words or phrases that represent a concept that is under authority control, e. g., the Library of Congress Subject Headings.

surrogate content
Metadata as a substitute for an actual resource.

switching language
A mediating language used to establish equivalencies among various indexing languages. Dublin Core has been viewed as a switching "language" between various metadata schemas.

syntactic interoperability
Achieved by marking up our data in a similar fashion so we can share the data and so that our machines can understand and take the data apart in sensible ways; for example, XML, EAD and MARC.

syntax
The form and structure with which metadata elements are combined. In the case of Dublin Core, the form and structure of how metadata elements and their components are combined to form a metadata record.

Syntax Encoding Schemes
Syntax Encoding Schemes indicate that the value is a string formatted in accordance with a formal notation, such as "2000-01-01" as the standard expression of a date.

T

TAP
A project developed at Stanford, TAP seeks to create a web of "machine-readable" (XML, RDF) data, not just human readable (HTML) data. A server which is queried for information about people or subjects, will collocate documents about people and concepts. See http://tap.stanford.edu/

taxonomy
In general terms, systematic classification according to principles or general laws. In digital terms, automated classification of documents in a hierarchy based on information gathered by a metacrawler. May refer to a classification of DCMI terms. A classification system such as Library of Congress Classification is an example of a taxonomy.

technical metadata
Metadata that documents the creation and the digital characteristics of the files.

TEI
See Text Encoding Initiative

temporal
An element refinement of coverage. Temporal characteristics of the intellectual content of the resource.

term
See DCMI term

Text Encoding Initiative (TEI)
An international project to develop guidelines for the preparation and interchange of electronic

texts for scholarly research as well as a broad range of other language industry uses. The TEI DTD is an SGML Document Type Definition for encoding literary works. For more information, seehttp://www.tei-c.org/

thesaurus

A structured vocabulary make up of names, words, and other information, typically including synonyms and/or hierarchical relationships for the purpose of cross-referencing in order to organize a collection of concepts for reference and retrieval. See the ANSI/NISO Standard for thesaurus construction Z39.19-2003 (R1998; ISO 2788). A controlled vocabulary of terms or concepts that are structured hierarchically (parent/child relationships) or as equivalences (synonyms), and related terms (associative). See also Subject headings and glossary. A thesaurus is a taxonomy.

Thesaurus of Geographic Names

The TGN is a controlled vocabulary containing around 1,000,000 names and other information about places. It includes physical features and administrative entities, such as cities and nations. The emphasis in TGN is on places important for art and architecture.

Title

The Dublin Core element used to designate the name given to the resource. Typically, a Title will be a name by which the resource is formally known. See also *"Using Dublin Core"*.

tokens

The means to denote the status of an element or qualifier within a registry; e.g., proposed, recommended, conforming (to the namespace), obsolete, or local.

Type

The Dublin Core element used to designate the nature or genre of the content of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary. See also *"Using Dublin Core"*.

U

ULAN

See Union List of Artist Names

Unicode

A universal encoding scheme designed to allow interchange, processing and display of the world's principal languages, as well as many historic and archaic scripts. Unicode supports and fosters a multilingual computing world community by allowing computers using one language to "talk" to computers using a different language. A registered trademark of Unicode, Inc.

Unicode Transformation Format, 8-bit (UTF-8)

A temporary form of Unicode that is well suited for routing data through systems that are not designed for Unicode, such as some email servers and Web clients. UTF-8 is an attractive way of storing multilingual data on the Internet, without requiring full Unicode compliance.

Uniform Resource Identifier (URI)

The syntax for all names/addresses that refer to resources on the World Wide Web. For information about Internet addressing, see http://www.w3.org/Addressing/Addressing.html.

Uniform Resource Locator (URL)

A technique for indicating the name and location of Internet resources. The URL specifies the name and type of the resource, as well as the computer, device and directory where the resource may be found. The URL for Dublin Core Metatdata Initiative is <http://dublincore.org/>. For information about Internet addressing, see http://www.w3.org/Addressing/Addressing.html.

Uniform Resource Name (URN)

A URI (name and address of an object on the Internet) that has some assurance of persistence beyond that normally associated with an Internet domain or host name. For information about Internet addressing, see http://www.w3.org/Addressing/Addressing.html.

Union Lists of Artists' Names (ULAN)

Union List of Artist Names. A controlled vocabulary of artists' names and biographical and bibliographic information produced by the Getty Vocabulary Program.

URI

See Uniform Resource Identifier

URL

See Uniform Resource Locator

URN

See Uniform Resource Name

USMARC

See MARC

UTF-8

See Unicode Transformation Format, 8-bit.

V

value qualifier

Value qualifier refers to either an encoding rule or controlled vocabulary that aids in the interpretation of the value within the metatag. See encoding scheme.

vCard

A standard for storing information about individuals or corporations; an electronic business card. For more information, check the Internet Mail Consortium page on personal data exchange.

Vocabulary Encoding Schemes

Vocabulary Encoding Schemes indicate that the value is a term from a controlled vocabulary, such

as the value "China - History" from the Library of Congress Subject Headings.

Vocabulary Terms

The Usage Board maintains the DCMI Type Vocabulary -- a general, cross-domain list of recommended terms that may be used as values for the Resource Type element to identify the genre of a resource. The member terms of the DCMI Type Vocabulary are called Vocabulary Terms.

W

Warwick Framework

An architecture for the interchange of metadata packages, or "containers"; designed to satisfy the need for competing, overlapping, and complementary metadata models. For more information, see http://www.dlib.org/dlib/july96/07weibel.html.

World Wide Web (WWW)

The panoply of Internet resources (text, graphics, audio, video, etc.) that are accessible via a Web browser.

World Wide Web Consortium (W3C)

An international industry consortium founded in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. For additional information see http://www.w3.org/Consortium/.

WWW

See World Wide Web

W3C

See World Wide Web Consortium

X

XML

See Extensible Markup Language

Y

Z

Z39.50

A NISO standard for an application layer protocol for information retrieval which is specifically designed to aid retrieval from distributed servers. http://lcweb.loc.gov/z3950/agency

Dublin Core Projects

Related Metadata Standards

AGLS (Australian Government Locator Service)

A set of 19 descriptive elements based on the Dublin Core which the Australian government departments and agencies can use to improve the visibility and accessibility of their services and information over the Internet.

EAD (Encoded Archival Description)

An SGML DTD that represents a highly structured way to create digital finding aids for a grouping of archival or manuscript materials. The standard is maintained in the Network Development and MARC Standards Office of the Library of Congress (LC) in partnership with the Society of American Archivists. For more information see http://lcweb.loc.gov/ead/.

GILS (Global Information Locator Service)

GILS embraces open standards to implement interoperable searching across diverse, decentralized information 'locators' to return references to all kinds of electronic and non-electronic information resources. Locators are implemented as common semantics for characterizing information resources, i.e. common metadata semantics. Formally known as Government Information Locator Service

IEEE LOM (Institute of Electrical and Electronics Engineers. Learning Object Metadata)

Standard jointly developed by IMS, IEEE, ARIADNE, and ADL/SCORM for describing, exchanging and managing, locating and evaluating learning objects, that is, instructional content, in a digital or non-digital format. The Draft standard dated 15 July 2002 includes nine categories for the metadata: general, life-cycle, meta-metadata, educational, technical, rights, relation, annotation and classification. Includes a mapping to Dublin Core Simple (Annex B, p. 44)

IMS (Instructional Management Systems)

A specification developed by EDUCAUSE (formerly EDUCOM), a consortium of U.S. institutions of higher learning and vendors, for the discovery and description of learning objects. The specification covers a wide range of e-learning related activities, e.g. vocabulary markup, learning design, content packaging, learner information. It became the basis for the IEEE Learning Object Meta-Data (LOM). The specification includes the element names, definitions, datatypes, and field lengths and defines a conceptual structure for the metadata.

INDECS (Interoperability of Data in D-Commerce Systems)

Standard that addresses the management of intellectual property rights and rights transactions for all media. Elements designed to faciliate the exchange of rights information between domain-specific standards.

MARC (Machine-Readable Cataloging Record)

The MARC formats are standards for the representation and communication of bibliographic and related information (authority, holdings, classification, community information) in machine-readable form. MARC 21 grew out of the harmonization of USMARC and CAN/MARC, formerly national standards, and has emerged as an international standard. MARC21 is an implementation of the American National Standard, Information Interchange Format (ANSI Z39.2) and its international counterpart, Format for Information Exchange (ISO 2709). UniMARC was originally designed for conversion between national formats but now has been adopted by some countries as their national standard.

METS (Metadata Encoding & Transmission Standard

"A standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML schema language of the World Wide Web Consortium. The standard is maintained in the Network Development and MARC Standards Office of the Library of Congress, and is being developed as an initiative of the Digital Library Federation."

MODS (Metadata Object Description Schema)

"Intended to be able to carry selected data from existing MARC 21 records as well as to enable the creation of original resource description records. It includes a subset of MARC fields and uses language-based tags rather than numeric ones, in some cases regrouping elements from the MARC 21 bibliographic format."

ONIX (ONline Information eXchange)

Developed by book publisher for the exchange of book trade information between publishers and wholesalers, e-tail and retail booksellers, other publishers, and anyone else involved in the supply chain. Standards are also being developed by publishers for serials. Mapping between ONIX and MARC exists to facilitate the exchange of content from publishers to library cataloging agencies. Consists of more than 236 elements.

SCORM (Sharable Content Object Reference Model)

eLearning metadata standards supported by ADL (Advanced Distributive Learning Initiative). See IEEE LOM

Acknowledgements

Many sources were consulted for the creation of this glossary:

BIBLINK: Objectives, Scope and Glossary

Clement, Gail and Peter Winn. A user guide for simple Dublin Core: glossary (draft). Last updated 05/12/99.

Baca, Murtha, ed. Introduction to metadata: glossary. Version 2.0
http://www.getty.edu/research/institute/standards/intrometadata/index.html

Caplan, Priscilla. *Metadata Fundamentals for all Librarians.* Chicago: American Library Association, 2003.

Lanzi, Elisa. Introduction to vocabularies: enhancing access to cultural heritage information. Los Angeles: Getty Information Institute, 1998. Updated by Patricia Harpring, 2000.
http://www.getty.edu/research/institute/vocabulary/introvocabs/

Moen, William. An Overview of Z39.50, Supplemented by a Case Study of Implementing the Zebra Server Under the Linux Operating System
http://www.unt.edu/wmoen/Z3950/GIZMO/appendix_d.htm

Schemas glossary
http://www.schemas-forum.org/related/glossary.html

Smith, Allison. Terms commonly used in authority control and thesaurus construction. Word document provided to DC-general listserv.

Other useful glossaries

Digital Library Initiative at the Univerisity of Illinois at Urbana-Champaign.

UKOLN Glossary

The online edition of Digital Libraries, by William Arms, (c) 2000 MIT Press, updated with additional material by the author.

Glossary Web Thesaurus Compendium. Provides listings of thesauri by alphabetical order and subject. Has links to related literature and software for building thesauri.

Metadata associated with this resource: http://dublincore.org/documents/usageguide/glossary.shtml.rdf

Glossary

This Web page is a placeholder for a revised version of the DCMI Glossary that was maintained through 2005 as part of "Using Dublin Core" (DCMI's "user guide"). As of August 2009, the expectation is that the content of this page will be developed on a Dublin Core Glossary Update wiki page by a planned DCMI Glossary Update Task Group. This page is intended to show the "look and feel" of a glossary page given the new (2009) design for DCMI Web pages.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

Abstract Model, DCMI

Application profile
Definition.

D

Description Set Profile
Definition.

O

one-to-one principle
The principle whereby related but conceptually different entities, for example a painting and a digital image of the painting, are described by separate metadata records

S

Singapore Framework

Definition.

Metadata associated with this resource: http://dublincore.org/beta/glossary/index.shtml.rdf

---

2008-07-03 From David Snyder <dsnyder@ASCD.ORG> on dc-education
Subject: Help with refinements

Very glad to join this list. I am developing an education-focused metadat a application profile, with the foundation of Dublin Core and the addition of several local elements. I'm finding the discussions, particularly regardi ng Audience vocabulary, to be very relevant to the work I'm doing.

In developing local elements, to be intermingled with DC elements, I'm running up against the question of whether to create elements or refinements - both in the case of a local element and its local refinement, and a local refinement to a DC element.

From DC documentation on refinements, I understand that the refinements provide a more specific or narrow meaning, should act as an 'adjective' to the element's 'noun', and don't expand the scope of the element - and that the "dumb down principle" means a value of a refinement should make sense if someone simply sees the value as that of the refinement's parent element.

What I'm grappling with is how this plays out in actual, real-world applications. For example, take the DC element Date and the various refinements - Date published, Date approved, etc. If the metadata for a resource had <dcterms.dateapproved>2000-04-01</dcterms.dateapproved>, for example, how precisely would a system that did not recognize this refinement know to roll up the value to Date?

Furthermore, what is the function of element refinements in a closed local system which the metadata creator controls (in other words, no need to 'dumb down' since all work is happening in one place) - if any - versus simply having separate elements? Does having this hierarchical relationship among elements/refinements serve any purpose?

Dumb-down

The notions of 'simple DC' and 'qualified DC' are widely used within DCMI documentation and discussion fora. This document does not present a definitive view of what these phrases mean because their usage is somewhat variable. However, in general terms, the phrase 'simple DC' is used to refer to DC metadata that only makes use of *properties* in the Dublin Core Metadata Element Set [DCMES], that does not make any use of *encoding schemes*, and in which each *statement* only contains a *value string*. The phrase 'qualified DC' is used to refer to metadata that makes use of all the features of the abstract model described here.

The process of translating qualified DC into simple DC is normally referred to as 'dumbing-down'. The process of dumbing-down can be separated into three parts: *description dumb-down*, *property* dumb-down and *value* dumb-down. Furthermore, each of these processes can be approached in one of two ways. Informed dumb-down takes place where the software performing the dumb-down algorithm has knowledge built into it about the *descriptions*, *properties* and *values* being used within a specific DCMI metadata application. Uninformed dumb-down takes place where the software performing the dumb-down algorithm has no prior knowledge about the *descriptions*, *properties* and *values* being used.

Based on this analysis, it is possible to outline a 'dumb-down algorithm' matrix, shown below:

	Description dumb-down	**Element dumb-down**	**Value dumb-down**
Uninformed	Generate a separate simple DC *description set* for each *description* in the qualified DC *description set*.	Discard any *statement* in which the *property URI* identifies a *property* that isn't in the Dublin Core Metadata Element Set [DCMES].	Use *value URI* (if present) or *value string* as new *value string*. Discard any *rich representations*. Discard any *encoding scheme URIs*. In cases where no *value URI* or *value string* can be determined, discard the *statement*.
Informed	Generate a separate simple DC *description set* for each of the key entities in the qualified DC *description set*, as defined by the DCMI metadata application.	Recursively resolve sub-property relationships until a recognised *property* is reached and substitute the *property URI* of that *property* for the existing *property URI* in the *statement*. If no recognised *property* is reached, then discard the *statement*. (In many cases, this process stops when a *property* is reached that is not an element refinement.)	Use knowledge of any *rich representations*, separate *descriptions* or the *value string* to create a new *value string*.

Note that software should make use of the DCMI term declarations represented in RDF schema language [DC-RDFS], the DC XML namespace URIs [DC-NAMESPACES] and the appropriate DCMI encoding guidelines (XHTML meta tags, XML, RDF/XML, etc.) [DCMI-ENCODINGS] to automate the resolution of sub-property relationships.

In cases where software is dumbing-down a *description set* containing multiple *descriptions*, it may either generate several 'simpler' *descriptions* (one per *description* in the original *description set*) or a single 'simple' *description* (in which case it will have to determine which is the 'primary' *description* in the original *description set*). This is an application-specific decision.

2009-07-18 DanBri to public-media-annotation@w3.org

> On 18/6/09 11:16, Pierre-Antoine Champin wrote:
> > According to your interpretation, the director of a movie would be
> > covered by the definition, which was indeed not intended (or did I miss
> > something? ;-)
>
> Yes, I was going on what was written, rather than what I thought you
> might've really meant!
>
> Other projects have danced around this problem. Early Dublin Core talked
> about "document like objects" with a strong emphasis towards online
> documents, but then you get the museums and archival community involved,
> and suddenly a physical painting, or a slide from a photos of that
> painting, or a roll of old movie film, all seem also in scope. And it
> turns out (eg. as concluded by http://dublincore.org/workshops/dc3/)
> that documents like objects, online and offline images, can usefully be
> described with many common properties.
>
> It seems you want to limit things to content / works and content-bearing
> objects, rather than admit the entire universe here?
>
> BTW re "Any Resource (as defined by [RFC 3986])", ... I wouldn't bother
> citing http://www.faqs.org/rfcs/rfc3986.html for a definition of
> resource. All it says is
>
> "This specification does not limit the scope of what might be a
> resource; rather, the term "resource" is used in a general sense
> for whatever might be identified by a URI."
>
> I suggest instead "thing", since that's all RFC3986 "resource" means
> these days.
>
> > Would "Any Resource (...) representing a media content" be clearer (this
> > is a question to the WG as well) ? It may seem a little too "concrete"
> > at first sight (e.g. implying "machine representation"), but I think the
> > following paragraph makes it clear it can be more abstract.
>
> Taking my "thing" suggestion, that gives ... "Any thing representing
> media content". This seems a reasonable statement of intent. Perhaps the
> best way to make this concrete (amongst ourselves, if not in the spec)
> is by example. Which of the following are intended to be Media Resources:
>
> 1. a library MARC record describing a DVD
> 2a. a HTML page describing that same DVD (some specific DVD)
> 2b. an Amazon HTML page describing the general class of that DVD (ie.
> that you can buy)
> 3. An MPEG movie embedding metadata describing the film (from the DVD)
> 4. A videotape of the film
> 5. A set of several thousand identical 35mm films with that same content
> 6. A rusty metal tin on a specific shelf of a national archive room,
> containing one of those films
> 7. The abstract notion of Shakespeare's Hamlet
> 8. The abstract notion of the W3C homepage
> 9. A printed, written transcript of the soundtrack to a film
> 10. Information about the creator of (9.).
>
> This is just a rough list, from top of my head. Main points: do you
> include physical artifacts directly? does there need to be a digital
> representation involved somewhere?
>
> Thinking again about
>
> "Any thing/resource representing media content", isn't that almost
> circular or somehow redundant, if "media" is taken to mean the carrier
> or means by which content/information is represented or transmitted?
>
> Can you give some more borderline counter-examples (apart from the
> Director) of situations where something is *not* a media object?

Issues with dcterms:coverage

This discussion of issues with dcterms:coverage follows on a discussion at an August 2007 meeting of the Usage Board in Singapore - see [1].

[1] http://stage.dublincore.org/usageboard/log/2007/2007-08-26.dcub-meeting-notes.html

Because of the problem between 'coverage' on the one hand and 'spatial' and 'temporal' on the other. We have two problems:

* 1. overlap between subject and coverage * 2. subproperty relations between spatial/temporal and coverage

We have four courses of action:

* 1. delete assertion spatial/temporal subproperty of coverage [problem for legacy and namespace policy]

* 2. change definition of spatial/temporal to fit coverage [problem for namespace policy]

* 3. do nothing and accept that spatial/temporal are not really subproperties

* 4. introduce a new coverage or recommend spatial/temporal instead

NOTED: 2004 definition of 'coverage' was significantly different from 2007 -- and we may have introduced a problem in the change in 2007 i.e., constraining 'coverage' by using the word 'topic'

NOTED: DCMI might want to build an AP that ameliorates the legacy problems of 1.1 and TERMS -- 'coverage', 'identifier', 'source', etc. -- this needs to be linked to discussion and decision of the AP of Simple DC.

AGREED: 'spatial' and 'temporal' subproperty assertions are inconsistent with the definition of 'coverage'.

As of 2007-12-03

* Comment: Spatial topic and spatial applicability may be a named place or a location specified by its geographic coordinates. Temporal topic may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies. Recommended best practice is to use a controlled vocabulary such as the Thesaurus of Geographic Names [TGN]. Where appropriate, named places or time periods can be used in preference to numeric identifiers such as sets of coordinates or date ranges.

As of 2006-12-18

* Definition: The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant. * Comment: Spatial topic may be a named place or a location specified by its geographic coordinates. Temporal period may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies. Recommended best practice is to use a controlled vocabulary such as the Thesaurus of Geographic Names [TGN]. Where appropriate, named places or time periods can be used in preference to numeric identifiers such as sets of coordinates

or date ranges.

As of 2002-10-04

* Definition: The extent or scope of the content of the resource. * Comment: Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN] and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges.

DCMI Frequently Asked Questions (FAQ)

What is metadata?

The simplest definition of metadata is " structured data about data."

Metadata is descriptive information about an object or resource whether it be physical or electronic. While metadata itself is relatively new, the underlying concepts behind metadata have been in use for as long as collections of information have been organized. Library card catalogs represent a well-established type of metadata that has served as collection management and resource discovery tools for decades.

Metadata can be generated either "by hand" or derived automatically using software.

Where is the metadata on the Dublin Core website?

We use separate .rdf files that contain the Dublin Core metadata information rather than including the metadata in the HTML. If you look at most of the pages on the DCMI Website, you'll see a link to the metadata at the bottom of each web page.

What is a resource?

In Web terminology, a resource is "anything addressable via a URL." However, Dublin Core implementations are not necessarily Web-based.

Dublin Core metadata can be used to describe any kind of resource - including various collections of documents and non-electronic forms of media such as a museum or library archive.

What is resource discovery?

A card catalog is the most common physical representation of a metadata system and is a perfect example of a system originally devised for locating resources that evolved into its own information system.

What is the Dublin Core Metadata Initiative?

The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to fostering the widespread adoption of interoperable metadata standards and promoting the development of specialized metadata vocabularies for describing resources to enable more intelligent resource discovery systems.

The first Dublin Core Series Workshop took place in Dublin, Ohio in 1995. Since that time, the DCMI has been committed to the continual refinement of a "core" foundation of property types and values to provide vertically specific (semantic)information about Web resources, much in the same way a library card catalog provide indexes of book properties.

What is the Dublin Core?

Dublin Core metadata is used to supplement existing methods for searching and indexing Web-based metadata, regardless of whether the corresponding resource is an electronic document or a "real" physical object.

The Dublin Core Metadata Element Set (DCMES) was the first metadata standard deliverable out of the DCMI was an IETF RFC 2413. DCMES provides a semantic vocabulary for describing the "core" information properties, such as "Description" and "Creator" and "Date".

Dublin Core metadata provides card catalog-like definitions for defining the properties of objects for Web-based resource discovery systems.

What is the Dublin Core Metadata Element Set?

The Dublin Core Metadata Element Set is a set of 15 descriptive semantic definitions. It represents a core set of elements likely to be useful across a broad range of vertical industries and disciplines of study.

The Dublin Core Metadata Element Set was created to provide a core set of elements that could be shared across disciplines or within any type of organization needing to organize and classify information.

Who can benefit from using Dublin Core metadata?

Anyone can use Dublin core metadata to describe the resources of an information system. Web pages are one of the most common types of resources to utilize the Dublin Core's descriptions, usually within HTML's meta tags however increasingly there are many digital archives of physical objects that are starting to make use of the Dublin Core.

Dublin Core metadata is being used as the basis for descriptive systems by several community interest groups such as:

- educational organizations
- libraries
- government institutions
- scientific research sector
- Web page authors
- businesses requiring more searchable sites
- corporations with vast knowledge management systems

(See also our listing of Dublin Core-based projects).

What is the difference between "Simple" ("unqualified") and "Qualified" Dublin Core?

"Simple Dublin Core" is Dublin Core metadata that uses no qualifiers; only the main 15 elements of the Dublin Core Metadata Element Set are expressed as simple attribute-value pairs without any "qualifiers" (such as encoding schemes, enumerated lists of values, or other processing clues) to provide more detailed information about a resource.

"Qualified Dublin Core" employs additional qualifiers to further refine the meaning of a resource. One use for such qualifiers are to indicate if a metadata value is a compound or structured value, rather than just a string.

Qualifiers allow applications to increase the specificity or precision of the metadata. They may also introduce complexity that could impair the metadata's compatibility with other Dublin Core software applications. With this in mind, designers should only select from the set of approved Dublin Core qualifiers that were developed by the Dublin Core community process.

Unfortunately, qualifiers often introduce additional complexity that can make metadata less interoperable unless approved DC Qualifiers developed within the DCMI are used with such interoperability considerations in mind.

A "date" is one example of a DC element that has the option of being further specified to identify it as a particular kind of date (date last modified, date published, etc.).

The use of a controlled vocabulary, such as Dewey Decimal Classification, is another method that could be used to further "qualify" the meaning of resource.

For examples of embedding qualified DC in HTML please read the "Qualified HTML Examples" section of the "Using Dublin Core" usage guide.

The Australian Government Locator Service is a good example of a major implementation using Dublin Core metadata that has added four local elements to Dublin Core in the proper, recommended way.

How do I begin implementing the Dublin Core Metadata Element set and where can I ask questions if I have questions regarding my implementation?

This is one of the most often asked questions and perhaps the most difficult to provide a catch-all answer. The wise ones say, "It depends." Unfortunately, that is the truth. It depends on your hardware, software, system resources and the goals you are trying to achieve. In a nutshell, we have a Usage Guide which provides some insight. Beyond that, we highly recommend posing questions to the DC-General mailing list. Most questions can be answered there or redirected to a place where the answer can be found.

What is the relationship between the DCMI and other Internet Standards groups?

The Dublin Core Metadata Initiative (DCMI) is a consensus building organization that has relationships with many standards activities. A number of people in the DCMI are active in the World Wide Web Consortium (W3C) (DC is the prototype application that drove the development of the Resource Description Framework, or RDF in the W3C).

Our own standardization activities take place in the IETF (RFC 2413 is reference description of the initial version of the Dublin Core), and there are currently formal DC standardization activities underway in CEN (the European Information Industry Standardization Forum) and in NISO (the North American Information Standardization Organization) and the IEEE (Institute of Electrical and Electronics Engineers).

Was Dublin Core metadata designed to be used only to describe digital and Web-based resources?

No. The scope of the Dublin Core was specifically designed to provide a metadata vocabulary of "core" properties able to provide basic descriptive information about any kind of resource, regardless of the media format, area of specialization or cultural origin. It is important that a semantic model used for resource discovery is not dependent on the medium of the resource it means to describe.

The Dublin Core metadata vocabulary is the result of many years of collaborative research to determine a common set of properties universal for describing any type of resource. The use of a standardized general classifications system also enables metadata of such collections to be combined and for knowledge contained within each collection to be shared.

Since most Dublin Core implementations only need to process a resource's descriptive metadata, the medium of that resource becomes a non-issue. This enables DC metadata to be used by museums and other organizations interested in cataloging specialized types of mixed-media collections, while maintaining an open framework preserving their ability to share metadata with other DC implementors.

How is Dublin Core metadata used?

The term "Dublin Core metadata" is usually in reference to the 15 elements of the Dublin Core Metadata Element Set (DCMES) and its limited set of optional approved qualifiers tested and approved as "safe" for use by the Dublin Core community.

The DCMES was published in 1998 as an IETF RFC 2413. Dublin Core metadata is used to supplement existing methods for searching and indexing Web-based resources, providing a semantic vocabulary to describing the "core" properties of a resource object (such as "Description" and "Creator" and "Date").

How is Dublin Core metadata stored?

Dublin Core metadata is often stored as name-value pairs within META tags, which are placed within the HEAD elements of an HTML document.

However, it can also be located in an external document or loaded into a database enabling it to be indexed and manipulated from within a propriety application.

How can I embed Dublin Core metadata within my HTML documents?

Dublin Core metadata can be stored using the meta element in the head of HTML documents. An informational IETF RFC (2731) titled "Encoding Dublin Core Metadata in HTML" defines the standard way to do this. A DCMI Note describes one method for storing Qualified DC in HTML.

What is the relationship between Dublin Core Metadata and RDF and XML?

Dublin Core Metadata Element Set (IETF RFC 2413) and RDF (Resource Description Framework) (W3C Recommendation) are two distinct specifications. Neither requires the other, but their co-evolution forms a natural complement within the Web's greater metadata architecture.

Both the Dublin Core and RDF communities have a number of members in common, and have evolved side-by-side. The Dublin Core community provided much of the basic requirements that were used to design RDF. In turn, the development of RDF provided the Dublin Core community with a much more formal underlying data model that has helped it to determine best practices and universal solutions (rather than ad hoc Band-Aids) for many of the detailed problems that were encountered during the deployment process.

Below is an example of how the Dublin Core vocabulary can be used to define additional semantics about the resources described within an RDF fragment:

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">

 <rdf:Description about="http://purl.org/DC/documents/notes-cox-816.htm">
```

```
    <dc:title>Recording qualified Dublin Core metadata in HTML</dc:title>
    <dc:description> We describe a notation for recording
    qualified Dublin Core metadata in HTML meta elements. The syntax
    includes recommended usage of the standard HTML syntax to record
    the different classes of qualification needed to represent the
    model.</dc:description>
  <dc:date>1999-08-18</dc:date>
  <dc:format>text/html</dc:format>
  <dc:language>en</dc:language>
  <dc:publisher>Dublin Core Metadata Initiative</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

What is the Dublin Core data model?

The basic Dublin Core data model is defined by its 15 elements and the relationships defined between the resource-of-interest and whatever other resource is "in-scope" for DC.

In particular, the "Relation" and "source" elements are used to indicate a connection with another resource of any type. "Creator", "Contributor" and "Publisher" elements relate the present resource to a party who has some responsibility for it. The "Coverage" element relates the present resource to a place or to a time-period. The value recorded for each of these elements, therefore, should normally be strictly an identifier of another resource, which could have its own DCMES description.

A completely abstracted DC data model must also include its two types of qualifiers. Value qualifiers (which store an identifier for the vocabulary, encoding or language of the value) and element qualifiers, which are used to further refine the semantic meaning of an element.

However, many users have found it useful to add extra information to Dublin Core descriptions. Particularly when values are frequently drawn from a controlled vocabulary (e.g. a keyword list, such as Library of Congress Subject Headings, LCSH), or written using a special notation (e.g. the ISO8601 format for dates and times).

Other techniques include using a particular natural language (in the case of values written in text-strings), which would make the information more useful if a client were informed of the source for this vocabulary, the definition of the notation, or the name of the language used.

How do I participate in a discussion about the Dublin Core?

Anyone may participate in discussions about Dublin Core metadata by simply joining the appropriate mailing list for the working group activity of interest.

The DC-General mailing list is the general forum for community participation and submitting general feedback.

What is the maximum length for each field in Dublin Core?

There are no limits to field length.

What search-engines support the Dublin Core Metadata Element Set?

Several commercial and non-commercial search engines will index META elements with just a little configuration.

A recent inquiry on the dc-general mailing list produced this list:

- Ultraseek
- Swish-E
- Microsoft's Index Server
- Blue Angel Technologies MetaStar
- Verity Search 97 Information Server

To get a good overview of what software is out there see Search Tools and Search Engine Watch.

The well-known "all the Web" search engines including AltaVista, Yahoo, HotBot, etc. tend to avoid using the information found in meta elements in their indexing. This is because, unless the pages are from guaranteed "trusted" servers, the meta information is commonly used by unscrupulous content-providers for spamming, to mislead the indexes into givingWeb-pages a misleading rating.

What is an attribute-value pair?

Attribute-value pairs are used within Dublin Core metadata to represent the properties of a resource or object.

Information such as "Author", "Creator" and "Date" are all examples of Dublin Core elements that are implemented as attribute-value pairs within HTML's META element to provide additional semantics about a resource.

Web resources are effectively information objects complete with properties that can be expressed in any number of ways. The Dublin Core Metadata Element Set (DCMES) enables properties to be assigned to a Web resource the exact same way we might fill in the blanks for a card catalog record.

What is the Warwick Framework?

The Warwick Framework is a set of design principles that have guided the development of the Dublin Core since the Second Dublin Core Metadata Workshop in Warwick, UK.

The Warwick provides a metadata-based school of thought that believes different kinds of metadata can be used to describe the same resource in disparate ways to accomplish different goals.

For more information about the Warwick Framework, please read "The Warwick Metadata Workshop: A Framework for the Deployment of Resource Description".

Can I add a new element to Dublin Core?

In theory, yes. A DCMES element specified with an element qualifier is, effectively, a "new" element or property- with a more specialized meaning than its parent element.

However, it is not possible to create a new Dublin Core element whose meaning goes beyond the scope of of the original elements in the DCMES.

It is expected that local or application-specific requirements may require additional qualifiers or elements that have not been approved by the Dublin Core community at large. Nevertheless, designers should employ additional qualifiers with both caution and the understanding that interoperability could suffer as a result.

In cases where additional qualifiers are being utilized, it may be helpful to bring this to alert the Dublin Core Directorate in order to promote the wider use of such qualifiers.

For examples of embedding qualified DC in HTML please read the "Qualified HTML Examples" section of the "Using Dublin Core" usage guide.

What is the Open Metadata Registry?

The DCMI's Open Metadata Registry is a Web-based semantic modeling tool that uses a form-driven user interface to enable an end-user to define relationships between different vocabularies.

The semantic framework for the application was created using Dublin Core metadata and RDF Schemas.

How do I store proper names in Dublin Core metadata?

The encoding of personal names is a difficult task within most metadata systems, and DC is no exception. The difficult part is that naming conventions tend to vary from culture to culture.

The recommendation for Dublin Core metadata is to encode the family name first, which supports effective collation of names and is consistent with most naming conventions globally.

What is the "Dumb-Down" Principle?

The so called "Dumb-Down Principle" simply means that in any use of a qualified DC element, the qualifier may be dropped and the remaining value of the element should still be a term that is useful for discovery.

For example, there are several date qualifiers that might be used to enhance the precision of various dates associated with a resource. Dropping the date qualifier (say, for example, Date-Created) will still leave a useful date for discovery, though perhaps not quite as useful as if the qualifier were included.

Similarly, the specification of a subject term from LCSH, for example, is still useful even if one does not know it was selected from a controlled vocabulary.

The basic idea is that qualifiers should improve the precision of a piece of metadata, but the metadata should still be useful even without that extra precision (that is, dropping the qualifier has 'dumbed-down' the metadata).

How can I use existing controlled vocabularies for DC Subject metadata?

One can assign a metadata value selected from a controlled vocabulary as the value of the element, and then qualify that element with the name or identifier of the scheme from which it is selected (the specifics of the encoding depends on the syntax being employed. Refer to DCMI specifications for details of encoding DC metadta in HTML, XML, or RDF/XML.

DCMI registers controlled vocabularies and encoding schemes to promote their use and to facilitate consistent identification within DC metadata. Application designers should review registered controlled vocabularies to determine if there is a suitable one for their application, and use the registered name of that vocabulary in their application. For example, "DDC" is the registered Name for the Dewey Decimal Classification, and should be used as the value of the qualifier. By using registered Names or tokens to designate schemes, metadata from different applications that use common controlled vocabularies are more likely to be interoperable.

The mechanics of selecting a value from such a vocabulary is dependent on the application. It is expected that metadata editors for some domains will have tools to support such selection, but it can also be done with conventional print-based references.

Can I use controlled vocabularies that are not approved by DCMI?

Yes. DCMI registers only those controlled vocabularies that have been brought to our attention. There are, of course, many others that are equally legitimate, and it has always been our intent that communities of expertise be able to leverage the value of such existing schemes in their metadata. To promote interoperability, it is recommended that application designers review registered controlled vocabularies for one that may be suitable for their application. If a controlled vocabulary of choice is not registered with DCMI, it is possible (but not mandatory) to register it. Registration assures that others who adopt this vocabulary use the same Name token in their metadta, thereby promoting interoperability.

It is important to note that DCMI 'registers' controlled vocabularies, rather than 'approving' them. Controlled vocabularies are generally the result of substantial community expertise. It is not in the purview of DCMI to approve or disapprove such works, but rather help to make them visible for others who might choose to adopt them, and to prevent Name collisions by assigning unique tokens to identify them within DC metadata.

What are the names of the DCMI namespaces?

http://purl.org/dc/elements/1.1/
http://purl.org/dc/terms/
http://purl.org/dc/dcmitype/

Metadata associated with this resource: http://dublincore.org/resources/faq/index.shtml.rdf

DCMI Mixing and Matching FAQ

Andy Powell
UKOLN, University of Bath

This document attempts to answer some of the practical questions that implementors ask when faced with a desire to incorporate their existing XML metadata semantics into DCMI metadata applications.

Question 1: My favorite XML schema contains an element type or attribute (e.g. my:price or my:currency) that I want to use in my Dublin Core metadata. How do I do it?

The bad news is that an existing XML element type or attribute **cannot** be used as is in DCMI metadata applications. This is a very important point, but is sometimes hard for people to understand. Before you can use your favorite element or attribute you must declare it as a DCMI-compatible term. The good news is that doing so need not be an overly onerous task. Here's what you have to do:

1. Decide whether your XML element type or attribute corresponds to DCMI's notion of a 'property' or an 'encoding scheme'. These notions are defined in the DCMI Abstract Model but, for convenience, the definitions are repeated here:

   A property is a specific aspect, characteristic, attribute, or relation used to describe resources.

   Encoding scheme is the generic name for vocabulary encoding scheme and syntax encoding scheme.
   A syntax encoding scheme indicates that the value string is formatted in accordance with a formal notation, such as "2000-01-01" as the standard expression of a date.
   A vocabulary encoding scheme is a class that indicates that the value of a property is taken from a controlled vocabulary (or concept-space), such as the Library of Congress Subject Headings.

2. Next, check whether an equivalent property or encoding scheme has already been defined by the DCMI (or elsewhere), for use in DCMI metadata. A good place to start checking is the list of DCMI Metadata Terms.
3. Next, assign a URI reference to your new property or encoding scheme - see question 3 below.
4. Finally, declare your new property or encoding scheme using the RDF Schema language (RDFS) and make this declaration available somewhere on the Web - see questions 4 and 5 below.

Question 2: Why can't I just re-use my XML element type or attribute as is?

Because XML element types and XML attributes are component parts of an XML language. Their meaning is determined solely by their placement in the XML tree structure of the given XML language and the semantics that the developers of that language chose to associate with that structure. On the other hand, DCMI properties and encoding schemes are conceptual entities within the DCMI Abstract Model - their meanings are defined by the model and by the semantic declarations that DCMI make available. Furthermore, XML element types and attributes are named using XML *expanded names* (a pair comprising an *XML Namespace Name* (which is a URI reference) and a *local name*). On the other hand, DCMI properties and encoding schemes are named using URI references.

So, although XML element types and DCMI properties may look superficially similar, for example `lom:title` looks similar to `dc:title` when the two are encoded in XML, in fact they are very different entities.

For those who are interested, the XML, RDF and DCAPs document provides a much more in-depth treatment of the differences between XML element types and RDF properties and the usage of both in the context of DCMI metadata.

Question 3: How do I assign a URI to my 'element'?

Unfortunately, there is little best-practice in this area to draw on at the time of writing. The Guidelines for assigning identifiers to metadata terms document lists some possible approaches.

One immediate issue to consider is whether to make the URI reference that is assigned to the new property or encoding scheme similar to the XML expanded name of the XML element or attribute. DCMI has chosen to keep the two things very similar. For example, the XML expanded name that is used to represent the DC Title property according to the Guidelines for encoding DC in XML recommendation is `dc:title` (correspoinding to the `http://purl.org/dc/elements/1.1/` XML namespace name and the `title` local name). The DC Title property is assigned the `http://purl.org/dc/elements/1.1/title` URI reference. Therefore, the property URI reference is simply a concatenation of the component parts of the XML expanded name.

On the other hand, the RDF encoding of the IEEE LOM (which has effectively made the LOM available for use with DCMI metadata because the DCMI Abstract Model is essentially the same as the RDF model) has chosen to keep the XML expanded names used in the LOM/XML encoding and the URI references assigned to the LOM/RDF properties completely separate. *Example to be provided*

Remember that your new property is likely to appear in the various DCMI encoding syntaxes using the name that is the final part of the URI reference you assign (usually the bit after the final '/' or '#'). For example, the URI reference `http://example.com/my/terms#color` is likely to appear as `my:color` (in XML syntaxes) or `MY.color` (in HTML syntaxes).

If in doubt, register with the PURL system and assign a PURL to your new property or encoding scheme.

Question 4: How do I declare my XML element type or attribute as a DCMI property?

Use the RDF Schema language to do this. Take a look at the DCMI RDFS terms declarations for inspiration! As a minimum, you'll need something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Property rdf:about="http://purl.org/my/terms/price">
    <rdfs:label xml:lang="en-US">Price</rdfs:label>
    <rdfs:comment xml:lang="en-US">The amount of money needed to purchase the resource.</rdfs:
comment>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/my/terms/"/>
    <dcterms:issued>2004-12-03</dcterms:issued>
    <dcterms:modified>2005-02-21</dcterms:modified>
    <dc:type rdf:resource="http://dublincore.org/usage/documents/principles/#element"/>
  </rdf:Property>
</rdf:RDF>
```

Make sure that this RDF/XML document is available (in this case) at both `http://purl.org/dc/terms/` and `http://purl.org/my/terms/price`.

Question 5: How do I declare my XML element type or attribute as a DCMI encoding scheme?

Use the RDF Schema language to do this. Take a look at the DCMI RDFS terms declarations for inspiration! As a minimum, you'll need something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:about="http://purl.org/my/terms/PoundsSterling">
    <rdfs:label xml:lang="en-US">Pounds Sterling</rdfs:label>
    <rdfs:comment xml:lang="en-US">A price in UK pounds sterling, formatted in the following way:
"P.pp"
    (where "P" representents one or more digits for the number of pounds and "pp" represents
```

```
    two digits for the number of pence).</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/terms/"/>
    <dcterms:issued>2003-07-11</dcterms:issued>
    <dcterms:modified>2004-06-15</dcterms:modified>
    <dc:type rdf:resource="http://dublincore.org/usage/documents/principles/#encoding-scheme"/>
  </rdfs:Class>
</rdf:RDF>
```

The examples used in questions 4 and 5 would allow the following XML fragment to be used in a DC/XML document that conforms to the Guidelines for implementing Dublin Core in XML recommendation:

```
<my:price xsi:type="my:PoundsSterling">2.99</my:price>
```

assuming that the appropriate namespace declarations are in place. Note that, by convention, the XML *local name* for an encoding scheme starts with an uppercase letter.

Question 6: I still don't understand! Do you have another example?

OK. Let's say that I want to start using DC metadata to describe car parts, and that my company (ZZ Motors) already uses an XML schema that allows for XML fragments like this:

```
<zz:carpart>
  <zz:engine>
    <zz:type>petrol</zz:type>
    <zz:capacity>2000cc</zz:capacity>
  </zz:engine>
  <zz:fueltank>
    <zz:capacity>25l</zz:capacity>
  </zz:fueltank>
<zz:carpart>
```

'zz' being associated with the `http://zz.com/carparts/` XML namespace name.

For the sake of argument, let's say that I want to start using a property in my DC metadata that indicates the engine capacity. Looking at my existing XML schema, I note that I already have an XML element type with the local name `capacity` (under `zz:engine`) that I might be able to use? But there's a problem! I also have an XML element type with the local name `capacity` elsewhere in my XML tree structure (under `zz:fueltank`). So I cannot simply use 'capacity' as the local name when I'm thinking about assigning a URI reference to my new property.

The semantics of my current XML element types called `capacity` are determined by the placement of those two element types in the XML tree. In fact I have two 'properties', which we'll call `engineCapacity` and `fueltankCapacity`. I'm interested in the first of these.

OK, so now I need to assign a URI reference to my new property called `engineCapacity`. I want this property to be widely used (because it'll make my supply chains work more smoothly if everyone else uses the same property) so I decide to name my new property using a PURL, rather than a URI reference somewhere under my company's DNS domain name. I choose:

```
http://purl.org/carparts/terms/engineCapacity
```

Now I need to declare my new property using RDFS. I create a file on my company's Web site that contains the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Property rdf:about="http://purl.org/carparts/terms/engineCapacity">
    <rdfs:label xml:lang="en-US">Engine Capacity</rdfs:label>
    <rdfs:comment xml:lang="en-US">The total combustion chamber size of an engine in cubic
centimetres.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/my/terms/"/>
```

```
    <dcterms:issued>2005-02-21</dcterms:issued>
    <dcterms:modified>2005-02-21</dcterms:modified>
    <dc:type rdf:resource="http://dublincore.org/usage/documents/principles/#element"/>
  </rdf:Property>
</rdf:RDF>
```

Finally, I register two PURLs, `http://purl.org/carparts/terms/` and `http://purl.org/carparts/terms/engineCapacity`, and configure them both so that they resolve to the RDF/XML document (above) on my Web server.

Content by: Andy Powell
Last updated: 22-Feb-2005

2009-02-26 from Pete to Jeff Albro <jalbro@BU.EDU> on dc-education

> I've been asked by a professor to help organize student
> website projects on our webserver. I immediately though of
> using meta tags in the header.
>
> After poking around the web for a while I found the Dublin
> core project.
> I figured using a meta tag schema that is documented and
> has wide support is much better than me making up my own.
>
> I also found out about, and am playing with mksearch, a
> spidering tool for Dublin core tagged webpages.
>
> I am a bit confused about some of the meta tag specifics. On
> http://dublincore.org/documents/2007/11/05/dc-html/ , it
> talks about using tags like this:
>
> <link rel="schema.DCTERMS" href="http://purl.org/dc/terms/"
> /> <meta name="DCTERMS.title" content="Services to Government" />
>
> How is DCTERMS.title different than DC.title?

Since January this year, for each property of the Dublin Core Metadata Element Set, i.e. the 15 properties with URIs of the form http://purl.org/dc/elements/1.1/xyz , there is a now a corresponding property in the DC Terms vocabulary with a URI of the form http://purl.org/dc/terms/xyz (Aside: note that the reverse is not true: there are terms with URIs of the form http://purl.org/dc/terms/xyz for which there is no corresponding terms with the URI http://purl.org/dc/elements/1.1/xyz)

The introduction of this "parallel" set of properties was primarily due to the decision to specify domains and ranges for DCMI properties i.e. to make formal assertions in the descriptions of its terms that DCMI publishes which enable an applocation to make inferences about the type of the described resource or the type of the value when it encounters a statement using a specified property. See [1] for more details.

In order not to risk creating contradictions for applications using the fifteen properties of the DCMES, which are frequently used with literal values, domain and range assertions were not made for these properties, but instead a new set of 15 properties were created, for which domain and range assertions _were_ made.

So while no range is specified for the property with the URI http://purl.org/dc/elements/1.1/creator and no inferences can be made about the class of a value of that property, there is a new property with the URI http://purl.org/dc/terms/creator for which a range is specified, and for which an application can conclude that the value is an instance of a class of "Agents" identified by the URI http://purl.org/dc/terms/Agent .

Formally the new property http://purl.org/dc/terms/creator is a subproperty - refinement - of the original property http://purl.org/dc/elements/1.1/creator

(As a very minor side effect of this, the creation of these 15 additional properties allows for some minor simplification in their use of concrete syntaxes since, if they can make use of the 15 new properties (i.e. their usage of those properties is consistent with the new domain/range assertions) then they require only one "namespace declaration" for the names of all the DCMI-owned properties.)

The case of DC.Title and DCTERMS.Title is arguably slightly anomalous, since - at this point in time - no range assertion is made for the new property with the URI http://purl.org/dc/terms/title and so essentially - at this point in time - there is no real difference between the property http://purl.org/dc/terms/title and the property

http://purl.org/dc/elements/1.1/title . However it is worth taking heed of the note in [1] regarding the new title property:

> In current practice, this term is used primarily with literal values; however, there are important uses with non-literal values as well. As of December 2007, the DCMI Usage Board is leaving this range unspecified pending an investigation of options.

It is expected that the range for the property http://purl.org/dc/terms/title will be changed in the near future.

For more details, see [1] and [2].

> If I have the link to the schema, is it okay for me to mix in
> my own tags if I don't use the DC or DCTERMS?

The short answer is "yes" :-)

To provide a longer answer and a bit of explanation, I probably need to take a step back and emphasise that - as that doc you refer to above [3] emphasises - the convention used in HTML meta and link elements is a convention for "encoding" a data structure known as a DC description set, the form of which is described by the DCMI Abstract Model [4]. So the "Expressing DC metadata using X/HTML meta & link elements" doc describes a mapping bewteen the components of that data structure and the components of X/HTML meta and link elements.

Now, in the DC description set, all metadata terms - properties, classes, vocabulary encoding schemes and syntax encoding schemes are referred to using URIs. Those can be URIs owned by anyone, not just URIs owned by DCMI - and it is perfectly possible for a DC description set to contain no references at all to terms owned by DCMI! :-)

And in the meta data profile described by the "Expressing DC metadata using X/HTML meta & link elements" doc, the "prefixed names" (like "DC.Title") used as the values of X/HTML name, scheme and rel attributes are abbreviations/shorthands for thoes term URIs i.e. the prefixed name "DC.Title" is (typically - it depends on the association between prefix and "namespace URI" provided by a link element) mapped to the URI http://purl.org/dc/elements/1.1/title (using the convention described in section 3.1.2 of [3]).

So if you wish to refer to terms other than those owned by DCMI, you need a URI for each of those terms, provided by the owner of those terms, either a third party (like the Library of Congress for the case of the MARC Relator Codes) or, as in this case, yourself. So you need to coin a URI in some URI-space where you can create URIs, and ideally if you expect your metadata instances to be used over the medium/lon term then that that URI would be chosen so that it was reasonably persistent.

And then to represent that URI as a name/rel/scheme attribute value, you need to choose a suitable prefix for abbreviating the term URIs and add a corresponding "namespace declaration" using the
link/@rel="schema.XYZ"
convention described in section 3.1.2 of [3] e.g.

<link rel="schema.JEFF" href="http://jeff.example.org/terms/" />
<link rel="JEFF.isRelatedInSomeWeirdAndWonderfulWayTo" href="doc123" />

where the "prefixed name" "JEFF.isRelatedInSomeWeirdAndWonderfulWayTo" is an abbreviation/shorthand for the URI http://jeff.example.org/terms/isRelatedInSomeWeirdAndWonderfulWayTo

Incidentally, although DCMI hasn't yet addressed this in any of its own specifications (but I think it probably will at some point!), you may also be interested in the recently published working draft from the W3C for RDFa [5], which specifies how to use a set of attributes to embed RDF metadata in XHTML documents (and other XML documents, I think, but the focus in the current draft is mainly on XHTML). Any RDF graph can be represented using RDFa, so it's very powerful and flexible, and once you become familiar with the basic RDFa "patterns", it is quite easy to use, I think.

Finally, it's probably worth adding that _if_ your _only_ requirement is

to provide data to a local search engine and you have control over that document set, you could just go ahead and use meta and link elements without worrying too much about the conventions recommended by DCMI. i.e. you could just use

<meta name="abcdef" content="blahblahblah" />

without worrying about the mapping of the @name attribute value to a URI via the link/@rel="schema.XX" mechanism.

[1] http://dublincore.org/documents/2008/01/14/domain-range/
[2] http://dublincore.org/usage/decisions/2008/dcterms-changes/
[3] http://dublincore.org/documents/2007/11/05/dc-html/
[4] http://dublincore.org/documents/2007/06/04/abstract-model/
[5] http://www.w3.org/TR/2008/WD-rdfa-syntax-20080221/

2007-07-23 From James Jeffery <cascadingstylez@GOOGLEMAIL.COM> on dc-education

Subject: Using MetaData in HTML

Over the weekend i began to read up more about the Dublin Core, its pretty new to me, although i have herd of it ive never actually implemented it into any of my Web Documents.

After reading some of the documents at the Dublin Core site i got lost rather quickly. I understand that the DC Elements are implemented within the <meta> tags in HTML documents, and have strong knowledge of HTML, so on that side of things im pretty clear, but below i have a few questions. If someone could dedicate a few moments of their time in order to help me, i would be greatly thankful.

1 - When embedding DC elements in HTML documents, do i need to include any document or vocabulary before defining DC. elements? For example, HTML pages include a doctype to point to a set of rules, does the DC need anything like this within my documents?

2 - A DC element is NOT the same as a HTML element right? Example, a HTML element would be <meta> whereas a DC element is not <meta> but instead DC.title (the name trailing the period, 1 of 15 i believe).

3) What is DCTERMS.somethinghere ? What does the DCTERMS mean? do i need to use it within my document?

At the moment im only concerned with using the 'standard' elements within my documents, to add some extra value to them.

I have read the documents, and find some of it confusing, especially the document that shows how to uses DC MetaData with HTML, i get easily confused by DC.element and DCTERMS.element, and dont understand the difference between them both.

James Jeffery
ClearVue Media Web Developer and Consultant

2009-06-20 Tom on public-esw-thes
To: Norman Gray <norman@astro.gla.ac.uk>
Cc: SKOS <public-esw-thes@w3.org>

On Fri, Jun 19, 2009 at 06:40:03PM +0100, Norman Gray wrote:
> The SKOS Primer illustrates annotating ConceptScheme instances using
> DCTerms properties. Is it a recommendation to use that rather than
> the DC elements properties from <http://purl.org/dc/elements/1.1/>;?

Yes. This point was raised and discussed in the SWD working group (e.g., [1]). As I sometimes put it, DCMI "gently promotes" the use of the dct: equivalents of the fifteen elements because of their formal ranges.

> This is possibly more of a DC question than a SKOS one, but this list
> is probably as good a place as any to ask the question, especially

> since I'm asking it in the context of trying to describe 'good
> practice' for developing vocabularies.
>
> DC elements have the advantage that they're probably more generally
> understood, and more things might be declared as rdfs:subClassOf DC
> elements.
>
> On the other hand, the DCTerms properties are more expressive, are
> declared as subclasses of the DC elements properties, and are newer
> (though I couldn't find an explicit statement that the older ones are
> deprecated, beyond the incidental description of the DC Elements
> properties as 'legacy' in the DC Terms documentation).
>
> My guess would be that the DC Terms properties are what I should
> recommend -- is there anything wrong with that?

DCMI has gotten alot of positive feedback from the Semantic Web community on the "makeover" of DCMI properties with domains and ranges. A longer explanation, with historical context:

dc:title [1] and dc:subject [2] (and the other thirteen Dublin Core properties) were among the first RDF properties declared anywhere. They were declared as RDF properties before W3C standardized the notion of "range" in the RDF Schema specification.

As RDF matured, the DC properties became criticized in SW circles for being underspecified. DCMI wanted to assign ranges, but in doing so did not want to "break" existing legacy data, which used "subject" (for example) both with literal and non-literal values.

As described in [4, paragraphs starting "Formal domains..."], DCMI resolved this dilemma by creating fifteen properties in the /terms/ namespace in parallel to the corresponding terms in the /elements/1.1/ namespace, and declared the former as subproperties of the latter.

It is not actually incorrect to continue using dc:subject and dc:title -- alot of Semantic Web data still does -- and since the range of those properties is unspecified, it is not actually incorrect to use (for example) dc:subject with a _literal_ value or dc:title with a _non-literal_ value. However, good Semantic Web practice is to use properties consistently in accordance with formal ranges, so implementers are encouraged to use the more precisely defined dcterms: properties.

[1] http://dublincore.org/documents/dcmi-terms/#elements-title
[2] http://dublincore.org/documents/dcmi-terms/#elements-subject
[3] http://dublincore.org/documents/dcmi-terms/#H1
[4] http://dublincore.org/documents/dcmi-terms/#terms-subject
[5] http://lists.w3.org/Archives/Public/public-swd-wg/2009Jan/0000.html

---

2009-06-20

> As I sometimes put it, DCMI "gently
> promotes" the use of the dct: equivalents of the fifteen
> elements because of their formal ranges.

Many thanks for this detailed and interesting summary.

I'll include a recommendation (in [1]) to use the DCTerms namespace, along with suitable examples and a validator.

[1] http://www.ivoa.net/Documents/latest/Vocabularies.html

2009-05-13 Tom on literal/non-literal

dc:title [1] and dc:subject [2] were among the first RDF properties declared anywhere. They were declared as RDF properties even before RDF schema standardized the notion of "range".

As RDF matured, the DC properties became criticized in SW circles for being underspecified. We wanted to assign ranges, but in doing so we did not want to "break" existing legacy data, which used "subject" (for example) both with literal and non-literal values.

As described in [4, paragraphs starting "Formal domains..."], we resolved this dilemma by creating fifteen properties in the /terms/ namespace in parallel to the corresponding terms in the /elements/1.1/ namespace, and declared the former as subproperties of the latter.

It is not actually incorrect to continue using dc:subject and dc:title -- alot of Semantic Web data still does -- and since the range of those properties is unspecified, it is not actually incorrect to use dc:subject with a _literal_ value or dc:title with a _non-literal_ value. However, good Semantic Web practice is to use properties consistently in accordance with formal ranges, so implementers are encouraged to use the more precisely defined dcterms: properties.

So to get back to your question: are the ranges for dcterms:title and dcterms:subject really so defined?

For dcterms:subject [4], we decided on the formulation "this term is intended to be used with non-literal values" as a Note. We did this because "non-literal resource" has not been declared by anyone as a formal class, we didn't think it was the role of DCMI to define such a class, and we felt that such an "informal" annotation was sufficient for now.

For dcterms:title, we decided: "In current practice, this term is used primarily with literal values; however, there are important uses with non-literal values as well. As of December 2007, the DCMI Usage Board is leaving this range unspecified pending an investigation of options" (see Note). We did investigate the options (for example, Japanese titles for which a single title resource is associated with multiple literal representations in parallel), but decided to follow the overwhelming preference of implementers for a literal range. Complex titles, e.g. in parallel writing systems (as in Japanese) would need to use dc:title or a new title property with a non-literal range would need to be created.

[1] http://dublincore.org/documents/dcmi-terms/#elements-title
[2] http://dublincore.org/documents/dcmi-terms/#elements-subject
[3] http://dublincore.org/documents/dcmi-terms/#H1
[4] http://dublincore.org/documents/dcmi-terms/#terms-subject
[5] http://dublincore.org/documents/dcmi-terms/#terms-title

Dublin Core Metadata Initiative

Making it easier to find information

Metadata Basics

The word "metadata" means "data about data". Metadata articulates a context for objects of interest -- "resources" such as MP3 files, library books, or satellite images -- in the form of "resource descriptions". As a tradition, resource description dates back to the earliest archives and library catalogs. The modern "metadata" field that gave rise to Dublin Core and other recent standards emerged with the Web revolution of the mid-1990s.

Background

Early Dublin Core workshops [/workshops/] popularized the idea of "core metadata" for simple and generic resource descriptions. The fifteen-element "Dublin Core" [/documents/dces/] achieved wide dissemination as part of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [http://www.openarchives.org/pmh/] and has been ratified as IETF RFC 5013 [http://www.ietf.org/rfc/rfc5013.txt] , ANSI/NISO Standard Z39.85-2007 [http://www.niso.org/standards/z39-85-2007/] , and ISO Standard 15836:2009 [http://www.iso.org/iso/search.htm?qt=15836&searchSubmit=Search&sort=rel&type=simple&published=on] .

Starting in 2000, the Dublin Core community focused on "application profiles" -- the idea that metadata records would use Dublin Core together with other specialized vocabularies to meet particular implementation requirements. During that time, the World Wide Web Consortium standardized a generic data model for metadata, the Resource Description Framework (RDF). As part of an extended set of DCMI Metadata Terms, Dublin Core became one of most popular vocabularies for use with RDF, more recently in the context of the Linked Data [http://linkeddata.org/] movement.

The consolidation of RDF motivated an effort to translate the mixed-vocabulary metadata style of the Dublin Core community into an RDF-compatible DCMI Abstract Model [/documents/abstract-model/] (2005). The DCMI Abstract Model was designed to bridge the modern paradigm of unbounded, linked data graphs with the more familiar paradigm of validatable metadata records like those used in OAI-PMH. A draft Description Set Profile [/documents/dc-dsp] specifies a language for expressing constraints in a generic, application-independent way. The Singapore Framework for Dublin Core Application Profiles [/documents/singapore-framework] defines a set of descriptive components useful for documenting an application profile for maximum reusability.

"Levels of interoperability"

From the perspective of the Dublin Core community, the metadata landscape is currently characterized in terms of four "levels" of interoperability: [/documents/interoperability-levels/]

Level 1 (Shared term definitions). At Level 1, interoperability among metadata-using applications is based on shared natural-language definitions. Within an application environment such as an intranet, library system, or repository federation, participants agree what terms to use in their

metadata and how those terms are defined. Terms are hard-wired into applications using specific implementation technologies, and interoperability with "the rest of the world" outside of the implementation environment is generally not a priority. Most existing metadata applications currently operate at this level of operability.

Level 2 (Formal semantic interoperability). At Level 2, interoperability among metadata-using applications is based on a shared, formal model for Linked Data. As defined in Wikipedia, the term "Linked Data" describes "a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs [Web addresses] and RDF." The properties and classes of DCMI Metadata Terms [/documents/dcmi-terms/] have been defined for compatibility with Linked Data principles. Over the past two years, vast amounts of commercial and public-sector data have been added to a growing to a linked data cloud. Search engines such as Yahoo and content-management platforms such as Drupal have implemented support for RDFa, a method for exposing linked data embedded in Web pages. In effect, the founding idea of Dublin Core -- "simple metadata for resource discovery" -- is being reinvented under the banner of "structured data for search engine optimization". Of the four interoperability levels, this one appears to be growing the fastest.

Level 3 (Description Set syntactic interoperability) and Level 4 (Description Set Profile interoperability). At Level 3, applications are compatible with the Linked Data model and, in addition, share an abstract syntax for validatable metadata records, the "description set". At Level 4, the records exchanged among metadata-using applications follow, in addition, a common set of constraints, use the same vocabularies, and reflect a shared model of the world. Levels 3 and 4 are more experimental than Levels 1 and 2 inasmuch they are not as well supported with software tools, though the problems addressed in this work are expected to grow in importance as producers of metadata records move their information into a linked-data environment.

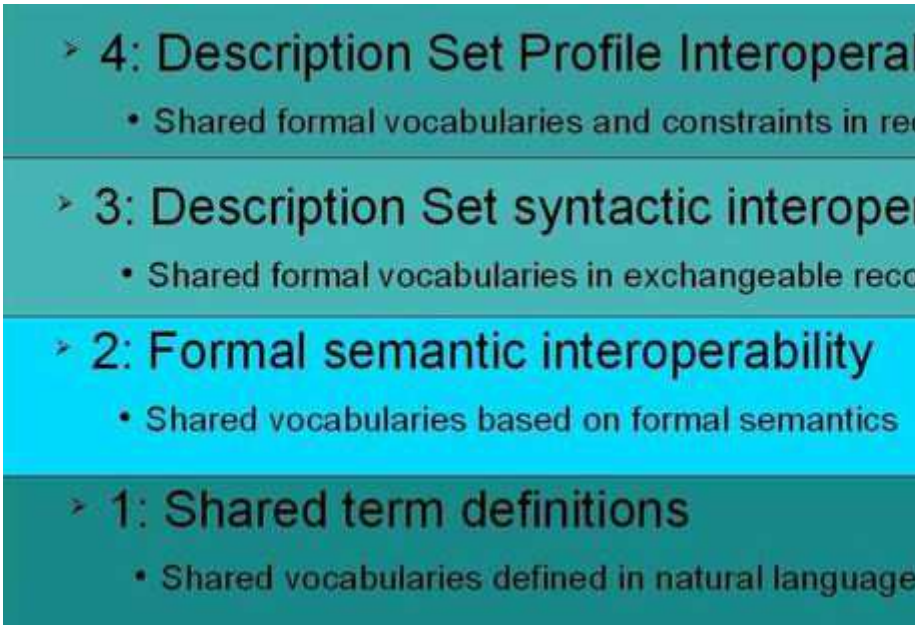
To the reader: If you are evaluating implementation options, it is good to start by defining your requirements:

- If the needed functionality can be met with a closed system, consider Level 1 solutions; DCMI Metadata Terms provides a useful starter set of elements, many good tools are available, and implementation is often straightforward.
- If exchangeability of metadata with "the rest of the world" is important, consider Level 2. Month by month, new platforms, tools, and data sources are coming online. Note that implementations do not need to use URIs and RDF natively in order to be compatible with the linked data cloud. With careful design and planning, just about any technology can be configured to export data in RDF.
- If your metadata needs to be designed both in

accordance with Level 2 *and* with validatable records, consider Levels 3 and 4 and join the community of pioneers on the DCMI Architecture Forum mailing list [http://www.jiscmail.ac.uk/lists/dc-architecture.html] .

Next steps:

- For a friendly introduction to the Dublin Core "mixed-vocabulary" style of metadata, see Guidelines for Dublin Core Application Profiles [/documents/profile-guidelines/] .
- Join the mailing list for one of DCMI's many communities of practice [/groups/#communities] and introduce yourself or ask a question. Read more... [/beta/community-and-events]
- Subscribe to DCMI's RSS news feed [/news.rss] or follow Dublin Core on Twitter [http://twitter.com/DublinCore] .
- Check out recent presentations and tutorials [/resources/training/] about Dublin Core metadata.
- Explore the available technical specifications and guidelines [/beta/specifications/] .



Dublin Core Metadata Initiative
Making it easier to find information

DCMI Specifications

As part of its mission, the Dublin Core Metadata Initiative develops and maintains specifications in support of resource description. Specifications developed and reviewed in the context of DCMI's formal approval process [/documents/approval/] are assigned a status (in ascending order of maturity and stability) of "DCMI Working Draft", "DCMI Proposed Recommendation", or "DCMI Recommendation". DCMI also provides pointers to guidelines and services developed outside of this formal review context ("Recommended Resources").

This selection highlights the specifications that currently attract the most attention in the Dublin Core community. Links to additional specifications (including superseded specifications) may be found at http://dublincore.org/documents/ [/documents/] . Some of the specifications have been translated [/resources/translations/] into one of twenty-five languages.

Semantic Recommendations

DCMI Metadata Terms [/documents/dcmi-terms/] [DCMI Recommendation]. This periodically updated document provides a one-stop source of up-to-date information on DCMI metadata terms, including the classic Dublin Core Metadata Element Set, the DCMI Type Vocabulary, and resource classes used as formal domains and ranges. Supporting documents includes

- DCMI Namespace Policy [/documents/dcmi-namespace/] [DCMI Recommendation]. This document describes how metadata terms are assigned URIs by DCMI and the policies governing changes to the documented meanings associated with those URIs.
- DCMI term declarations represented in RDF schema language [/schemas/rdfs/] . Since 2001, DCMI term declarations have been published as Web documents (with the status DCMI Recommendation) and, in parallel, as machine-processable RDF/XML documents.
- The Dublin Core Metadata Registry [http://purl.org/dcregistry/] [DCMI Recommended Resource]. This application, hosted at the University of Tsukuba as a service for the DCMI community, provides a navigational interface [http://dcmi.kc.tsukuba.ac.jp/dcregistry/navigateServlet] to DCMI's machine-processable RDF term declarations. The registry is under development as an open-source software project.
- Dublin Core Metadata Element Set [/documents/dces/] [DCMI Recommendation]. This document excerpts from DCMI Metadata Terms the fifteen elements of the classic "Dublin Core", which has been standardized as ISO Standard 15836:2009 [http://www.iso.org/iso/search.htm?qt=15836&searchSubmit=Search&sort=rel&type=simple&published=on] .

User guidelines

Interoperability Levels for Dublin Core Metadata [/documents/interoperability-levels/] [DCMI Recommended Resource]. This document articulates current thinking in the Dublin Core community about the nature of metadata interoperability. Read more... [/beta/metadata-basics]

Guidelines for Dublin Core Application Profiles [/documents/profile-guidelines/] [DCMI Recommended Resource]. This document provides guidelines for the creation of Dublin Core Application Profiles. The document explains the key components of a Dublin Core Application Profile and walks

through the process of developing a profile. The document is aimed at designers of application profiles -- people who will bring together metadata terms for use in a specific context.

Singapore Framework for Dublin Core Application Profiles [/documents/singapore-framework/] [DCMI Recommended Resource]. The Singapore Framework for Dublin Core Application Profiles is a framework for designing metadata applications for maximum interoperability and for documenting such applications for maximum reusability. The framework defines a set of descriptive components that are necessary or useful for documenting an Application Profile and describes how these documentary standards relate to standard domain models and Semantic Web foundation standards. The framework forms a basis for reviewing Application Profiles for documentary completeness and for conformance with Web-architectural principles.

Using Dublin Core [/documents/usageguide/] [DCMI Recommended Resource]. From January 1998 through the latest version in November 2005, this document was maintained as the main entry point for new users of Dublin Core metadata. While this document is accessibly written, with useful concrete examples, readers should be aware that the document is by now somewhat out of date and is best used in conjunction with Guidelines for Dublin Core Application Profiles [/documents/profile-guidelines/] .

Model-related specifications

DCMI Abstract Model [/documents/abstract-model/] [DCMI Recommendation]. This document specifies an abstract syntax for "description sets" (metadata records). Finalized as a DCMI Recommendation in March 2005, the DCMI Abstract Model (DC-AM) expressed the style of metadata that had emerged in the Dublin Core community over the previous decade as formal constructs based on the W3C Resource Description Framework (for example, by giving a formal definition to the DC-specific notion of a "vocabulary encoding scheme" and providing for the use of parallel labels in multiple languages). The Abstract Model was designed to bridge the modern paradigm of the unbounded, linked data graph and the more familiar paradigm of the validatable metadata record, providing a foundation for the development of application-independent syntax specifications and constraint languages.

Expressing Dublin Core metadata using the DC-Text format [/documents/dc-text/] [DCMI Working Draft]. DC-Text is a syntax for representing a DC-AM Description Set in plain text. Its primary use is in presenting metadata constructs for human consumption.

Description Set Profiles: A constraint language for Dublin Core Application Profiles [/documents/dc-dsp/] [DCMI Working Draft]. This document specifies an application-independent language for the constraints used in defining "templates" for metadata records -- for example, to specify a

template for records which describe exactly "one book", with "up to ten authors", using "subject headings from the Library of Congress". This specification represents work in progress and is likely to evolve in response to implementation experience.

Criteria for the Review of Application Profiles [/documents/profile-review-criteria/] [DCMI Recommended Resource]. This document was developed by the DCMI Usage Board to evaluate Application Profiles for documentary completeness and for conformance with linked-data principles. The criteria are based on the DCMI Abstract Model, Description Set Profile (DSP) language, and Singapore Framework. Even when conformance with these specifications is not a requirement, implementers may find these review criteria useful for uncovering flaws or weaknesses in the logic of an application profile.

A MoinMoin Wiki Syntax for Description Set Profiles [/documents/dsp-wiki-syntax/] [DCMI Working Draft]. Using this draft syntax, editors can embed formal constraint information about application profiles into normal Wiki documents and extract this information using an Open Source MoinMoin Wiki tool [/documents/2008/10/06/dsp-wiki-syntax/DescriptionSetProfile-dist.zip] into an XML representation. This functionality is illustrated by the MoinMoin Wiki document for the Scholarly Works Application Profile [http://dublincore.org/architecturewiki/EprintsApplicationProfile?action=DSP2XML] .

Syntax guidelines

DC-HTML [/documents/dc-html/] [DCMI Recommendation]. "Expressing Dublin Core metadata using HTML/XHTML meta and link elements" describes how a Dublin Core metadata description set can be encoded using the HTML/XHTML <meta> and <link> elements. This specification is also an HTML "meta data profile" as defined by the HTML specification. See also:

- RDFa [http://rdfa.info/] , another syntax for embedding Dublin Core descriptions in Web pages, which became a W3C Recommendation [http://www.w3.org/TR/rdfa-syntax/] in October 2008 and has strong tool support in platforms such as Drupal [http://drupal.org/] and Yahoo's Search Monkey [http://developer.yahoo.com/searchmonkey/] .
- The predecessor specification for DC-HTML, DCQ-HTML [/documents/dcq-html/] (last updated in 2003), consistently remains one of the most-accessed documents on the DCMI Web site despite its clearly marked status as a "Superseded Recommendation". Early versions of this specifications used the so-called "dotty" syntax for appending an additional "qualifier" to an element name. as with "DC.Coverage.temporal" -- an encoding style that is now discouraged. This and other related issues are discussed in "Notes on DCMI

specifications for Dublin Core metadata in HTML/XHTML meta and link elements" [/documents/dc-html-notes/] .

DC-DS-XML [/documents/dc-ds-xml/] [Proposed Recommendation]. "Expressing Dublin Core Description Sets using XML (DC-DS-XML)" specifies an XML format for representing a Dublin Core metadata description set. The specification supports all of the features of a Description Set as described by the DCMI Abstract Model. Related resources include:

- Notes on DC-DS-XML XML Format [/documents/dc-ds-xml-notes/] , which describes the development of the format and its relationship to the 2003 specification, DC-XML-GUIDELINES.
- DC-XML-GUIDELINES [/documents/dc-xml-guidelines/] [DCMI Recommendation]. The specification "Guidelines for implementing Dublin Core in XML" of April 2003 has been used in numerous implementations and remains popular despite its well-described limitations (see above). XMLS schemas [/schemas/] implementing the 2003 encoding conventions are available in several variants.

DC-RDF [/documents/dc-rdf/] [DCMI Recommendation]. "Expressing Dublin Core metadata using the Resource Description Framework (RDF)" describes how constructs of the DCMI Abstract Model may be expressed in RDF graphs. How this document relates to earlier styles for expressing Dublin Core metadata in RDF is discussed in "Notes on DCMI specifications for Dublin Core metadata in RDF". [/documents/dc-rdf-notes/]

[/documents/dc-rdf-notes/]

[/documents/dc-rdf-notes/]
[/documents/dc-rdf-notes/]

Old actions

ACTION 2008-09-21: Joe and Andrew to continue work on Coverage.
http://dublincore.org/usageboardwiki/IssuesWithCoverage

ACTION 2008-09-21: Tom (and Mikael) to continue work on Simple Dublin Core.

ACTION 2008-09-21: Tom to amend the naming policy (DCMI names should not differ only with respect to case).

ACTION 2008-09-21: Tom to correct RDF schemas of DCMI Metadata Terms to use blank node with publisher.

ISSUE 2008-09-21: Rendering of Wiki pages using the DSP syntax does not capture all of the DSP detail and uses different labelling. It would be useful to hyperlink labels in the DSP rendered page to the DSP documentation itself. Also, to hyperlink description template labels within the wiki page itself, e.g. creator to agent description.
ACTION 2009-04-22: Tom to turn Pete's proposal at
https://www.jiscmail.ac.uk/cgi-bin/webadmin?A2=ind0903&L=DC-USAGE&P=982
into a decision document for finalization on next telecon.

ACTION 2008-09-21: Joe and Andrew to continue work on Coverage.
http://dublincore.org/usageboardwiki/IssuesWithCoverage

Old actions

ACTION 2008-09-21: Joe and Andrew to continue work on Coverage.
http://dublincore.org/usageboardwiki/IssuesWithCoverage

ACTION 2008-09-21: Tom (and Mikael) to continue work on Simple Dublin Core.

ACTION 2008-09-21: Tom to amend the naming policy (DCMI names should not differ only with respect to case).

ACTION 2008-09-21: Tom to correct RDF schemas of DCMI Metadata Terms to use blank node with publisher.

ISSUE 2008-09-21: Is it good practice to provide a Value VES in addition to a Value URI, or is it arguably redundant? (Example: "Entity Type" under "Scholarly Work" in SWAP, http://dublincore.org/scholarwiki/SWAPDSP).
Guidance on this point may belong in the application profile guidelines.

ISSUE 2008-09-21: Rendering of Wiki pages using the DSP syntax does not capture all of the DSP detail and uses different labelling. It would be useful to hyperlink labels in the DSP rendered page to the DSP documentation itself. Also, to hyperlink description template labels within the wiki page itself, e.g. creator to agent description.