

Data and Metadata Standardisation

hussein suleman
uct cs honours 2006

Digital Object Types

Type	Example
Text	
Hypertext	
Image	
Video	
Audio	
3D Model	
Interactive Visualisation	
Software	

Data vs. Metadata

- Data refers to digital objects that contain useful information for information seekers.
- Metadata refers to standardised descriptions of objects, digital or physical.
- Many systems manipulate metadata records, which contain pointers to the actual data.
- The definition is fuzzy as metadata contains useful information as well and in some cases could contain all the data e.g., metadata describing a person.

An Example of Metadata

□ Object:



□ Metadata

- name: chalk
- owner: hussein
- colour: white
- size: 2.5
- description: used to write on board
- location: honours lecture room
- source: Waltons Stationers

Another Metadata Example

□ Object:



□ Metadata

- colour: white
- title: RG123
- owner: UCT
- lifetime: 2 months
- size: 1
- identifier: RG123
- description: white powdery stick

Metadata Comparisons

□ Metadata

- colour: white
- title: RG123
- owner: UCT
- lifetime: 2 months
- size: 1
- identifier: RG123
- description: white powdery stick

□ Metadata

- name: chalk
- owner: hussein
- colour: white
- size: 2.5
- description: used to write on board
- location: honours lecture room
- source: Waltons Stationers

What problems can occur?

Types of Metadata

- Descriptive
 - title, author, type, format, ...
- Structural
 - part, subpart, relation, child, ...
- Administrative
 - location, identifier, submitter, ...
- Preservation
 - resolution, capture device, watermark, ...
- Provenance
 - source archive, previous version, source format, ...

Creating Metadata

- Follow metadata guidelines.
- Use terms from controlled vocabularies.
- Avoid duplication of information across fields.
- Use accepted standards for common elements.
 - e.g., ISO 8601 for dates
 - 2005-03-03 instead of 03/03/05
- Use XML-based encoding according to standardised Schema/DTD.

Metadata Standards

- To promote interoperability among systems, use popular metadata standards to describe objects (both semantically and syntactically).
 - Dublin Core
 - 15 simple elements to describe anything.
 - MARC
 - Comprehensive system devised to describe items in a (physical) library.
 - RFC1807
 - Computer science publications format.
 - IMS Metadata Specification
 - Courseware object description.
 - VRA-Core
 - Multimedia (especially image) description.
 - EAD
 - Library finding aids to locate archived items.

Why didn't the CS folks use MARC?

Newer Metadata Standards

- METS
 - Descriptive, administrative and structural encoding for metadata of digital objects
- MODS
 - Richer than DC, subset of MARC21
- MPEG21-DIDL
 - Structural descriptions of complex multimedia objects

Dublin Core

- ❑ Dublin Core is one of the most popular and simplest metadata formats.
- ❑ 15 elements with recommended semantics.
- ❑ All elements are optional and repeatable.

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

DC in HTML

- ❑ `<META NAME=DC.Creator CONTENT="Tony Gill">`
- ❑ `<META NAME=DC.Title CONTENT="ADAM Quick Guide to Metadata">`
- ❑ `<META NAME=DC.Subject CONTENT="ADAM, Dublin Core, internet cataloguing, metadata">`
- ❑ `<META NAME=DC.Description CONTENT="A short ADAM guide to metadata, particularly Dublin Core.">`
- ❑ `<META NAME=DC.Date CONTENT="1997-11-21">`

Source: <http://adam.ac.uk/adam/metadata.html>

DC Metadata in XML

```
<title>02uct1</title>
<creator>Hussein Suleman</creator>
<subject>Visit to UCT </subject>
<description>the view that greets you as you
  emerge from the tunnel under the freeway -
  WOW - and, no, the mountain isnt that
  close - it just looks that way in 2-
  D</description>
<publisher>Hussein Suleman</publisher>
<date>2002-11-27</date>
<type>image</type>
<format>image/jpeg</format>
```

DC Metadata in Valid Qualified XML

```
<oaiddc:dc xmlns="http://purl.org/dc/elements/1.1/"
  xmlns:oaiddc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
  http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <title>02uct1</title>
  <creator>Hussein Suleman</creator>
  <subject>Visit to UCT </subject>
  <description>the view that greets you as you emerge from the tunnel
  under the freeway - WOW - and, no, the mountain isnt that close - it
  just looks that way in 2-D</description>
  <publisher>Hussein Suleman</publisher>
  <date>2002-11-27</date>
  <type>image</type>
  <format>image/jpeg</format>
  <identifier>http://www.husseinsspace.com/pictures/200230uct/02uct1.jpg
  </identifier>
  <language>en-us</language>
  <relation>http://www.husseinsspace.com</relation>
  <rights>unrestricted</rights>
</oaiddc:dc>
```

Why is there a separate namespace for the root element?

DC Qualifiers

- ❑ Dublin Core has been considered TOO simple for many applications – not enough semantics.
- ❑ Some DC terms have had qualifiers added to make the meaning more specific.
 - For example,
 - ❑ date.created instead of just date
 - ❑ relation.hasPart instead of just relation
- ❑ In general, qDC can be dumbed-down (*that's a technical term in interoperability*) to DC by ignoring qualifications.

What Metadata Format?

- ❑ Every project has its own metadata/data requirements, therefore most use a proprietary format.
- ❑ For maximum interoperability,
 - Map metadata to most descriptive format for use by close collaborators.
 - Map metadata to DC for use by all and sundry.
- ❑ How do we “map” metadata formats?

Do we actually store data in XML?

Metadata Transformation

- ❑ Use XML parser to parse data.
- ❑ Use SAX/DOM to extract individual elements and generate new format.
- ❑ Example (to convert UCT to DC):

```
■ my $parser = new DOMParser;
my $document = $parser->parsefile ('uct.xml')->getDocumentElement;
foreach my $title ($document->getElementsByTagName ('title'))
{
    print "<title>".$title->getFirstChild->getData."</title>\n";
}
foreach my $author ($document->getElementsByTagName ('author'))
{
    print "<creator>".$author->getFirstChild->getData."</creator>\n";
}
print "<publisher>UCT</publisher>\n";
foreach my $version ($document->getElementsByTagName ('version'))
{
    foreach my $number ($version->getElementsByTagName ('number'))
    {
        print "<identifier>".
            $number->getFirstChild->getData."</identifier>\n";
    }
}
```

Metadata Transformation (XSLT) 1/2

```
<stylesheet version='1.0'
  xmlns='http://www.w3.org/1999/XSL/Transform'
  xmlns:oaidc='http://www.openarchives.org/OAI/2.0/oai_dc/'
  xmlns:dc='http://purl.org/dc/elements/1.1/'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:uct='http://www.uct.ac.za'
>

<!--
  UCT to DC transformation
  Hussein Suleman
  v1.0 : 24 July 2003
-->

<output method="xml"/>

<variable name="institution"><text>UCT</text></variable>
```

Metadata Transformation (XSLT) 2/2

```
<template match="uct:uct">
  <oaiddc:dc xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title><value-of select="uct:title"/></dc:title>
    <apply-templates select="uct:author"/>
    <element name="dc:publisher">
      <value-of select="$institution"/>
    </element>
    <apply-templates select="uct:version"/>
  </oaiddc:dc>
</template>

<template match="uct:author">
  <dc:creator>
    <value-of select="."/>
  </dc:creator>
</template>

<template match="uct:version">
  <dc:identifier>
    <value-of select="uct:number"/>
  </dc:identifier>
</template>
</stylesheet>
```

Automatic Metadata Extraction

- ❑ Create metadata automatically from a digital object.
- ❑ Embedded Metadata
 - e.g., MP3 tags
- ❑ Heuristic Techniques
 - e.g., The first string that looks like a date is the date of publication
- ❑ Machine Learning
 - e.g., Neural networks
- ❑ Dictionary Techniques
 - e.g., If it looks like a name, it could be an author

References

- ❑ Dublin Core Metadata Initiative (2005). DCMI Metadata Terms. Available <http://dublincore.org/documents/dcmi-terms/>
- ❑ Dublin Core Metadata Initiative (2004). Dublin Core Metadata Element Set, Version 1.1: Reference Description. Available <http://dublincore.org/documents/dces/>
- ❑ Freed, N. and N. Borenstein (1996) Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC 2045, Network Working Group, IETF. Available <http://www.ietf.org/rfc/rfc2045.txt>
- ❑ Freed, N. and N. Borenstein (1996) Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, RFC 2046, Network Working Group, IETF. Available <http://www.ietf.org/rfc/rfc2046.txt>
- ❑ IMS Global Learning Consortium, Inc. (2001). IMS Learning Resource Meta-Data Information Model, Version 1.2.1 Final Specification. Available http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_infov1p2p1.html
- ❑ Lasher, R. and D. Cohen (1995). A Format for Bibliographic Records. Network Working Group, RFC1807. Available <http://www.ietf.org/rfc/rfc1807.txt>
- ❑ Library of Congress (2002). Encoded Archival Description (EAD), Official EAD Version 2002 Web Site. Website <http://www.loc.gov/ead/>
- ❑ Library of Congress (2005). MARC Standards. Website <http://www.loc.gov/marc/>
- ❑ Library of Congress (2005). Metadata Encoding and Transmission Standard. Website <http://www.loc.gov/standards/mets/>
- ❑ Library of Congress (2005). Metadata Object Description Schema. Website <http://www.loc.gov/standards/mods/>
- ❑ Visual Resources Association Data Standards Committee. (2002). VRA Core Categories, Version 3.0. Available <http://www.vraweb.org/vracore3.htm>
- ❑ XML Cover Pages (2005). MPEG-21 Part 2: Digital Item Declaration Language (DIDL). Website <http://xml.coverpages.org/mpeg21-didl.html>