

# The Role of XSLT in Digital Libraries, Editions, and Cultural Exhibits

David J. Birnbaum & Violeta Ilik

09/22/2013

TPDL 2013 Conference, Valletta, Malta

The use of eXtensible Stylesheet Language Transformation (XSLT) for repurposing, editing and reformatting metadata

Building block of XML are elements

```
<tag attribute-name="attribute-value">  
  (Malta is gorgeous)</tag>
```



Tag name

name of the attribute

## Rules of well-formedness:

- every start tag must have a closing tag

`<tag> </tag>` Or `<tag/>`

- Tags must nest cleanly

`<creator><name>Birnbaum</name></creator>`

- Attribute values must appear within quotation marks

`<page n="12"/>`

- Tags are case sensitive and they must match

`<creator></creator>` Or `<CREATOR></CREATOR>`

- Single root element

- The left angle bracket and ampersand are special characters

## XML Namespace

collection of element and attribute names  
uniquely identified by an explicit or implicit  
association to a Universal Resource  
Identifier (URI)

Mapping compares and analyzes two or more metadata schemas, while crosswalks are the product of the mapping process

## Challenges:

- ambivalent matches
- hybrid bibliographic records
- data mapping to multiple fields or combining into single fields during migration
- orphaned data parsed into incongruous fields
- mixed standards in original data
- MARC data loss during the migration
- flat structure versus hierarchical structures (Woodley, 2008)

## Challenges:

- reconciling metadata organization systems
- choice of unanalogous processes during metadata standards creation
- imprecise definitions or alternate naming choices that inhibit element to element mapping
- information being lost or combined during mapping
- unharmonious hierarchical structures

(St. Pierre & LaPlant, 2000)



## Two opposing views on crosswalks:

- crosswalks are a stopgap measure
- crosswalks represent an attempt to identify interoperable elements among standards

# XML & XSLT

each XSLT stylesheet describes how a set of XML documents (the source documents) should be converted to other documents (the result documents)

The Library of Congress (LC) developed MARCXML architecture and MARCXML toolkit to standardize the exchange of MARC structured data in XML

Difference between MARC and XML:  
XML uses beginning tags `< >` and ending tags  
`</>`

**MARCXML to Dublin Core XML-RDF**

```
<collection
xmlns="http://www.loc.gov/MARC21/slim"
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
xsi:schemaLocation="http://www.loc.gov/standa
rds/marcxml/schema/MARC21slim.xsd">
  <record
xmlns="http://www.loc.gov/MARC21/slim">
  <leader>00000cam a2200000l 4500</leader>
    <controlfield tag="001">ocm00795943
      </controlfield>
```

```
<datafield tag="245" ind1="0" ind2="0">
  <subfield code="a">Bdinski zbornik :</subfield>
  <subfield code="b">Ghent Slavonic Ms 408, A.D. 1360 :
    facsimile edition /</subfield>
  <subfield code="c">with a presentation by Ivan
    Dujčev.</subfield>
</datafield>
<datafield tag="260" ind1=" " ind2=" ">
  <subfield code="a">London :</subfield>
  <subfield code="b">Variorum Reprints,</subfield>
  <subfield code="c">1972.</subfield>
</datafield>
<datafield tag="300" ind1=" " ind2=" ">
  <subfield code="a">xi, 242 [i.e. 483] p. ;</subfield>
  <subfield code="c">21 cm.</subfield>
</datafield>
<datafield tag="490" ind1="1" ind2=" ">
  <subfield code="a">Variorum reprint ;</subfield>
  <subfield code="v">S11</subfield>
</datafield>
```

15 base Dublin Core elements:  
title; creator; subject;  
description; publisher;  
contributor; date; type; format;  
identifier; source; language;  
relation; coverage; rights



**Table 1 – Sample Metadata Profile**

<b>MARC field</b>	<b>Dublin Core field</b>
100, 110, 111	dc.creator
245	dc.title
260 subfield a	dc.publisher
260 subfield c	dc.date.created
260 subfield b	dc.publisher
500	dc.description
520	dc.description.abstract
546	dc.language.iso
600	dc.subject.lcsh
830	dc:relation.isPartOfSeries
In some cases added field with value: “Texas A&M University”	dc.publisher
Added field with value: “text”	dc.type.material
Added field with value: “reformatted digital”	dc.format.digitalOrigin
Added field with value: “electronic”	dc.format.medium

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
exclude-result-prefixes="xs"
  xpath-default-
namespace="http://www.loc.gov/MARC21/slim"
version="2.0"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<xsl:strip-space elements="*" />
<xsl:output method="xml" indent="yes" />
<xsl:template match="record">
```

```
<rdf:RDF>
  <rdf:Description>
    <xsl:apply-templates select="datafield[@tag='245']"/>
    <xsl:apply-templates select="datafield[@tag = ('100','700')]/>
    <xsl:apply-templates select="datafield[@tag =
'260']/subfield[@code=('b','c')]/>
    <xsl:apply-templates select="datafield[@tag =
'300']/subfield[@code=('a','b')]/>
    <xsl:apply-templates select="datafield[@tag =
('500','501','504','505')]/>
    <xsl:apply-templates select="datafield[@tag =
('650','610','653')]/>
    <xsl:apply-templates select="datafield[@tag='830']"/>
    <xsl:apply-templates select="leader"/>
    <xsl:apply-templates select="datafield[@tag='041']"/>
    <dc:publisher>Digital Publisher: Texas A&M
University</dc:publisher>
    <dc:format.digitalOrigin>reformatted
digital</dc:format.digitalOrigin>
    <dc:format.medium>electronic</dc:format.medium>
  </rdf:Description>
</rdf:RDF>
```

```
</xsl:template>
<xsl:template match="datafield[@tag='245']">
  <dc:title>
    <xsl:apply-templates select="subfield[@code = ('a','b')]" />
  </dc:title>
</xsl:template>
<xsl:template
match="datafield[@tag='245']/subfield[@code='a']">
  <xsl:text> </xsl:text>
  <xsl:value-of
    select="if (ends-with(., '/') or ends-with(., ';')) then
substring(.,1,string-length(.) - 1) else ."
  />
</xsl:template>
<xsl:template
match="datafield[@tag='245']/subfield[@code='b']">
  <xsl:text> </xsl:text>
  <xsl:value-of
    select="if (ends-with(., '/') or ends-with(., ';')) then
substring(.,1,string-length(.) - 2) else ."
  />
</xsl:template>
```

Thank you

Violeta Ilik [vilik@library.tamu.edu](mailto:vilik@library.tamu.edu)