

CEN

CWA 14855

WORKSHOP

November 2003

AGREEMENT

ICS 35.100.05; 35.240.60

English version

Dublin Core Application Profile Guidelines

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Luxembourg, Malta, Netherlands, Norway, Portugal, Slovakia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Contents

Contents.....	2
Foreword.....	3
Introduction	4
1 Scope.....	5
2 Definitions	6
3 Identifying terms with appropriate precision	7
4 Attributes of a Term Usage	9
4.1 Identifying attributes.....	9
4.2 Definitional attributes	9
4.3 Relational attributes.....	9
4.4 Constraints	10
5 Discussion.....	11
5.1 Descriptive Headers	11
5.2 Readability of Term Usages	11
5.3 "Using" Element Refinements, Encoding Schemes, and Vocabulary Terms	12
5.4 Attributes copied from external sources	12
5.5 Types of Comments	13
5.6 Term URIs versus Qualified Names.....	13
5.7 Declaring new elements	13
5.8 Documenting grouped or nested metadata elements	13
5.9 Documenting unorthodox practices.....	14
6 Examples	15
6.1 UK Resource Discovery Network OAI Application Profile	15
6.2 Renardus Application Profile	16
6.3 UK e-Government Metadata Standard Application Profile	17
Annex A: Metadata describing a DCAP	19
Annex B: Options for machine-interpretable DCAPs	20
Bibliography	21

Foreword

The production of this CEN Workshop Agreement (CWA) was formally accepted as part of the CEN/ISSS Workshop on Metadata for Multimedia Information - Dublin Core (WS/MMI-DC) in the Workshop's plenary meeting on 2002-03-07.

This CWA was agreed upon by the contributing partners in the CEN/ISSS Workshop on MMI-DC, representing a wide mix of interests, including administrations, libraries, on-line education and geographic information systems. The list of company individuals who have supported the document's contents may be obtained from the CEN/ISSS Secretariat.

The CWA was approved by the Workshop's plenary meeting on 2003-09-08.

The final text of this CWA was submitted to CEN for publication on 2003-09-09.

Introduction

A Dublin Core Application Profile (DCAP) 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.

A DCAP is designed to promote interoperability within the constraints of the Dublin Core model and to encourage harmonization of usage and convergence on “emerging semantics” around its edges. Historically, application profiles have emerged out of a need to share local domain- or application-specific refinements of or extensions to Dublin Core within particular application communities without necessarily seeking an extension of the core standard maintained by the Dublin Core Metadata Initiative (DCMI). Application profiles document how implementers use elements from Dublin Core along with elements from other vocabularies, customizing standard definitions and usage guidelines for local requirements [HEERY].

In practice, application profiles are created for a wide range of purposes: to document the semantics and constraints used for a set of metadata records (“instance metadata”); to help communities of implementers harmonize metadata practice among themselves; to identify emerging semantics as possible candidates for formal standardization; as guides for semantic crosswalks and format conversions; as specifications for formal encoding structures such as Document Type Definitions (DTDs); for interpreting or presenting legacy or proprietary metadata in terms of widely-understood standards; or for documenting the rules and criteria according to which a set of metadata was created. Application profiles often represent “work in progress”, providing foci for ongoing efforts to incrementally improve and clarify a body of shared metadata semantics within a particular user community.

In the absence of guidelines, creators of application profiles have hitherto invented a wide range of presentation formats. The present document distills the salient features of many existing profiles into a format that is as concise and simple as possible, yet as precise and detailed as is sometimes necessary to support the various uses identified above.

Semantic interoperability – the ultimate purpose of documents such as DCAPs – is a longer-term goal to be pursued as metadata vocabularies and related enabling technologies mature over time. In their current form, DCAPs are designed to document metadata usage in a normalized form that will lend itself to translation into common models, such as RDF, that can be processed by machines to automate such interoperability.

Machine-understandable representations will achieve this goal to the extent that metadata terms can be referenced using stable, well-documented identifiers. As discussed below, the practice of identifying metadata terms with Uniform Resource Identifiers (URIs) is currently gaining momentum. Maintaining a DCAP over time, then, may involve improving its precision incrementally by identifying its terms with URIs as the URIs become available; this is referred to here as the Principle of Appropriate Identification.

In the meantime, these guidelines aim at the more modest aim of providing system developers and information specialists with a normalized and readable view of Dublin-Core-based metadata models. A DCAP should include enough information to be of optimal usefulness for its intended audience – a Principle of Readability – even if this entails the redundant inclusion of information, which, in a formal system of machine-processable schemas, might otherwise be fetched dynamically from external sources.

Given the flexibility of presentation required by the Principle of Readability, no assumption is made that DCAPs will be convertible into future machine-understandable forms without the use of ad-hoc heuristics or manual intervention. Creators of DCAPs should bear in mind that a normalized form of documentation cannot itself address the deeper problems of interoperability in a world with a diversity of underlying metadata models – problems which will continue to challenge the metadata community as a whole, and the Dublin Core Metadata Initiative in particular, for the foreseeable future.

1 Scope

The present document gives guidance on how information should be structured and presented in Dublin Core Application Profiles. Principles and concepts underlying DCAPs as declarative metadata constructs are defined and explained.

The guidelines do not mandate a particular document format for DCAPs. DCAPs may be presented as plain text files or as Web pages, word-processing files, PowerPoint, or indeed as ink on paper. By providing a consistent presentation structure for such documents, these guidelines aim at making it easier for people to understand what others are doing in their metadata. The guidelines mandate enough structure to ensure that DCAPs will be convertible as straightforwardly as possible into expressions that use schema languages, such as RDF, for automatic processing by machines. In this sense, a normalized documentary form for DCAPs is a first step towards the more ambitious and long-term goal of automating semantic interoperability across a broad diversity of information sources.

2 Definitions

Dublin Core Application Profile (DCAP): A DCAP is a declaration specifying which metadata terms an organization, information provider, or user community uses in its metadata and how those terms have been customized or adapted to a particular application. By definition, a DCAP is based in part on Dublin Core and follows DCMI Grammatical Principles [DCMI-PRINCIPLES]. A DCAP consists of a Descriptive Header and one or more Term Usages.

DCMI Grammatical Principles: As maintained by the Dublin Core Metadata Initiative, DCMI grammatical principles specify a typology of metadata terms – Elements, Element Refinements, Encoding Schemes, and Vocabulary Terms – along with their interrelationships and functions [DCMI-PRINCIPLES]. A DCAP is based on the simple model of a resource described with a flat set of properties. This is consistent with DCMI grammatical principles, which do not themselves specify more elaborate models.

Descriptive Header: A Descriptive Header places the DCAP into an interpretive context by specifying, at a minimum, a Title, Creator, Date, Identifier, and Description for the DCAP. An optional Preamble may comment on any technical or stylistic conventions followed in the DCAP.

Term Usage: A Term Usage is a description of a metadata term, which, at a minimum, identifies a metadata term in accordance with the Principle of Appropriate Identification by using one or more identifying attributes – Term URI, Defined By, Name, Label – as described in Section 3. Optionally, a Term Usage may also describe or annotate a term in more detail by providing additional definitional attributes, relational attributes, or constraints, as described in Section 4.

Principle of Appropriate Identification: The Principle of Appropriate Identification dictates that metadata terms be identified as precisely as possible. As established in the so-called CORES Resolution of December 2002, the preferred method for identifying a metadata term is to cite its Uniform Resource Identifier (URI). All DCMI metadata terms are identified with URIs, and URIs are currently being assigned to the terms of other major semantic standards such as MARC21 and IEEE/LOM [CORES-RESOLUTION]. Whenever such URIs are available they should be cited as an attribute of a Term Usage (its “Term URI”). Terms to which URIs have not (or not yet) been assigned should be identified using other attributes as appropriate, as described in Section 3.

Principle of Readability: The Principle of Readability dictates that a DCAP should include enough information in Term Usages to be of optimal usefulness for the intended audience of the DCAP – even if this entails the redundant inclusion of information which, in a formal system of machine-processable schemas, might otherwise be fetched dynamically from external sources. Conversely, the Principle of Readability allows unused attributes simply to be omitted from display.

3 Identifying terms with appropriate precision

Application profiles serve to clarify who is declaring and maintaining the metadata semantics that a group wants to share. This section describes how a metadata term used in a Term Usage can be identified with appropriate precision (the Principle of Appropriate Identification).

At present, the preferred method for identifying a metadata term is to cite its Uniform Resource Identifier (URI) if such is available. A URI is "a compact string of characters for identifying an abstract or physical resource" constructed according to a generic and flexible syntax [URI]. The World Wide Web Consortium has promoted the notion that "All important resources should be identified by a URI" [WEBARCH] and has specifically promoted the use of URIs for identifying metadata elements. In the CORES Resolution of December 2002, the maintainers of seven leading metadata standards – Dublin Core, IEEE/LOM, DOI, CERIF, MARC21, ONIX, and GILS -- pledged to assign URIs to their elements and to articulate policies for the persistence of those URIs [CORES-RESOLUTION]. (Note that a URI, when used to identify a metadata term, often functions as a Web address for accessing information about that term, such as a Web page or machine-processable schema. However, the CORES Resolution does not require that such identifiers resolve to such resources, and URIs that result in "file not found" messages are not necessarily "broken" as identifiers.)

For metadata terms to which a URI has been officially assigned – for example, by DCMI or by another signatory of the CORES Resolution – that URI should be cited in the field "Term URI". For example, the Dublin Core element "Audience" should be cited as "<http://purl.org/dc/terms/audience>". As this form of identification is precise and sufficient on its own, other identifying fields may be left blank:

Term URI	http://purl.org/dc/terms/audience
Name	-
Label	-
Defined By	-

In accordance with the Principle of Readability, other identifying attributes such as Name and Label could be added here to make the DCAP more "reader-friendly". If the DCAP is intended as a guide for processing metadata records, it may indeed be necessary to provide a Name (i.e., the string actually used in the metadata records). If the Name or Label of a term are considered more "reader-friendly" as captions for a Term Usage than the Term URI, the order of these attributes may be changed to put these first. See Sections 5.4 ("Attributes copied from external sources") and 5.2 ("Readability of Term Usages") for further discussion.

A term that has been declared or documented somewhere but not assigned a URI (as far as one knows) should be identified as precisely as possible by providing its name and pointing to a declarative document or schema in which it has been defined. The declarative document or schema should be cited with URI, Web address, or bibliographic reference in the field "Defined By". The term itself can be cited using either a string identifier or token (in the field "Name", which by default is assumed to be case-sensitive) or a natural-language label (in the field "Label"), or both, taken from the declarative document or schema:

Term URI	-
Name	AttendancePattern
Label	Attendance Pattern
Defined By	http://someones-project.org/schema.html

CWA 14855:2003 (E)

For a term that has not already been defined in any other declarative document, the field Defined By should simply cite the URI of the DCAP itself (as assigned with Identifier in the DCAP Descriptive Header). For example, in a DCAP with the URI "http://my-project.org/profile.html", a new local term called Star Ratings could be defined as follows:

Term URI	-
Name	StarRatings
Label	Star Ratings
Defined By	http://my-project.org/profile.html

A creator of a DCAP wishing to declare locally coined terms in a way that makes them citable with precision, and thus re-usable by others, may undertake the additional step of assigning them URIs. At present, the technical conventions and "Web etiquette" for naming metadata terms with URIs have yet to establish themselves in common practice, though at a minimum it seems both polite and sensible not to promote new URIs unless it is expected they will be maintained. For the purposes of DCAPs, DCMI itself provides models of practice, and further options are likely to emerge as the CORES Resolution is implemented [DCMI-NAMESPACE, DCMI-TERMS, DCMI-SCHEMAS]. Note that the CORES Resolution itself addresses the use of URIs as identifiers only and is silent on whether the URIs should resolve to informational Web pages or schemas [CORES-RESOLUTION].

4 Attributes of a Term Usage

Attributes for describing the metadata terms "used" in a DCAP are listed below. Note that they are called "attributes" here simply to avoid confusingly recursive formulations such as "terms for describing terms".

Use of Identifying Attributes in Term Usages (see Section 4.1) is governed by the Principle of Appropriate Identification. According to this principle, a Term Usage should use one or more of the four Identifying Attributes to identify a term as precisely as appropriate – i.e., with a formally assigned URI if available, or alternatively by citing a name or label for the term along with a reference to a document, schema, or Web page in which that term is defined.

All of the other attributes of Term Usages are optional and should be used as local needs may dictate. As discussed in Section 5.4, "local" and "source" attributes may be distinguished as necessary.

4.1 Identifying attributes

Term URI	A Uniform Resource Identifier used to identify the term.
Name	A unique token assigned to the term.
Label	A human-readable label assigned to the term.
Defined By	An identifier of a namespace, pointer to a schema, or bibliographic reference for a document within which the term is defined.

4.2 Definitional attributes

Definition	A statement that represents the concept and essential nature of the term.
Comments	Additional information about the term or its application.
Type of term	A grammatical category of the term (e.g., "Element", "Element Refinement", or "Encoding Scheme").

4.3 Relational attributes

Refines	The described term semantically refines the referenced term.
Refined By	The described term is semantically refined by the referenced term.
Encoding Scheme For	The described term, an Encoding Scheme, qualifies the referenced term.
Has Encoding Scheme	The described term is qualified by the referenced Encoding Scheme.
Similar To	The described term has a meaning the same as, or similar to, that of the referenced term.

4.4 Constraints

Obligation	Indicates whether the element is required to always or sometimes be present (i.e., contain a value). Examples include "Mandatory", "Conditional", and "Optional".)
Condition	Describes the condition or conditions according to which a value shall be present.
Datatype	Indicates the type of data that can be represented in the value of the element.
Occurrence	Indicates any limit to the repeatability of the element.

5 Discussion

5.1 Descriptive Headers

By definition, a DCAP consists of a Descriptive Header and one or more Term Usages (see Section 2). The Description Header should include the following:

- A brief description of the DCAP based on Dublin Core. At a minimum, the description should specify a Title, Contributor, Date, Identifier, and Description, as explained in more detail in Annex A. Ideally, the description of the DCAP will elaborate on the context in which the DCAP is intended to be used.
- Optionally, a Preamble for the DCAP should describe any technical or formatting conventions used in the DCAP. For example, if namespace prefixes are used in the Name field (see Section 5.6), these prefixes should be documented here. The Preamble can also cite Web pages or schemas in which the terms used in the DCAP are documented and defined so that such information does not need to be repeated in the “Defined By” field of each individual Term Usage; see the example in Section 6.1.1.

5.2 Readability of Term Usages

By default, each term cited in a DCAP should be described with its own Term Usage – a table with a full set of attributes on the left and attribute values on the right. In accordance with the Principle of Readability, however, the intended use of a DCAP may dictate a different presentational style: while DCAPs intended for use by software developers will need to be explicit and detailed, DCAPs intended primarily as informational documents for human consumption can (and often should) be much terser. The following are several ways in which Term Usages may be formatted for readability:

- Instead of creating a separate Term Usage for every Element Refinement and Encoding Scheme used in an application, such terms may simply be cited in the attributes Refined By or Has Encoding Scheme of the Term Usages of the Elements to which they refer (see Section 5.3). Note that this terser style does not support the addition of usage notes, local definitions, or annotations, for which a full Term Usage must be used.
- Attributes not needed for Term Usages can simply be omitted. At one extreme, a Term Usage might legitimately consist of just a Name and a Term URI; see the example in Section 6.1.2).
- The order of attributes presented in Section 4 is significant only for the usability of DCAPs as documents – not for future machine-processable representations of DCAPs. Authors of DCAPs may therefore change the order of attributes in the interest of readability, though they should bear in mind that any such changes may make it more difficult for people to compare two DCAPs visually. For examples of how Name (in boldface) is placed before Term URI, see Section 6.1.2 and DCMI-TERMS.
- In the interest of readability, it might make sense to describe Elements, Element Refinements, and Encoding Schemes with different subsets of relevant attributes. Indeed, these different types of terms might be grouped under separate sections of the DCAP document. (For example, see how Elements and Element Refinements are separated from Encoding Schemes in the document “DCMI Metadata Terms” [DCMI-TERMS].)
- In the interest of readability and of future machine-parsability, attributes should be repeated when necessary (as opposed to listing multiple values for a single attribute).

5.3 “Using” Element Refinements, Encoding Schemes, and Vocabulary Terms

5.3.1 Using Vocabulary Terms

According to DCMI Grammatical Principles, a Vocabulary Term is a member of a controlled vocabulary of values, and a controlled vocabulary of values (as a whole) is named by an Encoding Scheme [DCMI-PRINCIPLES].

In general, it is not the role of application profiles to declare controlled vocabularies of values, either in the sense of creating lists of potential values or in the sense of giving that list (as a whole) a name and URI. Sets of Vocabulary Terms are most appropriately declared in separately citable documents external to a DCAP.

However, if the creator of a DCAP merely wishes to specify a short list of possible values (e.g., "Animal, Vegetable, or Mineral"), these can be simply listed in a "Comment" field.

5.3.2 Using Encoding Schemes

There are three ways to cite Encoding Schemes in a DCAP:

- The most concise way is to use the attribute Has Encoding Scheme (or Comment) for a blanket reference to a set of encoding schemes documented elsewhere. The DCAP for Resource Discovery Network, for example, simply cites “RDN Subject Encoding Schemes” and gives a URL where the list of those encoding schemes may be found; see the example in Section 6.1.2.
- A more precise way is to use the attribute Has Encoding Scheme, repeated as necessary, to cite each Encoding Scheme by its URI (or by Name and URI); for example, see the example in Section 6.3.2.
- In addition to citing Encoding Schemes in the Has Encoding Scheme attribute of Elements, creators of DCAPs may want to describe Encoding Schemes in stand-alone Term Usages in order to annotate their usage, for example by specifying a Datatype, Occurrence, or Local Definition. The attribute Encoding Scheme For points back to the Element or Element Refinement qualified.

5.3.3 Using Element Refinements

The options for Element Refinements are analogous to those for Encoding Schemes:

- Statements such as “all terms in Vocabulary D can be used as element refinements for Contributor” can be simply recorded in a Refined By attribute (or as a Comment).
- Element Refinements can be cited one-by-one using the attribute Refined By; see the example in Section 6.3.2.
- Element Refinements can additionally be described in separate Term Usages.

5.4 Attributes copied from external sources

Ideally, application profiles would be dynamically up-dated with information on the terms they use directly from schemas on the Web and this information would be integrated with local annotations into a “one-stop” document for the convenience of users. The use of machine-understandable DCAPs may some day make this possible.

In the meantime, however, creators of DCAPs who wish to include definitions or other such information from original source documents in their Term Usages have no choice but to copy that information from the source. While the Principle of Readability specifically permits this, authors of DCAPs should bear in mind that copied information, if not maintained, can go out of alignment with the official source.

Where information copied from external sources is supplied, this fact should be reported in the Preamble as described in Section 5.1. Where it is necessary to distinguish in a DCAP between attributes defined locally and attributes copied from an external source, the DCAP should establish its own document-internal

convention, such as distinguishing between a Local Definition and a Source Definition; see the example in Section 6.3.2.

5.5 Types of Comments

Past creators of application profiles for Dublin Core have invented many types of annotation, the most popular of which have been Notes, Best Practice, Usage, Scope, Open Questions, Examples, Purpose, Guidelines, and Don't Confuse With. While the present guidelines lump all of the above into a generically named Comments field, creators of DCAPs may wish to repeat this field with different labels as needed. The needs of future machine processing do not now seem to dictate tighter uniformity in this area.

5.6 Term URIs versus Qualified Names

In the sense intended here, Qualified Names are names of metadata terms that are “qualified” with a prefix standing for a namespace with which the terms are associated (a “namespace prefix”). For example, the Dublin Core element “Title” is sometimes referenced in metadata records and usage documentation using a namespace prefix such as “DC.” or “dc:” as in “DC.Title” or “dc:title”. As straightforward as this citation method may seem, it is based on assumptions about the nature of “namespace” that cannot be assumed to hold across different application environments (e.g., HTML versus RDF versus relational databases) or metadata communities (e.g., for citing elements from standards other than Dublin Core), and at any rate it presupposes an additional mechanism or declaration for associating prefixes with the proper namespaces.

For such reasons, it is far better to cite an element with a full URI – indeed, this is the only method supported by the CORES Resolution and by DCMI policy [CORES-RESOLUTION, DCMI-NAMESPACE]. According to the Principle of Appropriate Identification followed in these guidelines, a Term URI must be cited when available.

On the other hand, long strings such as “http://purl.org/dc/elements/1.1/title” are not very readable and may be misunderstood by the average reader of a DCAP. In accordance with the Principle of Readability, therefore, the author of a DCAP may choose to use qualified names (e.g., “dc:title”) in the “Name” field – as long as any prefixes used are explained in the Preamble of the DCAP, and as long as any available Term URIs are cited as well.

5.7 Declaring new elements

There is nothing to restrain the creator of a DCAP from creating new URIs as identifiers for locally coined metadata terms. For reasons discussed above in Section 3, one should perhaps pause for reflection before taking this step, and if URIs are declared, this step should perhaps be documented separately and not embedded “in passing” into a DCAP full of Term Usages. Any URIs declared for use in a DCAP might best be formed by following the DCMI algorithm and concatenating the URL of the DCAP (e.g., “http://myproject.org/profile/”) and the Name of the term (e.g., “starRatings”) into a single string (e.g., “http://myproject.org/profile/starRatings”) [DCMI-NAMESPACE]. Other models for forming URIs as identifiers for metadata elements are emerging with the implementation of the CORES Resolution [CORES-RESOLUTION].

5.8 Documenting grouped or nested metadata elements

In order to be usable across a diversity of application environments, Dublin Core was designed as a flat set of attributes for describing a resource. In implementation practice, however, Dublin Core elements may be embedded in more elaborate models that group or nest the elements in locally specific ways.

In the absence of a clear and widely accepted data model beyond that of the flat set of attributes, however, applications for integrating metadata from many different sources may be able only to extract and interpret the metadata in terms of Simple Dublin Core, losing any application-specific modelling context. An application designer wishing to document nesting or grouping constructs in a DCAP will need to extend the guidelines described here in order to do so and should bear in mind that documenting such constructs will not in itself guarantee that they will be understood or correctly processed by other applications.

5.9 Documenting unorthodox practices

For reasons both of history and of expedience, a significant number of applications have metadata based on interpretations of the Dublin Core model that are unsound from the standpoint of today's grammatical principles. For example, an application may use `CreatorDateOfBirth` – an element representing the birth date of a creator of a resource that does not, however, semantically "refine" `Creator` as its name may imply.

Rather than incorrectly asserting "`CreatorDateOfBirth`" to be an Element Refinement refining `http://purl.org/dc/elements/1.1/creator`, the Term Usage in the DCAP should simply record the local name of the element and identify the URI of the DCAP itself as its source. For example, if the DCAP itself is identified by "`http://myproject.org/profile/2003/03/17/`", the Term Usage should declare the following, leaving empty any fields (such as "Term URI" and "Refines") that would make incorrect assertions about the element:

Term URI	-
Local Name	<code>CreatorDateOfBirth</code>
Defined By	http://my-project.org/profile.html
Refines	-

Whether "errors" such as "`CreatorDateOfBirth`" will be of negative consequence for interoperability will depend on how they are interpreted and used in the context of particular applications. The analytical effort involved in creating a DCAP is in effect an important first step towards putting such applications onto a more interoperable foundation.

6 Examples

6.1 UK Resource Discovery Network OAI Application Profile

6.1.1 Descriptive Header

Title	RDN OAI Application Profile
Contributor	Andy Powell
Date	2003-03-23
Identifier	URL for this document - to be assigned
Description	This document expresses the application profile established by the Resource Discovery Network (RDN) to be used by RDN partners for harvesting of records using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). The Application Profile is expressed according to guidelines published by the CEN/ISSS [Reference]. Full user documentation for the Application Profile, together with associated XML schemas, is available at http://www.rdn.ac.uk/oai/rdn_dc/ . All Dublin Core terms are fully documented at http://www.dublincore.org/documents/dcmi-terms/ .

6.1.2 A Term Usage

Name	Subject
Term URI	http://purl.org/dc/elements/1.1/subject
Has Encoding Scheme	DC Subject Encoding Schemes
Has Encoding Scheme	RDN Subject Encoding Schemes
Comment	RDN Subject Encoding Schemes are available from http://www.rdn.ac.uk/publications/cat-guide/subject-schemes/
Obligation	Recommended

6.1.3 Commentary

The DCAP for the UK Resource Discovery Network is formatted in the tersest possible style [RDN]. Note in particular the following:

- The Descriptive Header puts the DCAP into a specific usage context.
- Only the attributes actually used in a given Term Usage are shown. Indeed, most of the Term Usages in this DCAP consist of just a Name and a Term URI.
- Instead of listing all of the DCMI- and RDN-maintained Encoding Schemes as separate Has Encoding Scheme entries (or as separate Term Usages), this DCAP uses shorthand references to “DC Subject Encoding Schemes” and “RDN Subject Encoding Schemes”. Pointers to documentation are given in the Descriptive Header (for the former) and in a Comment field of the Term Usage (for the latter).
- This DCAP increases the readability of Term Usages by listing the Name, in boldface, before the Term URI. This option is discussed in Section 5.2.

6.2 Renardus Application Profile

6.2.1 Descriptive Header

Title	Renardus Application Profile
Contributor	Metadata Working Group SUB Göttingen
Date	18-04-2002
Identifier	http://renardus.sub.uni-goettingen.de/renap/renap.html
Description	Cross search and cross browse European quality controlled subject gateways.

6.2.2 A Term Usage

Term URI	http://purl.org/dc/elements/1.1/language
Name	Language
Label	Language
Defined By	-
Definition	-
Comments	Renardus: The language code is the ISO 639-2, three-letter code. SUB will provide a mapping between the two letter and three letter language code but this will also be found on the LoC site - ISO 639-2: http://lcweb.loc.gov/standards/iso639-2/englangn.html
Type of term	Element
Refines	-
Refined By	-
Encoding Scheme For	-
Has Encoding Scheme	http://purl.org/dc/terms/ISO639-2
Similar To	-
Obligation	Mandatory
Condition	-
Datatype	String
Occurrence	Repeatable

6.2.3 Commentary

The DCAP for the Renardus Project has been formatted in a somewhat more verbose style [RENARDUS]. Note in particular:

- The DCAP uses its own URL as an identifier.

6.3 UK e-Government Metadata Standard Application Profile

6.3.1 Descriptive Header

Addressee	Metadata Working Group, Interoperability Working Group
Contributor	Drafted by Interoperability and Metadata Analyst, Office of the e-Envoy, Cabinet Office, UK farah.ahmed@e-envoy.gsi.gov.uk
Contributor	Metadata Working Group
Coverage.spatial	UK
Creator	Senior Policy Advisor, Interoperability and Metadata, Office of the e-Envoy, Cabinet Office, UK
Date.issued	2003-08-05
Description	The elements and refinements that provide the structure for metadata used by the UK public sector, designed to complement the e-GMS.
Format	Text/MS Word 2003
Identifier	http://purl.oclc.org/NET/e-GMS-AP_v1
Language	Eng
Publisher	Office of the e-Envoy, Cabinet Office, UK. govtalk@e-envoy.gsi.gov.uk
Rights.copyright	http://www.hmsso.gov.uk/docs/copynote.htm Crown Copyright
Source	http://purl.oclc.org/NET/e-GMS_v2
Status	Version 1.0 For publication
Subject	Metadata
Subject.category	Information management
Title	UK e-government metadata standard application profile version 1

6.3.2 A Term Usage

Term URI	http://purl.org/dc/elements/1.1/date
Defined by	http://purl.org/dc/elements/1.1/
Name	Date
Label	Date
Source Definition	A date associated with an event in the life cycle of the resource.
Source Comment	-
Local Definition	-
Local Comment:	To enable the user to find the resource by limiting the number of search hits according to a date, e.g. the date the resource was made available.
Purpose	
Local Comment: Notes	Dates need to appear in a format that is recognisable to people all over the world, and that can be interpreted by computer software. The W3C format allows accurate searching, and makes it clear which is the year, month or day. The format is 'ccyy-mm-dd', where 'ccyy' is the year, 'mm' is the month and 'dd' the day. When the time is also needed, add 'hh:mm', where 'hh' is the hour (using the 24 hour clock), 'mm' is minutes. More about this notation can be found at http://www.w3.org/TR/NOTE-datetime .

Local Comment: Not to be confused with	Coverage - Date refers to dates relevant to the information resource itself, not the information held within the resource. For example, for a document about the civil service in the 18 th century, put '18 th century' in Coverage and put the date published in Date.
Type of term	Element
Refines	-
Refined by	http://www.govtalk.gov.uk/terms/dateAcquired
Refined by	http://purl.org/dc/terms/dateAccepted
Refined by	http://purl.org/dc/terms/Available
Refined by	http://purl.org/dc/terms/dateCopyrighted
Refined by	http://purl.org/dc/terms/created
Refined by	http://purl.org/dc/terms/issued
Refined by	http://purl.org/dc/terms/dateSubmitted
Refined by	http://purl.org/dc/terms/valid
Refined by	http://purl.org/dc/terms/modified
Refined by	http://www.govtalk.gov.uk/terms/cutOffDate
Refined by	http://www.govtalk.gov.uk/terms/dateDeclared
Refined by	http://www.govtalk.gov.uk/terms/dateClosed
Refined by	http://www.govtalk.gov.uk/terms/nextVersionDue
Refined by	http://www.govtalk.gov.uk/terms/updatingFrequency
Encoding Scheme For	-
Has Encoding Scheme	http://purl.org/dc/terms/W3CDTF
Has Encoding Scheme	http://purl.org/dc/terms/Point
Similar To	-
Constraints	The value must always be taken from the specified encoding scheme, with the exception of the 'updatingFrequency' refinement.
Obligation	Mandatory
Condition	A value must be given either for the unqualified date or at least one date refinement
Datatype	-

6.3.3 Commentary

The DCAP for the UK e-Government Metadata Standard is formatted in the most detailed and specific possible style [EGMS]. While this results in a significantly longer document than the DCAPs for RDN and Renardus, such specificity may be helpful to developers of applications that need to create or process metadata based on the DCAP. Note in particular the following:

- Encoding Schemes and Element Refinements are listed using repeated fields in the Term Usage of the Element to which they refer. In addition, each Encoding Scheme and Element Refinement is also described in its own Term Usage, which allows information about each of them, such as Definition and Constraints, to be recorded in the DCAP as well.
- The Term Usage marks information coming from outside sources: the "Source Definition" copies the definition of Date from DCMI documentation, while the "Local Comment" supplies usage information local to this DCAP.

Annex A: Metadata describing a DCAP

A DCAP should itself be described with Dublin Core metadata, either in a header or in a separate metadata record. At a minimum, this description should include:

Title	A name for the Application Profile.
Contributor	A creator or maintainer of the Profile.
Date	The date of last modification.
Identifier	An unambiguous reference to the Profile. Best practice is to provide a URL by which a copy of the document or schema can be retrieved over the Web.
Description	A concise description of the Profile. As appropriate, the description should elaborate on the context and purposes in which the DCAP is intended to be used; the organizations or individuals involved in its development; any arrangements, policies, or intentions regarding the future development and maintenance of the DCAP; or technical characteristics of the instance metadata or database described.

Annex B: Options for machine-interpretable DCAPs

DCAPs can be expressed in machine-interpretable schema languages, and such machine-interpretable schemas can be manipulated by software applications. This CWA does not give detailed recommendations on how such schemas should be structured, as a number of issues are still open for debate. The scope of this CWA is limited to recommending how application profiles can be expressed as text documents. Future options for machine-interpretable DCAPs are outlined below.

Currently, two schema languages specified by W3C might be considered: XML Schema [XML-SCHEMA] and RDF Schema [RDF-SCHEMA]. The choice of schema language will be influenced by the functionality that the schema is intended to support – for example, whether it is required as a predictable format for data exchange or intended to support inferences about existing metadata. Such different objectives imply different choices between the two schema languages. There has been some discussion on ways to combine XML Schema and RDF Schema to more fully express characteristics of application profiles [HUNTER]. More recently there has been an attempt within the W3C to differentiate RDF Schema as a vocabulary description language and XML Schema as a basis for providing structured data exchange.

An XML schema provides a structured expression that supports validation of instance metadata. In effect, an XML schema provides a document "template" which acts as an exchange format for metadata instances. An XML Schema serves the same function as an XML DTD with additional capability for extensibility and namespace handling.

An RDF schema expresses relationships between terms, providing a data model for expressing the semantics of terms – their properties, classes, and definitions. The underlying RDF data model combined with the use of unique identifiers allows software to infer relationships between terms and perform data aggregation.

RDF Schemas are effective for expressing the semantics of application profiles, whilst XML Schemas are more effective for expressing cardinality, data-typing, and constraints. Possible approaches to the expression of application profiles in RDF have been explored within projects such as SCHEMAS [BAKER] and MEG [MEG-REGISTRY].

Bibliography

- [BAKER] Thomas Baker, Makx Dekkers, Rachel Heery, Manjula Patel, Gauri Salokhe, What terms does your metadata use? Application profiles as machine-understandable narratives. Journal of Digital Information 2:2 (November 2001), <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker>.
- [CORES-RESOLUTION] Thomas Baker, Makx Dekkers, Identifying Metadata Elements with URIs: the CORES Resolution. D-Lib Magazine (July 2003), <http://www.dlib.org/dlib/july03/baker/07baker.html>.
- [DC-LIBRARY] Library Application Profile, <http://dublincore.org/documents/2002/09/24/library-application-profile/>.
- [DCMI-NAMESPACE] Andy Powell, Harry Wagner, Stuart Weibel, Tom Baker, Tod Matola, Eric Miller, Namespace policy for the Dublin Core Metadata Initiative, <http://dublincore.org/documents/dcmi-namespaces/>.
- [DCMI-PRINCIPLES] DCMI Grammatical Principles, <http://dublincore.org/usage/documents/principles/>.
- [DCMI-SCHEMAS] DCMI Schemas, <http://dublincore.org/schemas/>.
- [DCMI-TERMS] DCMI Metadata Terms, <http://dublincore.org/documents/dcmi-terms/>.
- [EGMS] Office of the e-Envoy – Cabinet Office, UK e-Government Metadata Standard Application Profile Version 1, <http://purl.oclc.org/NET/eGMSAPv1>.
- [HEERY] Rachel Heery, Manjula Patel, Application profiles: mixing and matching metadata schemas, Ariadne 25, September 2000, <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>.
- [HUNTER] Jane Hunter, Carl Lagoze, Combining RDF and XML Schemas to enhance interoperability between metadata application profiles. WWW10, May 1-5, 2001, Hong Kong, <http://www10.org/cdrom/papers/572/index.html>.
- [MEG-REGISTRY] Rachel Heery, Pete Johnston, Dave Beckett, Damian Steer, The MEG Registry and SCART: Complementary Tools for Creation, Discovery and Re-use of Metadata Schemas. In: Proceedings of the International Conference on Dublin Core and Metadata for e-Communities, 2002. Florence: Firenze University Press, 2002, pp. 125-132, <http://www.bncf.net/dc2002/program/ft/paper14.pdf>.
- [RDF-SCHEMA] Brickley, Dan and Guha, R.V, editors. RDF Vocabulary Description Language 1.0: RDF Schema W3C Working Draft 23 January 2003, <http://www.w3.org/TR/rdf-schema/>.
- [RDN] Powell, Andy, RDN OAI Application Profile, [URL to be assigned].
- [RENARDUS] Metadata Working Group SUB Goettingen, Renardus Application Profile, <http://renardus.sub.uni-goettingen.de/renap/renap.html>.
- [URI] T. Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identifiers (URI): Generic Syntax, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>.
- [WEBARCH] Ian Jacobs, ed., Architecture of the World Wide Web, <http://www.w3.org/TR/webarch/>.
- [XML-SCHEMA] Thompson, Henry S. et al., editors. XML Schema Part 1: Structure. W3C Recommendation 2 May 2001, <http://www.w3.org/TR/xmlschema-1/>.