```
Glossary and User Guide Task Groups - face-to-face meeting

Date:         2010-10-21, Thursday, 14:00-15:30
In program:   http://www.asis.org/Conferences/DC2010/program-sessions.html#userdoc
Expected:     Tom, Mary, Stefanie, Marcia, Pete (remote)

1. 14:00-14:30 - User Guide (Stefanie)

   Discuss (see meeting PDF)
   -- http://colab.mpdl.mpg.de/mediawiki/UsingDC
   -- http://colab.mpdl.mpg.de/mediawiki/CreatingMetadata
   -- http://colab.mpdl.mpg.de/mediawiki/PublishingMetadata
   -- Pete comments on User Guide

   Note: guidance on specific properties in the User Guide may be
   discussed in the Libraries Task Group meeting on Wednesday, see
   http://www.asis.org/Conferences/DC2010/program-sessions.html#librarytaskgroup

2. 14:30-15:00 - Glossary and FAQ (Tom and Mary)

   Discuss glossary entries (see meeting PDF)
      DCMI Metadata Terms
      Dublin Core
      Dumb-Down
      Namespace Policy
      One to One Principle
      Open World Mindset
      RDF
      Resource Discovery
      Simple Dublin Core

   Discuss FAQ entries (see meeting PDF)
      On reusing XML elements

3. 15:00-15:30 - Issues and Next Steps

   General principles
   -- One list, with both current and obsolete technology.
   -- Legacy or archaic terminology can be marked as such.

   Issues
   -- Cross-references and redundancies among User Guide, Glossary, FAQ
   -- Form of publication (wiki document?)
   -- Next steps, work plan
```

UsingDC

From MPDLMediaWiki

Contents

Purpose and Scope of this Guide

"The Dublin Core" (aka the Dublin Core Metadata Element Set), created in 1995, is a set of fifteen generic elements for describing resources. These are: Creator, Contributor, Publisher, Title, Date, Language, Format, Subject, Description, Identifier, Relation, Source, Type, Coverage, and Rights. Today the Dublin Core is a formal standard [1][2][3], used in countless implementations, and one of the top metadata vocabularies on the Web.

"Dublin Core metadata" is about more than fifteen elements. It is best described as a *style of metadata* that has evolved from efforts to put the fifteen elements into the context of a coherent approach to metadata on the World Wide Web generally. Since the late 1990s, the Dublin Core style has evolved in the context of a Dublin Core Metadata Initiative (DCMI) -- now incorporated as a non-profit organization hosted at the National Library Board of Singapore -- in tandem with a generic approach to metadata developed in the World Wide Web Consortium under the banner "Semantic Web". The Semantic Web approach achieved a breakthrough in 2006 with the Linked Data movement, which uses DCMI Metadata Terms as one of its key vocabularies. "DCMI Metadata Terms" is a larger set that includes the fifteen elements along with several dozen related properties and classes. "Dublin Core application profile" is the key expression of the Dublin Core style. An application profile uses DCMI metadata terms together with terms from other, more specialized vocabularies to describe a specific type of information -- and it does this in the framework of the Semantic Web.

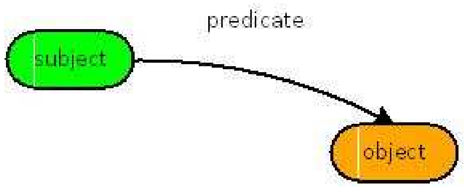
These guidelines provide an entry point for users of Dublin Core -- i.e., for users of DCMI Metadata Terms in the "Dublin Core style". For catalogers, it will show how to create metadata "descriptions" for information resources such as documents, images, data, etc. Implementers will it support publishing Dublin Core Metadata as Linked Data. The guidelines will neither show how to create or Dublin Core Application Profiles -- for this see the Guidelines for Dublin Core Application Profils (http://dublincore.org/documents/profile-guidelines/) -- nor how to express Dublin Core Terms in different syntaxes -- for this see the section "Syntax Guidelines" of DCMI Specifications (http://dublincore.org/specifications/) .

What is Metadata?

Metadata has been with us since people made lists on clay tablets and scrolls. The term "meta" comes from the Greek for "alongside" or "with". Over time, "meta" was also used to denote something transcendental, or beyond nature. "Meta-data", then, is "data about data", such as the contents of catalogs, inventories, etc. Since the 1990s, "metadata" most commonly denotes machine-readable descriptions of things, most commonly in the context of the Web. The structured descriptions of metadata help find relevant resources in the undifferentiated masses of data available online. Anything of interest can be described with metadata, from book collections to football leagues and stuff you want to sell. Describing different types of resources requires different types of metadata and metadata standards.

What is Linked Data?

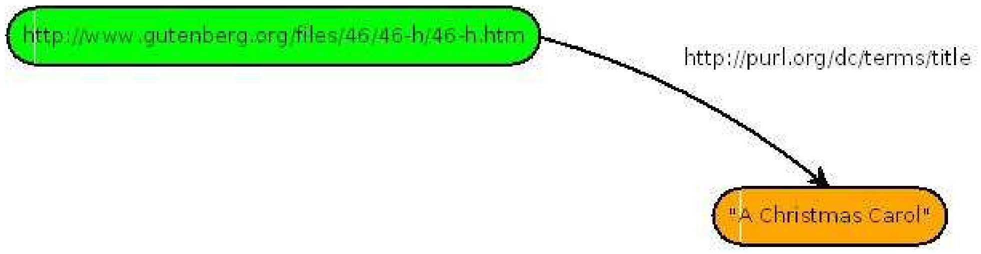
Linked Data is a method of exposing, sharing, and connecting data on the Semantic Web using URIs and RDF (see http://linkeddata.org/). Metadata are the backbone of this method, making statements about data and how they relate to each other. In Linked Data these statements have to be expressed in RDF triples, which break statements in three parts:



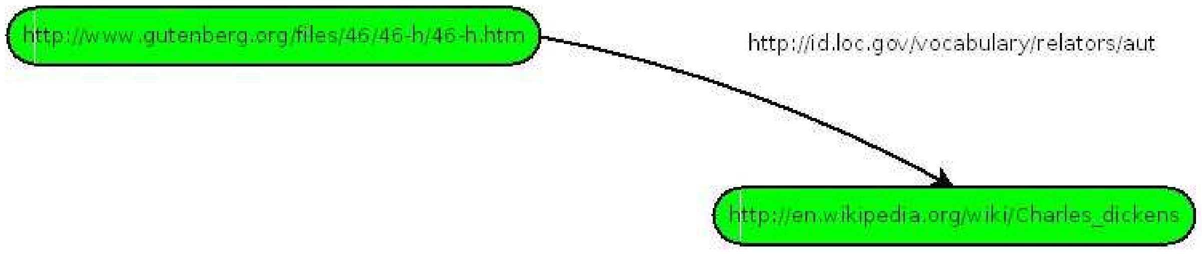
- the subject - the part that identifies the thing the statement describes,
- the predicate - the part that identifies a property of the described thing.
- the object - the part that identifies the value this property has when describing this thing.

(http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#statements)

Another "must" when publishing metadata in Linked Data is the usage of URIs. In Linked Data you need URIs referencing to things by identifying, localizing and interlinking them. Considering this a triple graph describing Charles Dickens "A Christmas Carol" might look this way:


http://www.gutenberg.org/files/46/46-h/46-h.htm
http://purl.org/dc/terms/title
"A Christmas Carol"

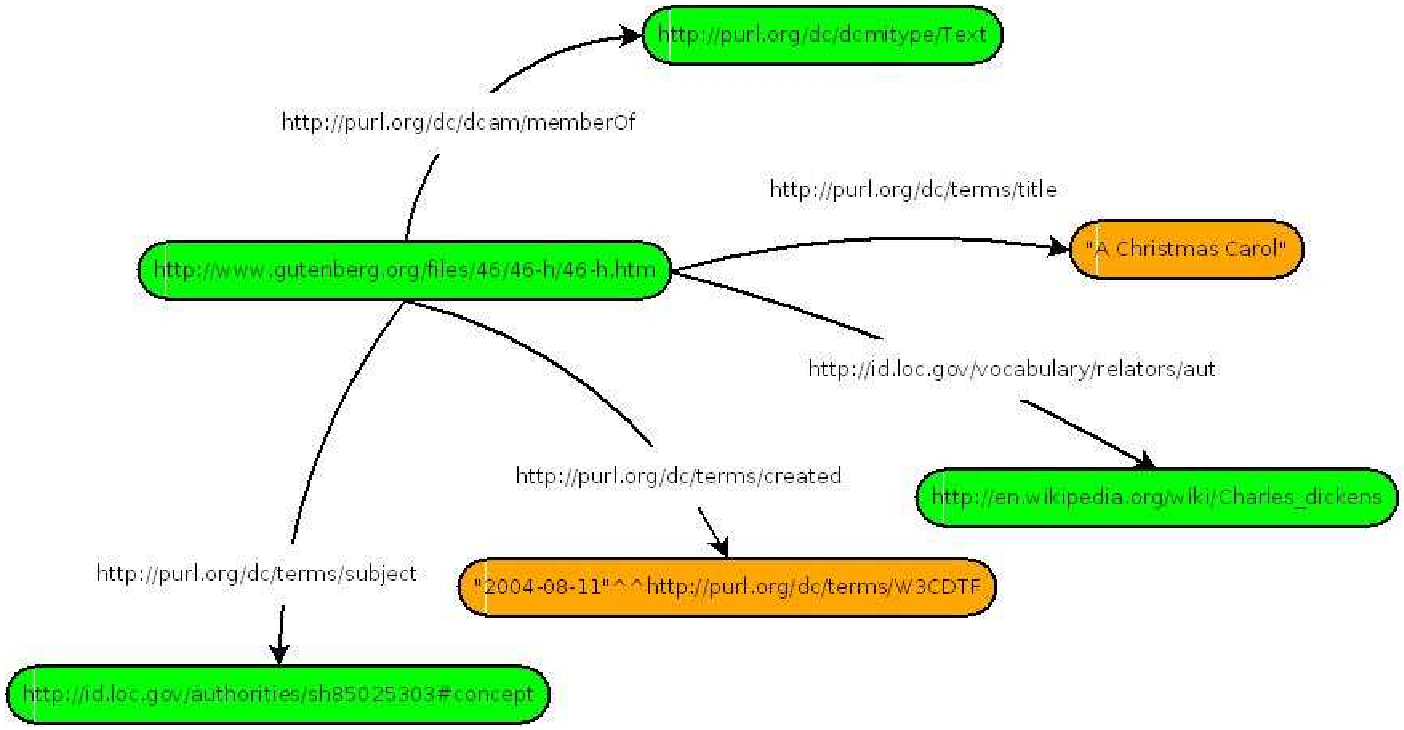
Here the value "A Christmas Carol" is a simple string or literal value. Another sort of values used in a triple are non-literal values, which means you use a URI that references to another description - the description of a thing that is the object of your statement.


http://www.gutenberg.org/files/46/46-h/46-h.htm
http://id.loc.gov/vocabulary/relators/aut
http://en.wikipedia.org/wiki/Charles_dickens

Based on the above said a metadata description in Linked Data consists of:

- **a reference to a described resource**, i.e. to the thing that the metadata is *about*, the *subject* of the metadata. This reference take the form of a Uniform Resource Identifier (URI) or of an unnamed placeholder that is inferred by context (e.g., that metadata embedded in a document is about the enclosing document).
- **references to properties**: typed relationships between the described resource and various bits of descriptive information, or between the described resource and another resource. Examples of properties, which are also known as *predicates*, include the fifteen elements of the Dublin Core.
- **values**: bits of descriptive information (string *literals*) or references to other entities (resources), such as people or concepts, which are related to the described resource via the properties.
- **references to classes**, i.e. to types, or categories, of things, such as the category *books* or the category *people*.
- **references to syntax encoding schemes (RDF datatypes)**, i.e. specifications of how value strings map refer to things in the world, such as *2010-09-24*, which uses the YYYY-MM-DD pattern specified in ISO 8601 to represent the date 22 September 2010.
- **references to vocabulary encoding schemes (VES)**, enumerated sets of resources of which the things referenced as values are members, the way a subject heading belongs to the VES *Library of Congress Subject Headings*.
- **language tags** indicating the language of words used in literal string values.

In a RDF graph this might look this way:


http://purl.org/dc/dcmitype/Text
http://purl.org/dc/dcam/memberOf
http://purl.org/dc/terms/title
"A Christmas Carol"
http://www.gutenberg.org/files/46/46-h/46-h.htm
http://id.loc.gov/vocabulary/relators/aut
http://purl.org/dc/terms/created
http://en.wikipedia.org/wiki/Charles_dickens
http://purl.org/dc/terms/subject
"2004-08-11"^^http://purl.org/dc/terms/W3CDTF
http://id.loc.gov/authorities/sh85025303#concept

In this example the **described resource** is a web page referenced by the URI http://www.gutenberg.org/files/46/46-h/46-h.htm. We may reference to a **class** - in our example its the class "text" of the Dublin Core Type Vocabulary - using the property memberOf. Further **properties** of the resource are "title", "author", "created" and "subject". We used literal - "A Christmas Carol" and "2004-08-11" - and non-literal **values** -<http://en.wikipedia.org/wiki/Charles_dickens> and <http://id.loc.gov/authorities/sh85025303#concept>. Literal values may be constrained by **datatypes** (in our example values describing dates have to be conform with W3CDTF), non-literal values by **vocabulary encoding schemes** (in our example values used describing the topic of a resource have to be concepts of the Library of Congress Subject Headings). A **language tag** can be used to describe the language of a literal value (in our example the value of the title is English).

Levels of Interoperability

Metadata is most helpful when used to navigate the *information jungle* of the Web -- to find, identify, use resources -- or to share and exchange structured information. However there is a tension between requirements of applications and of people:

- people prefer customized descriptions which reflect their understanding of the world
- applications need interoperability between descriptions in order to process them efficiently.

Metadata *vocabularies* help bridge this gap. *Vocabularies* define properties and classes that can be used to describe resources in a coherent way within or between communities. A vocabulary provides the shared basis for exchanging descriptions within groups of people. They support the interoperability of different metadata applications, and they support the ability of applications to change data with systems with no or minimal loss of information. The Dublin Core distinguishs four levels of interoperability (http://dublincore.org/documents /interoperability-levels/) :

- **Level 1**: Shared term definitions: At this level, a community uses the same classes and properties, for example DCMI Metadata Terms (http://dublincore.org/documents/dcmi-terms/) .
- **Level 2**: Formal semantic interoperability: At this level, the description of resources is based on, or automatically mappable to, the shared formal model for metadata provided by W3C's Resource Description Framework (RDF), the basis of Linked Data.
- **Level 3**: Description Set syntactic interoperability. At this level, resource descriptions are based on a shared notion of RDF-based metadata *records* (based, specifically, on the DCMI Abstract Model (http://dublincore.org/documents/abstract-model)).
- **Level 4**: Description Set Profile interoperability. At this level, a community uses not only the same classes and properties, based on the same underlying RDF model, but also uses, or *constrains* those classes and properties based on an agreed model of the things being described and on shared usage patterns. The Description Set Profile is the key component of what the Dublin Core community calls an *application profile*.

As of Fall 2010, the DCMI approach to defining *records* and *constraints* -- Interoperability Levels 3 and 4 -- is under review in light of the rapid evolution of alternative approaches to documenting metadata usage patterns in application-profile-like constructs for ensuring the consistency and quality of Linked Data.

At the other extreme, Level 1 interoperability is so open-ended that it quickly leads to a proliferation of custom-built solutions incompatible with each other, such as metadata expressed in document formats that require customized software to read and data models that cannot easily be mapped to generic, interoperable representations such as those expressed in RDF.

This User Guide therefore focuses on Level 2 interoperability -- the sweet spot between ad-hoc implementations and shared models of records and constraints, which remain the object of experimentation and research. Given the state of play in 2010, implementers can design their metadata for compatibility Linked Data target with the confidence that this will ensure its formal compatibility with an explosively growing Cloud of Linked Data (http://lod-cloud.net/) .

Users can explore the Dublin Core approach to shared record constraints (Levels 3 and 4) in the DCMI documents "Singapore Framework for Dublin Core Application Profiles" (http://dublincore.org/documents/singapore-framework/) , the draft "Description Set Profiles: A constraint language for Dublin Core Application Profiles" (http://dublincore.org/documents/dc-dsp/) , and the user-oriented "Guidelines for Dublin Core Application Profiles" (http://dublincore.org/documents/profile-guidelines/) . A well-developed example of an application profile developed according to these principles may be found in the "Scholarly Works Application Profile" (http://dublincore.org/scholarwiki /SWAPDSP) .

What is Dublin Core?

In the mid 1990s Dublin Core started with the idea of "core metadata" for simple and generic resource descriptions. An international, cross-disciplinary group of professionals from librarianship, computer science, text encoding and museum community, and other related fields of scholarship and practice developed such a core standard – the fifteen Dublin Core elements. But this was just the first steps – since then the World Wide Web has changed in some ways and has broken new ground on the way to a semantic web. Dublin Core followed this path developing further standards for metadata based on the World Wide Web Consortium's work on a generic data model for metadata, the Resource Description Framework (RDF). So **Dublin Core metadata standards** today are:

- The DCMI Abstract Model (http://dublincore.org/documents/abstract-model/) (DCAM) – an RDF based syntax independent model supporting mappings and cross-syntax translations.
- The Sinagpore Framework for Dublin Core Application Profiles (http://dublincore.org/documents/singapore-framework/) – a framework which defines the components that are necessary for documenting a Dublin Core compatible Application Profile.
- **The Dublin Core Metadata Vocabularies**
  - DCMI Metadata Terms (http://dublincore.org/documents/dcmi-terms/) in the /terms/ Namespace
  - DCMI Type Vocabulary (http://dublincore.org/documents/dcmi-type-vocabulary/)
  - Metadata terms related to the DCMI Abstract Model (http://dublincore.org/documents/abstract-model/)
  - Dublin Core Metadata Element Set, Version 1.1 (http://dublincore.org/documents/dces/)

Dublin Core and Linked Data

Since the emergence of the Semantic Web and Linked Data approaches, implementers face a wider range of choices in designing applications. Traditional approaches based on metadata records and descriptive tags embedded in Web pages remain effective alternatives within closed, controlled implementation environments, while Linked Data approaches are designed to provide metadata in a generic form that is easily reusable by other applications for "mashing up" your data with related data published by others. Linked Data has given new meaning to old ideas, such as embedded metadata, which are being reinvented with new Web technologies and tools to solve practical problems of resource discovery and navigation. "The Dublin Core" (and DCMI Metadata Terms) provides a solid basis for bridging these traditional and modern approaches, lookin at DC terms as a "small language for making a particular class of statements about resources". In this language terms are arranged into a simple pattern of statements and DCMI metadata properties are used as predicates in subject-predicate-object triples.

```
ex:myPicture dc:title "Milking the goats" ;
             dc:creator <http://d-nb.info/gnd/131724126> .
```

In this example there are two statements about the picture:

- it has a title, which is „Milking the goats
- it has a creator, represented by the URI http://d-nb.info/gnd/131724126

The object or value we use in the title statement is a simple string called a literal value. The value used in the creator statement is a non-literal value, a URI that leads to another metadata description, where the person who is the creator is described in a more detailed way.

We call the value used here a non-literal value. We want to say more about the creator and we need another description to do so. There are two statements about this person in our example:

- it's a link to the homepage of her workplace
- and her name

```
<http://d-nb.info/gnd/131724126> foaf:name "Rühle, Stefanie" ;
                                 foaf:workplaceHomepage <http://www.sub.uni-goettingen.de/> .
```

Though the statements we make here are not using Dublin Core properties they are non the less DC compatible, because in the namespace these terms are explicitly declared to be properties - one of the four metadata terms DCAM knows - and have been assigned a proper URI. This means we may use other properties than Dublin Core Terms as long as these terms are Dublin Core compatible (see above). This way Dublin Core provides the grammar for a "metadata pidgin for digital tourists", a grammar that allows to merge terms from different vocabularies of different communities in one language and may be used to display both simple and complex issues. With this grammar Dublin Core provides a mechanism for extending the Dublin Core term set for additional resource discovery needs expecting that other communities will create and administer additional metadata sets, specialized for the need of their community.

Dublin Core namespaces / URIs

! work in progress!

Dublin Core Properties

These guidelines list all properties defined in the following two namespaces of DCMI Metadata Terms (http://dublincore.org/documents/dcmi-terms/) :

- http://purl.org/dc/elements/1.1/ (referred to here using the short prefix "dc:")
- http://purl.org/dc/terms/ ("dcterms:").

and illustrate their usage by examples. Examples are offered for two points of view: for the "cataloger" creating metadata descriptions, typically with help from a software interface, and for the "technician" responsible for publishing the data created as linked data.

- CreatingMetadata: describes how to create content for DCMI Metadata listing each property by name, abbreviated URI, and definition, and groups together related properties -- i.e. properties that can be described with similar usage guidelines and illustrated with similar examples.

- PublishingMetadata: describes how to use DCMI Metadata as linked data listing the properties by namespaces.
  - The terms namespace
    - used with literal values
    - used with non-literal values
  - The legacy namespace

Retrieved from "http://colab.mpdl.mpg.de/mediawiki/UsingDC"

Category: KIM

CreatingMetadata

From MPDLMediaWiki

Go to UsingDC | PublishingMetadata

How to create content for DCMI Metadata

Contents

About the examples

We are presenting the examples in tables. To interpret these tables please consider that:

- yellow rows are standing for statements describing a resource,
- white rows are standing for statements describing a resource that is the value of another statement but is not described by another record,
- bold strings are standing for properties,
- italicized strings are standing for values,
- red strings are standing for values linking to records.

Titles

Title

Title is a property that refers to the name or names by which a resource is formally known.

Alternative

Alternative is a property that refers to a name or names of a resource used as a substitute or alternative to the formal title. These are secondary titles, abbreviations, translations of a title, etc.

Guidelines for the creation of title content

For the title use the name given to the resource.

title	*Alvar Aalto Chair No. 66*

as linked data

In most databases title is one of the main criteria to identify search results. So **if there is no formal title resp. name** you should formulate an adequate one by yourself.

title	*Data from a survey about the usage of metadata*

as linked data

If there is **more than one title resp. name** you should repeat the title property

title	*Autumn Leaves*
title	*The Dead Leaves*

as linked data

or use the alternative property.

Typically alternative is used for **secondary titles**,

title	*Passion for Pulses*
alternative	*A Feast of Beans, Peas and Lentils from Around the World*

as linked data

or for **abbreviations**.

title	*American Meteorological Association Newsletter*
alternative	*AMA Newsletter*

as linked data

If the title resp. name is expressed in **different languages** you should use language tags.

title	*La Joconde*	fre
title	*Mona Lisa*	eng
title	*La Gioconda*	ita

as linked data

the same is true for alternative

title	*EU Stability Programm of Belgium*	eng
alternative	*Council Opinion on the Updated Stability Programm of Belgium 2009 - 2010*	eng
alternative	*Stellungnahme des Rates zum aktualisierten Stabilitätsprogramm Belgiens für 2009 - 2012*	ger

as linked data

It is recommended to describe titles with plain text (as done in the examples above). But sometimes it is necessary to **create a relationship between the described resource and a more detailed title description**.

This should be used to refer to **a title with different transliterations**,

title		
	in greek	*Οιδίπους Τύραννος*
	in latin	*Oidipous Tyrannos*

as linked data

or to refer to a **title authority**.

title	*Nibelungenlied Handschrift B*
title	*nibelungenlied*

title	*Nibelungenlied Handschrift C*
title	*nibelungenlied*

identifier	*nibelungenlied*
label	*Der Nibelunge Not*

alternative label	*Nibelungenlied*
alternative label	*Nibelungenklage*
alternative label	*Nibelungensage*
description	*The Nibelungenlied exists of 39 aventiuren created between 1180 and 1210*

as linked data

Relationships between Resource and Agents

Using agent properties in Dublin Core

Persons, organizations and services can relate to resources in various ways. DCMI defined only a few common properties to describe the relationship between resources and agents. So when necessary other vocabularies should be used describing these relationships more detailed. In these cases we recommend to use the marcrelator codes (http://id.loc.gov/vocabulary/relators.html) . How to use them is described in

- MARC Relator terms and Dublin Core (http://dublincore.org/usage/documents/relators/)
- MARC Relator properties in Dublin Core metadata (http://www.ukoln.ac.uk/metadata/dcmi/marcrel-ex/)

Contributor

The contributor property represents a relationship between the resource and a person, an organization, or a service making a contribution to a resource.

Creator

The creator property represents a relationship between the resource and a person, an organization, or a service primarily responsible for making the content of a resource.

Publisher

The publisher property represents a relationship between the resource and a person, an organization, or a service responsible for making the resource available and provide access to the resource.

RightsHolder

This property represents a relationship between the resource and a person or an organization owning or managing rights over this resource.

Guidelines for the creation of content for agents

If you know there is a **URI standing for a person or organization** you should use it. For further information you should handle the person or organization like another resource.

Example with an **organization**:

rights holder	gnd	*39454-3*
	name	*Bundesarchiv Koblenz*
	homepage	*http://www.bundesarchiv.de/index.html.de*

as linked data

Example with a **person**:

contributor	gnd	*135066719*

	family name	*Elliott*
	given name	*Missy*
	nick name	*Missy E*

as linked data

Regardless of the existence of a URI **personal names** should be grouped in family name resp. surname as one part of the name and forename resp. given name as the other part .

You should handle the person name like **another resource**,

creator		
	family name	*Shakespeare*
	given name	*William*

as linked data

or you could devide these names **by comma**.

creator	*Shakespeare, William*

as linked data

When you **in doubt about family name and given name** give the name as it appears.

contributor	*Snoop Dogg*

as linked data

If there is **more than one contributor/creator/publisher/rightsHolder**, each should be listed separately,

like another resource,

publisher	gnd	*2125990-2*
	name	*Rossijskaja Gosudarstvennaja Biblioteka*
	homepage	*http://www.rsl.ru/*
publisher		
	name	*Knižnaja Palata*

as linked data

or with plain text.

creator	*Hubble Telescope*
publisher	*University of Nowhere*
publisher	*All Your Data Inc.*

as linked data

Type

The type property refers to a description of the nature or genre of the content of a resource (e.g. a stylistic category, a function or an aggregation level). To describe the physical or digital manifestation, use format.

Guidelines for the creation of type content

We recommend to **select a value from a controlled vocabulary** (e. g. DCMI Type Vocabulary (http://dublincore.org/documents/dcmi-type-vocabulary/)).

type	dctype	
		Still Image

as linked data

But you may also use **plain text**.

type	*Conference*

as linked data

If **no formal controlled vocabulary** exists, you could create a domain specific one.

type	*conference*

identifier	*conference*	
label	*Conference*	eng
label	*Tagung*	ger
label	*съезд*	rus

as linked data

Is the resource composed of **multiple components of different types** the property should be repeated.

type	dctype	
		Interactive Resource
type	dctype	
		Text

as linked data

Different communities use a variety of type vocabularies. To ensure interoperability you can use terms from different vocabularies side by side - e.g. a type of the DCMI Type vocabulary in addition to **a non-controlled or domain specific type term**.

type		
		PC Game
type	dctype	
		Software

as linked data

Format

The property format refers to the file format, the physical medium (e.g. the data storage medium), or the dimension (the size or duration) of a resource. The information can be relevant to determine the equipment needed to display or operate a resource (e.g. if the described resource has format pdf you need a pdf reader to use it). To specify the different categories of format you should use extent and/or. To reference to the nature or genre of the content use type.

Guidelines for the creation of format content

For the description of the **file format** we recommend to use a controlled vocabulary - e.g. the list of Internet Media Types (http://purl.org/NET/mediatypes/) (MIME).

format	mime	*jpeg*

as linked data

You should repeat the properties when **more than one type of value exists**.

format	mime	*jpeg*
format		*40 x 512 pixels*

as linked data

Extent

This property refers to the size (e.g. bytes, pages, inches, etc.) or duration (e.g. hours, minutes, days, etc.) of a resource.

Guidelines for the creation of extent content

Typically the value used for the description of the extent consists of a **numeric value and a caption** to specify it. You may use a text string to present it

extent	
	21 minutes

as linked data

or use controlled values for the caption.

extent		
	minutes	*21*

as linked data

Medium

This property refers to the physical carrier of the resource and may only be used if the resource is of physical nature (e.g. a painting, a sculpture, etc.)

Guidelines for the creation of medium content

Note that the **media types must not be used with the property medium** because medium describes only physical objects.

We recommend to **use a controlled vocabulary**. If no formal controlled vocabulary exist you should nonetheless handle the media type like another resource.

medium	
	oil on wood
	oil
	wood

as linked data

Language

This properties refers to the language of the intellectual content of the resource.

Guidelines for the creation of language content

For the identification of languages please follow RFC 4646 (http://www.ietf.org/rfc/rfc4646.txt) . Best practice would be to **select a value from the three letter language tags of ISO 639** (e.g. http://www.sil.org/iso639-3/codes.asp).

title	*A great deliverance*	
language	ISO 639-3	*eng*

as linked data

or

title	*A great deliverance*	
language		
	RFC 4646	*eng*

as linked data

If the content is in **more than one language**, the property should be repeated.

title	*Charlie Wilson's War*	
language	ISO 639-3	*eng*
language	ISO 639-3	*hun*
language	ISO 639-3	*tur*

as linked data

But if **every language version has it's own identifier**, they have to be treated like single resources.

Video1

title	*Medieval helpdesk with English subtitles*	
identifier	*http://www.youtube.com/watch?v=pQHX-SjgQvQ&feature=player_embedded*	
isVersionOf	*Video2*	
language	ISO 639-3	*nor*
language	ISO 639-3	*eng*

Video2

title	*Book help (better verson)*	
identifier	*http://www.youtube.com/watch?v=UOorZQLsmuA&feature=related*	
isVersionOf	*Video1*	
language	ISO 639-3	*nor*

as linked data

You could also use plain text

title	*The Power of Orange Knickers*
language	*English*

as linked data

or create your own language vocabulary.

title	*The Power of Orange Knickers*
language	*english*

identifier	*english*
label	*English*
ISO 639-1	*en*
ISO 639-3	*eng*

as linked data

Identifiers

Identifier

An identifier is an unambiguous reference to a resource. Examples of formal identification systems include the Uniform Resource Identifier (URI) - including the Uniform Resource Locator (URL), the Digital Object Identifier (DOI) or the International Standard Book Number (ISBN).

BibliographicCitation

- dcterms:bibliographicCitation

BibliographicCitation is a bibliographic reference to a resource identifying the resource by bibliographic details. Typically the described resource is the child in a parent/child relationship where the bibliographicCitation is not describing the relationship but the location of the described resource within the parent resource (e.g. the bibliographic citation of an article consists of the name of a journal as well as the number of volume, issues and even page references). For the description of parent/child relations see hasPart or isPartOf. BibliographicCitation may only be used to describe bibliographic resources like books, articles or other documentary resource.

Guidelines for the creation of identifier content

Best practice is to **declare the identification system** from which an identifier is selected.

title	*What's a URI and why does it matter?*	
identifier	*http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/*	URI

as linked data

Identifiers should be selected from formal identification systems as above but can also be **local identifiers** as long as there is a proper declaration of these.

title	*Small and medium sized companies in Kathmandu*	
identifier	*03KTM147*	local ID

as linked data

Note that the identifier of the description of a resource (e. g. of the metadata record) is not the same as the identifier of the resource itself.

metadata record		
created	*2010*	W3CDTF
identifier	*013234098*	database ID
my Video		
title	*Medieval helpdesk with English subtitles*	
created	*2007*	W3CDTF
identifier	*http://www.youtube.com/watch?v=pQHX-SjgQvQ&feature=player_embedded*	URI

as linked data

If no identifier from a formal identification system exist, the identifier can be generated by a **bibliographic citation**. The bibliographic citation can be created as **text citation**,

title	*Prototyping Digital Library Technologies in zetoc*
bibliographicCitation	*Lecture Notes in Computer Science 2458, 309-323 (2002)*

as linked data

or as **machine readable citations**.

title	*Prototyping Digital Library Technologies in zetoc*
bibliographicCitation	*&ctx_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.jtitle=Lecture Notes in Computer Science&rft.volume=2458&rft.spage=309"^^info:ofi/fmt:kev:mtx:ctx*

as linked data

You can also **structure the bibliographic information**.

title	*My first article about metadata*	
identifier		
	journal title	*My Favorite Journal*
	volume	*3*
	issue	*2*
	start page	*14*
	date	*2010*

as linked data

For additional information on bibliographicCitation see "Guidelines for Encoding Bibliographic Citation Information in Dublin Core Metadata" (http://dublincore.org/documents/dc-citation-guidelines/index.shtml) .

Descriptions

Description

This property refers to the description of the content of a resource. The description is a potentially rich source of indexable terms and assist the users in their selection of an appropriate resource. To refine the character of a description use abstract or tableOfContent.

Abstract

This property is used when the description of a resource is a formal abstract.

TableOfContent

This property is used when the description of a resource is a structured list of the contents of a resource.

Guidelines for the creation of descriptions

A description may be **a free text account**,

title	*Bugs from New Zealand*
description	*A box of ten bugs collected in New Zealand between 1845 and 1846*

as linked data

an **abstract**,

title	*The Foundations of Programm Verification*
abstract	*This revised edition provides a precise mathematical background to several program verification techniques. It concentrates on those verification methods that have now become classic, such as the inductive assertions method of Floyd, the axiomatic method of Hoare, and Scott's fixpoint induction. The aim of the book is to present these different verification methods in a simple setting and to explain their mathematical background. In particular the problems of correctness and completeness of the different methods are discussed in some detail and many helpful examples are included.*

as linked data

a **table of contents**,

title	*Remains of Claire Klawitter*
table of content	*Diary 1822 - 1824 -- 20 pictures of Indian farmers -- 5 letters to Rudi Ratlos -- 1 map of North India*

as linked data

or a **reference to a description**.

title	*Thriller*
description	*http://en.wikipedia.org/wiki/Thriller_(album)*

as linked data

You can give some **information about the description you refer to**.

title	*Coriandrum sativum*	
description	*http://en.wikipedia.org/wiki/Coriander*	
	title	*Coriander*
	contributor	*Wikipedia*

as linked data

If **descriptions in different languages** exist, the property should be repeated with language tags.

title	*Delvig and Kjuchelbeker*	
abstract	*The book is a collection of works of two poets - contemporaries and friends of Pushkin - A. A. Delvig and V. K. Kjuchelbeker. It includes poems and prosa by Kjuchelbeker, parts of his diary, the poem Jurij and Xenia and reviews by Delvig. The attachment presents some retrospections to Delvig and Kjuchelbeker. A detailed biographic description tells us something about the life of the poets.*	eng

abstract	*Сборник впервые обединяет произведения двух поетов - современиков и друзей Пушкина - А. А. Дельвига и В. К. Кюхельбекера. Наряду со стихотворениями в книгу вклучены проза Кюхельбекера, фрагменты из его дневника, поема "Юрий и Ксения", а также рецензии Дельвига. В "Приложении" печатаются воспоминания о Дельвиге и Кюхельбекерею. Подробные биографические очерки рассказывают о жизненном пути поэтов.*	rus

as linked data

Subject

The property subject represents a relationship between a resource and another resource which is a topic of the first resource rsp. describes the intellectual content of the first resource. If the topic of the resource has a spatial or temporal character, use coverage, spatial or temporal.

Guidelines for describing the subject of a resource

To express the topic of a resource we recommend to **use a URI representing another resource describing this topic**,

title	*Inviato alla Biennale : Venezia, 1949 - 2009*	
subject	*http://www.labiennale.org/en/Home.html*	
	title	*La Biennale di Venezia*

as linked data

or a URI representing the value of a controlled vocabulary,

- like **entries from authority file systems** (e.g. the Library of Congress Subject Headings (http://id.loc.gov/authorities) or the authority files of the German National Library (http://d-nb.info/gnd/) , etc.),

title	*My Winter Wonderland*	
subject	*http://id.loc.gov/authorities/sh88004323#concept*	
	label	*Cross-country skiing--Skating*

as linked data

- like **entries from classification systems** ((e.g. the Dewey Decimal Classification (http://www.oclc.org/dewey/) , or Linnaean taxonomy, etc.).

title	*Transports in Kazakhstan 2000 - 2010*
subject	*W03_7*
subject	*G06_3*

label	*W03.7*
name	*Freight Transport*
broader	*W03*
narrower	*W03.72*
narrower	*W03.75*

label	*G06_3*
name	*Kazakhstan*

broader	*G06*
narrower	*G06_31*
narrower	*G06_35*

as linked data

You may also use plain text. But if you need **more than one entry to describe the content** you should repeat the property.

title	*How to get an aircraft*
subject	*aircraft*
subject	*leasing*

as linked data

If you want to use a keyword or keyphrase in **different languages**, you should use language tags.

title	*KONSTYTUCJA RZECZYPOSPOLITEJ POLSKIEJ*	
subject		
	Rzeczpospolita Polska	pol
	Republic of Poland	eng

as linked data

If the **subject is a person or organization** you should use names from formal name authorities (e.g. from the Library of Congress Name Authority Headings [1] (http://id.loc.gov/authorities) , or from the Virtuell International Authority File [2] (http://www.viaf.org/)).

title	*Candle in the wind*	
subject	gnd	*118583549*
	family name	*Monroe*
	given name	*Marilyn*
	born	*1926*
	died	*1962*

as linked data

Coverage

Coverage

The property coverage describes a relationship between a resource and another resource which represents the extent or scope of the content of the first resource. This includes the spatial locations (a place name or geographic co-ordinates), temporal periods (a period label, a date or a date range), or jurisdictions (states, counties, or other administrative entities). If you want to make a destinction between the temporal or spatial character of the content use temporal or spatial.

Temporal

This property describes the relationship between a resource and another resource which represent the temporal characteristics of the intellectual content of the first resource expressed by period labels or date encoding. If you want to describe date of the lifecycle of a resource use the date properties.

Spatial

This property describes the relationship between a resource and another resource which represents spatial characteristics of the intellectual content of the first resource expressed by by geographic names, latitude/longitude, or other established georeferencing.

Guidelines for describing the coverage, spatial or temporal character of a resource

To describe the **temporal** characteristic of a resource

you may use plain text,

title	*Transports in Kazakhstan 2000 - 2010*
coverage	*2000 - 2010*

as linked data

or **structure your entry** using dates,

title	*Transports in Kazakhstan 2000 - 2010*		
temporal			
	start	*2000*	W3CDTF
	end	*2010*	W3CDTF

as linked data

or **period labels**.

title	*Analysis of rocks collected in Perth*	
temporal		
	start	*Cambrian period*
	scheme	*Geological timescale*
	name	*Phanerozoic Eon*

as linked data

Further information on encoding temporal characteristics you will find in the DCMI Period Encoding Scheme (http://dublincore.org/documents/dcmi-period/) .

To describe the **spatial** character of a resource, you could use plain text,

title	*Analysis of rocks collected in Perth*
coverage	*Perth, W. A.*

as linked data

or express it by **georeferencing**,

title	*Analysis of rocks collected in Perth*	
spatial		
	east	*115.85717*
	north	*-31.95301*
	name	*Perth, W. A.*

as linked data

or **reference to a formal encoding**.

title	*The growth of trees in the suptropical highlands*	
spatial		
	label	*Cwb*
	source	*Köppen-Geiger Climate Classification*
	main Climates	*warm temperate*
	precipitation	*winter dry*
	temperature	*warmest month averaging below 22°C*

as linked data

Further information on encoding spatial characteristics you will find in the DCMI Box Encoding Scheme (http://dublincore.org/documents /dcmi-box/) and the DCMI Point Encoding Scheme (http://dublincore.org/documents/dcmi-point/) .

Dates

Date

The property date refers to a description of any dates or ranges in the lifecycle of a resource and is typically associated with the creation or availability. If the destinction between different sorts of date is necessary, the following subproperties should be used. If a date is describing the content of a resource the properties coverage or temporal have to be used.

Created

This property refers to a description of the date or range of the creation of a resource. According to the one-to-one principle this has to be the creation date of the resource being described and not the creation date of any other resource from which the described resource derives (e.g. a former version or a superior resource). So a resource is created only once, every other date of creation belongs to another resource that has to be described on its own.

Issued

This property refers to a description of the date of the formal issuance resp. publication of a resource. A resource is issued only once, every other issuance belongs to another resource that has to be described on its own. If the issuance of a resource is not formal the property "available" should be used.

Available

This property refers to a description of the date a resource did become or will become available. A resource becomes available only once, every other availability belongs to another resource that has to be described on its own. If the availability of a resource starts with the formal issuance resp. publication use "issued".

Modified

This property refers to a description of the date a resource was changed. You may record every date a resource was modified by repeating this property or record only one date (this should be the last one).

Valid

This property refers to a description of the date or range a resource is, was or will be valid. This property should be used if a resource is only valid resp. relevant until a particular date.

DateCopyrighted

This property refers to a description of the date or range of the copyright of the resource.

DateSubmitted

This property refers to a description of the date a resource was submitted (e.g. a thesis at a university department, an article at the editorial board of a journal, etc.).

DateAccepted

This property refers to a description of the date a resource was accepted (e.g. a thesis by a university department, an article by the editorial board of a journal, etc.)

Guidelines for the creation of content for dates

For the structure of date properties we recommend the usage of the **W3CDTF profile of ISO 8601** [W3CDTF]. It allows to sort search results by date and facilitates the merging of metadata of different applications.

You should use this encoding for **a point in time**

created	*2003-04-10*	W3CDTF

as linked data

but must not use it with **a range**,

valid	*2007-05-06/2007-07-15*

as linked data

or when the date is located **before the common area**.

created	*-500*	gYear

as linked data

If you need to **structure range data**, make a distinction of start and end date.

date			
	start	*2007-05-06"*	W3CDTF
	end	*2007-07-15*	W3CDTF

as linked data

If the **complete date is unknown** you should use

month and year

available	*2006-07*	W3CDTF

as linked data

or only the year

issued	*2009*	W3CDTF

as linked data

If **more than one date of the same type** (e.g. modified) is recorded, the property must be repeated.

modified	*2009-12-22*	W3CDTF
modified	*2010-01-08*	W3CDTF
modified	*2010-02-15*	W3CDTF

as linked data

Since a resource has only one date of creation, issuance, availability and/or copyright you may repeat the properties created, issued, available and dateCopyrighted only if you want to **provide the same date in another structure**.

created	*1752*	W3CDTF
created	*probably after 1752*	

as linked data

If the described date is **only approximately known**, you may use plain text,

created	*aprox. 500 B.C.*

as linked data

or describe it.

date		
	year	*500*
	qualifier	*approx.*
	epoch	*B.C.*

as linked data

Another date of creation, issuance, availability and/or copyright however belongs to another resource that has to be described on its own. The relation of both resources may be described by one of the relation properties or by source.

title	*Population estimates in Scandinavia*		
created	*2004*	W3CDTF	
source			
	title	*World health report 2002 statistical annex*	
	created	*2002*	W3CDTF

as linked data

Source and Relations

Relation

Relation represents the relationship between the described resource and another resource, that is related to the described resource in some way. Such relationship may be expressed reciprocally but this is not required and depends on the sort of relation. If the relation shall be specified more precicely use one of the following properties.

Source

Source describes the relationship between the described resource and another resource, from which the described resource is derived in whole or in part (e.g. data of a climate centre are the source of a forecast, a book or journal is the source of a scan, etc.)

IsPartOf

This property describes the relationship between the described resource and another resource of which the described resource is a physical or logical part (e.g. a painting as part of a collection, an article as part of a journal, etc.). The described resource is like a "child" in a hierarchical or "parent/child" relationship. For the reciprocal statement use hasPart.

HasPart

This property describes the relationship between the desribed resource and another resource which is a physical or logical part of the described resource (e.g. the described resource is a collection of paintings, or a journal with different articles, etc.). The described resource is like the "parent" in a hierarchical or "parent/child" relationship. For the reciprocal statement use isPartOf.

IsVersionOf

This property describes the relationship between the described resource and another resource, that is a former version, edition or adaptation of the described resource (e.g. the described resource is the revision of a book, or another recording of a song, etc.). Another version implies changes in the content of a resource. For resources with different formats use isFormatOf. For the reciprocal statement use hasVersion.

HasVersion

This property describes the relationship between the desribed property and another property, that is a later version, edition or adaptation of the described resource (e.g. the described resource is the older version of a revised book, or of a song, etc.). Another version implies changes in the content of a resource. For resources with different formats use hasFormat. For the reciprocal statement use isVersionOf.

IsFormatOf

This property describes the relationship between the described resource and another resource, that is a former version of the described resource with the same intellectual content but presented in another format (e.g. the described resource is the microfilm version of a printed book, or the pdf version of a doc document). For intellectual changes between resources use isVersonOf. For the reciprocal statement use hasFormat.

HasFormat

This property describes the relationship between the described resource and another resource, that is a later version of the described resource with the same intellectual content but presented in another format (e.g. the desribed resource is a printed book that is also availabel as a microfilm, or a doc document that is also available as pdf). For intellectual changes between resources use hasVersion. For the reciprocal statement use isFormatOf.

Replaces

This property describes the relationship between the described resource and another resource, that has been supplanted, displaced or superseeded by the described resource. It is used for the valid version in chain of versions (e.g. the described resource is the the last draft of a contract, or the current version of guidelines). For the reciprocal statement use isReplacedBy.

IsReplacedBy

This property describes the relationship between the described resource and another resource, that supplants, displaces or superseedes the described resource. It is used, when in chain of versions only one version is valid (e.g. the described resource is one of the former drafts of a contract, or a former version of guidelines). For the reciprocal statement use replaces.

Requires

This property describes the relationship between the described resource and another resource supporting the function, delivery or coherence of the content of the described resource (e.g. the described resource is an application that can be used only with a particular software, or hardware). For the reciprocal statement use isRequiredBy.

IsRequiredBy

The described resource is necesssary for the function, delivery or coherence of the content of the resource the property references to (e.g. the described resource is a software or hardware necesssary to use a particular application). For the reciprocal statement use requires.

References

This property describes the relationship between the described resource and another resource that is cited, referenced, or otherwise pointed to by the described resource (e.g. the described resource is an article citing a book, or an interview pointing to a play). For the reciprocal statement use isReferencedBy.

IsReferencedBy

This property describes the relationship between a resource and another resource that points to the described resource by citation, acknowledgement, etc (e.g. the described resource is a book cited in an article, or a play pointed to in an interview, etc.). For the reciprocal statement use references.

ConformsTo

This property describes the relationship between a resource and an established standard, to which the described resource conforms (e.g. a metadata record that conforms to the RDA standard, or a pipe that conforms to ISO 3183, etc.)

Guidelines for the creation of content for relations and source

You may refer to the related resource by plain text or by a URI representing the related resource. If you use plain text, you should **use a formal citation**.

issued	*2009*	W3CDTF
isFormatOf		
	Eike von Repgow: Sachsenspiegel, Auffs newe vbersehen mit Summarijs vnd Additionen ...; Leipzig 1561/1563	

as linked data

However recommended best practice is to **use an identifier instead of text**,

conformsTo	*http://www.w3.org/2001/XMLSchema*	-

as linked data

or to **describe the related resources like another resource**.

references			
	creator	*Black, Carl*	
	contributor	*White, Stuart*	
	title	*Black and White*	
	date	*1988*	W3CDTF

as linked data

If there is **more than one relation of the same sort** you have to repeat the property:

requires	
	audio
	video

as linked data

If **both resources of a relationship are described** the relation could be expressed reciprocally whereupon reciprocality could be generated automatically

identifier	*mySong1*	
title	*Candle in the wind*	
issued	*1973*	W3CDTF
description	*Portayal of the life of Marilyn Monroe*	
hasVersion	*mySong2*	

as linked data

identifier	*mySong2*	
title	*Candle in the wind*	
alternative	*Goodbye England's Rose*	
issued	*1997*	W3CDTF
description	*Tribut to the dead princess of Wales*	
isVersionOf	*mySong1*	

as linked data

Rights

Rights

The rights property represents the relationship between a resource and information about rights held in and over this resource. This includes information like access rights, Intellectual Property Rights (IPR), copyrights, references to legal documents describing how to use a resource, etc. To specify rights more precicely use accessRights or license.

AccessRights

This property represents the relationship between a resource and information about who can access a resource or an indication of its security status. Access rights provides information about restrictions to view, search or use a resource based on attributes of the resource itself or the category of user.

License

This property represents the relationship between a resource and a legal document giving official permission to do something with the resource (e.g. an otherwise free resource may not be used for reproduction within commercial applications). Examples of such licenses you will find at http://creativecommons.org/.

Guidelines for the creation of rights content

A rights statement may be **a text**,

title	*Data from my last evaluation*
accessRights	
	My colleagues only

as linked data

or **a URI** referencing to formal rights information,

title	*You and me*
rights	*http://creativecommons.org/licenses/by/3.0/legalcode*

as linked data

or a **combination of both**.

title	*GeoNetwork - Geographic Metadata Catalog*
license	*http://www.gnu.org/licenses/gpl.html*
	GNU General Public License

as linked data

If there are no formal rights statements to use you may also **create your own rights statement**.

title	*Diaries of Juanita Ramirez*
rights	*accessConditions*

identifier	*accessConditions*
title	*Access to my stuff*
description	*Resources under this right can only be read, searched and used by members of the myProject*

as linked data

Special properties for the description of education material

Audience

The property audience represents the relationship between a resource and the class of persons for whom the resource is intended or useful (e.g. the resource is a textbook for psychologists, etc.). To specify an audience more precisely use mediator or educationLevel.

Mediator

This property represents the relationship between a resource and the class of persons who mediate access to the resource and for whom it is intended or useful. This might be teachers, parents etc. (e.g. teachers are mediators for a resource intended to be used in elementary school lessons)

EducationLevel

This property refers to information about the progress of an audience through the educational or training context, for which the described resource is intended (e.g. the resource is an English workbook for students of the 4th - 5th grade).

InsctructionalMethod

This property refers to the process used to engender knowledge, attitudes and skills, that the described resource is designed to support. Typically it includes 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 the creation of content for properties describing education material

You should use formal or informal **controlled vocabularies**. Though none are registered by DCMI, implementors are encouraged to develop local lists of values, and to use them consistently.

title	*Advances Physics*
audience	
	elementary school pupils

as linked data

mediator	
	schoolteacher

as linked data

educationLevel	
	3rd - 4th grade

as linked data

InstructionalMethod	
	experimental learning

as linked data

Special properties for the description of collections

AccrualMethod

This property refers to the method by which items are added to a collection.

AccrualPeriodicity

This property refers to the frequency with wich items are added to a collection.

AccrualPolicy

This property refers to the policy governing the addition of items to a collection.

Guidelines for the creation of content for properties describing collections

Resources described by these properties **have to be collections**. We recommend to use values of formal or informal **controlled vocabularies**.

title	*Pottery in Scandinavia in the second half of 19th century*
accrualMethod	
	purchase

as linked data

accrualPeriodicity	
	irregular

as linked data

accrualPolicy	
	Objects of this collection have to be Scandinavian ceramics from 1940s to 1999s.

as linked data

Provenance

This property refers to a description of changes in ownership and custody. The statement should include any changes of the resource that are significant for its authenticity, integrity and interpretation.

Guidelines for the creation of content for provenance

The description of the provenance of a resource includes all changes made to a resource.

The provenance can be **described by text**,

title	*Luxor Obelisk*
provenance	
	Originally located at the entrance to the Luxor temple the obelisk came to Paris in 1836 as a gift by Muhammad Ali Pasha.

as linked data

or **desribed like another resource**.

title	*The flea circus*	
provenance		
	ownedBy	*1829 - 1833; Jim Button*
	ownedBy	*1833 - 1915; My Library*
	ownedBy	*since 1915; Flea Academy*

as linked data

Retrieved from "http://colab.mpdl.mpg.de/mediawiki/CreatingMetadata"

- This page was last modified 13:07:40, 2010-10-11.
-

PublishingMetadata

From MPDLMediaWiki

Go to UsingDC | CreatingMetadata

How to use DCMI Metadata as linked data

Contents

About the linked data examples

To present these examples in a concise form we use the Turtle syntax for RDF (http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/) . The examples therefore appear in code lines such as:

```
@prefix ex: <http://www.example.com/>.
ex:aResource ex:aProperty "An RDF Literal" .
```

or

```
@prefix ex: <http://www.example.com/>.
ex:aResource ex:aProperty ex:anotherResource .
ex:anotherProperty "An RDF Literal"@en .
```

Within the examples we use different namespaces. The following prefixes are used to specify the namespaces we use for our examples:

For the **Dublin Core namespaces** we use:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> (DCMI legacy namespace used for the 15 core properties)
@prefix dcterms: <http://purl.org/dc/terms/> (DCMI terms namespace of the DCMI terms - properties, classes and datatypes)
@prefix dctype: <http://purl.org/dc/dcmitype/> (DCMI type namespace of the DCMI classes of types)
```

Further namespaces used are:

```
@prefix ex: http://www.example.org/ (an exemplary namespace)
@prefix xsd: http://www.w3.org/2001/XMLSchema# (namespace of the XML Schema language)
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# (namespace of the rdf vocabulary)
@prefix rdfs: http://www.w3schools.com/RDF/rdf-schema.xml (namespace of the rdf schema vocabulary)
@prefix foaf: http://xmlns.com/foaf/0.1/ (namespace of the "Friend of a Friend" vocabulary)
@prefix gnd: http://d-nb.info/gnd/ (namespace of the authority files of the German National Library)
@prefix mime: http://purl.org/NET/mediatypes/ (namespace for MIME media types vocabulary)
@prefix skos: http://www.w3.org/2004/02/skos/core# (namespace of the SKOS vocabulary)
```

Properties of the terms namespace

The main difference between the legacy namespace and the terms namespace is the definition of domains and ranges for most terms of the last. The definition of a domain governs the entities for which a property may be used. The definition of a range governs the usage of literal and non-literal values in context with a property.

Properties of the terms namespace used only with literal values

dcterms:alternative

The range for dcterms:alternative is rdfs:Literal. So you can use dcterms:alternative only with literal values

```
ex:myBook dc:terms: title "Passion for Pulses" ;
        dcterms:alternative "A Feast of Beans, Peas and Lentils from Around the World" .
```

```
ex:myStuff dcterms:title "American Meteorological Association Newsletter" ;
         dcterms:alternative "AMA Newsletter" .
```

```
ex:myDocument dcterms:title "EU Stability Programm of Belgium"@eng ;
            dcterms:alternative "Council Opinion on the Updated
                                 Stability Programm of Belgium 2009 -
                                 2010 "@eng ,
                                "Stellungnahme des Rates zum
                                 aktualisierten Stabilitätsprogramm
                                 Belgiens für 2009 - 2012"@ger .
```

dcterms:available

The range defined for dcterms:available is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

```
ex:myMusic dcterms:available "2006-07"^^dcterms:W3CDTF .
```

dcterms:bibliographicCitation

The range defined for dcterms:bibliographicCitation is the class of rdfs:Literal. Values used with dcterms:bibliographicCitation have to be instances of this class. Therefore this property can only be used with literal values.

Moreover a domain of the class dcterms:BibliographicResource is declard for dcterms:bibliographicCitation. Thus dcterms:bibliographicCitation may only be used describing bibliographic resources.

```
ex:myArticle dcterms:title "Prototyping Digital Library Technologies in zetoc" ;
           dcterms:bibliographicCitation "Lecture Notes in Computer Science 2458, 309-323 (2002)" .
```

```
ex:myArticle dcterms:title "Prototyping Digital Library Technologies in zetoc" ;
           dcterms:bibliographicCitation "&ctx_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
           &rft.jtitle=Lecture Notes in Computer Science&rft.volume=2458&rft.spage=309"^^info:ofi/fmt:kev:mtx:ctx .
```

dcterms:created

The range defined for dcterms:created is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

```
ex:myPicture dcterms:created "2003-04-10"^^dcterms:W3CDTF .
```

```
ex:mySculpture dcterms:created "-500"^^xsd:gYear
```

```
ex:myScript dcterms:created "1752"^^dcterms:W3CDTF ,
                            "probably after 1752" .
```

```
ex:mySculpture dcterms:created "approx. 500 B.C."
```

```
ex:myData dcterms:title "Population estimates in Scandinavia" ;
          dcterms:created "2004-07-19"^^dcterms:W3CDTF ;
          dcterms:source ex:oldData .

ex:oldData dcterms:title "World health report 2002 statistical annex" ;
           dcterms:created "2002-09-28"^^dcterms:W3CDTF .
```

dcterms:date

The range defined for dcterms:date is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

dcterms:dateAccepted

The range defined for dcterms:dateAccepted is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

dcterms:dateCopyrighted

The range defined for dcterms:dateCopyrighted is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

dcterms:dateSubmitted

The range defined for dcterms:dateSubmitted is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

dcterms:identifier

The range defined for dcterms:identifier is the class of rdfs:Literal. Values used with dcterms:identifier have to be instances of this class. Therefore this property can only be used with literal values.

```
ex:myWebsite dcterms:title "What's a URI and why does it matter?" ;
             dcterms:identifier "http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/"^^dcterms:URI .
```

```
ex:myFile dcterms:title "Small and medium sized companies in Kathmandu" ;
          dcterms:identifier "03KTM147"^^ex:mylocalID .
```

```
ex:myVideo dcterms:title "Medieval helpdesk with English subtitles" ;
           dcterms:created "2007"^^dcterms:W3CDTF ;
           dcterms:identifier "http://www.youtube.com/watch?v=pQHX-SjgQvQ&feature=player_embedded"^^dcterms:URI .

ex:myMetadata dcterms:created "2010"^^dcterms:W3CDTF ;
              dcterms:identifier "013234098"^^ex:myDatabaseIdentifier .
```

dcterms:issued

The range defined for dcterms:issued is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

```
ex:myBook dcterms:issued "2009"^^dcterms:W3CDTF .
```

dcterms:modified

The range defined for dcterms:modified is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

```
ex:mySoftware dcterms:modified "2009-12-22"^^dcterms:W3CDTF ,
                               "2010-01-08"^^dcterms:W3CDTF ,
                               "2010-02-15"^^dcterms:W3CDTF .
```

dcterms:title

The range for dcterms:title is rdfs:Literal. So you can use dcterms:title only with literal values.

```
ex:myFurniture dcterms:title "Alvar Aalto Chair No. 66" .
```

```
ex:mySong dcterms:title "Autumn Leaves",
                        "The Dead Leaves" .
```

```
ex:myPainting dcterms:title "La Joconde"@fre ,
                            "Mona Lisa"@eng ,
                            "La Gioconda"@ita .
```

```
ex:myData dcterms:title "Data from a survey about the usage of metadata" .
```

dcterms:valid

The range defined for dcterms:valid is the class of rdfs:Literal. Values used with this property therefore have to be literal values.

```
 ex:myDraft dcterms:valid "2007-05-06/2007-07-15" .
```

Properties of the terms namespace used only with non-literal values

dcterms:accrualMethod

The range of dcterms:accrualMethod is the class dcterms:MethodOfAccrual. All values used with dcterms:accrualMethod have to be instances of this class. Therefore the property may only be used with non-literal values.

The domain of dcterms:accrualMethod is the class dcmitype:Collection. So dcterms:accrualMethod may be used only for the description of collections.

```
ex:myCollection dcterms:title "Pottery in Scandinavia in the second half of 19th century" ;
                dcterms:accrualMethod [ rdfs:label "purchase" ] .
```

dcterms:accrualPeriodicity

The range of dcterms:accrualMethod is the class dcterms:Frequency. All values used with dcterms:accrualPeriodicity have to be instances of this class. Therefore the property may only be used with non-literal values.

The domain of dcterms:accrualPeriodicity is the class dcmitype:Collection. So dcterms:accrualPolicy may be used only for the description of collections.

```
ex:myCollection dcterms:title "Pottery in Scandinavia in the second half of 19th century" ;
                dcterms:accrualPeriodicity [ rdfs:label "irregular" ] .
```

dcterms:accrualPolicy

The range of dcterms:accrualPolicy is the class dcterms:Policy. All values used with dcterms:accrualPolicy have to be instances of this class. Therefore the property may only be used with non-literal values.

The domain of dcterms:accrualPolicy is the class dcmitype:Collection. So dcterms:accrualPolicy may be used only for the description of collections.

```
ex:myCollection dcterms:title "Pottery in Scandinavia in the second half of 19th century" ;
                dcterms:accrualPolicy [ rdfs:label "Objects of this collection have to be
                                       Scandinavian ceramics from 1940s to 1999s." ] .
```

dcterms:accessRights

The range of dcterms:accessRights is the class dcterms:RightsStatement. Values used with this property have to be instances of the class RightsStatement and therefore non-literal values.

```
ex:myDatabase dcterms:title "Data from my last evaluation" ;
             dcterms:accessRights [ rdfs:label "My colleagues only" ] .
```

dcterms:audience

dcterms:audience has a range of the class dcterms:AgentClass. Values used with this property therefore have to be non-literal values.

```
ex:myTutorial dcterms:title "Advanced physic" ;
             dcterms:audience [ rdfs:label "elementary school pupils" ] .
```

dcterms:conformsTo

dcterms:conformsTo has a range of the class dcterms:Standard. Values used with this property therefore have to be non-literal values.

```
ex:myData dcterms:conformsTo <http://www.w3.org/2001/XMLSchema> .
```

dcterms:contributor

The range for dcterms:contributor is dcterms:Agent. All values used with this property have to be instances of the class [dcterms:Agent] . **dcterms:contributor must not be used with literal values**.

You may use dcterms:contributor only with non-literal values.

```
ex:myMusic dcterms:contributor gnd:135066719 .

gnd:135066719 foaf:familyName "Elliott" ;
             foaf:givenName "Missy" ;
             foaf:nick "Missy E" .
```

dcterms:coverage

dcterms:coverage has a range of the class dcterms:LocationPeriodJurisdication. All values used with dcterms:coverage have to be instances of this class. Therefore the property must only be used with non-literal values.

dcterms:creator

The range for dcterms:creator is dcterms:Agent. All values used with this property have to be instances of the class [dcterms:Agent] . **dcterms:creator must not be used with literal values**. You may use it only with non-literal values.

```
ex:myBook dcterms:creator _shakespearesName .

_:shakespearesName foaf:familyName "Shakespeare" ;
                  foaf:givenName "William" .
```

dcterms:educationLevel

dcterms:educationLevel has a range of the class dcterms:AgentClass. Values used with this property therefore have to be non-literal values.

```
ex:myTutorial dcterms:title "Advanced physic" ;
             dcterms:educationLevel [ rdfs:label "3rd - 4th grade" ] ;
```

dcterms:extent

The range of dcterms:extent is the class dcterms:SizeOrDuration. All values used with dcterms:extent have to be instances of this class. Therefore the property may only be used with non-literal values.

```
ex:myVideo dcterms:extent [ rdf:value "21 minutes" ] .
```

```
ex:myVideo dcterms:extent [ rdf:value "PT21M"^^xsd:duration ] .
```

dcterms:format

The range of dcterms:format is the class dcterms:MediaTypeOrExtent. All values used with dcterms:format have to be instances of this class. Therefore the property may only be used with non-literal values.

```
ex:myPicture dcterms:format mime:jpeg .
```

dcterms:hasFormat

This property is intended to be used with non-literal values.

dcterms:hasPart

This property is intended to be used with non-literal values.

dcterms:hasVersion

This property is intended to be used with non-literal values.

```
ex:mySong1 dcterms:identifier "mySong1"
           dcterms:title "Candle in the wind" ;
           dcterms:issued "1973"^^dcterms:W3CDTF;
           dcterms:description "Portayal of the life of Marilyn Monroe" ;
           dcterms:hasVersion ex:mySong2
```

dcterms:instructionalMethod

dcterms:instructionalMethod has a range of the class dcterms:MethodOfInstruction. Values used with this property therefore have to be non-literal values.

```
ex:myTutorial dcterms:title "Advanced physic" ;
              dcterms:InstructionalMethod [ rdfs:label "experimental learning" ] .
```

dcterms:isFormatOf

This property is intended to be used with non-literal values.

```
ex:myScan dcterms:issued "2009"^^dcterms:W3CDTF ;
          dcterms:isFormatOf [ rdfs:label "Eike von  Repgow: Sachsenspiegel, Auffs newe vbersehen mit
                             Summarijs vnd Additionen ...; Leipzig 1561/1563" ] .
```

dcterms:isPartOf

This property is intended to be used with non-literal values.

```
ex:myPainting dcterms:title "Still Life" ;
              dc:creator "Mignon, Abraham" ;
              dcterms:isPartOf http://www.rijksmuseum.nl/meesterwerken
http://www.rijksmuseum.nl/meesterwerken dcterms:title "The Masterpieces special"
                                        dc:contributor "Rijksmuseum Amsterdam"
```

dcterms:isReferencedBy

This property is intended to be used with non-literal values.

dcterms:isReplacedBy

This property is intended to be used with non-literal values.

dcterms:isRequiredBy

This property is intended to be used with non-literal values.

dcterms:isVersionOf

This property is intended to be used with non-literal values.

```
ex:mySong2 dcterms:identifier "mySong2"
           dcterms:title "Candle in the wind" ;
           dcterms:alternative "Goodbye England's Rose" ;
           dcterms:issued "1997"^^dcterms:W3CDTF ;
           dcterms:description "Tribut to the dead princess of Wales" ;
           dcterms:isVersionOf ex:mySong1 .
```

dcterms:language

The range of dcterms:language it the class dcterms:LinguisticSystem. All values used with dcterms:language have to be instances of this class. Therefore the property may only be used with non-literal values.

```
ex:myBook dcterms:title "A great deliverance" ;
          dcterms:language [ rdf:value "eng"^^dcterms:RFC4646 ] .
```

or

```
ex:myBook dcterms:title "A great deliverance" ;
          dcterms:language <http://www.lexvo.org/page/iso639-3/eng>
```

```
ex:myVideo dcterms:title "Charlie Wilson's War" ;
           dcterms:language <http://www.lexvo.org/page/iso639-3/eng> ,
                            <http://www.lexvo.org/page/iso639-3/hun> ,
                            <http://www.lexvo.org/page/iso639-3/tur> .
```

```
ex:myVideo1 dcterms:title "Medieval helpdesk with English subtitles" ;
            dcterms:identifier "http://www.youtube.com/watch?v=pQHX-SjgQvQ&feature=player_embedded"^^dcterms:URI .
            dcterms:isVersionOf ex:myVideo2 ;
            dcterms:language <http://www.lexvo.org/page/iso639-3/nor> ,
                             <http://www.lexvo.org/page/iso639-3/eng> .

ex:myVideo2 dcterms:title "Book help (better verson)" ;
            dcterms:identifier "http://www.youtube.com/watch?v=UOorZQLsmuA&feature=related"^^dcterms:URI .
            dcterms:hasVersion ex:myVideo1 ;
            dcterms:language <http://www.lexvo.org/page/iso639-3/nor> .
```

```
ex:mySong dcterms:title "The Power of Orange Knickers"
          dcterms:language _:eng

_:eng rdfs:Label "English"
      ex:639-1 "en"
      ex:639-2 "eng"
```

dcterms:license

The range of dcterms:license is the class dcterms:LicenseDocument. Values used with this property have to be instances of the class LicenseDocument and therefore non-literal values.

```
ex:mySoftware dcterms:title "GeoNetwork - Geographic Metadata Catalog" ;
              dcterms:license <http://www.gnu.org/licenses/gpl.html> .

<http://www.gnu.org/licenses/gpl.html> rdfs:label "GNU General Public License" .
```

dcterms:mediator

dcterms:mediator has a range of the class dcterms:AgentClass. Values used with this property therefore have to be non-literal values.

```
ex:myTutorial dcterms:title "Advanced physic" ;
              dcterms:mediator [ rdfs:label "schoolteacher" ] ;
```

dcterms:medium

The range of dcterms:medium is the class dcterms:PhysicalMedium. All values used with dcterms:medium have to be instances of this class. Therefore the property may only be used with non-literal values.

The domain of dcterms:medium is the class dcterms:PhysicalResource. So dcterms:medium may be used only for the description of physical resources.

```
ex:myPainting dcterms:medium _:oilOnWood

_:oilOnWood rdfs:label "oil on wood"
            rdfs:label "oil"
            rdfs:label "wood"
```

dcterms:provenance

The range of the provenance property is the class dcterms:ProvenanceStatement. So you may use this property only with non-literal values.

```
ex:myResource dcterms:title "Luxor Obelisk"
              dctems:provenance [ rdfs:label "Originally located at the entrance to the Luxor temple the
                                 obelisk came to Paris in 1836 as a gift by Muhammad Ali Pasha." ] .
```

```
ex:myBook dcterms:title "The flea circus" ;
          dcterms:provenance _:thisProvenance

_:thisProvenance ex:ownedBy "1829 - 1833; Jim Button" ,
                            "1833 - 1915; My Library" ,
                            "since 1915; Flea Academy" .
```

dcterms:publisher

The range for dcterms:publisher is dcterms:Agent. All values used with this property have to be instances of the class [dcterms:Agent] . **dcterms:publisher must not be used with literal values**. You may use it only with non-literal values.

```
ex:myBook dcterms:publisher gnd: 2125990-2 ,
                            _:kniznajaPalata

gnd:2125990-2 foaf:name "Rossijskaja Gosudarstvennaja Biblioteka ;
              foaf:homepage "http://www.rsl.ru/" .

_:kniznajaPalata foaf:name "Knižnaja Palata"
```

dcterms:references

This property is intended to be used with non-literal values.

```
ex:myArticle dcterms:references _:articlesReference .
_:articlesReference dc:creator "Black, Carl" ;
                    dc:contributor "White, Stuart" ;
                    dc:title "Black and White"
                    dc:date "1988"^^dcterms:W3CDTF
```

dcterms:relation

This property is intended to be used with non-literal values.

dcterms:replaces

This property is intended to be used with non-literal values.

dcterms:requires

This property is intended to be used with non-literal values.

```
ex:myGame dcterms:requires [ rdfs:label "audio" ] ,
                           [ rdfs:label "video" ].
```

dcterms:rights

The range of dcterms:rights is the class dcterms:RightsStatement. Values used with this property have to be instances of the class RightsStatement and therefore non-literal values.

```
 ex:myPicture dcterms:title "You and me" ;
             dcterms:rights <http://creativecommons.org/licenses/by/3.0/legalcode> .
```

```
ex:myDocuments dcterms:title "Diaries of Juanita Ramirez"
               dcterms:rights _:accessConditions

_:accessConditions dcterms:title "Access to my stuff"
                   dcterms:description "Resources under this right can only be read, searched and
                   used by members of the myProject" .
```

dcterms:rightsHolder

The range for dcterms:rightsHolder is dcterms:Agent. All values used with this property have to be instances of the class [dcterms:Agent] . **dcterms:rightsHolder must not be used with literal values**. You may use it only with non-literal values.

```
ex:myFilm dcterms:rightsHolder gnd:39454-3 .

gnd:39454-3 foaf:name "Bundesarchiv Koblenz" ;
            foaf:homepage "http://www.bundesarchiv.de/index.html.de" .
```

dcterms:source

This property is intended to be used with non-literal values.

dcterms:spatial

dcterms:spatial has a range of the class dcterms:LocationPeriodJurisdication. All values used with dcterms:spatial have to be instances of this class. Therefore the property must only be used with non-literal values.

```
ex:myData dcterms:title "Analysis of rocks collected in Perth" ;
          dcterms:spatial _:thisPlace .

_:thisPlace ex:east "115.85717" ;
            ex:north "-31.95301" ;
            ex:name "Perth, W. A." .
```

```
ex:myData dcterms:title "The growth of trees in the suptropical highlands" ;
          dcterms:spatial _:Cwb .

_:Cwb rdfs:label "Cwb"
      dc:source "Köppen-Geiger Climate Classification" ;
      ex:mainClimates "warm temperate" ;
      ex:precipitation "winter dry" ;
      ex:temperature "warmest month averaging below 22°C" .
```

dcterms:subject

This property is intended to be used with non-literal values.

```
ex:myBook dcterms:title "Inviato alla Biennale : Venezia, 1949 - 2009" ;
          dcterms:subject <http://www.labiennale.org/en/Home.html> .

<http://www.labiennale.org/en/Home.html> dcterms:title "La Biennale di Venezia"
```

```
ex:myVideo dcterms:title "My Winter Wonderland" ;
           dcterms:subject <http://id.loc.gov/authorities/sh88004323#concept> .

<http://id.loc.gov/authorities/sh88004323#concept> rdfs:label "Cross-country skiing--Skating" .
```

```
ex:myData dcterms:title: "Transports in Kazakhstan 2000 - 2010"
          dcterms:subject ex:W03_7 ,
          dcterms:subject ex:G06_3

ex:W03_7 rdf:type skos:Concept ;
         prefLabel "W03.7" ;
         skos:altLabel "Freight Transport" ;
         skos:broader ex:W03 ;
         skos:narrower ex:W03_72 ,
                       ex:W03_75 .

ex:G06_3 rdf:type skos:Concept ;
         prefLabel "G06_3 ;
         skos:altLabel "Kazakhstan" ;
         skos:broader ex:G06 ;
         skos:narrower ex:G06_31 ,
                       ex:G06_35 .
```

```
ex:myLaw dcterms:title "KONSTYTUCJA RZECZYPOSPOLITEJ POLSKIEJ"
         dcterms:subject _:polska

_:polska rdfs:label  "Rzeczpospolita Polska"@pol ,
                     "Republic of Poland"@eng .
```

```
ex:mySong dcterms:title "Candle in the wind" ;
          dcterms:subject gnd:118583549 .
gnd:118583549 foaf:_familyName "Monroe" ;
              foaf:_givenName "Marilyn" ;
              http://rdvocab.info/ElementsGr2/dateOfBirth "1926"^^dcterms:W3CDTF ;
              http://rdvocab.info/ElementsGr2/dateOfDeath "1962"^^dcterms:W3CDTF .
```

dcterms:temporal

dcterms:temporal has a range of the class dcterms:LocationPeriodJurisdication. All values used with dcterms:temporal have to be instances of this class. Therefore the property must only be used with non-literal values.

```
ex:myData dcterms:title "Transports in Kazakhstan 2000 - 2010"
          dcterms:temporal _:thisPeriod .

_:thisPeriod ex:start "2000"^^dcterms:W3CDTF ;
             ex:end "2010"^^dcterms:W3cDTF .
```

```
ex:myData dcterms:title "Analysis of rocks collected in Perth" ;
          dcterms:temporal _:thisPeriod .

_:thisPeriod ex:start "Cambrian period" ;
             ex:scheme "Geological timescale" ;
             ex:name "Phanerozoic Eon" .
```

dcterms:type

The range of dcterms:type is rdfs:Class. Values used with dcterms:type may only be non-literal values.

```
ex:myPainting dcterms:type dctype:StillImage .

dctype:StillImage rdfs:label "Still Image" .
```

```
ex:myStuff dcterms:type dctype:InteractiveResource ,
                        dctype:Text .

dctype:InteractiveResource rdfs:label "Interactive Resource" .

dctype:Text rdfs:label "Text" .
```

```
ex:myStuff dcterms:type _:PCGameType ,
                        dctype:Software .

_:PCGameType rdfs:label "PC Game" .

dctype:Software rdfs:label "Software" .
```

```
ex:myEvent dcterms:type _:conference .

_:conference rdfs:label "Conference"@eng ,
                        "Konferenz"@ger ,
                        "съезд"@rus .
```

Properties of the terms namespace, that may be used with literal or non-literal values

dcterms:abstract

There is no range defined for dcterms:abstract. Therefore you can use it either with literal values or with non-literal values.

```
ex:myBook dcterms:title "The Foundations of Programm Verification" ;
          dcterms:abstract "This revised edition provides a precise mathematical background to
                            several program verification techniques. It concentrates on those
                            verification methods that have now become classic, such as the
                            inductive assertions method of Floyd, the axiomatic method of Hoare,
                            and Scott's fixpoint induction. The aim of the book is to present these
                            different verification methods in a simple setting and to explain their
                            mathematical background. In particular the problems of correctness and
                            completeness of the different methods are discussed in some detail and
                            many helpful examples are included." .
```

```
ex:myBook dcterms:title "Delvig and Kjuchelbeker"
          dcterms:abstract "The book is a collection of works of two poets - contemporaries
                            and friends of Pushkin - A. A. Delvig and V. K. Kjuchelbeker. It
                            includes poems and prosa by Kjuchelbeker, parts of his diary, the
                            poem Jurij and Xenia and reviews by Delvig. The attachment presents
                            some retrospections to Delvig and Kjuchelbeker. A detailed biographic
                            description tells us something about the life of the poets"@eng ,
                            "Сборник впервые обединяет произведения двух поетов - современиков
                            и друзей Пушкина - А. А. Дельвига и В. К. Кюхельбекера. Наряду со
                            стихотворениями в книгу вклучены проза Кюхельбекера, фрагменты из его
                            дневника, поема Юрий и Ксения, а также рецензии Дельвига. В Приложении
                            печатаются воспоминания о Дельвиге и Кюхельбекерею. Подробные
                            биографические очерки рассказывают о жизненном пути поэтов."@rus
```

dcterms:description

There is no range defined for dcterms:description. Therefore you can use it either with literal values

```
ex:myObjects dcterms:title "Bugs from New Zealand" ;
             dcterms:description "A box of ten bugs collected in New Zealand between 1845 and 1846" .
```

or with non-literal values.

```
ex:myMusic dcterms:title "Thriller" ;
           dcterms:description <http://en.wikipedia.org/wiki/Thriller_(album)> .
```

dcterms:tableOfContents

There is no range defined for dcterms:tableOfContents. Therefore you can use it either with non-literal values and with literal values.

```
ex:myFile dcterms:title "Remains of Claire Klawitter" ;
          dcterms:tableOfContent "Diary 1822 - 1824 -- 20 pictures of Indian farmers
          -- 5 letters to Rudi Ratlos -- 1 map of North India" .
```

Legacy namespace

For properties of the legacy namespace neither domain nor range is defined. So all properties of the legacy namespace can be used either with literal values or with non-literal values.

Properties of the legacy namespace

dc:contributor

You may use dc:contributor either with literal values or with non-literal values.

```
ex:myMusic dc:contributor "Snoop Dogg" .
```

dc:coverage

You may use dc:coverage either with literal values or with non-literal values.

```
ex:myData dcterms:title "Transports in Kazakhstan 2000 - 2010"
          dc:coverage "2000 - 2010"
```

```
ex:myData dcterms:title "Analysis of rocks collected in Perth" ;
          dc:coverage "Perth, W. A." .
```

dc:creator

You may use dc:creator either with literal values or with non-literal values.

```
ex:myBook dc:creator "Shakespeare, William" .
```

dc:date

You may use dc:date either with literal or with non-literal values.

```
ex:myDraft dc:date _:thisValidity .

_:thisValidity ex:start "2007-05-06"^^dcterms:W3CDTF ;
               ex:end "2007-07-15"^^dcterms:W3CDTF .
```

```
 ex:mySculpture dc:date _:mySculptureDate .
 _mySculptureDate ex:year "500" ;
                  ex:qualifer "approx." ;
                  ex:epoch "B.C." .
```

dc:description

You may use dc:description either with literal values,

```
ex:myHerbs dc:title "Coriandrum sativum"
          dc:description "Coriandrum sativum or Coriander is a herb used in Europe, North Africa
          and Asia. It belongs to the familiy of Apiaceae and ist growing to 20 inch."
```

or with non-literal values

```
ex:myHerbs dc:title "Coriandrum sativum"
          dc:description http://en.wikipedia.org/wiki/Coriander

http://en.wikipedia.org/wiki/Coriander dc:title "Coriander"
                                       dc:contributor "Wikipedia"
```

dc:format

You may use dc:format either with litarel values or with non-literal values.

```
ex:myPicture dc:format mime:jpeg ,
                       "40 x 512 pixels" .
```

dc:identifier

You may use dc:identifier either with literal values

```
 ex:myVideo dc:title "My first article about metadata"
           dc:identifier "My Favorite Journal 3 (2), 14-25 (2010)" .
```

or with non-literal values

```
ex:myVideo dc:title "My first article about metadata"
           dc:identifier _:myCitation .

_:myCitation ex:jtitle "My Favorite Journal"
             ex:volume "3"
             ex:issue "2"
             ex:spage "14"
             ex:date "2010" .
```

dc:language

You may use dc:language either with literal values or with non-literal values.

```
ex:mySong dc:title "The Power of Orange Knickers"
          dc:language "English"
```

```
ex:mySong dc:title "The Power of Orange Knickers"
          dc:language "eng"^^dcterms:RFC4646 .
```

dc:publisher

You may use dc:publisher either with literal values or with non-literal values.

```
ex:myData  dc:creator "Hubble Telescope";
           dc:publisher "University of Nowhere" ,
                        "All Your Data Inc. .
```

dc:subject

There is no range defined for dc:subject. Therefore you can use it either with non-literal values and with literal values.

```
ex:myManual dc:title "How to get an aircraft"
            dc:subject "aircraft" ;
                       "leasing" .
```

dc:rights

You may use dc:rights either with literal values or with non-literal values.

```
ex:myVideo dc:Rights "May be used only by members of the myProject" .
```

```
ex:myBook dcterms:title "News from the South" ;
          dc:rights "http://creativecommons.org/licenses/by-nd/3.0/legalcode"^^dcterms:URI .
          dc:rights "Attribution-NoDerivs 3.0 Unported" .
```

dc:title

It is recommended to use dc:title with literal values. But there are important **uses with non-literal values** as well. To do so you have to use dc:title.

```
ex:myBook dc:title _:thisTitle .

_:thisTitle ex:greekAlphabet "Οιδίπους Τύραννος" ;
            ex:latinAlphabet "Oidipous Tyrannos" .
```

```
ex:myScript1 dcterms:title "Nibelungenlied Handschrift B" ;
             dc:title _:nibelungenlied .

ex:myScript2 dcterms:title "Nibelungenlied Handschrift C" ;
             dc:title _:nibelungenlied

_:nibelungenlied skos:prefLabel "Der Nibelunge Not" ;
                 skos:altLabel "Nibelungenlied" ,
                               "Nibelungenklage" ,
                               "Nibelungensage" ;
                 dcterms:description "The Nibelungenlied exists of 39 aventiuren created between 1180 and 1210" .
```

dc:type

The range of dcterms:type is rdfs:Class. Values used with dcterms:type may be either literal or non-literal values.

```
ex:myEvent dc:type "Conference" .
```

Retrieved from "http://colab.mpdl.mpg.de/mediawiki/PublishingMetadata"

Category: KIM

```
----------------------------------------------------------------------
Date:         Tue, 12 Oct 2010 17:27:48 +0100
From:         Pete Johnston <Pete.Johnston@EDUSERV.ORG.UK>
Subject:      Re: Using Dublin Core - next generation
To: To:       DC-GLOSSARY@JISCMAIL.AC.UK
```

Just a couple of quick points on the graphs:

1. I notice that you use a colour convention (yellow-ish v green) to distinguish between literal nodes and URI nodes. The usual convention for RDF graphs is to use rectangles for literals, and I think it might be a good idea to follow that?

2. The larger graph includes the triple

<http://www.gutenberg.org/files/46/46-h/46-h.htm> dcam:memberOf
<http://purl.org/dcmitype/Text> .

While this isn't "wrong", I think it is probably a slightly unusual example and might be confusing? i.e. dcmitype:Text is a class and would typically be referenced in an "is-a"/is-instance-of-class relationship (i.e. rdf:type or dcterms:type).

dcam:memberOf is typically used with things which are "sets" but aren't necessarily classes (i.e. "Vocabulary Encoding Schemes")

So I suggest

- replacing http://purl.org/dc/dcam/memberOf with http://www.w3.org/1999/02/22-rdf-syntax-ns#type

And if you do want to illustrate the use of a VES and the property dcam:memberOf, then add a new arc, hanging off the concept i.e. add

<http://id.loc.gov/authorities/sh85025303#concept>
<http://purl.org/dc/dcam/memberOf>
<http://id.loc.gov/authorities#conceptscheme>

3. I think the URI

http://en.wikipedia.org/wiki/Charles_dickens

identifies a document, i.e. if I do a GET on that URI, the server returns 200 and an HTML page.

But the relator properties are meant to be relations between things and agents, not things and documents - they don't have an rdfs:range, but I think that is the intent, anyway.

So I'd suggest using the DBpedia URI for the Person i.e. use

http://dbpedia.org/resource/Charles_Dickens

instead of http://en.wikipedia.org/wiki/Charles_dickens

Terms, DCMI Metadata Terms

The RDF Properties, Classes, Vocabulary Encoding Schemes, and Datatypes declared and maintained by the Dublin Core Metadata Initiative are collectively referred to as DCMI Metadata Terms [1]. Of these term types, only the notion of a Vocabulary Encoding Scheme is unique to DCMI; Properties, Classes, and Datatypes are exactly as defined in RDF (see glossary entry for RDF).

It hasn't always been this way. When Elements were introduced in a 1995 workshop, and Qualifiers in 1997, the W3C specification for RDF did not yet exist. The path from "native Dublin Core" term types to RDF only really ended with the publication of a revised DCMI Abstract Model in 2007 [2]. Starting with the Canberra Qualifiers of 1997 [3] -- Language, Scheme, and Type (Sub-Element) -- and continuing with the transitional Types A and B of 1998 [4], the path to RDF took the following turns:

-- The Dublin Core Qualifiers published in 2000 were of two types -- Element Refinements and Encoding Schemes.

-- By 2003, Elements began to be called Properties. The type Encoding Scheme was differentiated into Vocabulary Encoding Schemes and Syntax Encoding Schemes. [6]

-- In the meantime, 2004 saw the publication of a major revision of RDF and laying the groundwork for modern Semantic Web implementations. [7]

-- In the revised DCMI Abstract Model of 2007, Property replaced Element as the designation of choice and was explicitly equated with RDF Property. The term Element Refinement was dropped with the explanation that an Element Refinement was no more than a Property which happened to be a sub-property of another. Syntax Encoding Schemes were declared to be RDF Datatypes. [2]

[1] http://dublincore.org/documents/dcmi-terms/
[2] http://dublincore.org/documents/2007/06/04/abstract-model/#sect-5
[3] http://www.dlib.org/dlib/june97/metadata/06weibel.html
[4] http://dublincore.org/documents/1998/10/07/datamodel/index.shtml
[5] http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/
[6] http://dublincore.org/usage/documents/2003/02/07/principles/
[7] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

Dublin Core, The Dublin Core, Dublin Core Metadata Element Set

"The Dublin Core", also known as the Dublin Core Metadata Element Set, is a set of fifteen "core" elements (properties) for describing resources. The elements are: Creator, Contributor, Publisher, Title, Date, Language, Format, Subject, Description, Identifier, Relation, Source, Type, Coverage, and Rights. The fifteen-element Dublin Core, originally drafted with thirteen elements in 1995, has been formally standardized as ISO 15836:2009, ANSI/NISO Z39.85, and the IETF RFC 5013 (@cites). Metadata based on the fifteen elements is used in countless implementations, and the Dublin Core is one of the top metadata vocabularies in use today, particularly for Semantic-Web and Linked-Data applications. The Dublin Core Metadata Element Set is part of a larger set of DCMI Metadata Terms (see related entry "DCMI Metadata Terms").

The Dublin Core was the first result of a process which engendered a wider "Dublin Core community" and led to the creation of a Dublin Core Metadata Initiative. (See entries for "Dublin Core Metadata Initiative" and "Dublin Core, used as an adjective").

Andy Powell

"Dublin Core (DC) is a metadata standard that provides a vocabulary for describing the "core" attributes of resources, often in the context of resource discovery and/or management."

Dublin Core, used as an adjective

Besides The Dublin Core, Dublin Core Metadata Element Set, and Dublin Core Metadata Initiative (DCMI), described in separate entries, "Dublin Core" is commonly used as an adjective in the following:

-- "Dublin Core community". Participants in DCMI mailing lists, task groups, specialist communities, and annual conferences constitute a loosely defined Dublin Core community.

-- "Dublin Core metadata". In the late 1990s, "Dublin Core metadata" referred to metadata based on the fifteen-element Dublin Core, has come to refer what might best be described as a "style", "form", or "flavor" of metadata -- a style that has evolved from efforts to put the fifteen elements into the context of a coherent approach to metadata on the World Wide Web generally. As Dublin Core metadata now denotes a type of metadata based on a generic model, it is often pointed out that Dublin Core metadata does not actually require use of The Dublin Core or of any other DCMI metadata terms. The definitive characteristic of the Dublin Core style is the "Dublin Core application profile".

-- "Dublin Core application profiles". Analogously to "Dublin Core Metadata", the term "Dublin Core application profiles" referred in circa 2000 through 2004 to any metadata specification that "used" the Dublin Core, usually in combination with other metadata vocabularies. With the development of the DCMI Abstract Model in 2004 through 2007, the term "Dublin Core application profile" was narrowed in scope to refer specifically to application profiles based on the DCMI Abstract Model. Weak uptake of the DCMI Abstract Model, together with the emergence of alternative methods for documenting Linked Data patterns, may be loosening the dependence of the "DC application profile" notion on the DCMI Abstract Model per se in favor of a basis in Linked Data methods more generally.

Dublin Core Metadata Initiative (DCMI)

The Dublin Core Metadata Initiative (DCMI) is a public, not-for-profit organization, incorporated in Singapore, with a mission to develop interoperable metadata standards for a broad range of purposes and business models. DCMI is hosted by the National Library Board of Singapore and supported by institutional members and sponsors. From its start in 1995 with a workshop at OCLC in Dublin, Ohio, through

```
incorporation in Singapore in 2009, the Dublin Core Metadata
Initiative was managed as an activitiy within OCLC's Office
of Research.  DCMI publishes metadata specifications, holds
annual conferences, and hosts numerous discussion forums.
```

Dumb-Down Principle

The Dumb-Down Principle, which entered Dublin Core discourse in 1998 [1], has traditionally denoted a principled way of viewing a complex metadata description through the lens of a simpler, human-readable representation, typically Simple Dublin Core.

The meaning of "dumb-down" has evolved over the years:

-- In 1998, the "dumb-down procedure" discussed by the Dublin Core Data Model Working Group involved ignoring contextual information such as datatypes and vocabulary encoding schemes and resolving URIs, if possible, to readable value strings.

-- By 2000, the notion of "dumb-down" had been extended to include the resolution of "resource" placeholders in RDF triples to an intelligible text representation -- a "simple default name" by which the resource as a whole could be characterized. (This was done by resolving -- if necessary recursively -- triples using the predicate rdf:value.) Tools were supposed to "simply ignore any resources originating from intermediate structural nodes in the node-and-arc diagram, and follow the chain of rdf:value arcs until they terminate in a character string. The content of that character string is then returned as the value of the Dublin Core element."

-- In 2005, the notion of "informed dumb-down" introduced the idea of using the sub-property relationships encoded in RDF schemas to infer statements using more-general properties from statements using more-specific properties [2].

As the notion of Simple Dublin Core becomes relatively less salient in the context of Linked Data, the dumb-down principle is increasingly understood as referring to the more general notion of partial interoperability among imperfectly aligned data sets in an open Web environment on the basis of Semantic Web principles. This notion is less about converting metadata into simpler forms and more about using formal definitions to infer additional information that can be used to align metadata descriptions based on different vocabularies. (See also the glossary entries for "Simple Dublin Core" and "Open World Mindset".)

[1] http://dublincore.org/archives/1998/1998-09-24.decisions.html
[2] http://dublincore.org/documents/2005/03/07/abstract-model/

Namespace, DCMI Namespace, DCMI Namespace Policy

The DCMI Namespace Policy, in 2001 among the first such policies of its kind to be articulated for any vocabulary, declares the principles by which DCMI Metadata Terms are identified and published as an RDF vocabulary.

The DCMI Namespace Policy defines a DCMI Namespace as "a collection of DCMI term URIs where each term is assigned a URI that starts with the same 'base URI'. The 'base URI' is known as the DCMI namespace URI."[1] The policy defines the following four base URIs:

-- http://purl.org/dc/elements/1.1/ for the original fifteen properties of the Dublin Core Metadata Element Set, Version 1.1

-- http://purl.org/dc/dcmitype/ for classes in the DCMI Type Vocabulary

-- http://purl.org/dc/dcam/ for terms used in the DCMI Abstract Model (of which there are currently two)

-- http://purl.org/dc/terms/ for all other DCMI properties, classes, vocabulary encoding schemes, and datatypes

The policy explains how a DCMI namespace URI is used together with a name, such as "extent", to form the URI "http://purl.org/dc/terms/extent".

Like other notoriously polysemous terms, the term "namespace" has been a source of confusion. The Namespace Policy clarifies as follows:

-- Even though a DCMI namespace URI is sometimes used in XML formats -- arguably incorrectly -- as an XML namespace URI, a DCMI namespace is not the same as an XML namespace [2]. Technically, an XML namespace is a collection of two-part "expanded names", often abbreviated as "qualified names", or XML QNames, in which the namespace name is bound to a abbreviated prefix. An XML component with the same expanded name or QName can be used in multiple XML formats in the context of substantially different content models. In contrast, an RDF property with a given URI is designed to be interpreted a consistent way independently of the contexts in which it may appear. In RDF syntaxes such as RDF/XML and Turtle, prefixing mechanisms merely provide a way to abbreviate these URIs.

-- The grouping of term URIs into DCMI Namespaces is orthogonal to the grouping of terms into sets designed to meet other functional needs, as in various types of vocabularies and formats.

The DCMI Namespace Policy declares guidelines for maintenance changes to DCMI terms: The correction of editorial errata (e.g., updated URLs pointing to documentation external to DCMI) result in no changes to DCMI term URIs, while semantic changes judged likely to have a substantial impact on machine processing will trigger the creation of a new term with a new URI.

In one of the historically earliest illustrations of Linked Data principles, DCMI namespace and term URIs have since 2001 dereferenced to machine-processable DCMI term declarations, so that "clicking on" a URI in a browser will retrieve a term representation in RDF.

[1] http://dublincore.org/documents/dcmi-namespace/
[2] http://www.w3.org/TR/xml-names/

One-to-One Principle (current)

The One-to-One Principle, first formulated in the earliest Dublin Core meeting, dictates that a metadata description should refer to just one resource. This principle was articulated in the recognition that most existing metadata records, in practice, combined descriptions of what might conceptually be seen as multiple distinct entities. For example, metadata records about books routinely included information such as the affiliation of an author (i.e., a property of the author), or the metadata record about a painting (e.g., "Mona Lisa") might include description of the photograph itself (a JPEG image).

The basic idea behind the One-to-One Principle became a fundamental feature of the RDF data model, which required that distinct resources be identified, distinguished, and described separately, as in the case of the original "Mona Lisa", created in 1506 by Leonardo da Vinci, and a photograph of "Mona Lisa" created in 2008 by John Smith. Accepting RDF as the foundational model for metadata allowed Dublin Core descriptions both to respect the One-to-One Principle and to transcend the limitations of "flat" single-resource descriptions.

What constitutes a "resource", hence a meaningful object of description, lies of course in the eye of the beholder. In a simple bibliography, a book may be described as a single resource, whereas a trained library cataloger might perceive the same book as the Expression of a Work, the particular version of which (Manifestation) is available in multiple copies (Items) -- in effect, as four separate resources requiring four separate, though related, metadata descriptions, potentially retrieved from four separate sources. The One-to-One Principle, therefore, is relative to the variety of subjective viewpoints in the world, where some people make distinctions while others do not, and others yet may draw the boundaries differently.

Open World Mindset (a very rough draft!!)

The modern Web environment has created possibilities for knowledge management that are fundamentally new with respect to pre-Web ("normal") information technology.

Traditional information technology environments are "closed worlds" inasmuch their data sources are carefully controlled; data formats are custom-defined for specific applications. Traditional databases, for example, require agreement on a schema as a precondition for storing and querying data. Closed world technologies work efficiently for well-defined, bounded systems.

Since the early 1990s, the Web has placed local sources of information into a new global context, giving rise to what Mike Bergman calls an "open world mindset". [1] The Web provides a context for integrating different sources of content.

@@@ Ideas from Mike Bergman [1] - would need to summarized, condensed...

* "By design, tolerance of incomplete information.
* "Incremental, low-risk means to knowledge systems and management.
* "Domains can be analyzed and inspected incrementally
* "Systems designed to integrate information incrementally.
* "Systems designed to incorporate new information incrementally.
* "Schema can be incomplete and developed and refined incrementally
* "Database design and management can be more agile, with schema evolving incrementally.
* "Lower risk, lower cost, faster deployment, and more agile responsiveness.
* "The process of describing an open, semantic Web world can proceed incrementally, sequentially asserting new statements or conditions.
* "Start small, initial focus on a few applications with high returns.
* "Designed to tolerate incomplete information. Partially known.
* "Knowledge is never complete
* "Knowledge is found in structured, semi-structured and unstructed forms.
* "The data and the structures within these open world frameworks can be used and expressed in a piecemeal or incomplete manner
* "We can readily combine data with partial characterizations with other data having complete characterizations
* "Systems built with open world frameworks are flexible and robust; as new information or structure is gained, it can be incorporated without negating the information already resident, and
* "Open world systems can readily bridge or embrace closed world subsystems.
* "Reusable and extensible.
* "Information can be combined about similar objects or individuals even though they have different or non-overlapping attributes.
* "What is important to describe (the attributes) about certain information also varies by context and perspective.
* "What distinguishes knowledge from information is that knowledge makes the connections between disparate pieces of relevant information. As these relationships accrete, the knowledge base grows. Again, RDF and the open world approach are essentially "connective" in nature.

"By contrast, systems based on the closed-world assumption are more brittle - new connections and relationships tend to break relational models.

"This makes CWA and its related assumptions a very poor choice when attempting to combine information from multiple sources, to deal with uncertainty or incompleteness in the world, or to try to integrate internal, proprietary information with external data.

"However, many of the new knowledge economy challenges are anything but defined and bounded. These applications all reside in the broad category of knowledge management (KM), and include such applications as data federation, data warehousing, enterprise information integration, business intelligence, competitive intelligence, knowledge representation, and so forth.

"Open world does not necessarily mean open data and it does not mean open source. Open world is simply a way to think

about the information we have and how we act on it. Open world
technologies can be applied to internal, closed, proprietary
data and structures.

[1] http://www.mkbergman.com/852/the-open-world-assumption-elephant-in-the-room/

Resource Description Framework (RDF)

RDF is "a standard model for data interchange on the Web" [1]. First standardized as a W3C Recommendation in 1999 [2], RDF was re-released as a revised W3C Recommendation in 2004 [3]. RDF underpins current approaches to Semantic Web and Linked Data [4].

The Dublin Core approach to metadata, which began two years before work started on RDF, has evolved in close interaction with the RDF community. DCMI Metadata Terms are defined as RDF Properties, Classes, and Datatypes (plus a DCMI-specific type of term, the Vocabulary Encoding Scheme), and DCMI properties are currently among the most widely used properties in Linked Data. (See glossary entry for DCMI Metadata Terms.)

Metaphorically, RDF may be seen as a "grammar" for a "language of data". Uniform Resource Identifiers (URIs) -- in essence, Web addresses used as unique identifiers for things real or conceptual -- constitute the "words" of that language. Like natural-language words, the words of Dublin Core -- i.e., their URIs -- belong to grammatical categories, where "properties" (e.g., "references" or "isReferencedBy") are a bit like verbs or "predicates"; "classes" are a bit like nouns; "vocabulary encoding schemes" a bit like proper nouns; and "datatypes" a bit like adjectives.

Aside from being "words" in this data language, URIs double as "footnotes" indicating ownership and maintenance responsibility for the words by way of ownership of the domain names under which the "words" (URIs) are coined, as recorded in the globally managed Domain Name Service (DNS). For example, the subdomain http://purl.org/dc/ is "owned" (in the sense of "controlled") by the organization Dublin Core Metadata Initiative. Inasmuch URIs resolve to official representations of "words" (see glossary entry on Namespace), this globally managed space of unique identifiers functions as a continually updated "dictionary" of the RDF data language. For example, the URI http://purl.org/dc/terms/title resolves (by redirection) to an RDF schema ([1] at the time of writing) which says in a machine-understandable way that dcterms:title is an RDF property.

In its early years, the fifteen-element Dublin Core was likened to a "pidgin" -- a lexicon of generic predicates good enough for the sort of rough but serviceable communication one hears from intermediate-level speakers of foreign languages. As in natural languages, "sentences" make no sense without a shared sentence grammar which gives them meaning as complete thoughts, such as "Book A is a translation of Book B". The grammar of RDF statements follows a simple and consistent three-part grammar of "subject", "predicate", and "object". Statements all have the form: "Resource A is related to Resource B", where the "is related to" part is a predicate from an RDF vocabulary such as DCMI Metadata Terms.

Taken individually, an RDF statement does not say much, but when sentences are aggregated into "paragraphs" (RDF "graphs"), the statements can convey just about any kind of the information one might express using spreadsheets, databases, Web tags, or Web links. Just as pidgin vocabularies go only so far when expressing more specialized knowledge, a healthy ecosystem of RDF vocabularies needs to include both specialized lexicons -- e.g., for use by biologists, manufacturers, or library catalogers among themselves -- and generic lexicons for rough communication with the world in general.

RDF is a language designed by humans for processing by machines. RDF -- the grammar together with its vocabularies -- does not itself engineer a solution to the inherent problems of human communication any more than the English language guarantees world understanding. However, RDF does support the process of connecting dots -- "knowledge" -- by providing a shared basis for expressing information in a comprehensible way.

Just as English also provides a basis for communicating among non-native English speakers, RDF provides an idiom for coherently merging or linking data among systems which may not be based on RDF natively but can export or expose data on the basis of the RDF grammar using known RDF vocabularies.

RDF supports the longevity of information by expressing knowledge a generic form using a meaningful grammar -- assuming that society can find robust methods for preserving the parts of the Web where its dictionaries and resource identifiers are defined. Aside from supporting data interchange in the here and now, RDF provides a response to the ongoing and inevitable obsolescence of our computer applications and customized data formats.

[1] http://www.w3.org/RDF/
[2] http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
[3] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
[4] http://linkeddata.org/
[5] http://dublincore.org/2010/10/11/dcterms.rdf

Resource Discovery, Resource Description

As of 2010, the Dublin Core Metadata Initiative's mission includes the goal of "developing and maintaining international standards for describing resources".

It has not always been this way. The foundational workshop of March 1995 in Dublin, Ohio aimed at improving the "discovery" of electronic resources on a rapidly growing World-Wide Web. The approach taken was that of defining "elements" for simple "records" describing "document-like objects" [1]. Implicit was the traditional model of a search application indexing text fields (value strings) for retrieval.

What actually happened, as we know, was that Google developed a radically different, wildly successful, solution to the problem of resource discovery. The Dublin Core community's understanding of metadata's role subsequently evolved in several ways:

-- The focus on "resource discovery" came to be seen as artificial -- what descriptive element might not be conceived as "useful for discovery"? -- and the objective was redefined as that of "resource description".

-- The scope of Dublin Core metadata was broadened from "electronic resources" to encompass, in principle, any object that can be identified, whether electronic, real-world, or conceptual, and particularly including resources of the sort named in the DCMI Type Vocabulary [2], such as physical objects, software, services, and sound.

-- Attention shifted from the "record" as an aggregate information object to individual metadata "statements" designed to be merged and recombined in an emerging Linked Data environment (see glossary entry "records or statements").

-- Metadata came to be seen as valuable less for the value strings it carried than for its use of URIs -- controlled identifiers for subject headings, Web documents, people, and the like -- and for the role of properties (such as Dublin Core properties) in defining explicit links and cross-references between resources. URI-rich metadata came to be seen as the basis for richly interlinked browsing and data mashups.

[1] http://www.dlib.org/dlib/July95/07weibel.html
[2] http://dublincore.org/documents/dcmi-type-vocabulary/
[3] http://dublincore.org/about/#mission

Simple Dublin Core, Qualified Dublin Core

With the invention of Qualifiers in 1997 (see glossary entry Terms), a distinction was made between Simple and Qualified Dublin Core. Qualified Dublin Core referred to metadata that used the Dublin Core with DCMI qualifiers. Simple Dublin Core was typically taken to refer to:

-- a description of just one resource,
-- using only the fifteen properties of the Dublin Core Metadata Element Set,
-- all optionally and repeatably,
-- with string values,
-- and without qualifiers.

This pattern was the product of a time when the Dublin Core Metadata Element Set was widely understood, even within the Dublin Core community, as the specification of a record format. At the time, it was also assumed that metadata should consist of textual values indexed for retrieval. This pattern was widely deployed after 2000 in systems supporting the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), which requires that an OAI-PMH repository expose records using an XML format, oai_dc [1], which may be seen as a serialization of Simple Dublin Core. Due to the popularity of such formats, many people still refer to the Dublin Core Metadata Element Set and Simple Dublin Core interchangeably.

Both Simple Dublin Core and Qualified Dublin Core are defined in XML schemas owned and maintained by DCMI [2] -- the latter following "Guidelines for implementing Dublin Core in XML" of 2003 [3]. While these formats have proven to be useful in numerous XML-based implementations over the years, such fixed formats show limitations in the heterogenous context of Linked Data. The availability of more flexible Semantic Web solutions cast Simple and Qualified Dublin Core into a different light:

-- In a Semantic Web perspective, the Dublin Core Metadata Element Set -- i.e., the set of fifteen properties -- is understood as just one vocabulary, among many, available for use in data. The Simple Dublin Core pattern constrains the use of its properties in one particular way and can therefore be seen as an application profile that happens to be limited to properties from one DCMI namespace.

-- The limitation of Simple Dublin Core to string values looks dated at a time when the Linked Data approach is emphasizing the use of URIs in order to make richly interlinked connections with other resources and metadata descriptions.

-- In practice, implementers of Dublin-Core-based metadata more often create application profiles that use a sub-set of Dublin Core properties, or use Dublin Core properties together with properties from other vocabularies. In a Linked Data environment, and in accordance with an Open World Mindset, perfect agreement on fixed formats is not imperative, making the limitation to properties from one DCMI namespace seem a bit arbitrary. (See glossary entry on Open World Mindset.)

-- Providing interoperable metadata as Linked Data, on the other hand, does not mean that data must be expressed natively using RDF. It may indeed be more practical to manage data in fixed XML formats on the backend and transform or convert the data for publication in a Linked Data context.

[1] http://www.openarchives.org/OAI/2.0/oai_dc.xsd
[2] http://dublincore.org/schemas/xmls/
[3] http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/

e:/u/folders/XPAD/Agenda.pdf
e:/u/folders/XPAD/UsingDC.pdf
e:/u/folders/XPAD/CreatingMetadata.pdf
e:/u/folders/XPAD/PublishingMetadata.pdf
e:/u/folders/XPAD/pete-comments-on-user-guide.pdf
e:/u/folders/XPAD/DCMI-Metadata-Terms.pdf
e:/u/folders/XPAD/Dublin-Core.pdf
e:/u/folders/XPAD/Dumb-Down.pdf
e:/u/folders/XPAD/Namespace-Policy.pdf
e:/u/folders/XPAD/One-to-One-Principle.pdf
e:/u/folders/XPAD/Open-World-Mindset.pdf
e:/u/folders/XPAD/RDF.pdf
e:/u/folders/XPAD/Resource-Discovery.pdf
e:/u/folders/XPAD/Simple-Dublin-Core.pdf
e:/u/folders/XPAD/index.txt
e:/u/folders/XPAD/FAQ-reusing-XML-elements.pdf

"In my metadata, I would like to use Dublin Core elements together with elements from the XML-based Standard XYZ in my metadata format. How can I do this?"

An XML format is a binding for an information structure constructed according to a specified model. XML formats define an ad-hoc "language" of elements and attributes nested within a hierarchical structure. The meaning of these components is determined solely by their placement in the tree structure of the given XML language and the interpretation that the developers of that language define in accompanying documentation. Prefixes bound to "namespace" URIs are used to avoid name collision between like-named elements, but there is no notion of XML elements themselves being identified by URIs.

In RDF, "elements" -- which in RDF are called "properties" -- are identified and referenced using URIs, and there is no notion of element containment or nesting. In contrast, there is a notion of URIs as components of "statements". The meaning of properties is considered to be global in scope, independently of any specific information structure, such as a particular record format. Statements using those properties are designed to be interpretable beyond the context of a specific information structure.

Terms defined as RDF properties are therefore not directly usable as elements in XML formats, and vice versa -- the two types are like apples and oranges.

In practice, the two can be used together if RDF properties (or classes or datatypes) are created which correspond to -- are based on or map to -- XML elements and attributes, or vice versa. In very simple cases, mappings may be so obvious as to seem trivial. But complex, highly nested XML formats can be intellectually challenging to interpret in RDF, especially if the tree structures are at all ambiguous about how particular elements or attributes explicitly relate to resources described.

In short: To "use" an XML element in RDF statements, a corresponding RDF property with a compatible meaning must be coined and assigned a URI -- but then it is no longer an XML element. To "use" an RDF property in an XML format, an RDF property can be taken as the inspiration for an XML element of comparable meaning -- but then it is no longer an RDF property. In other words, because XML elements and RDF properties are fundamentally different, they cannot be "used" together directly, but only via translation from one model into the other. Both uses may base themselves on DCMI Metadata Terms -- seen as a set of concepts with (human-readable) definitions -- but only as RDF properties are DCMI Metadata Terms expressed in a form directly usable for interoperability in a Linked Data context.

The differences between the RDF and XML models, and their consequences for interoperability, are elaborated in a 2005 discussion paper by Pete Johnston [1].

[1] http://www.ukoln.ac.uk/metadata/dcmi/dc-elem-prop/