

Agents and the DC Abstract Model

Andy Powell

UKOLN, University of Bath

DC Agents WG Meeting

DC-2005, Madrid



DCMI abstract model

- a *description* is made up of
 - one or more *statements* (about one, and only one, *resource*) and
 - optionally, the URI of the *resource* being described (*resource URI*)
- each *statement* is made up of
 - a *property URI* (that identifies a *property*)
 - a *value URI* (that identifies a *value*) and/or one or more representations of the *value* (*value representations*)



Value strings

- each *value representation* may take the form of a *value string*, a *rich value* or a *related description*
- each *value string* is a simple, human-readable string that represents the *resource* that is the *value* of the *property*
- each *value string* may have an associated *value string language* that is an ISO language tag (e.g. en-GB)



Encoding schemes

- *values and value strings can be 'qualified' by using encoding schemes*
- *a vocabulary encoding scheme is used to indicate the class of the value*
 - *e.g. the value is taken from LCSH*
- *a syntax encoding scheme is used to indicate how the value string is structured*
 - *e.g. the value string is a date structured according to the W3CDTF rules ("2004-10-12")*



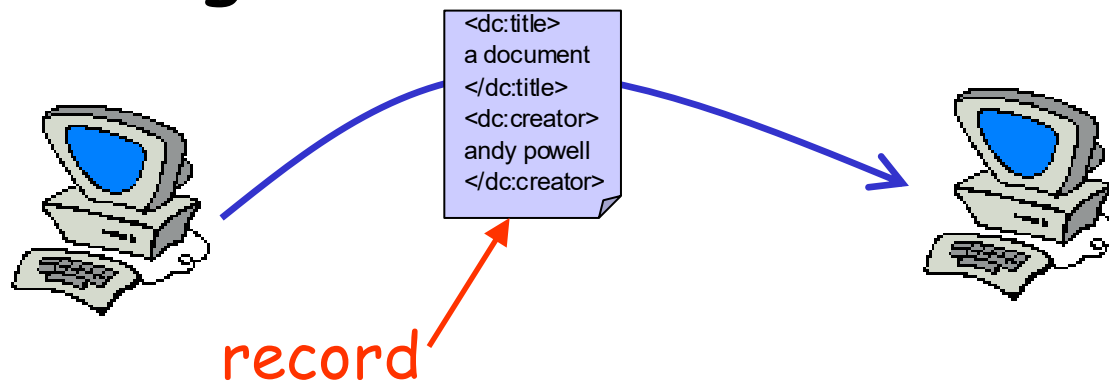
Description sets

- real-world metadata applications tend to be based on loosely grouped sets of *descriptions* (where the described *resources* are typically related in some way)
- known in the abstract model as *description sets*
- for example, a *description set* might comprise *descriptions* of both a painting and the artist



Records

- *description sets* are instantiated, for the purposes of exchange between software applications, in the form of *metadata records*
- each *record* conforms to one of the DCMI encoding guidelines (XHTML meta tags, XML, RDF/XML, etc.)



Model summary

record (encoded as XHTML, XML or RDF/XML)

description set

description (about a resource (URI))

statement

property (URI)

vocabulary encoding scheme (URI)

value (URI)

syntax encoding scheme (URI)

language
(e.g. en-GB)

representation

value string

OR rich value

OR related description



Example

- I write a Blog entry about DC-2005
- I want to create some metadata that provides the title of the Blog entry and my name and email address
- two 'related descriptions'
 - one about the Blog
 - one about me (the creator of the Blog)
- need to build up two descriptions and tie them together somehow



Description 1

description

- resourceURI = <http://myblog.com/madrid/>
- property = dc:title
 - valueString = "My travels to Madrid"
- property = dc:creator
 - valueString = "Andy Powell"



Description 2

description

- resourceURI =
- property = my:name
 - valueString = "Powell, Andy"
- property = my:email
 - valueString = "a.powell@ukoln.ac.uk"



descriptionSet

description

- resourceURI = <http://myblog.com/madrid/>
- property = dc:title
 - valueString = "My travels to Madrid"
- property = dc:creator
 - valueString = "Andy Powell"

description

- resourceURI =
- property = my:name
 - valueString = "Powell, Andy"
- property = my:email
 - valueString = "a.powell@ukoln.ac.uk"

descriptionSet

description

- resourceURI = <http://myblog.com/madrid/>
- property = dc:title
 - valueString = "My travels to Madrid"
- property = dc:creator
 - valueString = "Andy Powell"
 - valueURI = <http://andypowell.name/>

description

- resourceURI = <http://andypowell.name/>
- property = my:name
 - valueString = "Powell, Andy"
- property = my:email
 - valueString = "a.powell@ukoln.ac.uk"

descriptionSet

description

- resourceURI = <http://myblog.com/madrid/>
- property = dc:title
 - valueString = "My travels to Madrid"
- property = dc:creator
 - valueString = "Andy Powell"

description =

- resourceURI =
- property = my:name
 - valueString = "Powell, Andy"
- property = my:email
 - valueString =
"a.powell@ukoln.ac.uk"

RDF/XML example 1

```
<rdf:RDF>
  <rdf:Description about=http://myblog.com/madrid/>
    <dc:title>My travels to Madrid</dc:title>
    <dc:creator>
      <rdf:Description about=http://andypowell.name/>
        <rdfs:label>Andy Powell</rdfs:label>
      </rdf:Description>
    </dc:creator>
  </rdf:Description>
  <rdf:Description about=http://andypowell.name/>
    <my:name>Powell, Andy</my:name>
    <my:email>a.powell@ukoln.ac.uk</my:email>
  </rdf:Description>
</rdf:RDF>
```



RDF/XML example 2

```
<rdf:RDF>  
  <rdf:Description about=http://myblog.com/madrid/>  
    <dc:title>My travels to Madrid</dc:title>  
    <dc:creator>  
      <rdf:Description>  
        <rdfs:label>Andy Powell</rdfs:label>  
        <my:name>Powell, Andy</my:name>  
        <my:email>a.powell@ukoln.ac.uk</my:email>  
      </rdf:Description>  
    </dc:creator>  
  </rdf:Description>  
</rdf:RDF>
```



Agent application profile

- need to declare (or find) a set of properties for creating description 2
- name, phone, email, date of birth, sex, etc., etc.
- DCMES, FOAF, vCard, ...
- note: to be usable in DC application profiles, terms must be
 - declared in appropriate ways, e.g. as RDF properties
 - assigned URIs

