Dublin Core Metadata Initiative
Making it easier to find information

Guidelines for implementing Dublin Core in XML

Creator: Andy Powell [mailto:a.powell@ukoln.ac.uk]
UKOLN
University of Bath

Creator: Pete Johnston [mailto:p.johnston@ukoln.ac.uk]
UKOLN
University of Bath

Date Issued: 2003-04-02

Identifier: http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/ [/documents/2003/04/02/dc-xml-guidelines/]

Replaces: http://dublincore.org/documents/2002/12/02/dc-xml-guidelines/ [/documents/2002/12/02/dc-xml-guidelines/]

Is Replaced By: Not Applicable

Latest version: http://dublincore.org/documents/dc-xml-guidelines/ [/documents/dc-xml-guidelines/]

Status of document: This is a Dublin Core Metadata Initiative Recommendation [/documents/#recommendations] .

Description of document: This document provides guidelines for people implementing Dublin Core metadata applications using XML. It considers both simple (unqualified) DC and qualified DC applications. In each case, the underlying metadata model is described (in a syntax neutral way), followed by some specific guidelines for XML implementations. Some guidance on the use of non-DC metadata within DC metadata applications is also provided.

1. Introduction

This document provides guidelines for people implementing Dublin Core [DCMI [#DCMI]] metadata applications using XML [XML [#XML]]. It considers both simple (unqualified) DC and qualified DC applications. In each case, the underlying metadata model is described (in a syntax neutral way), followed by some specific guidelines for XML implementations. Some guidance on the use of non-DC metadata is also provided.

This document does **not** provide guidelines for encoding Dublin Core in RDF/XML [RDF [#RDF]]. Nor does it take a position on the relative merits of encoding metadata in 'plain' XML rather than RDF/XML. This document provides guidelines in those cases where RDF/XML is not considered appropriate. Mechanisms for encoding Dublin Core metadata in RDF/XML are being developed elsewhere [DCARCH [#DCARCH]].

2. Terminology

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*.

Property
: a *property* is a specific aspect, characteristic, attribute, or relation used to describe a *resource*.

Record

a *record* is some structured metadata about a *resource*, comprising one or more *properties* and their associated *values*.

Note that Dublin Core metadata elements are *properties* (as defined above). Note also that there is potential confusion between the XML usage of the terms 'element' and 'attribute' and the usage of those terms in a more general metadata context.

3. General implementation guidelines

Recommendation 1. Implementors should base their XML applications on XML Schemas [XMLSCHEMA [#XMLSCHEMA]] rather than XML DTDs. Approaches based on XML Schemas are more flexible and are more easily re-used within other XML applications. In some cases it may be sensible to provide both an XML Schema and a DTD for the application. Where XML Schemas are not used, a DTD should be provided instead. The DCMI maintains a list of XML schemas that are in use in projects or products using DCMI metadata [DCXMLS [#DCXMLS]].

Recommendation 2. Implementors should use XML Namespaces [XMLNS [#XMLNS]] to uniquely identify DC elements, element refinements and encoding schemes. DC namespaces are defined in the DCMI Namespace Recommendation [DCMINS [#DCMINS]].

Note that it is anticipated that *records* will be encoded within one or more container XML element(s) of some kind. This document makes no recommendations for the name of any container element, nor for the namespace that the element should be taken from. Candidate container element names include `<dc>`, `<dublinCore>`, `<resource>`, `<record>` and `<metadata>`.

4. Simple Dublin Core

4.1 Abstract model

- A *simple DC record* is made up of one or more *properties* and their associated *values*.
- Each *property* is an attribute of the *resource* being described.
- Each *property* must be one of the 15 DCMES [DCMES [#DCMES]] elements.
- *Properties* may be repeated.
- Each *value* is a literal string.
- Each literal string *value* may have an associated language (e.g. en-GB).

Note that there is no formal linkage between a *simple DC record* and the *resource* being described. Such a linkage may be made by encoding the URI of the *resource* as the *value* of the DC Identifier element, however this is not mandatory.

Note that while the *value* of a *property* may be a URI, there is nothing in the simple DC model that indicates this is the case. At their own risk, implementations may choose to guess which *values* are URIs and which are not.

4.2 Implementation guidelines

(Implementors should follow the general guidelines.)

Recommendation 3. Implementors should encode *properties* as XML elements and *values* as the content of those elements. The name of the XML element should be an XML qualified name (QName) which associates the element name with the appropriate DCMI namespace name. For example, use

```
<dc:title>Dublin Core in XML</dc:title>
```

rather than

```
<dc:title value="Dublin Core in XML" />
```

Recommendation 4. The *property* names for the 15 DC elements should be all lower-case. For example, use

```
<dc:title>Dublin Core in XML</dc:title>
```

rather than

```
<dc:Title>Dublin Core in XML</dc:Title>
```

Recommendation 5. Multiple *property values* should be encoded by repeating the XML element for that *property*. For example:

```
<dc:title>First title</dc:title>
<dc:title>Second title</dc:title>
```

4.3 Example - a simple DC record

```
<?xml version="1.0"?>

<metadata
  xmlns="http://example.org/myapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/myapp/ http://example.org/myapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <dc:title>
    UKOLN
  </dc:title>
  <dc:description>
    UKOLN is a national focus of expertise in digital information
    management. It provides policy, research and awareness services
    to the UK library, information and cultural heritage communities.
    UKOLN is based at the University of Bath.
  </dc:description>
  <dc:publisher>
    UKOLN, University of Bath
  </dc:publisher>
  <dc:identifier>
    http://www.ukoln.ac.uk/
  </dc:identifier>

</metadata>
```

Note that the `http://example.org/myapp/schema.xsd` XML schema does not exist - this is a ficticious example.

5. Qualified Dublin Core

5.1 Abstract model

- A *qualified DC record* is made up of one or more *properties* and their associated *values*.
- Each *property* is an attribute of the *resource* being described.
- Each *property* must be either:
  - one of the 15 DC elements,
  - one of the other elements recommended by the DCMI (e.g. audience) [DCTERMS [#DCTERMS]],
  - one of the *element refinements* listed in the DCMI Metadata Terms recommendation [DCTERMS [#DCTERMS]].
- *Properties* may be repeated.
- Each *value* is a literal string.

- Each *value* may have an associated *encoding scheme*.
- Each *encoding scheme* has a *name*.
- Each literal string *value* may have an associated language (e.g. en-GB).

Note that for encoding schemes currently recommended by the DCMI, the name is specified in the DCMI Metadata Terms recommendation [DCTERMS [#DCTERMS]] (listed as the 'Term name', not the 'Label'). It is anticipated that the DCMI will develop other mechanisms for registering agreed names for schemes in the future.

5.2 Implementation guidelines

(Implementors should follow the general guidelines and the guidelines for simple Dublin Core.)

Recommendation 6. *Element refinements* should be treated in the same way as other *properties*. The name of the XML element should be an XML qualified name (QName) which associates the element refinement name with the appropriate DCMI namespace name. For example:

```
<dcterms:available>2002-06</dcterms:available>
```

Element refinements are elements in their own right and are therefore best encoded in a similar way to other DC elements. In particular, it should be noted that element refinements may have further refinements of their own (e.g. 'format' is refined by 'extent' which might be further refined by 'duration').

See also Note 1 [#note1] below.

Recommendation 7. *Encoding schemes* should be implemented using the 'xsi:type' attribute of the XML element for the *property*. The name of the encoding scheme should be given as the attribute value, and should be in the form of an XML qualified name (QName) which associates the scheme name with the appropriate namespace name. For example:

```
<dc:identifier xsi:type="dcterms:URI">http://www.ukoln.ac.uk/</dc:identifier>
```

See also Note 2 [#note2] below.

Recommendation 8. *Element refinements* and *encoding schemes* should use the names specified in the DCMI Metadata Terms recommendation [DCTERMS [#DCTERMS]] (listed as the 'Term name', not as the 'Label'). As a general rule, element and element refinement names may be mixed-case but should always have a lower-case first letter; encoding scheme names may be mixed-case but should always start with an upper-case letter; encoding scheme names are often all upper-case. For example:

```
<dcterms:isPartOf xsi:type="dcterms:URI">
  http://www.bbc.co.uk/
</dcterms:isPartOf>

<dcterms:temporal xsi:type="dcterms:Period">
  name=The Great Depression; start=1929; end=1939;
</dcterms:temporal>
```

Recommendation 9. Where the language of the *value* is indicated, it should be encoded using the 'xml:lang' attribute. For example:

```
<dc:subject xml:lang="en">seafood</dc:subject>
<dc:subject xml:lang="fr">fruits de mer</dc:subject>
```

5.3 Example - a qualified DC record

```
<?xml version="1.0"?>

<metadata
```

```
xmlns="http://example.org/myapp/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://example.org/myapp/ http://example.org/myapp/schema.xsd"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/">

<dc:title>
  UKOLN
</dc:title>
<dcterms:alternative>
  UK Office for Library and Information Networking
</dcterms:alternative>
<dc:subject>
  national centre, network information support, library
  community, awareness, research, information services,public
  library networking, bibliographic management, distributed
  library systems, metadata, resource discovery,
  conferences,lectures, workshops
</dc:subject>
<dc:subject xsi:type="dcterms:DDC">
  062
</dc:subject>
<dc:subject xsi:type="dcterms:UDC">
  061(410)
</dc:subject>
<dc:description>
  UKOLN is a national focus of expertise in digital information
  management. It provides policy, research and awareness services
  to the UK library, information and cultural heritage communities.
  UKOLN is based at the University of Bath.
</dc:description>
<dc:description xml:lang="fr">
  UKOLN est un centre national d'expertise dans la gestion de l'information
  digitale.
</dc:description>
<dc:publisher>
  UKOLN, University of Bath
</dc:publisher>
<dcterms:isPartOf xsi:type="dcterms:URI">
  http://www.bath.ac.uk/
</dcterms:isPartOf>
<dc:identifier xsi:type="dcterms:URI">
  http://www.ukoln.ac.uk/
</dc:identifier>
<dcterms:modified xsi:type="dcterms:W3CDTF">
  2001-07-18
</dcterms:modified>
<dc:format xsi:type="dcterms:IMT">
  text/html
</dc:format>
<dcterms:extent>
  14 Kbytes
</dcterms:extent>
```

```
</metadata>
```

6. Mixing DC metadata with other metadata schemas

Many metadata applications will mix Dublin Core metadata with *properties* taken from other metadata schemas. There are several reasons for wanting to do this including the need for DC-based metadata applications to incorporate semantics that are not available within the DCMES and the desire to incorporate DCMES elements within other metadata applications.

Where possible, all use of DC metadata in XML should follow the guidelines above.

Non-DC properties can be encoded as XML elements alongside DC elements. Here are two simple examples:

6.1 Example - mixing DC and IMS metadata

This example adds an IMS (IEEE LOM) [IMS [#IMS]] TypicalLearningTime property to a simple DC record:

```
<?xml version="1.0"?>

<record
  xmlns="http://example.org/learningapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/learningapp/ http://example.org/learningapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ims="http://www.imsglobal.org/xsd/imsmd_v1p2">

  <dc:title>
    Frog maths
  </dc:title>
  <dc:identifier>
    http://somewhere.com/frogmaths/
  </dc:identifier>
  <dc:description>
    Simple maths games for 5-7 year olds.
  </dc:description>
  <ims:typicallearningtime>
    <ims:datetime>
      0000-00-00T00:15
    </ims:datetime>
  </ims:typicallearningtime>

</record>
```

Note that DC case conventions for element names may not apply to other metadata schemas.

6.2 Example - mixing DC and ODRL metadata

This example adds a machine-readable Open Digital Rights Language [ODRL [#ODRL]] statement and a DC Rights element to the example above:

```
<?xml version="1.0"?>

<record
  xmlns="http://example.org/learningapp/"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/learningapp/ http://example.org/learningapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:oex="http://odrl.net/1.0/ODRL-EX"
  xmlns:odd="http://odrl.net/1.0/ODRL-DD"
  xmlns:ims="http://www.imsglobal.org/xsd/imsmd_v1p2">

  <dc:title>
    Frog maths
  </dc:title>
  <dc:identifier xsi:type="dcterms:URI">
    http://somewhere.com/frogmaths/
  </dc:identifier>
  <dc:description>
    Simple maths games for 5-7 year olds.
  </dc:description>
  <ims:typicallearningtime>
    <ims:datetime>
      0000-00-00T00:15
    </ims:datetime>
  </ims:typicallearningtime>
  <dc:rights>
    Permission is granted for anyone to display, copy, modify and annotate
    this software.
  </dc:rights>

  <oex:rights>
   <oex:asset>
     <oex:context>
        <odd:uid idscheme="URI">
          http://somewhere.com/frogmaths/
        </odd:uid>
     </oex:context>
   </oex:asset>
   <oex:permission>
     <odd:display/>
     <odd:modify/>
     <odd:annotate/>
   </oex:permission>
  </oex:rights>

</record>
```

Note that ODRL uses structural conventions (i.e. the nesting of XML elements) that are not used in the DC XML encoding proposed in this document. This may also be true for other metadata applications, such as IMS.

Notes

[Note 1] Recommendation 6 assumes that it is **not** required that Qualified Dublin Core metadata records include explicit information on the relationships between refinements and the elements they refine. Such relationships are well documented elsewhere and may be expressed in machine understandable ways in schema definitions.

Some implementors, however, prefer to represent this relationship within metadata records.

Some suggest the use of a 'refines' attribute. For example:

```
<dcterms:alternative refines="dc:title">foo</dcterms:alternative>
```

Others suggest using element containment, with XML elements representing refinements nested within XML elements representing the elements they refine. For example

```
<dc:title>
  <dcterms:alternative>foo</dcterms:alternative>
</dc:title>
```

Neither of these approaches is recommended by these guidelines. However, it may be sensible for software applications that consume DCQ in XML to accept either of these alternative representations, assuming it is correctly and consistently applied.

[Note 2] Recommendation 7 specifies the use of 'xsi:type' to implement encoding schemes in Qualified Dublin Core. This allows implementors to take some advantage of the data typing functionality provided by the W3C XML Schema specifications [XMLSCHEMA [#XMLSCHEMA]].

Some implementors have suggested instead the use of conventions which are independent of a specific schema definition language. One such convention is the use of a 'scheme' attribute of the XML element for the property. For example:

```
<dc:identifier scheme="dcterms:URI">http://www.ukoln.ac.uk/</dc:identifier>
```

This approach is **not** recommended by these guidelines. However, it may be sensible for software applications that consume DCQ in XML to accept this alternative representation, assuming it is correctly and consistently applied.

References

[DCMI]
Dublin Core Metadata Initiative
http://dublincore.org/ [/]

[XML]
Extensible Markup Language (XML)
http://www.w3.org/XML/ [http://www.w3.org/XML/]

[DCMES]
Dublin Core Metadata Element Set, Version 1.1: Reference Description
http://dublincore.org/documents/dces/ [/documents/dces/]

[DCTERMS]
DCMI Metadata Terms
http://dublincore.org/documents/dcmi-terms/ [/documents/dcmi-terms/]

[RDF]
Resource Description Framework (RDF)
http://www.w3.org/RDF/ [http://www.w3.org/RDF/]

[DCARCH]
DCMI Architecture Working Group
http://dublincore.org/groups/architecture/ [/groups/architecture/]

[XMLSCHEMA]
XML Schema
http://www.w3.org/XML/Schema [http://www.w3.org/XML/Schema]

[DCXMLS]
DCMI Metadata expressed in XML Schema Language

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

[XMLNS] Namespaces in XML
http://www.w3.org/TR/1999/REC-xml-names-19990114/ [http://www.w3.org/TR/1999/REC-xml-names-19990114/]

[DCMINS]
Namespace Policy for the Dublin Core Metadata Initiative (DCMI)
http://dublincore.org/documents/dcmi-namespace/ [/documents/dcmi-namespace/]

[IMS]
IMS Learning Resource Meta-data Information Model - Version 1.2 Final Specification
http://www.imsglobal.org/metadata/imsmdv1p2/imsmd_infov1p2.html [http://www.imsglobal.org/metadata/imsmdv1p2/imsmd_infov1p2.html]

[ODRL]
The Open Digital Rights Language Initiative
http://odrl.net/ [http://odrl.net/]