Tutorial 1: Basic Semantics

Stuart A. Sutton
University of Washington
DC-2007, Singapore
27 August 2007

Tutorial schedule

- Tutorial 1: Basic Semantics (Stuart Sutton)
- Tutorial 2: Basic Syntaxes (Mikael Nilsson)
- Tutorial 3: Vocabularies (Alistair Miles)
- Tutorial 4: Application Profiles (Diane Hillmann)

What's covered in this tutorial

- An introduction to metadata
- Key features of the Dublin Core
- Dublin Core metadata in broader context
- Important aspects of the DCMI community

What do we mean by “Dublin Core”?

1. An international **community** interested in interoperable metadata (DCMI)
2. New **ways of thinking** about interoperable metadata
   - DCMI
   - Application Profiles
3. A **core set of 15 metadata elements**
   - ISO Standard 15836-2003 (February 2003)
   - NISO Standard Z39.85-2007 (May 2007)
4. A number of **additional elements and element refinements** beyond the core—e.g., audience—reflecting needs of different discourse and practice communities
5. All of the above!

Why did the Dublin Core come to be in 1995?

- Dramatic increase in the number of document-like resources on the net
- Little improvement in indexing services made resources hard to discover
- Belief that descriptive metadata would improve discovery
- Perceived need for a descriptive standard that was simple to apply (even by non-professionals)

An Introduction to Metadata

–Definition
–Types & Functions
–Metadata Building Blocks

What is metadata?

- Metadata consists of *statements* we make about resources to help us find, identify, use, manage, evaluate, and preserve them.
- 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 and functions of metadata

Types of Metadata

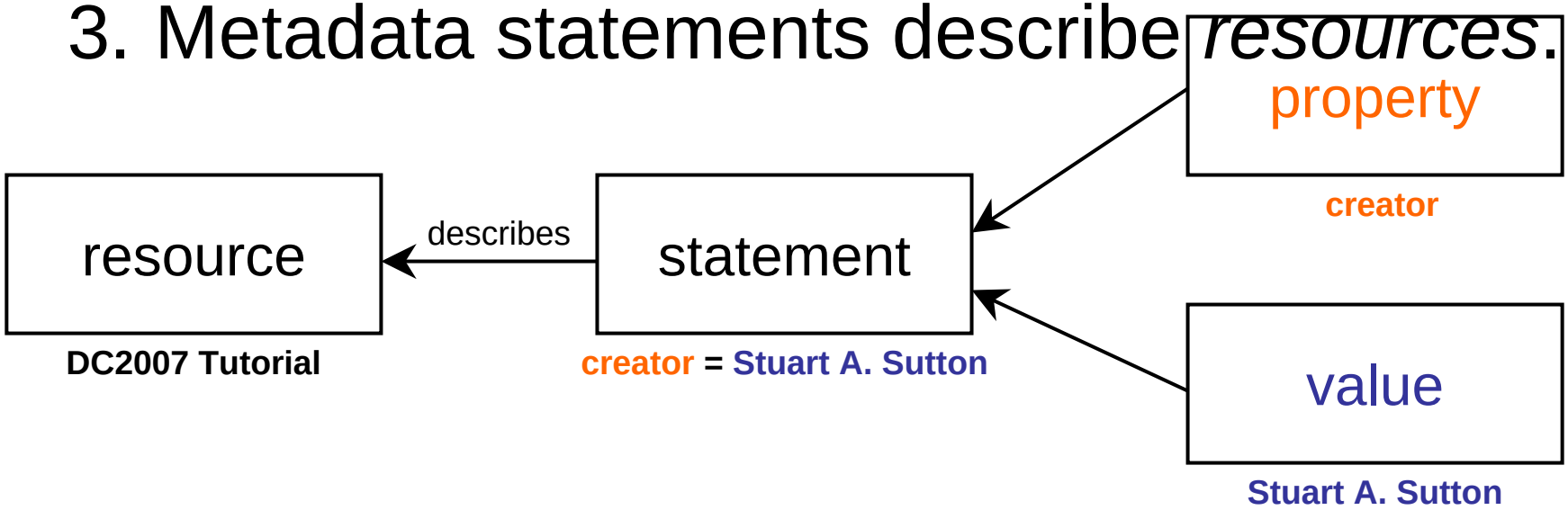
- Administrative
- Descriptive
- Access/Use
- Preservation
- Technical/Structural
- Relational
- Etc. …

Functions of Metadata

- Discover resources
- Mark content structure
- Identify versions
- Manage resources
- Certify authenticity
- Situate geospatially
- Control IP rights
- Indicate status
- Describe processes
- Etc. ….

Metadata building blocks

1. The basic unit of metadata is a *statement*.
2. A statement consists of a *property* (element) and a *value* (i.e., a property/value pair)
3. Metadata statements describe *resources*.



Properties and values in action?

245 00 $a Romeo and Juliet $h [videorecording]

<title>Gone with the wind</title>

<type>MovingImage</type>

Key Features of Dublin Core

- General Characteristics
- Dublin Core Principles

A note of caution before beginning…

Use of the word *element*—e.g., “Dublin Core Element Set”

> *“An Element is a property of a resource. As intended here, "properties" are attributes of resources -- characteristics that a resource may "have", such as a Title, Publisher, or Subject.”*
> http://dublincore.org/usage/documents/principles/ (2003)

- It is *not* used in the XML sense of *element* as a unit of XML data, delimited by tags
- *Element* is frequently used interchangeably in the Dublin Core community with the word *property*

The *DCMI Abstract Model*

- DC community realized early that rational development and machine-processing requires a coherent data model
  - *DCMI Grammatical Principles*
    http://dublincore.org/usage/documents/principles/ (2003)
  - *DCMI Abstract Model*
    http://dublincore.org/documents/abstract-model/ (2005)
    - Defines *resources* in terms of semantic relationships among *classes*, *properties*, and *values*
    - Defines a model for DCMI *descriptions*, *description sets*, and *records*
    - Serves as a foundation for future DCMI developments
    - Serves as a conceptual model for metadata initiatives outside DCMI

Characteristics of (or abstractions from) Dublin Core metadata

- Syntax independent
  - HTML/XHTML, XML, RDF/XML
- Flat, non-hierarchical element structure
- Extensible
  - By refinement (sub-properties of existing properties)
  - By combination (application profiles)
- Optional
- Repeatable
- Elements may occur in any order

Simple DC: Fifteen core elements (1996)

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

ISO 15836-2003
NISO Z39.85-2007

Resources for which DC is often used

DCMI Type Vocabulary

http://dublincore.org/documents/2006/12/18/dcmi-terms/

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

DC elements and refinements since 2000

- DC Simple (15 core elements)

abstract	coverage	hasFormat	isVersionOf	requires
accessRights	created	hasPart	language	rights
accrualMethod	creator	hasVersion	license	rightsHolder
accrualPeriodicity	date	identifier	mediator	source
accrualPolicy	dateAccepted	instructionalMethod	medium	spatial
alternative	dateCopyrighted	isFormatOf	modified	subject
audience	dateSubmitted	isPartOf	provenance	tableOfContents
available	description	isReferencedBy	publisher	temporal
bibliographicCitation	educationLevel	isReplacedBy	references	title
conformsTo	extent	isRequiredBy	relation	type
contributor	format	issues	replaces	valid

http://dublincore.org/documents/2006/12/18/dcmi-terms/

Element refinements

- Element refinements narrow the meaning of DC elements
  - *medium* refines *format*
  - *bibliographicCitation* refines *identifier*
  - *tableOfContents* refines *description*
- Refinements are properties just like the properties they refine…they can stand alone
  - <alternative>Nine queens<alternative>
  - <dateCopyrighted>2000-07-11</dateCopyrighted>

  NOT
  - <*title*.alternative>Nine queens</*title*.alternative>
  - <*date*.dateCopyrighted>2000-07-11</*date*.dateCopyrighted>

Qualification of element values: encoding schemes

- Encoding schemes give context to element values
  - ***Vocabulary encoding schemes***
    - Indicate that a value comes from a controlled vocabulary (e.g., that “Spanish American literature” is an LCSH term)
  - ***Syntax encoding schemes***
    - Indicate that a string is formatted in a standard way (e.g., that “1956-11-12” follows ISO 8601)
- For the DC core elements, DCMI recommends using encoding schemes with *coverage*, *date*, *format*, *language*, *subject*, and *type*

Summary: Simple and qualified DC

- *Simple DC*: Varying definitions
  - 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 *DCMI Abstract Model*
    - Element Refinements
    - Value Encoding Schemes

Dublin Core Principles

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

Dumb-Down

- **Simple DC** does not use *element refinements* or *encoding schemes* and statements only contain *value strings*
- **Qualified DC** uses features of the DCMI Abstract Model, particularly *element refinements* and *encoding schemes*
- *Dumbing-down* is translating qualified DC to simple DC (*property dumb-down* and *value dumb-down*)

The One-to-One Principle

- Create one metadata *description* for one and only one resource
  - Do not describe a digital image of the Mona Lisa as if it were the original painting
  - Do not describe both a song and the song's composer in the same *description*
    - Describe the composer and the work in two separate *descriptions*
- Group related *descriptions* into a *description set* (record)

Appropriate Values

- Use elements, element refinements and qualifiers to meet the needs of your local context, but . . .
- Remember that your metadata may be 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

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

Dublin Core metadata in a broader context

–Metadata Creation and Distribution

–Application Profiles and Interoperability

Metadata creation and distribution models

- *Federation*
  - Extensive specifications, standards, protocols, training
- *Harvesting*
  - Basic agreements, reliance on best practices
- *Gathering*
  - Automated indexing of content, algorithms yield results from search terms, less likely to use descriptive metadata per se

Harvesting model key features

- Integrating metadata from many sources calls for common element sets, record structures, and harvesting protocols
- Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) serves as a framework for sharing metadata and mandates ‘simple DC’ as a common metadata format
- Harvesting promotes metadata reuse
- Best practices balance cost and interoperability
- Communities add value to basic infrastructure (more complex metadata, new uses for protocol)

Application profiles & interoperability

"Application profiles consist of data elements drawn from one or more namespace schemas combined together by implementers and optimised for a particular local application." http://www.ariadne.ac.uk/issue25/app-profiles/

Application profiles enable:

- Implementers to use DC metadata in conjunction with non-DC metadata
- Implementers to benefit from the experience of their peers
- Communities to harmonize metadata usage for greater interoperability

Important aspects of the DCMI community

Dublin Core grows and changes

- DCMI emphasizes *open participation*
  - Conferences, Communities, Task Groups, and discussion lists
- DCMI element set evolves as implementers coin new terms and usage patterns emerge
- DCMI Usage Board reviews proposals for new metadata terms

Dublin Core Usage Board

- Considers proposals for new terms (elements, refinements, encoding schemes, DCMI Type Vocabulary terms)
  - Evaluates proposals in light of the requirements of the *DCMI Abstract Model*
- Evaluates constructs that use DCMI terms, such as *application profiles*

Finding out more about DC

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

Acknowledgement

Many thanks to Diane Hillmann and Marty Kurth for sharing with me the slides from earlier versions of this tutorial.

Questions?

Thank you for your attention!

Stuart A. Sutton
University of Washington
sasutton@u.washington.edu