An Introduction to Dublin Core

Diane I. Hillmann

Cornell University

Tutorial 2: Basic Semantics at DC-2005, Madrid
13 September 2005

What is Metadata?

- Answers come from three traditions:
  - Database Management Systems (“Schemas of relational databases”)
  - Library Cataloging Traditions (MARC & AACR2)
  - The World Wide Web (since the mid-1990’s)
    - The context for Dublin Core

Types of Metadata

- Administrative
- Descriptive
- Access/Use
- Preservation
- Technical/Structural
- Other?

Comparing Libraries & the Web

- Library tradition:
  - More than 100 years of experience, from catalog cards to MARC
  - MARC Records include:
    - Description (all objects)
    - Subjects and classification (topical context)
    - Holdings information (location)
    - Administrative information required to manage records
  - Distributed metadata creation based on common consensus on standards

The Web: One distributed Library?

- The situation in the mid-1990s:
  - Thousands of information providers, using a variety of metadata schemas (if anything at all)
  - Search engines providing too many hits and very little precision
  - Volatile resources changing addresses, disappearing, etc.
  - Exponential growth in numbers and types of resources available

Fifteen Core Elements (1996)

Creator	Title	Subject
Contributor	Date	Description
Publisher	Type	Format
Coverage	Rights	Relation
Source	Language	Identifier

Functions of Metadata

Discover resources	Manage documents	Control IP Rights
Identify versions	Certify authenticity	Indicate status
Mark content structure	Situate geospatially	Describe processes

Characteristics of the Dublin Core

- A flat file structure, with:
  - All elements optional
  - All elements repeatable
- Elements may be displayed in any order
- Extensible
- International in scope

Dublin Core Principles

- Dumb-Down
- One-to-One
- Appropriate Values

Dumb-Down

- The fifteen core elements are usable with or without qualifiers
- Qualifiers make elements more specific:
  - Element Refinements narrow meanings, never extend
  - Encoding Schemes give context to element values
- If your software encounters an unfamiliar qualifier, look it up – or just ignore it!

The One-to-One Principle

- Describe one manifestation of a resource with one record
  - Ex.: a digital image of the Mona Lisa is not described as if it were the same as the original painting
- Separate descriptions of resources from descriptions of the agents responsible for those resources
  - Ex.: email addresses and affiliations of creators are attributes of the creator, not the resource

Appropriate Values

"Best practice for a particular element or qualifier may vary by context, but in general an implementor cannot always predict that the interpreter of the metadata will always be a machine. This may impose certain constraints on how metadata is constructed, but the requirement of usefulness for discovery should be kept in mind."

-- from *"Using Dublin Core"*

Moving Towards a Data Model

- Collective realization that machine-processability requires a coherent data model
- 1996: “Warwick Framework” proposed at DC-2 workshop: DC as one specialized module (“resource discovery”) among many
- 1997: “Qualifiers” proposed for specifying meanings
  - Some early adopters take this to unintended extremes: “DC.Creator.telephone-number”
- 1998: DCMI involvement in emerging Resource Description Framework and clarification of simple data model for Dublin Core
- 2000: First set of qualifiers officially approved

DC Data Model Finalized (2005)

- Provides explicit definitions of resources
- Relates DC principles and practices to the developments outside DCMI
- Makes clear the relationship of DC "packages" of information to other metadata "packages"
- Paves the way for future progress for DCMI

Terms and Values: a simple view

- <term> is an element or property
- <value> is a resource represented by a string or a URI
- <term> = <value>
- Called “property/value pairs” in the DCMI Abstract Model document
- Vocabularies appear on both sides of the equal sign
- Best: <term [general bucket]> = <value [specialized vocabulary]>

Simple and Qualified DC

- Varying definitions for Simple DC:
  - Only the original 15 elements, or
  - All available elements, without encoding schemes or refinements
  - In each case only making use of *value strings*
- Qualified DC
  - Metadata that makes use of some or all the features of the abstract model
    - Element Refinements
    - Value Encoding Schemes

Element Refinements

- Make element meanings narrower, more specific:
  - a Date *Created* versus Date *Modified*
  - an *IsReplacedBy* versus *Replaces* Relation
- Depending on syntax chosen, refinements may appear as stand-alone tags instead of with elements:
  - **<dct:created>2002-10-04</dct:created>**, instead of:
  - **<dc:date><dct:created>2002-10-04 </dct:created><dc:date>**
    - Requires a schema to dumb-down *Date Created* to *Date*
  - Dublin Core is simple enough to support both usages

Value Encoding Schemes

- Indicate that the value is:
  - a term from a controlled vocabulary (e.g., Library of Congress Subject Headings)
  - a string formatted in a standard way (e.g., that "05/02" means May 2nd, not February 5th)
- Even if a scheme is not known by software, the value should be "appropriate" and usable for resource discovery.

Application Profiles & Interoperability

- Implementers want to know how their peers design metadata – to avoid "reinventing the wheel"
- Information providers need to harmonize metadata usage for improved access within domains, e.g.:
  - Between countries (Nordic Metadata Project)
  - Preprint repositories (Open Archives Initiative)
  - Subject gateways (Renardus)
  - Mathematics and physics (MathNet, PhysNet)

Creating Metadata Records

- The “Library Model”
  - Trained catalogers, one-at-a-time metadata records
- The “Submission Model”
  - Authors create metadata when submitting resources
- The “Automated Model”
  - Automated tools create metadata for resources

Distributing Metadata Records

- Mid-1990s: HTML tags embedded in Web pages
  - Simple, easy to deploy, but inflexible, hard to maintain
  - Bad tags like *DC.Creator.eyecolor* imply a non-existent support for nesting and for entity distinctions
- 2000+: Better XML/RDF alternatives
  - RDF metadata supports complex structures without breaking simple DC grammar
  - Open Archives Initiative promotes mass adoption of an XML schema for simple, unqualified Dublin Core records – along with a protocol to make them available

Distribution Models

- Database with dedicated search engine
  - Good control, little need for interoperability
  - Not much re-use in other contexts
- Metadata exposure and harvesting
  - Standards, agreements and interoperability necessary
- Embedded metadata
  - Generally websites or individual documents
  - Doesn’t scale well, maintenance difficult

Distribution Model: Metadata Harvesting

- Service Providers harvest and integrate metadata from diverse Content Providers
  - Presupposes standard element sets, record formats, and harvesting methods
- 1999+: Open Archives Initiative began as a federation of scholarly pre-print providers
  - Today: aggregations using OAI-PMH creating new models of metadata services

Why the Harvesting Model Works

- Exposing metadata facilitates:
  - Reuse of available metadata
  - The creation of value-added services
- Low entry cost is key to deployable digital library infrastructure
  - New Static Repositories version of OAI lowers entry bar even more
- Individual communities have begun to customize the common infrastructure
  - Simple DC minimum, but supports any metadata

Dublin Core Grows and Changes

- DCMI Community emphasizes open participation
  - Conferences, working groups, discussion lists
- DCMI term set evolves as implementers coin new terms and usage patterns emerge
- DCMI Usage Board reviews proposals for new metadata terms

Dublin Core Usage Board

- Usage Board receives proposals for new elements, refinements, encoding schemes, Type terms
  - Evaluates proposals in light of grammatical principles, usefulness, clarity of definition, overlap with existing terms
- Moving away from review of terms “in isolation” towards review of Application Profiles

DCMI Namespaces and Policies

- All DCMI metadata terms are given unique identity within three namespaces:
  - http://purl.org/dc/elements/1.1/ - the legacy DC-15
  - http://purl.org/dc/terms/ - all other elements/qualifiers
  - http://purl.org/dc/dcmitype/ - a Type vocabulary
  - Example: http://purl.org/dc/elements/1.1/title
- Policies promote long-term stability of namespace URIs
  - Changes not substantially "semantic" (i.e., corrections) will not result in change of namespace URIs

Finding Out More about DC

- DCMI Web Site
  - http://dublincore.org
- “Using Dublin Core”
  - http://dublincore.org//documents/usageguide/
- Partiticipating in a Working Group
  - http://dublincore.org/groups
- Ask a question!
  - http://askdcmi.askvrd.org/

Questions?

Thank you for your attention!

Diane I. Hillmann

Cornell University

DIH1@cornell.edu (fourth character is “one”)