Dublin Core: DCAM, Syntax, and Semantics (part two)

DC-2010 INTERNATIONAL CONFERENCE ON DUBLIN CORE AND METADATA APPLICATIONS
Pittsburgh, PA, October 20, 2010

Jon Phipps, Lead Scientist Internet Strategies
JES & Co.
jonp@jesandco.org
In the beginning was
The Statement
The Scope of This Tutorial

Traditionally, at this stage we've talked about 'encodings'

But we need to go back further, and start with statements ... ...and how they're aggregated into records, and how the information gets managed, and shared, and maintained ...

The goal is to provide context for thinking about these issues, not a 'cookbook' to build implementations
Models

XML vs. RDF
Two Models

We're going to discuss two 'generic', but very different models:

- RDF
  Resource Description Framework
- XML
  Extensible Markup Language
Why...

are we talking about XML at all?

- Many of us 'think' in XML
- RDF often looks like XML (RDF/XML sure does)
- Many of us think RDF is a flavor of XML

but...

RDF is a data model
XML is a document model
XML vs. RDF

Syntax vs. Semantics
Two Models of 'the world'

• XML assumes a 'closed' world (domain), usually defined by a schema:
  "We know all of the data describing this resource. The single description must be a valid document according to our schema. The data must be valid."

• RDF assumes an 'open' world:
  "There's an infinite amount of unknown data describing this resource yet to be discovered. It will come from an infinite number of providers. There will be an infinite number of descriptions. Those descriptions must be consistent."
Two Models of 'the world'

- XML's document model provides a neat equivalence to a metadata 'record'
- RDF's statement-oriented data model has no notion of 'record'
Syntax vs. Semantics

Structure vs. Meaning
Syntax vs. Semantics -- XML

- Meaning is conveyed by the schema.
- The schema defines a document (record).
- A statement's inclusion in a record that conforms to the schema conveys the meaning (semantics)

"This document describes this 'thing' because I have determined that this thing is a member of the class of things that this document describes. This class of things have the properties defined by the schema that defines this document."
Thing?

aka 'resources', 'non-literals'...
have a distinct identifier (URI)

• RDF
  "anything addressable via a URL."
• OWL:Thing
  "Every 'individual' in the (OWL) world is a member of the 'class' owl:Thing."
• DC
  "Anything that might be identified... Not all resources are network 'retrievable' e.g. human beings"
Syntax vs. Semantics -- RDF

- Meaning is conveyed by the 'name' of the property used in the statement.
- If the 'name' is a web protocol URI (HTTP), then the URI will 'resolve' to an 'ontology' that will convey the semantics directly.
- The ontology may define a 'domain' for the property -- the class of things that this property describes.

"This thing is a member of the class of things that this property defines, because someone, somewhere, has published a statement describing this thing as having this property."
Properties are also 'things' (resources) and the 'name' of a property thing is a URI.

Many people think the URI 'name' should convey the meaning of the property.
<http://purl.org/dc/elements/1.1/title>

Sometimes this leads people astray.
<dc:title>Mr.</dc:title>
Resolvable?

Sometimes called 'machine-actionable'... if used as a web (http) address, a web server will return some document (information resource) that describes the resource identified by the URI.

In the case of a property URI, this resource would be an ontology or RDF Schema that describes the 'meaning' of that property.
Ontology?

"In computer science and information Science, an ontology is a formal representation of knowledge as a set of concepts within a 'domain', and the relationships between those concepts. It is used to ‘reason’ about the entities within that domain, and may be used to describe the domain."

--Wikipedia
'Statements' exist solely in the context of a metadata 'document' where property value pairs are aggregated but must be understood in the context of the document, which includes the identifier for the resource described.

*Examples from: http://www.loc.gov/standards/marcxml/Sandburg/*
XML Statement

A DC alternative...

```xml
<?xml version='1.0' ?>
<dc xmlns="http://purl.org/dc/elements/1.1/">
  <creator>Sandburg, Carl, 1878-1967.</creator>
</dc>
```
XML Statement

But what exactly is being described?

<?xml version='1.0' ?>
<dc xmlns="http://purl.org/dc/elements/1.1/">
  <creator>Sandburg, Carl, 1878-1967.</creator>
</dc>

You need the whole record to know that (maybe).
RDF Statement

A 'triple'...

<subject><predicate><object>

OR

<resource><property><value>
RDF Statement

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about="http://lcnn.loc.gov/92005291">
        <dcterms:creator resource="http://authorities.loc.gov/cgi-bin/Pwebrecon.cgi
            ?AuthRecID=489180&v1=1&HC=1
            &SEQ=20101016101536
            &PID=0pZq1Sd2ZEWxYLAD0VJFJMzF4Spy"/>
    </rdf:Description>
</rdf:RDF>

(RDF/XML)
OR...
RDF Statement

<http://lccn.loc.gov/92005291>  <http://purl.org/dc/terms/creator>  <http://authorities.loc.gov/cgi-bin/Pwebrecon.cgi?AuthRecID=489180&v1=1&HC=1&SEQ=20101016101536&PID=0pZq1Sd2ZEWxYLAB0VjJFGMzF4Spy>

(Raw Triple)
OR...
RDF Statement

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. @prefix dcterms: <http://purl.org/dc/terms/>.
<http://lccn.loc.gov/92005291> dcterms:creator <http://authorities.loc.gov/cgi-bin/Pwebrecon.cgi?AuthRecID=489180&v1=1&HC=1&SEQ=20101016101536&PID=0pZq1Sd2ZEWxYLAB0VjJFGMzF4Spy>
(Notation3)
RDF Statement

No document (record) context needed to convey the "meaning" (semantics) of the statement

Emphasis is on the independence of the statement because the URI for the thing that the statement is describing is the explicit subject of the statement.
Quality Control

What is metadata quality?
  • XML
    Does it conform to schema?
  • RDF
    Is it consistent with a definition of 'true'?

Need to change the context of the discussion about quality and differentiate between what humans and machines see and do in response to good and bad quality.

Not the same idea of quality as in the context of MARC data, though that is closer to XML notions (with similar limitations for machine validation)
Metadata 'records'

- **XML**
  predefined, limited collections of schema-conformant statements

- **RDF**
  limitless aggregations of independent, ontology-defined statements
XML

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about="http://dublincore.org/"
  
  <dc:title>Dublin Core Metadata Initiative - Home Page</dc:title>
  <dc:description>The Dublin Core Metadata Initiative Web site.</dc:description>
  <dc:date>1998-10-10</dc:date>
  <dc:format>text/html</dc:format>
  <dc:language>en</dc:language>
  <dc:contributor>The Dublin Core Metadata Initiative</dc:contributor>
  <dc:contributor xml:lang="fr">L'Initiative de métadonnées du Dublin Core</dc:contributor>
  <dc:contributor xml:lang="de">der Dublin-Core Metadata-Diskussionen</dc:contributor>
</rdf:Description>
</rdf:RDF>
Metadata 'records'

RDF

<table>
<thead>
<tr>
<th>Subject</th>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
</table>

Here's the statements (that we know at the moment).

Where's the record?
Data Validation

• XML
  Constraints and restrictions (cardinality, range, datatype) enforced by domain-specific schema, and structural constraints (well-formed). Data is either valid or not.
• RDF
  Logical constraints and restrictions (domain, cardinality, range, datatype) control what can be logically inferred. All data is valid. Some data may be illogical (inconsistent), especially in a global, 'Open World' context.
Creating Metadata

...is the primary place to exert control of conformance to a structural, syntactic domain model as a measure of metadata quality.

- **XML**
  - Has input restrictions for validity
  - Schema conformance for testing validity

- **RDF**
  - Can't be validated -- all data is 'valid'
  - Quality (truthiness) must be judged in aggregation
  - Some data may be globally inconsistent with some predefined notion of 'true'. Or not.
Publishing Metadata

...is the primary place to communicate the semantics of your domain model.

- XML
  - Easy to publish
  - Semantics must be implied by structure

- RDF
  - Easy to publish
  - Explicit semantics
Distributing Metadata

- Think globally -- anyone (any system) can make any statement, about any thing, at any time.
- The Global Web of Data will aggregate it all
- Provenance* -- 'who' made the statement, and when -- will be increasingly important

*http://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance
Consuming Metadata

...is the primary place to 'validate' the semantics of your domain model.

• XML
  o Consumer must agree to validate against schema
  o Aggregation can be a multi-schema struggle
  o Data loses meaning when removed from structural context

• RDF
  o Data is inherently valid (*this is good?*)
  o Designed for easy aggregation by consumers
  o Each statement can be 'understood' in isolation
Aggregating Metadata

- **XML**
  - Easy if everyone uses the same schema (shared meaning)
  - Hard when dealing with multiple schema (domain meaning)
- **RDF**
  - Each statement contains and transfers meaning
  - Domain meaning well supported and transferable
DCAM

DCMI Abstract Model
Dublin Core Abstract Model (DCAM)

- A conceptual model for Dublin Core metadata
- Initial work begun in 2003
- Recommendation in 2005
- Revised recommendation in 2007

In the process of being reviewed at this meeting
Why DCAM?

Officially...

- specify the components and constructs used in DC metadata
- describes how those components are combined to create information structures
- provides an information model independent of any particular encoding syntax.
Why DCAM?

Suggest/specify a document (record) model for RDF
DCAM Description Set

"The DCMI Abstract Model defines an "abstract syntax" based on a data structure it calls the "description set". The Expressing Dublin Core metadata using the Resource Description Framework (RDF) specification defines a mapping from the "description set" model to the RDF abstract syntax."
DCAM Resource Model
Description Set

...is made up of one or more descriptions, each of which describes one resource

```xml
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:foaf="http://xmlns.com/foaf/0.1/
         xmlns:gr="http://purl.org/goodrelations/v1/
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about="http://example.org/book">
        <dc:title>Antlers in the Trees</dc:title>
    </rdf:Description>
    <rdf:Description rdf:about="http://example.org/author">
        <foaf:name>Hoogoos Damoos</foaf:name>
    </rdf:Description>
    <rdf:Description rdf:about="http://example.org/publisher">
        <gr:BusinessEntity>Phantastic Stories</gr:BusinessEntity>
    </rdf:Description>
</rdf:RDF>
```
DCAM Description Set

• Each described resource is described using one or more property-value pairs.
• Each property-value pair is made up of one property and one value.
• Each value is either a literal value or a non-literal value:
  o A literal value is a value which is a literal.
  o A non-literal value is a value which is a physical, digital or conceptual entity.
• A literal is an entity which uses a Unicode string as a lexical form, together with an optional language tag or datatype, to denote a resource (i.e. "literal" as defined by RDF [RDF]).
DCAM Resource Model
DCAM Resource Model
Literals?

aka strings, value strings

• May be 'typed'
  "A value string with an associated syntax encoding scheme URI."
  <dct:modified xsi:type="dct:W3CDTF">1945-03-15</dct:modified>

• or 'untyped' (plain)
  "A value string without an associated syntax encoding scheme URI."
  <dc:modified xml:lang="en">March 15, 1945</dc:modified>
DSP

Description Set Profiles
DSP Basic Structure

A DSP describes the structure of a Description Set by using the notions of "templates" and "constraints". A template describes the possible metadata structures in a conforming record. There are two levels of templates in a Description Set Profile:

- **Description templates**, which contain the statement templates that apply to a single kind of description as well as constraints on the described resource.
- **Statement templates**, which contain all the constraints on the property, value strings, vocabulary encoding schemes, etc. that apply to a single kind of statement.
Description Templates

see http://dublincore.org/documents/dc-dsp/#sect-3

• What class of resource is being described
• Which properties are allowed to describe this class of resource
Statement Templates

see http://dublincore.org/documents/dc-dsp/#sect-3
• statement (property) cardinality and allowed values
• how to validate the data on input (xml schema, javascript, form validation)
• how to express the semantics for redistribution in an open world i.e. how can you tell if this data make sense (xml schema, rdf schema, owl ontology)
DSP Templates

...represent an abstraction that can be expressed as:

- Data validation restrictions (XML)
- Logical inference constraints (RDF)
Restriction vs. Inference 'constraints'

A statement template can be used...

- as the basis of an XML schema -- "how to validate the data on input"
- and an OWL(2) ontology -- "how can you tell if this data makes sense" and what data can/must be logically inferred
Application Profiles (AP)

Where's the value?

• Documentation (one aspect of quality)
• Basis for creation of metadata emphasizing predictability, consistency, validity
• Basis for correct interpretation and reuse of metadata
The Singapore Framework (SF)
Domain Model

Be careful about adopting other people's domain models!

Hint: They're *other people's* (organization's) domain models.
Speaking of 'domain'

In RDF the 'domain' of a property identifies the class of things that property describes:

```xml
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Description
    rdf:about="http://RDVocab.info/Elements/bookFormatManifestation">
    <rdfs:label xml:lang="en">Book format (Manifestation)</rdfs:label>
    <rdfs:subPropertyOf
    <rdfs:domain
      rdf:resource="http://RDVocab.info/uri/schema/FRBRentitiesRDA/Manifestation"/>
  </rdf:Description>
</rdf:RDF>
```

Applying this property to a resource will infer that the resource is a member of the class 'FRBRentitiesRDA/Manifestation'. 
Speaking of 'domain'

In XML the 'domain' of a property says that the property may only be used when describing a member of the class 'FRBRentitiesRDA/Manifestation'.
So...

DCMI: Provides a suite of recommendations based on...

DCAM: A document model for RDF
Description Sets
Description Set Profiles
Application Profiles: A documentation framework for metadata
Thank You!

Questions?

jonp@jesandco.org