Metadata

How do we?

- Enable discovery of material
- Enable harvesting of material by external systems
- Organize content
- Support archiving and preservation

Metadata

- *Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use or manage an information resource.*
  - NISO (2004) Understanding Metadata
- Used for
  - Organization and management of content
  - Support discovery
  - Direct content in channels
  - Enable automated discovery/manipulation

What is Metadata?

- Structured data about “something”
  - Text
  - Images
  - Sound,
  - Movement
  - Objects
  - Events
  - Services

What is Metadata

- Encountered every day
  - Timetables
  - Directories
  - Internet shopping sites, etc.

What IS Metadata?

Data ‘reporting’

- **WHO** created the data?
- **WHAT** is the content of the data?
- **WHEN** was it created?
- **WHERE** is it geographically?
- **WHY** was the data developed?
- **HOW** was the data developed?

Who?

- **Name of the author**
- **Contact person**
- **Email address**
- **Copyright owner**

- **Who is that person?**
- **Who is this about?**
- **Who created this metadata?**
- **Who identified the organism?**

What?

- What is this about?
- What species is it?
- What part is it?
  - Leaf, root, fruit, flower, cell…
- What type of view?
  - Dorsal, ventral, lateral
  - Close-up, microscopic, in situ
  - Cross-section, stain…
- What is it doing?
  - Behaviors, interactions…

When?

- **When was it made?**
- **When was it collected?**
- **When was it processed?**
- **When did you answer all of these questions?**

Where?

- Where was it made?
- Where was it collected?
- Where was it processed?
- What coordinate system are you using?
- How precise are your coordinates?

Why?

- **Why was it made?**
- **Why was it made that way?**

How?

- How was it made?
  - Equipment used, special techniques employed
- How was it (the object of the photo) prepared?
  - What stains?
  - What sectioning?
  - In situ versus studio?
  - Spontaneous or staged?

Metadata is

- Stored in
  - Databases, repositories
  - Webpages
- Carriers
  - Formats (MARC – machine readable cataloging)
  - Markup languages (e.g. HTML, SGML, XML)

Types of Metadata

- Descriptive
- Structural
- Administrative
  - Rights management metadata
  - Preservation metadata

Types of Metadata We Have Already Seen

- File headers
- Database schema
- Are these interoperable?

Dublin Core

- Metadata to improve information retrieval of internet resources
- Developed predominantly by the bibliographic community. Elements similar to bibliographic surrogate

Dublin Core

- Dublin Ohio, 1995
- Core – concept of a ‘core’ metadata elements set
- The core set can be expanded

Characteristics of Dublin Core

- Simplicity
- Semantic Interoperability
- International Consensus
- Extensibility
- Metadata Modularity on the Web

Unqualified Dublin Core is the *Pidgin* metadata language

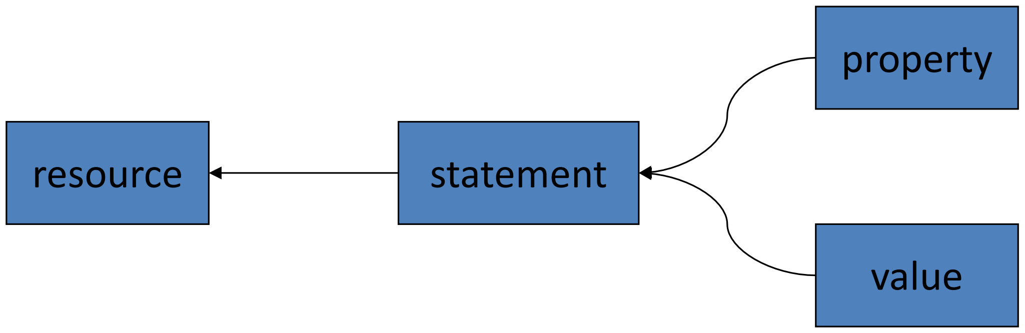
- Metadata is language
- Dublin Core is a small and simple language -- a pidgin -- for finding resources **across domains** using the internet**.**
- Speakers of different languages naturally "pidginize" to communicate

A Grammer of Dublin Core

- By design not as rich as mother tongues, but easy to learn and useful in practice
- Pidgins: small vocabularies (Dublin Core: fifteen special nouns and lots of optional adjectives)
- Simple grammars: sentences (statements) follow a simple fixed pattern...

Basic Structures in Dublin Core Metadata

- The basic unit of metadata is a *statement*:
  - *Statements* consist of a *property* (a metadata element) and a *value*
  - Metadata *statements* describe *resources*



What is a resource?

- W3C definition:
  - "anything that has identity… electronic document, an image, a service"
  - "not all resources are network retrievable; e.g. human beings, corporations, and bound books can also be considered *resources*"
- In other words, a *resource* is *anything* we can identify:
  - Physical things (books, people, airplanes….)
  - Digital things (Images, web pages, services….)
  - Concepts (colors, subjects, eras, places)
- In the DC context, the DCMI Type list describes the *stuff* we describe with DC metadata

Resource types for which DC is often used

DCMI *TYPE* Vocabulary

Collection	Dataset	Event
Image	Interactive Resource	Moving Image
Physical Object	Service	Software
Sound	Still Image	Text

Dublin Core Elements

Content

- Coverage
- Description
- Type
- Relation
- Source
- Subject
- Title

Intellectual Property

- Contributor
- Creator
- Publisher
- Rights

Dublin Core Elements

Instantiation

- Date
- Format
- Identifier
- Language

Core elements to describe content?

- Title: The name given to the resource.
- Description: An account of the content of the resource. Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
- Subject: The topic of the content of the resource. Typically, a subject will be expressed as keywords or key phrases or classification codes that describe the topic of the resource.
- Coverage: The extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity).
- Source: A reference to a resource from which the present resource is derived. The present resource may be derived from the Source resource in whole or part.
- Type: The nature or genre of the content of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content.
- Relation: A reference to a related resource.

Core elements do describe Intellectual Property

- ***Creator***: An entity primarily responsible for making the content of the resource. Examples of a Creator include a person, an organization, or a service.
- ***Contributor***: An entity responsible for making contributions to the content of the resource. Examples of a Contributor include a person, an organization or a service. Typically, the name of a Contributor should be used to indicate the entity.
- ***Publisher***: The entity responsible for making the resource available. Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
- ***Rights***: Information about rights held in and over the resource. Typically a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

Core Elements for "instantiation" of the resource:

- Language: A language of the resource. Recommended best practice is to use a controlled vocabulary such as ISO 639-2. Example: "eng" for English.
- Identifier: An unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string conforming to a formal identification system, such as an ISBN.
- Date: A date associated with the creation or availability of the resource. Recommended best practice is defined in a profile of ISO 8601 that includes (among others) dates of the forms YYYY and YYYY-MM-DD.
- Format: The file format, physical medium, or dimensions of the resource. Examples of dimensions include size and duration. Recommended best practice is to use a controlled vocabulary such as the list of Internet Media Types [MIME]. Example: image/jpeg.

So two instantiations of the same resource will require two different metadata records!!

Examples: you have a text in both French and English; or an image of a museum object as well as a 3-D video of it… or two versions of a scanned photograph.

Qualifiers and Domain-specific Extensions

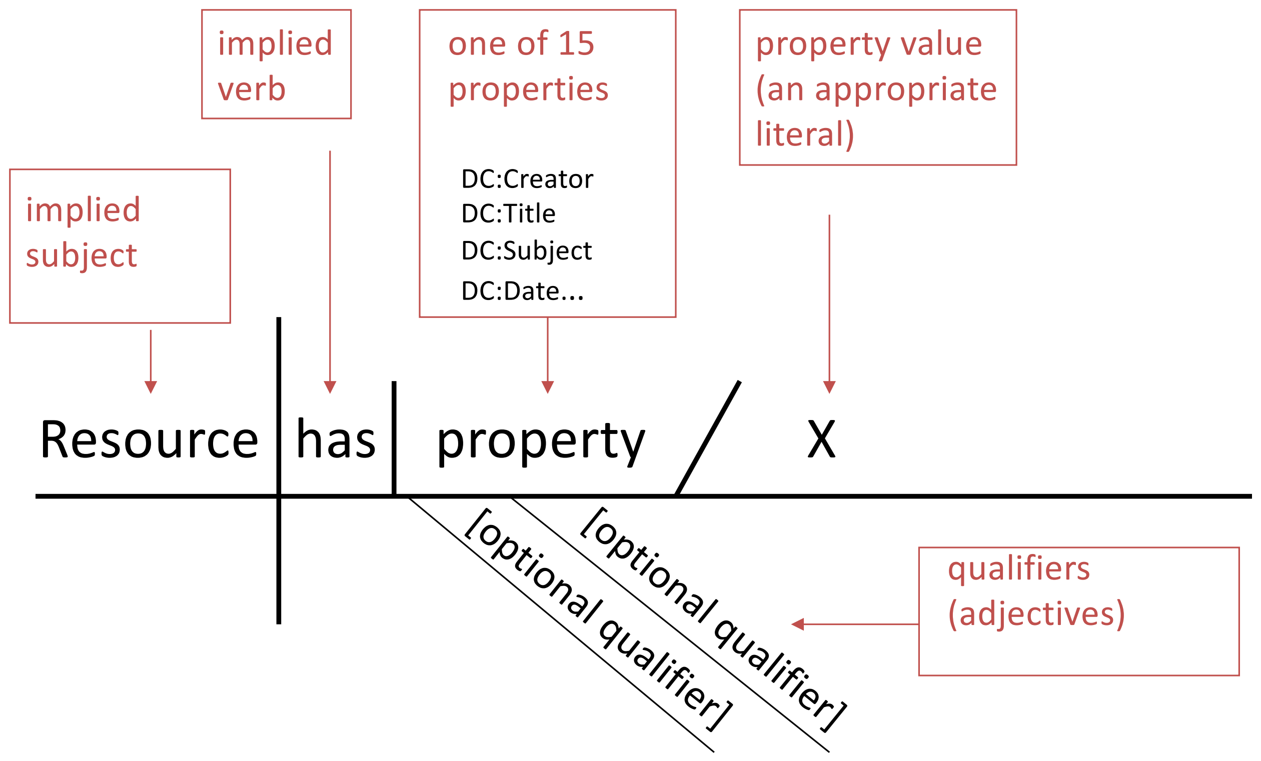
- The Dublin Core architecture supports more sophisticated metadata solutions through the addition of:
  - Qualifiers
  - Domain-specific extensions
  - Application Profiles of involving mixed namespaces (more on this later)
- Increased sophistication comes at the cost of some degree of interoperability

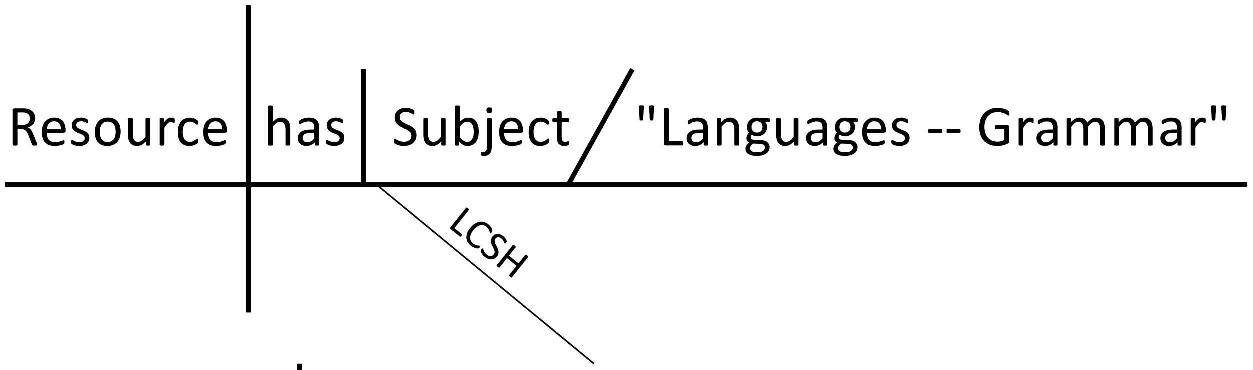
Varieties of Qualifiers: Value Encoding Schemes

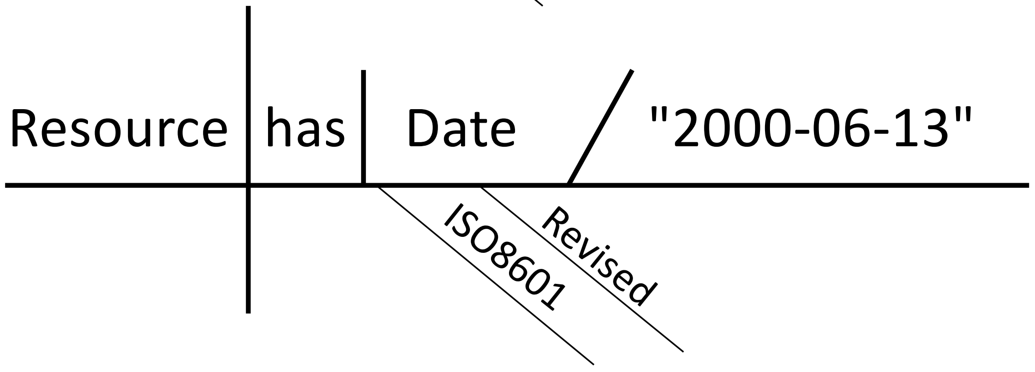
- Says that the value is
  - a term from a controlled vocabulary (e.g., Library of Congress Subject Headings)
  - a string formatted in a standard way (e.g., "2001-05-02" means May 2, not February 5)
- Even if a scheme is not known by software, the value should be "appropriate" and usable for resource discovery.

Varieties of qualifiers: Element Refinements

- Make the meaning of an element narrower or more specific.
  - a *Date Created* versus a *Date Modified*
  - an *IsReplacedBy Relation* versus a *Replaces* Relation
- If your software does not understand the qualifier, you can safely ignore it.


implied verb
one of 15 properties
DC:Creator
DC:Title
DC:Subject
DC:Date...
property value (an appropriate literal)
implied subject
Resource
has
property
X
[optional qualifier]
[optional qualifier]
qualifiers (adjectives)


Resource
has
Subject
"Languages -- Grammar"
LCSH


Resource
has
Date
"2000-06-13"
ISO8601
Revised

Some example qualifiers…

Type of Qualifier	Element	Example Qualifiers
Element Refinement	Description	Abstract, tableOfContents
	Coverage	Spatial, Temporal
	Date	Available, Created, dateCopyrighted, dateAccepted, dateSubmitted
	Relation	hasPart, hasVersion, isPartOf, isReferencedBy, isReplacedby, isVersionOf
Encoding Schemes	Subject	DDC (Dewey Decimal Classification), LCC (Library of Congress Classification), LCSH (Library of Congress Subject Headings), MESH (Medical Subject Headings)…
	Language	ISO639-2 (such as eng, for English), RFC1766 (such as en-us for US English)
	Date	W3CDTF (such as 1997-12-04 for 4 Dec. 1997)
	Type	DCMIType, such as: Collection, Dataset, Event, Image, InteractiveResource, MovingImage, PhysicalObject, Service, Software, Sound, StillImage, Text.

Dublin Core Principles

- Dumbdown
- The one-to-one principle
- Appropriate values

Dumb-Down Principle for Qualifiers

- The fifteen elements should be usable and understandable with or without the qualifiers
- Qualifiers refine meaning (but may be harder to understand)
- Nouns can stand on their own without adjectives
- If your software encounters an unfamiliar qualifier, look it up -- or just ignore it!

The One to One Principle

- Each resource should have one metadata description
  - For example, do not describe a digital image of the Mona Lisa as if it were the original painting
- Group Related descriptions into description sets
  - Describe an artist and his or her work separately, not in a single description

Example of “dumb down”

- ***Example***: **“dcterms:temporal” refines “dc:coverage.”**

```
<dcterms:temporal>
    The Great Depression, 1929-1939
</dcterms:temporal>
```

becomes

```
<dc:coverage>
    The Great Depression, 1929-1939
</dc:coverage>
```

- There is nothing to indicate now that it is a time period…but it’ s still “coverage”

Appropriate Values

- Use elements and qualifiers to meet the needs of your local context, but . . .
- Remember that your metadata maybe interpreted by machines and people, so . . .
- Consider whether the values you use will aid discovery outside your local context and . . .
- Make decisions about your local practices accordingly

Example xml record

Created by Melanie Feltner-Reichert for the Arrowmont Project at UT Libraries

```
<record xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/">
     <dc:title> An Impression of the Pi Beta Phi Settlement School and its Vicinity</dc:title>
     <dc:creator> Moorehead, Rosemary</dc:creator>
     <dcterms:dateCreated> 1936-11-07</dcterms:dateCreated>
     <dc:type> Mixed material</dc:type>
     <dc:type> Scrapbook</dc:type>
     <dc:format xsi:type="dcterms:IMT"> image/jpeg</dc:format>
     <dc:description> Scrapbook on the Pi Beta Phi Settlement School and Gatlinburg area.
            Compiled by Pi Beta Phi teacher Rosemary Moorehead. </dc:description>
     <dc:subject> Student life</dc:subject>
     <dc:subject> Settlement schools </dc:subject>
     <dc:subject> Gatlinburg, Tennessee</dc:subject>
     <dc:subject> Pi Beta Phi Settlement School</dc:subject>
     <dc:subject> Expansion and Growth of Pi Beta Phi Settlement School, 1928-1943</dc:subject>
     <dc:source> Great Smoky Mountains Regional Project, The University of Tennessee
            Libraries. </dc:source>
     <dc:rights> For rights relating to this resource, visit:
    http://idserver.utk.edu/?id=200500000001941</dc:rights>
     <dc:identifier> rms00000</dc:identifier>
     <dc:relation> From Pi Beta Phi to Arrowmont:
    http://idserver.utk.edu/?id=200500000001049</dc:relation>
    <dc:language xsi:type="dcterms:ISO639-2"> eng</dc:language>
</record>
```

Using DC with other vocabularies

- Specialized *application profiles* may need to:
  - Use **general-purpose** Dublin Core elements
  - Use elements from another, more **domain-specific** standard
  - **Narrow standard definitions** of DC elements for specific local uses
  - **Invent local elements** outside the scope of existing standards

What is an Application Profile?

- A metadata schema incorporating a set of elements from one or more metadata element sets
- A set of policies defining how the elements should be applied to the domain of the application
- A set of guidelines that make the policies concerning elements explicit

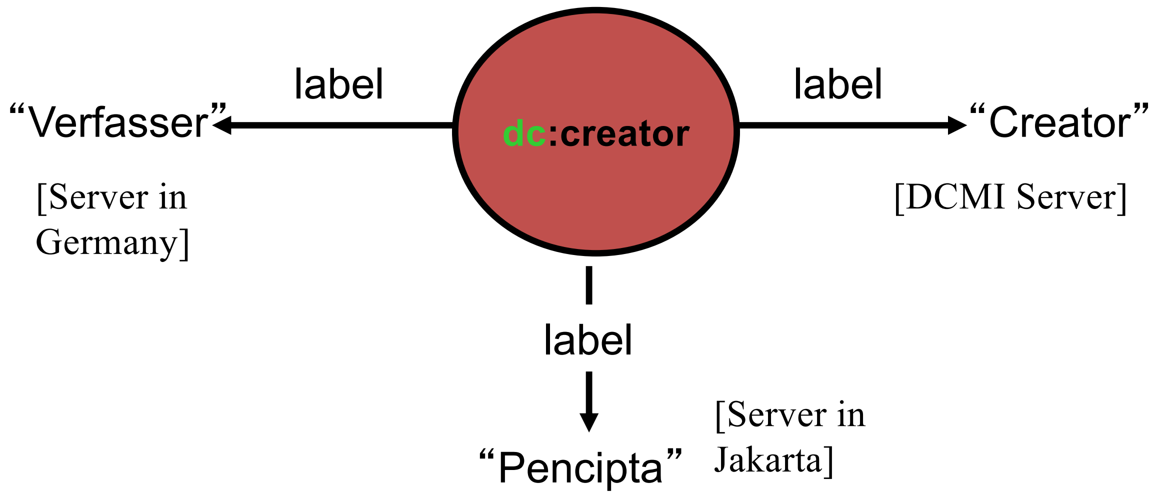
Multiple Namespace Fragment

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:co="http://purl.org/rss/1.0/modules/company/"

<dc:publisher>The O'Reilly Network</dc:publisher>
<dc:creator>Rael Dornfest</dc:creator>
<dc:rights>Copyright &#169; 2000 O'Reilly &amp; Associates,
           Inc.</dc:rights>
<dc:date>2000-01-01T12:00+00:00</dc:date>
<dc:description> XML is placing increasingly heavy loads on the existing
                 technical infrastructure of the Internet. </dc:description>

<co:name>XML.com</co:name>
<co:market>NASDAQ</co:market>
<co:symbol>XML</co:symbol>
```

Namespaces and Translation

- Dublin Core has been translated into 26 languages
  - machine-readable tokens are shared by all
  - human-readable labels are defined in different languages
  - translations are distributed, maintained in many countries
  - eventually linked in DCMI registry

One token – labels in many languages



Metadata languages are "multilingual"

- Metadata is not a spoken language
- The words of metadata -- "elements" -- are symbols that stand for concepts expressible in multiple natural languages
- Standards may have dozens of translations
- Are concepts like "title", "author", or "subject" used the same way in English, Finnish, and Korean?

Syntax Alternatives HTML… XML… RDF/XML

- Three Web-based models for deploying metadata
- Each has advantages and disadvantages
- What is ‘best’ depends on local constraints
  - What is the objective of the system? How do these syntax alternatives support local functional requirements?
  - Are there services and software to ‘consume’ the metadata created?
  - Are trained practitioners available to create and support the systems?

Syntax Alternatives: HTML

- Advantages:
  - Simple – META tags embedded in content
  - Widely deployed tools and knowledge
  - Resource carries its metadata around with it
  - Metadata is openly harvestable

Syntax Alternatives: HTML (continued)

- Disadvantages
  - Limited structural richness (does not support hierarchical, tree-structured data
  - Management of metadata is less reliable (the metadata is out in the wild)

- Describe one thing (the HTML document) and no more!

Dublin Core in HTML (example)

```
<head>
<link rel="schema.DC" href="http://purl.org/dc">

<meta name="DC.title"
   content=“DC Metadata Tutorial”
<meta name="DC.creator"
   content=“Stuart L. Weibel">
<meta name="DC.subject" xml:lang= “en-US’
   content=“Metadata">
<meta name="DC.date" scheme=“DCTERMS.W3CDTF"
   content=“2007-07-08">
<meta name=“DCTERMS.audience”
   content =“technical librarians”
</head>
<body>
… [ rest of html document ]
```

The namespaces for HTML encoding

- All DCMI terms (elements, element refinements, and encoding schemes) are found in:

  DCMI Metadata Terms

  http://dublincore.org/documents/dcmi-terms/

- The namespaces are a result of historical developments
  - DC: [original elements]
  - DCTERMS: [later elements]

Syntax Alternatives: XML

- XML = eXtensible Markup Language
- *The* standard for networked text and data
- Wide-spread tool support
  - Parsers are widely available
  - Extensibility (XML namespaces)
  - Type definitions (XML Schema)
  - Transformation and Rendering (XSLT)
  - Rich linking semantics (XLINK)

XML Schema

- Rich XML-based language for expressing data-type semantics
- Replaces arcane and limited DTD (origin in SGML)
- Facilities:
  - Data typing (both complex and primitive)
  - Constraints (ranges, cardinality…)
  - Defaults (specify defaults for certain properties)

Dublin Core fragment in XML

```
<metadata xmlns:dc="http://www.openarchives.org
   /OAI/dc.xsd">
   <dc:creator>Carl Lagoze</dc:creator>
   <dc:title>Accommodating Simplicity and
   Complexity in Metadata</dc:title>
   <dc:date>2000-07-01</dc:date>
   <dc:publisher>Cornell University,    Computer
Science</dc:publisher>
</metadata>
```

Where is the rest of the stuff? In the schema!

Syntax Alternatives: RDF

- RDF (Resource Description Format)
- Syntax expressed in XML
- W3C recommendation for encoding metadata (a semantic Web technology)
- Enabling technology for richly-structured metadata
- Rich data model (the DC Abstract Model is a constrained version of RDF)
- Metadata can be shared easily among independent applications that understand RDF
- W3C – Resource Description Framework (RDF)
  http://www.w3.org/RDF/

Resource Description Framework

- A framework (not a language) for describing resources
- Model for data
  - Metadata
- Syntax to allow exchange and use of information stored in various locations
- The point is to facilitate reading and correct use of information by *computers*, not necessarily by people

RDF Components

- RDF Model and Syntax WG
  - Formal data model
  - Syntax for interchange of data
- RDF Schema (RDFS)
  - Type system (schema model)

RDF Basics

- RDF is based on the idea of identifying resources using Web identifiers and describing resources in terms of simple properties and property values.
- To identify resources, RDF uses Uniform Resource Identifiers (URIs) and URI references (URIrefs).
- Definition: A resource is anything that is identifiable by a URIref.

Identification and description

- RDF identifies resources with URIs
  - Often, though not always, the same as a URL
  - Anything that can have a URI is a RESOURCE
- RDF describes resources with properties and property values
  - A property is a resource that has a name
    - Ex. Author, Book, Address, Client, Product
  - A property value is the value of the Property
    - Ex. "Joanna Santillo," http://www.someplace.com/, etc.
    - A property value can be another resource, allowing nested descriptions.

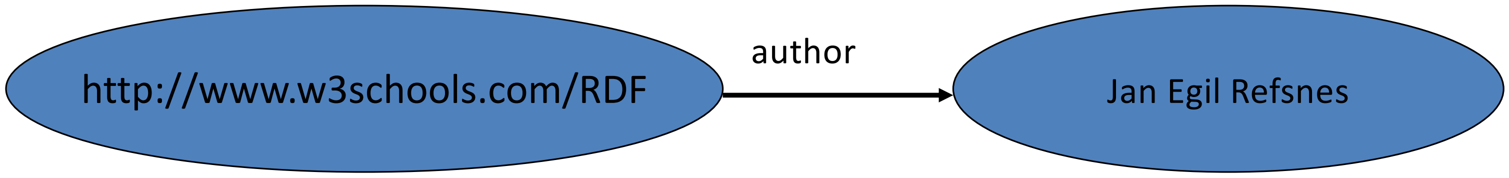
Statements

- Resource, Property, Property Value
- Aka subject, predicate, object of a statement
- Predicates are not the same as English language verbs.
  - Specify a relationship between the subject and the object

Binary predicates

- RDF offers only binary predicates.
- Think of them as P(x,y) where P is the relationship between the objects x and y.

- From the example,
- X = http://www.w3schools.com/RDF
- Y = Jan Egil Refsnes
- P = author



Examples

- Statement: "The author of http://www.w3schools.com/RDF is Jan Egil Refsnes".
- Subject: http://www.w3schools.com/RDF
- Predicate: author
- Object: Jan Egil Refsnes

- Statement: "The homepage of http://www.w3schools.com/RDF is http://www.w3schools.com".
- Subject: http://www.w3schools.com/RDF
- Predicate: homepage
- Object: http://www.w3schools.com

RDF Triples

- RDF statements can be written down using triple notation. In this notation, a statement is written as follows:
- subject predicate object .
- Example:
  - `<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740> .`
  - `<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .`
  - `<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .`
- Note: In this notation URIs are written out completely, in angle brackets.

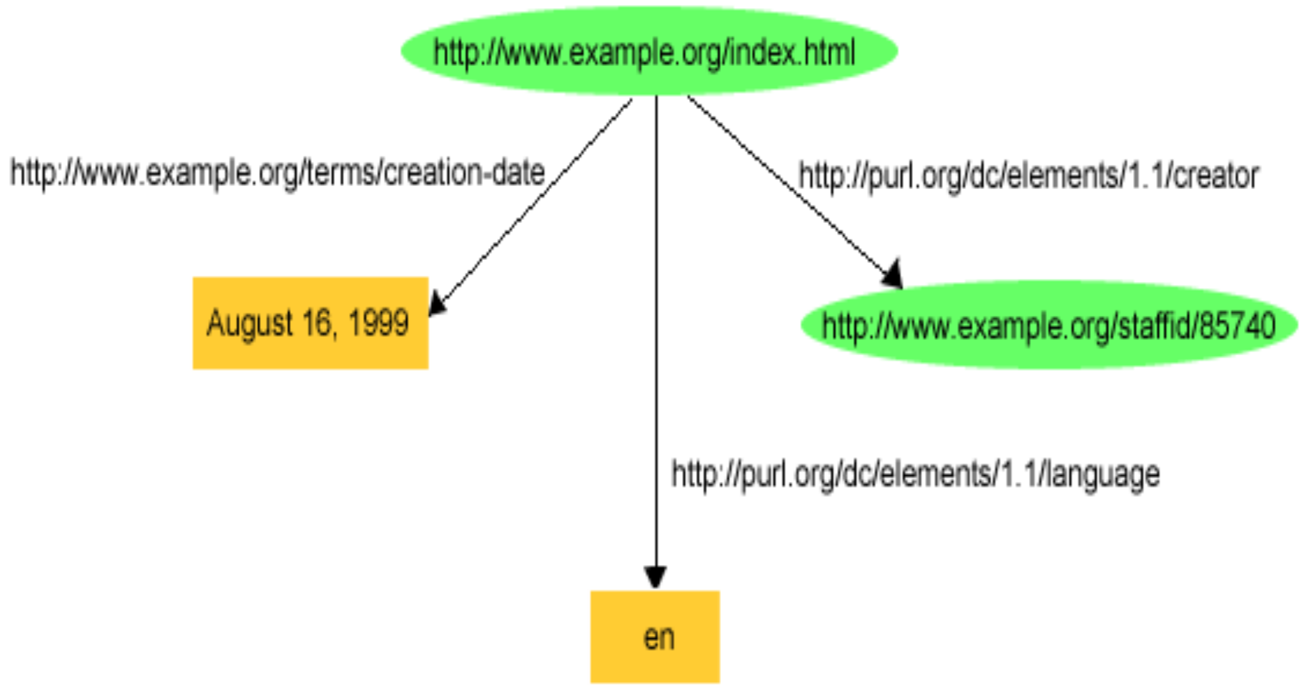
RDF Graphs

- Graphically, RDF models statements by nodes and arcs in a graph.
- In the RDF graph notation, a statement is represented by:
  - a node for the subject
  - a node for the object
  - an arc for the predicate, directed from the subject node to the object node.
- A node may be identified by a URIref or it can be a literal (it can also be a blank node; we will explain this later).
- An arc is identified by a URIref.
- Note: We will draw RDF graphs as directed graphs. But strictly speaking, directed graphs are not sufficient for capturing all of RDF (e.g., directed graphs assume that the sets of nodes and arcs are disjoint but RDF allows a property as a subject of a statement).

Example

- Consider the following statements:
  - `http://www.example.org/index.html` has a creation-date whose value is August 16, 1999.
  - `http://www.example.org/index.html` has a language whose value is English.

The RDF Graph of the Example


http://www.example.org/index.html
http://www.example.org/terms/creation-date
http://purl.org/dc/elements/1.1/creator
August 16, 1999
http://www.example.org/staffid/85740
http://purl.org/dc/elements/1.1/language
en

RDF and Related Data Models

- In terms of the relational model, an RDF statement is similar to a tuple in a relation called Triples or Graph with attributes Subject, Predicate and Object.
- In terms of first-order logic, an RDF statement is similar to an atomic formula
  - triple(subj,pred,obj)

where triple is a first-order logic predicate and subj, pred and obj are constants.

Example

- Consider the triple `ex:index.html dc:creator exstaff:85740 .`

  The predicate `dc:creator`, when fully expanded as a URIref, is an unambiguous reference to the "creator" attribute in the Dublin Core metadata attribute set, a widely-used set of attributes (properties) for describing a wide range of networked resources (see http://dublincore.org/documents/usageguide/).

  The writer of this triple is effectively saying that the relationship between the Web page and the creator of the page is exactly the concept identified by `http://purl.org/dc/elements/1.1/creator`.

  Another person familiar with the Dublin Core vocabulary, or who finds out what `dc:creator` means (say by looking up its definition on the Web) will know what is meant by this relationship. In addition, based on this understanding, people can **write programs to behave in accordance with that meaning** when processing triples containing the predicate `dc:creator`.

Machine Readable Formats for RDF

RDF has a number of machine readable formats:

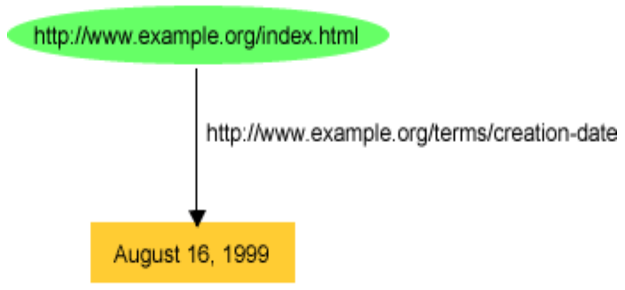
- XML/RDF
- Turtle (Terse RDF Triple Language)
- N3
- …

An XML Syntax for RDF: RDF/XML

- The conceptual model for RDF is a graph.
- RDF provides an XML syntax for writing down and exchanging RDF graphs, called RDF/XML.
- RDF/XML is the normative syntax for writing RDF.

Example

http://www.example.org/index.html has a creation-date whose value is August 16, 1999.


http://www.example.org/index.html
http://www.example.org/terms/creation-date
August 16, 1999

Example in XML/RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:exterms="http://www.example.org/terms/">
     <rdf:Description rdf:about="http://www.example.org/index.html">
        <exterms:creation-date>August 16, 1999</exterms:creation-date>
     </rdf:Description>
</rdf:RDF>
```

RDF Schemas (RDFS) -- W3C standard

- A **dictionary format for metadata terms:**
  - Simple XML format for namespaces, terms and definitions
- Example: "Title" (Dublin Core)
  - Human-readable label and definition:
    - Title: A name given to the resource.
  - Unique, machine-readable identifiers
    - dc:title
- Support for **cross-references**
  - Between multiple language renditions of a namespace
  - between terms in related standards
  - between local adaptations and related standards

RDF Schemas

- Declaration of vocabularies
  - properties defined by a particular community
  - characteristics of properties and/or constraints on corresponding values
- Schema Type System - Basic Types
  - Property, Class, SubClassOf, Domain, Range
  - Minimal (but extensible) at this time
  - minimize significant clashes with typing system designed for XML Schema WG
- Expressible in the RDF model and syntax