Information Management Resource Kit

Module on Management of Electronic Documents

UNIT 3. METADATA STANDARDS AND SUBJECT INDEXING

LESSON 3. METADATA STANDARDS FOR THE WEB: PRACTICAL APPLICATIONS

NOTE

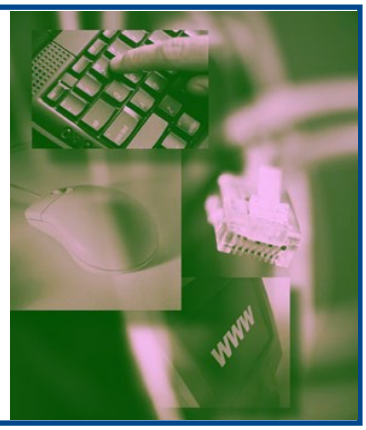
Please note that this PDF version does not have the interactive features offered through the IMARK courseware such as exercises with feedback, pop-ups, animations etc.

We recommend that you take the lesson using the interactive courseware environment, and use the PDF version for printing the lesson and to use as a reference after you have completed the course.

Objectives

At the end of this lesson you will able to:

- understand the purpose of **element qualifiers**;
- differentiate between **namespaces** and **application profiles**; and
- understand when it is necessary to **create new elements**.



Dublin Core qualifiers

The Dublin Core (DC) metadata set provides important information to describe resources such as books, articles and web pages.

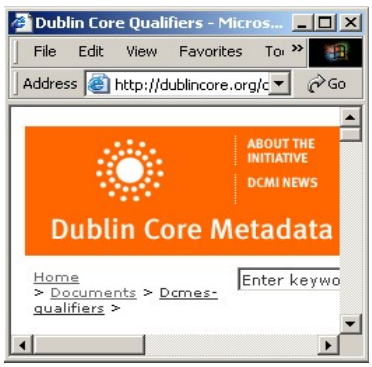
However, since different communities applied the DC differently, working groups were set up in the growing DC community to investigate **how the elements are further qualified in local implementations**.

Some of these groups are DC-Education, DC-Libraries, DC-Government, each exploring needs in their own domain.

The working groups propose domain-specific or generic lists of elements to the DC Metadata Initiative (DCMI) Usage Board, which evaluates these proposals and makes the final decision.

This procedure ensures orderly evolution of Dublin Core Metadata Element Set (DCMES).0

Dublin Core qualifiers



These further qualifications take the form of either:

- element refinement, or
- encoding scheme

Both of these qualifiers further describe the elements, similar to how adjectives are used in our natural languages.

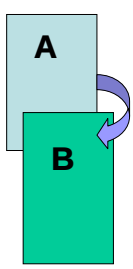
Let's now have a look at them in detail...

View the list of refinements and schemes at **http://dublincore.org/usage/terms/dc/current-elements**

Element Refinements

Let's have a look at an example of an **element refinement**.

Let's say we would like to update the metadata of the old version of an online paper (A) with information about the updated version (B).



Looking at the DC elements, we can use the **relation** element, defined as "A reference to a related resource".

The HTML metadata code for **resource A** would be as follows:

<META NAME="DC.**Relation**" CONTENT="B">

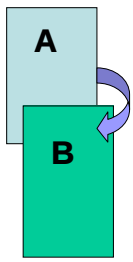
The above statement indicates that **resource A** has a relationship to a **resource B**.

However, this does not give us any information about "**how**" the two resources are related.

Element Refinements

We would like to show to a user that resource A is being **replaced by** resource B.

Let's take a look at the list of qualifiers for **Relation**.



The refined pairs of "**Replaces / isReplacedby**" seem closest in indicating the "**how**" relationship!

The HTML metadata code for **resource A** then would be as follows:

<META NAME="DC.**Relation.isReplacedBy**" CONTENT="B" >

The above statement indicates two things:
1. A is related to B, and
2. A is replaced by B

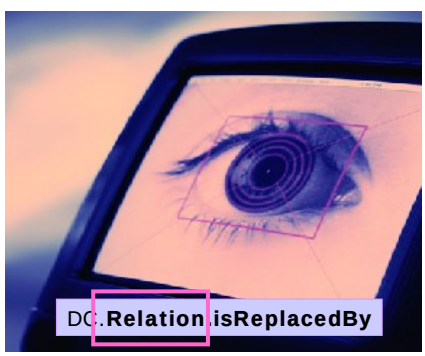
In this case, the qualifier "isReplacedby" **refines** the meaning of the element "Relation" to specify the **type of relation**.

Other possible refinements of DC element "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

Element Refinements

To summarize, **element refinements** are qualifiers that make the meaning of an element either **narrower** or more **specific**.



It is important to remember that a refined element **shares the meaning** of the unqualified element, but with a more restricted scope.

If a client or a system does not understand an element refinement, then it should be able to ignore the qualifier and treat the value as if it were for the refined (broader) element.

Encoding Schemes

Encoding schemes are another type of qualifiers. They **identify schemes** that help to interpret the value of an element (or its refinements). These schemes can either be **controlled vocabularies** or **formal notations**.

For example:

Video games and teenagers

EXAMPLE OF CONTROLLED VOCABULARY
The following metadata statement allows us to interpret the value "Video games and teenagers" as a heading from Library of Congress Subject Headings (LCSH).

<META NAME="DC.Subject" **SCHEME="LCSH"** CONTENT=" Video games and teenagers">

2001-05-26

EXAMPLE OF FORMAL NOTATION
This date has been written using the **YYYY-MM-DD** format, also known as W3CDTF (W3 Consortium Date and Time Formats). Thus, if you follow this format, the metadata statement should be written to indicate the scheme "W3CDTF".

<META NAME="DC.Date" **SCHEME="W3CDTF"** CONTENT="2001-05-26">

Encoding Schemes

To summarize, **encoding schemes** aid in the **interpretation** of an **element value**.

Even if a system does not understand the encoding scheme, the value is still useful for a human reader because they can see, as in the previous example, that the string "Video games and teenagers" is taken from the Library of Congress Subject Headings.

Here is a table showing the schemes that have been approved by the DC for the subject element.

DCMES Element	Element Encoding Scheme(s)
Subject	LCSH [Library of Congress Subject Headings] MeSH [Medical Subject Headings] DDC [Dewey Decimal Classification] LCC [Library of Congress Classification] UDC [Universal Decimal Classification]

A complete list of endorsed encoding schemes for other elements and their definitions are provided at: http://dublincore.org/usage/terms/dc/current-elements/.

Element Refinements

Now, let's see if you can generate qualified metadata!

Language scheme:
- **ISO639-2**

Date refinements:
- **Created**
- **Valid**
- **Available**
- **Issued**
- **Modified**

Imagine you would like to add qualified metadata on your Web Page written in **Spanish** on **15 August 2002**.

You already know that **date** can be presented using W3CDTF. By clicking on and looking at Date refinements, you should be able to choose the correct qualifier for your date. Look also at ISO language scheme to indicate **language**.

Then, try to type in the correct HTML metadata statements for your Web Page.

<META NAME="DC.Language" SCHEME="----------" CONTENT="---">

<META NAME="----------" SCHEME="W3CDTF" CONTENT="---- ---- ----">

Type the text in the relevant boxes.

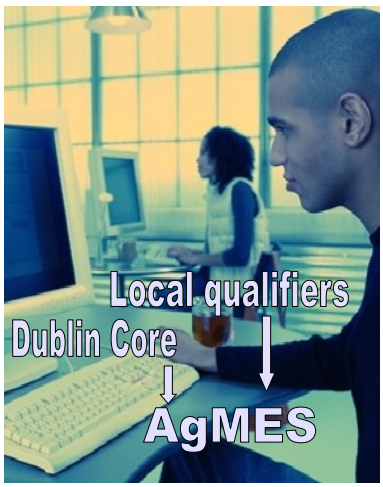
Namespaces

Agriculture Standards (AgStandards) is an initiative which aims to promote common standards within the domain of Agriculture.

The **Agricultural Metadata Element Set (AgMES)** is part of this initiative and aims to encompass issues of semantic standards in the domain of agriculture with respect to description, resource discovery, interoperability and data exchange for different types of information resources in this domain.

AgMES is a proposal that **defines only the new elements and refinements** necessary to sufficiently describe all types of resources in the domain of Agriculture.

Namespaces

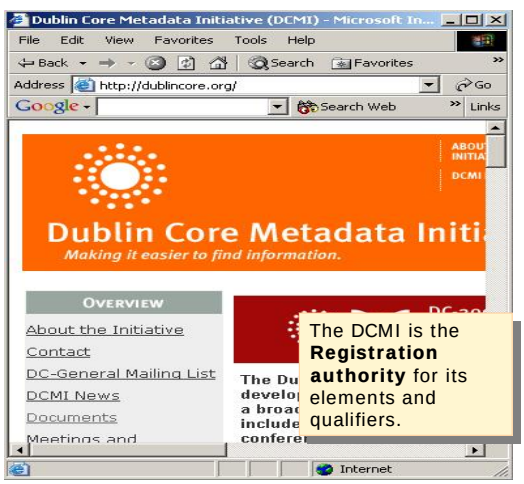


As more and more information becomes available on the web, it becomes important to provide **easy access to that information**. It is, therefore, the aim of AgMES to provide accurate data to search engines and consequently relevant results to users.

AgMES does **not re-create** the elements already provided by other communities such as DC, but instead supplements them with domain specific ones to help improve accessibility and visibility of information in today's more open environment.

These **new elements, refinements** and **encoding schemes** allow us to make the meaning of the DC elements clearer and more **domain specific**.

Namespaces



AgMES is an example of a **namespace**. Dublin Core is another example.

In the metadata community, namespaces are used to identify "**newly defined**" elements and their qualifiers.

A namespace normally has a **registration authority**, that is the entity authorized to register the new elements and qualifiers in a given namespace.

Any organization can create their own namespace as long as they are committed to its maintenance.

Namespaces

For example, let's look at how the existing DC element **Subject** has been extended in AgMES.

In DC the Subject element has schemes. However, often it is necessary to distinguish which particular **Classification** or **Thesaurus** the subject value comes from. To meet this requirement, the Subject element can be refined as either "subjectClassification" or "subjectThesaurus".

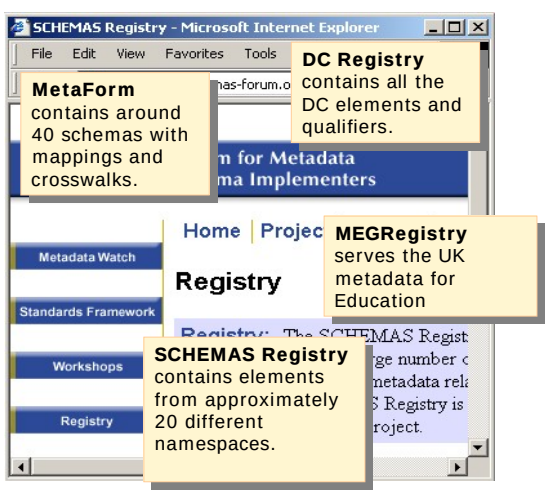
(DC) = defined in the DC namespace

(AGS) = defined in the AgMES namespace

Element	AgMES Element Refinements	AgMES Encoding Schemes	
(DC) Subject	(AGS) subjectClassification	(AGS) ASC (AGS) CABC	Classification schemes
	(AGS) subjectThesaurus	(AGS) AGROVOC (AGS) CABT (AGS) ASFA (AGS) NAL	Thesaurus schemes

Furthermore, agriculture specific classifications and thesauri have been added as encoding schemes: two classifications (ASC and CABC) and four thesauri (AGROVOC, CABT, ASFA and NAL).

Namespaces



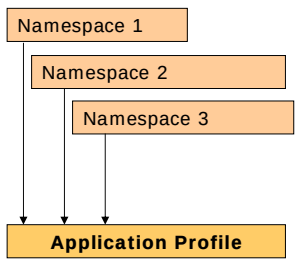
Often, a registration authority can give credibility to the elements or refinements.

There are **several metadata namespace registries** currently under development.

A **metadata registry** contains definition of terms (elements, element refinements and encoding schemes), informs us of newly available terms, controls version changes in terms, serves as a promoter of terms for reuse.

These registries serve the purpose of providing a **one-stop view** of what elements are currently available and what their definitions are.

Application Profiles



If you need metadata elements that will sufficiently describe your resources, you can look through metadata registries that contain already declared elements and **choose elements that meet your needs**.

This way, you save lot of valuable time that you might have otherwise spent in creation of you data model.

This process, of picking elements from different namespaces, results in the creation of an **application profile**.

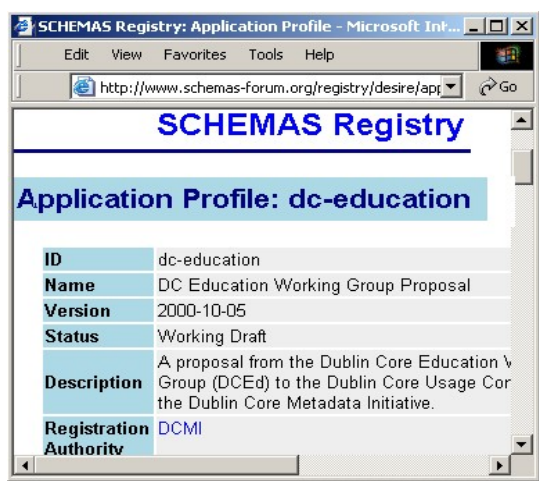
Let's have a look at an example…

Application Profiles

For example, in the DCMI Registry you can find the **DC Education Application Profile (DC-ED AP)**. This has been proposed by the DC-Education Working Group for describing **educational resources**.

It takes elements from other namespaces: Dublin Core, IEEE LOM (Institute of Electrical and Electronics Engineers Learning Object Metadata), as well as its own DC-ED namespace.

Another example is the **AGRIS Application Profile**, created to promote an xml based common metadata format for exchange within the Agricultural Community.



Application Profiles

Application profiles should allow the implementers to declare:

- a limited set of existing elements from different namespaces
- the cardinality for an element
- particular schemes that must be used with a particular element
- a customised definition of an element from existing namespace
- rules for content (usage guidelines)

*Click on each feature to view an example from the **AGRIS Application Profile (AGRIS AP)***

AGRIS AP takes existing elements from the following namespaces:

- **DC Elements**,
- **DC Qualifiers and Schemes**,
- **AgLS** (Australian Government Locator Service Metadata Element Set), and
- **AgMES**.

Application Profiles

the cardinality for an element	Commonly expressed as {**repeatable, not repeatable**}. In AGRIS AP, the element **Creator** is repeatable whereas the AGRIS Record Number, which uniquely identifies each metadata record, is not.
particular schemes that must be used with a particular element	In AGRIS AP, values for **subject** element should come from the **AGROVOC** Thesaurus.
a customised definition of an element from existing namespace	Although an application profile is allowed to slightly modify the meaning of an element or its refinement, AGRIS AP does not make use of this possibility.
rules for content (usage guidelines)	Each element/refinement can have content guidelines. One form of correcting the content is by providing scheme information; the other, is by providing specific guidelines on their format. For example, the name of the Author (if it is a person), should be in the form of: “surname, forename initial(s), prefixes, particles, role, affiliation”

Namespaces vs. Application Profiles

Let's try and see if you have spotted the important difference between namespaces and application profiles.

	1		
Namespace			
Application Profile			

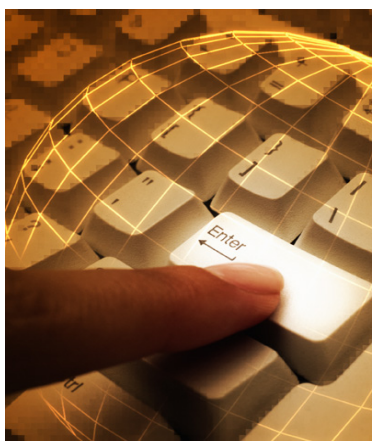
2

a	b	c
Allows for declaration of used elements	Generic and therefore all-purpose	One or more registration authorities of elements
Allows for definition of new elements	Catered to specific applications	One registration authority for all elements
d	e	f

Click each option, drag it and drop it in the corresponding box, in the same column.

When you have finished click on the **Confirm** button.

When should you create a new term?



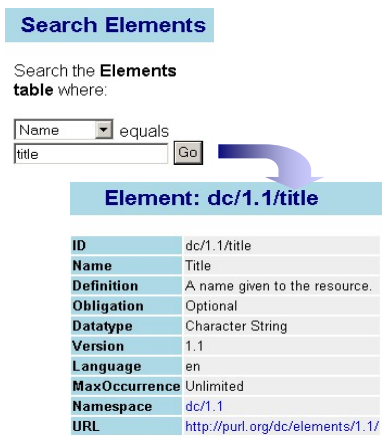
The goal of DC and other such metadata standards is to promote **interoperability** through **reuse** of a common metadata element set. This facilitates easy exchange and sharing of information in the current networked environment.

To be able to understand each other we need to speak the same metadata tags, at least some basic common ones.

Therefore: **when possible, reuse a well-accepted metadata standard**.

As more and more communities start adopting a single standard, they become more and more interoperable.

When should you create a new element?



To reuse elements, you need to be aware of them. This is where metadata registries come into play.

Case 1: You need the **TITLE** element to give "title of a resource."

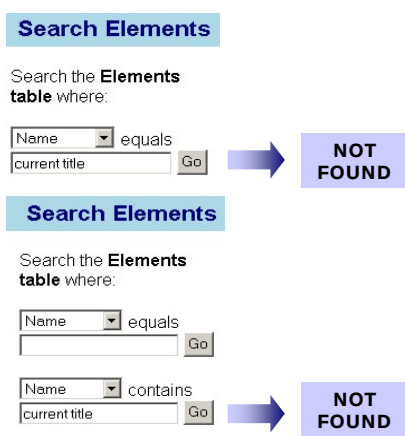
You are aware that there are several registries that might save you some valuable time. You decide to use the SCHEMAS metadata registry and see what it offers.

After searching for the word "Title" in the registry, you get one result showing an element "Title".

Since the definition of this term meets yours, you decide to use this in your application.

Remember, using this "Title" defined by DC, will ensure that every system capable of understanding DC will understand your tags.

When should you create a new refinement?

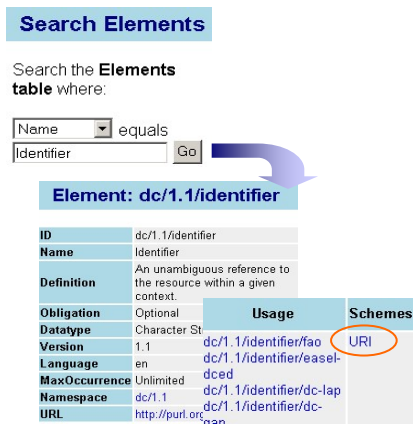


Case 2: Let's imagine you also require a refinement to the Title element. You would like to distinguish the **current title** from **previous title**.
You search the registry for an **exact match** of "current title" and receive no results. You also give a second try to see if there are any elements that may **contain** your title, but get no results.

You already know that you should reuse, and know that DC has already defined **title element,** you decide that you will modify this title with the refinement CURRENT.

As you know that new elements can be defined in a namespace, you create your own namespace and define the refinement **CURRENT** in it. You can now register this namespace in a registry, like SCHEMAS, so that others can make use of your terms.

When should you create a new scheme?



Case 3: You need the IDENTIFIER element with URN (Universal Record Number) as a scheme.

Many elements and refinements have schemes. Before creating one yourself, look for what is already there. If your needs are not met by the existing encoding schemes, only then should you **declare a new encoding scheme**.
Remember: You can declare qualifiers, **both refinements and encoding schemes**, for any existing element.

You find IDENTIFIER on SCHEMAS Registry, but the only scheme available is a URI.

Since this does not meet your needs, you decide to declare URN and add it to the already created namespace (that you created previously).

Benefits of using common metadata

Using common data allows us to:

• **give lexical words a meaning** (e.g. differentiate between "Title" of a book from the "Title" of a person, like "Sir" - Book Title vs. Personal Title -),

• facilitate **easy exchange** between systems since they use the same element set,

• facilitate **resource discovery** and request access for it,

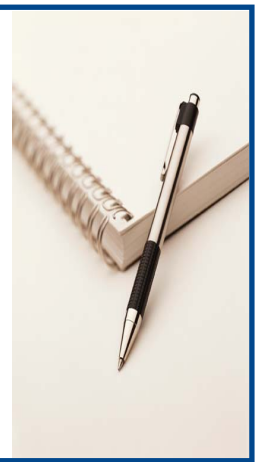
• combine **content for reuse**,

• **reduce cost** by using standardized tools (generic resources such DC and AgMES, automatic metadata creation tools such as DC.Dot),

• facilitate **automatic processing and manipulation** of information, e.g., allowing you to send an email using all <email> fields.

Summary

- **Element refinements** are qualifiers that make the meaning of an element either **narrower** or more **specific**.
- **Encoding schemes** are qualifiers that **identify schemes** that aid in the interpretation of the value of the element and/or its refinements.
- In the metadata community, namespaces are used to identify "**newly defined**" elements and their qualifiers.
- An **application profile** is created by taking existing elements that may come from one or more namespaces registered by one or more authorities.
- As more and more communities start adopting a single standard, they become more and more interoperable; therefore, **when possible, reuse a well-accepted metadata standard**.



Exercises

The following four exercises will help you test your understanding of the concepts that were covered in the lesson and will provide you with feedback.

Good luck!

Exercise 1

Which of the following examples uses an element refinement?

- ○ <META NAME="DC.Subject" SCHEME="AGROVOC" CONTENT="oryza">
- ○ <META NAME="DC.Subject" CONTENT="production increase">
- ○ <META NAME="DC.Title.Alternative" CONTENT=" Brucellosis control in cyprus">

Click on your answer

Exercise 2

What is a benefit of using an encoding scheme?

- ○ It aids in the interpretation of the value of the element and refined element.
- ○ It makes the meaning of an element either narrower or more specific.

Click on your answer

Exercise 3

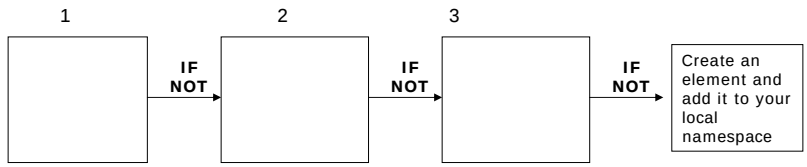
Indicate which of the following are properties of an application profile.

- ☐ It allows for definition of new elements.
- ☐ It allows for declaration of used elements.
- ☐ It specifies the allowed schemes for a particular element.
- ☐ It is generic and therefore all-purpose.

Click on your answers

Exercise 4

You want to describe your resources, PowerPoint presentations. You want to create a metadata set that describes this resource. Which sequence of questions should you ask yourself before creating a **new local element**?



a	b	c
Can I use elements and qualifiers from **more than one** namespace?	Can I **qualify existing** elements with refinements or schemes?	Can I use all the elements and qualifiers from a single **schema** without changes?

Click each option, drag it and drop it in the corresponding box.

When you have finished, click on the confirm button.

If you want to know more...

DC Qualifiers
http://dublincore.org/usage/terms/dc/current-elements/

Namespaces in XML
http://www.w3.org/TR/REC-xml-names/

Application profiles: mixing and matching metadata schemas
http://www.ariadne.ac.uk/issue25/app-profiles/

Difference between namespaces and application profiles
http://www.fao.org/agris/agmes/Documents/nsvsap.doc

Machine Understandable Application Profiles
http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker/

AgMES
http://www.fao.org/agris/agmes/

SCHEMAS Registry
http://www.schemas-forum.org/registry/desire/index.php3

DESIRE Registry
http://desire.ukoln.ac.uk/registry/index.php3

DC Dot Tool (metadata created in HTML, XML, RDF, XHTML)
http://www.ukoln.ac.uk/cgi-bin/dcdot.pl