Dublin Core Metadata Initiative Abstract Model

JENN RILEY
METADATA LIBRARIAN
DIGITAL LIBRARY PROGRAM

DL BROWN BAG SERIES
FEBRUARY 11, 2009

This is not your father's DC

DCAM ≠ DCMES

The DCAM is a *different way of looking at data* than what libraries are used to.

The DCAM is…

An “abstract model for Dublin Core metadata”

AND

“an information model which is independent of any particular encoding syntax”

History of the DCAM

* Remember: the DCMES (and DC Terms) *never* intended to be the be all, end all, of metadata

- Qualified Dublin Core first released in 2000
  - Dumb-down principle a great idea, but challenge comes when one tries to make statements about, for example, creator roles
  - Rise of RDF 1999-2004 starts folks thinking about self-descriptive models
- (Lots of community debate)
- The term "abstract model" appears in DC documentation *at least* as early as January 2002
- (Lots of community debate)
- Abstract Model first released as stable DCMI Recommendation in March 2005
- Current version of Abstract Model released as stable DCMI Recommendation in June 2007

DCMI Resource Model

- *Resources* are described using *property-value pairs*
  - This concept is familiar to libraries; a field and its content
  - DCAM applies additional constraints, however
- Types of values
  - *literal value*: represents something by means of a string
  - *non-literal value*: the something itself, not a reference to it
- **A *value* is also itself a *resource***

DCMI Description Set Model (1)

- *Description set*: collection of description(s)
- *Description*
  - Makes *statements* (which contain *property-value pairs*)
  - *Can* contain a URI for the described resource
- *Property-value pairs*
  - Properties are kind of like elements (but wait 'til later!)
  - Properties must be represented by URIs
  - The values are where this gets complicated!

DCMI Description Set Model (2)

- *Value surrogates*
  - *Literal value surrogate*:
    - Representation of a literal value by means of a string
    - From RDF, a literal value is generally something like a number or date
  - *Non-literal value surrogate*
    - Representation of a non-literal value
    - *Can* have a URI referring to the value
    - *Can* have a vocabulary encoding scheme URI
    - *Can* have a value string (literal representing the non-literal value)
- *Value strings*
  - *Plain value string*: just a string, but can have an associated language code
  - *Typed value string*: also associates the string with a syntax encoding scheme via a URI

DCMI Vocabulary Model

- Not just what we in libraries consider to be controlled vocabularies!
- Vocabularies can contain:
  - *Properties*
  - *Classes*
  - *Vocabulary encoding schemes*
  - *Syntax encoding schemes*

What libraries are used to

DC encodings

- Don’t have to implement entire DCAM
  - But do need to make clear which parts are supported
- Current encoding statuses
  - RDF encoding implementing DCAM now a Recommendation
  - XML encoding implementing DCAM still a Working Draft (since May 2006)*
  - XHTML <meta> and <link> encoding implementing DCAM now a Recommendation

* I think this says something interesting.

XML DC encodings


```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://example.org/myapp/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://example.org/myapp/
    http://example.org/myapp/schema.xsd"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>DCMI Home Page</dc:title>
    <dc:publisher>Dublin Core Metadata Initiative</dc:publisher>
    <dc:identifier>http://dublincore.org/pages/home</dc:identifier>
    <dc:subject xsi:type="dcterms:LCSH">Metadata</dc:subject>
</metadata>
```

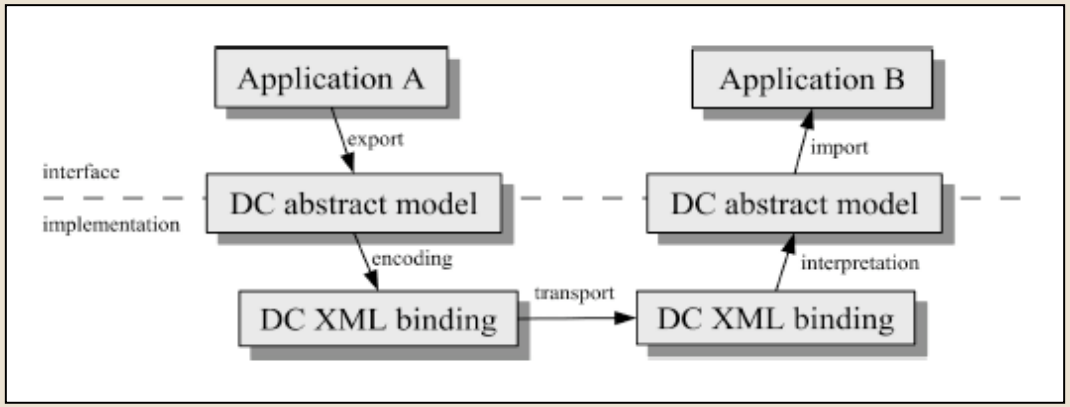
← Old

New →

```
<?xml version="1.0" encoding="UTF-8"?>
<dcx:descriptionSet xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcx="http://purl.org/dc/xml/">
    <dcx:description dcx:resourceURI="http://dublincore.org/pages/home">
        <dc:title>
            <dcx:valueString>DCMI Home Page</dcx:valueString>
        </dc:title>
        <dc:publisher dcx:valueURI="http://example.org/agents/DCMI">
            <dcx:valueString>Dublin Core Metadata Initiative</dcx:valueString>
        </dc:publisher>
        <dc:subject dcx:vocabEncSchemeURI="http://purl.org/dc/terms/LCSH">
            <dcx:valueString>Metadata</dcx:valueString>
        </dc:subject>
    </dcx:description>
</dcx:descriptionSet>
```

So why go to all of this trouble?

"Interoperability"

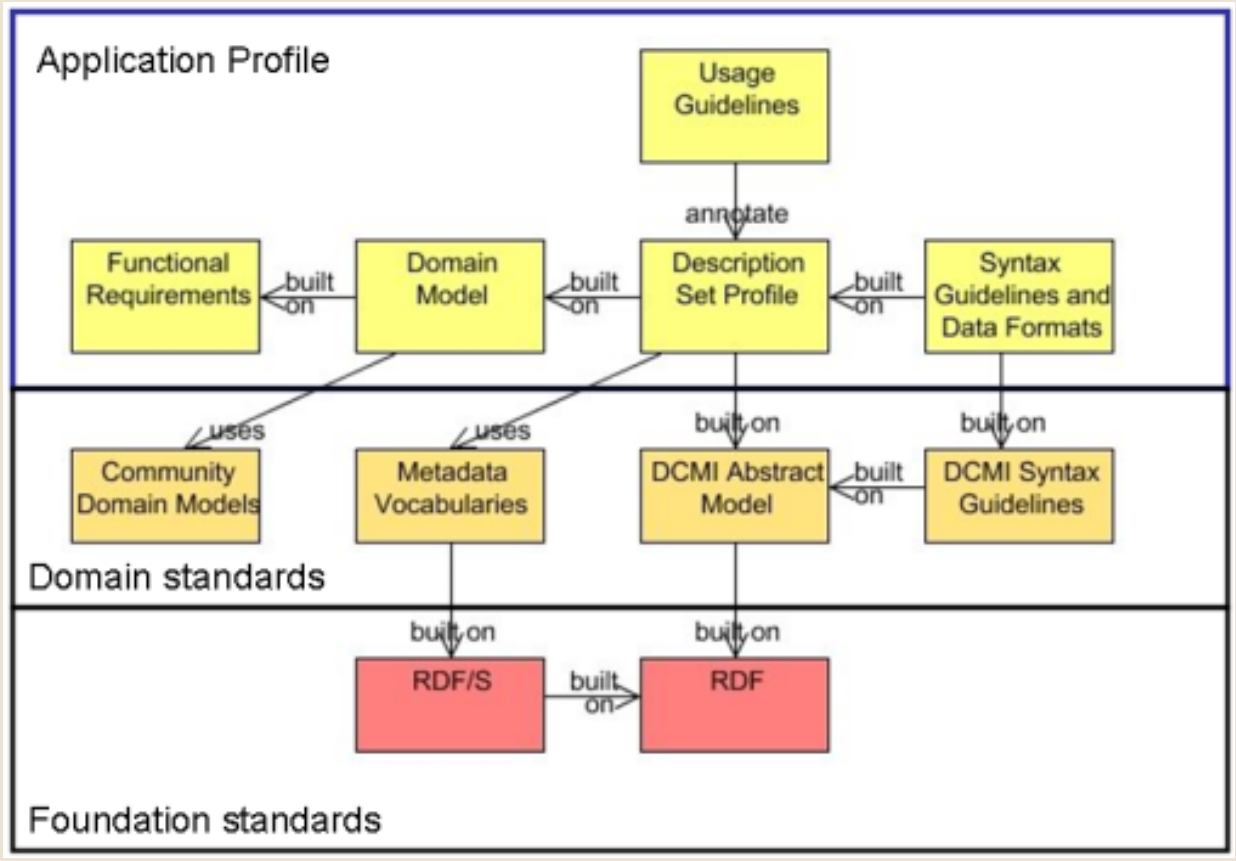


Nilsson, Mikael, Pete Johnston, Ambjörn Naeve, and Andy Powell. "The Future of Learning Object Metadata Interoperability." In: Harman, Keith and Alex Koohang (eds.). *Learning Objects: Standards, Metadata,* Repositories*, and LCMS*. Santa Rosa, California: Informing Science Press, 2007. http://kmr.nada.kth.se/papers/SemanticWeb/FutureOfLOMI.pdf

- DCAM potentially promotes interoperability by allowing for the building of effective *application profiles*

Singapore Framework

- Created at 2007 DC Conference
- No endorsed DC Application Profile exists yet that implements this framework



Whoa. What, now?

- Layers would allow communities to define their own needs but still base structures on the common abstract model
- Some possible benefits of the DCMI Abstract Model
  - Much easier metadata interoperability between systems
  - Less re-inventing the wheel in multiple places
  - Increased utility of library metadata in non-library environments
  - Better integration of authority data into bibliographic discovery systems
- But setting it all up is a *lot* of work!

Is this really going to work?

I’m afraid I don’t know.

MODS elements in DC Application Profiles?

- MODS terms suggested by the DCMI Usage Board in 2002 for the DC Libraries Application Profile
- 2 issues arose as the DCAM evolved:
  - The terms suggested were not MODS top-level elements and not directly addressable via URI (solvable?)
  - MODS "elements" ≠ DC "elements"
    - DC "element" really a "property"
    - MODS subelement values have shades of meaning affected by parent elements and parent element attribute values
- Is this really a problem?
  - Can MODS be understood in terms of the DCAM? Or MODS elements as "properties"?
  - Or is the difference fundamental in the XML vs. RDF approach?
  - And is the distinction meaningful to those who would use MODS?
  - Current DCMI position is that this is not allowable

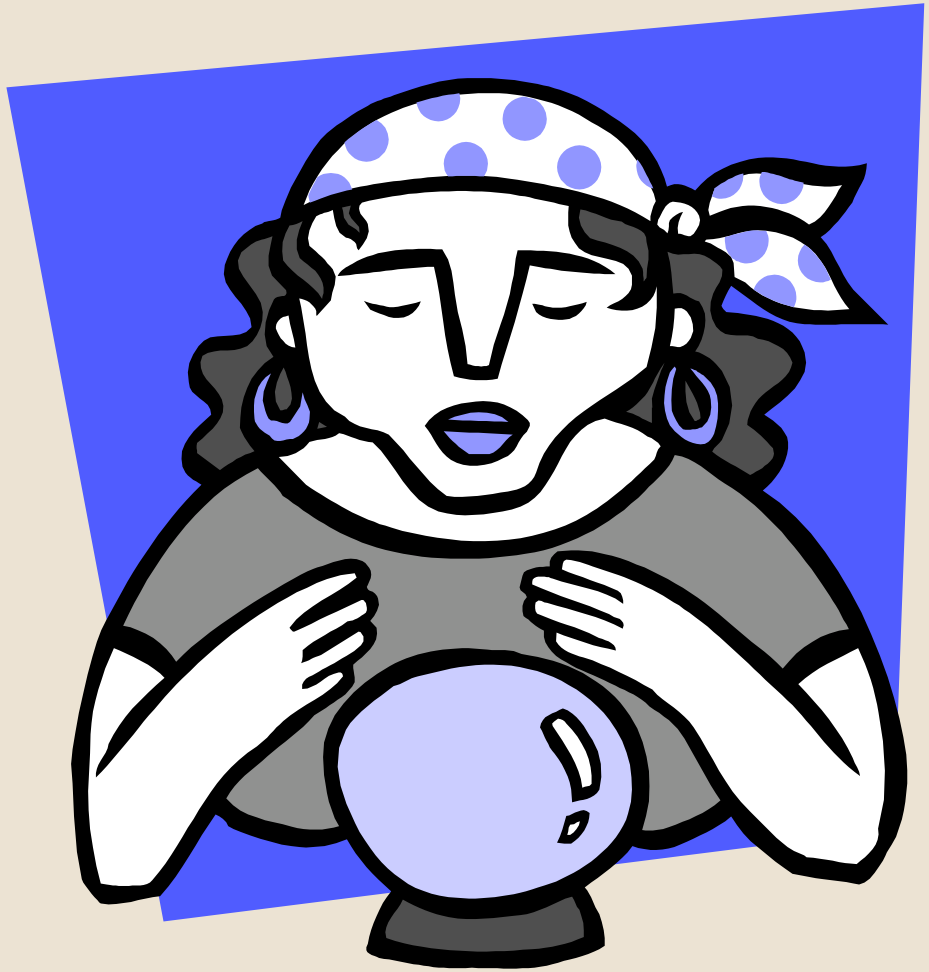
RDA/DCMI Task Group

- Attempting to facilitate utility of library-generated data in DCAM-focused applications
- Goal: “To define components of the draft standard "RDA - Resource Description and Access" as an RDF vocabulary for use in developing a Dublin Core application profile.”
  - Define RDA modeling entities as an RDF vocabulary (properties and classes).
  - Identify in-line value vocabularies as candidates for publication in RDFS or SKOS.
  - Develop a DC Application Profile for RDA based on FRBR and FRAD.
- Vocabularies being defined in the NSDL Metadata Registry
- It is still unclear where responsibility will lie in the long term, and what role the registry will play in the production version of RDA

Should libraries care about this?

- Yes, if just to be aware
  - Because this *could* be the prevailing model in the future
  - Although that's far from clear right now
- It depends, on…
  - How RDA and the RDA/DCMI vocabulary registry are received
  - How quickly metadata creation systems in libraries develop infrastructure to support making these distinctions
  - If we can overcome the terminological challenges currently separating the two communities
  - How effective mashups of library and non-library data are in the short- and medium-term, sparking interest in this area
  - How quickly Semantic Web-style applications emerge that can make good use of this data
  - How the balance between intelligence in data and intelligence in applications goes over the next few years

Do you have one? Can I borrow it?



Thank you!

- Questions?
- For more information:
  - DCMI Abstract Model home page: <http://dublincore.org/documents/abstract-model/>
  - These presentation slides: <http://www.dlib.indiana.edu/~jenlrile/presentations/bbspr09/dcam/dcmi-am.ppt>
  - Today's handout: <http://www.dlib.indiana.edu/~jenlrile/presentations/bbspr09/dcam/handout.pdf>
  - jenlrile@indiana.edu