OLAC Metadata

Steven Bird
University of Melbourne /
University of Pennsylvania

OLAC Workshop
10 December 2002

OLAC Metadata

- OLAC Metadata - Simons & Bird
  http://www.language-archives.org/OLAC/metadata.html
- Draft standard
- Purpose:
  - Define the metadata format
  - Define the extension mechanism

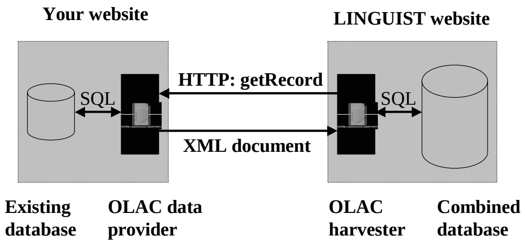
OLAC Metadata

1. Introduction
2. Metadata elements
3. Metadata format
4. OLAC extensions
5. Defining a third-party extension
6. Documenting an extension

1. Introduction

- XML
- OAI framework
- From data provider to service provider
  - How we ship the metadata around
  - Data is stored/presented in other ways

Aside: OAI Protocol



Your website

LINGUIST website

SQL

HTTP: getRecord

XML document

SQL

Existing database

OLAC data provider

OLAC harvester

Combined database

2. Metadata Elements

- 15 DC elements - dublincore.org
- Need to describe language resources with greater precision
- Follow DC recommendation for qualifying elements
  - *Dublin Core Qualifiers*
    http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/
  - Refinements: meaning of element is narrower, more specific
  - Encoding schemes: controlled vocabularies and standardized formats

Community-specific qualifiers
aka “OLAC Extensions”

- Access rights
  dc:rights
- Discourse type
  dc:type
- Language identification
  dc:language
  dc:subject

- Linguistic field
  dc:subject
- Linguistic data type
  dc:type
- Participant role
  dc:creator
  dc:contributor

Vocabularies to be discussed this afternoon…

Refinements vs encoding schemes

Refinement:

- Role vocabulary, e.g. annotator; translator
  role of contributor is more specific

Encoding scheme:

- Linguistic data type, e.g. lexicon; dataset
  free-text description is summarized with a restricted term, facilitating precision and recall

Both:

- Subject language, e.g. es; x-sil-BAN
  subject is more specific (about language)
  restricted vocabulary

3. Metadata format

- Follows guidelines for DC/DCQ in XML
  1. *Guidelines for implementing DC in XML*
     http://dublincore.org/documents/2002/09/09/dc-xml-guidelines
  2. *Recommendations for XML Schema for DCQ*
     http://www.ukoln.ac.uk/metadata/dcmi/xmlschema/20021007/
- Application profile
  - Metadata schema
  - Combines elements from multiple sources
- OLAC = DC application profile for LRs
  1. DC: dc.xsd
  2. DCQ: dcterms.xsd
  3. OLAC extensions

Tour of an OLAC record

```
<olac:olac
  xmlns:olac="http://www.language-archives.org/OLAC/1.0/“
  xmlns="http://purl.org/dc/elements/1.1/“
  xmlns:dcterms="http://purl.org/dc/terms/“
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance“
  xsi:schemaLocation="
    http://www.language-archives.org/OLAC/1.0/
    http://www.language-archives.org/OLAC/1.0/olac.xsd">
  <creator>Bloomfield, Leonard</creator>
  <date>1933</date>
  <title>Language</title>
  <publisher>New York: Holt</publisher>
</olac:olac>
```

(1) Container and namespace

```
<olac:olac
  xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
  xmlns="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.language-archives.org/OLAC/1.0/
    http://www.language-archives.org/OLAC/1.0/olac.xsd">
  <creator>Bloomfield, Leonard</creator>
  <date>1933</date>
  <title>Language</title>
  <publisher>New York: Holt</publisher>
</olac:olac>
```

(2) XML Schema information

```
<olac:olac
  xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
  xmlns="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.language-archives.org/OLAC/1.0/
    http://www.language-archives.org/OLAC/1.0/olac.xsd">
  <creator>Bloomfield, Leonard</creator>
  <date>1933</date>
  <title>Language</title>
  <publisher>New York: Holt</publisher>
</olac:olac>
```

(3) DC namespace & content

```
<olac:olac
 xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
 xmlns="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="
   http://www.language-archives.org/OLAC/1.0/
   http://www.language-archives.org/OLAC/1.0/olac.xsd">
 <creator>Bloomfield, Leonard</creator>
 <date>1933</date>
 <title>Language</title>
 <publisher>New York: Holt</publisher>
</olac:olac>
```

Using DC Qualifiers

- Extra namespace declaration:
  xmlns:dcterms="http://purl.org/dc/terms/"
- Qualified element:

```
<dcterms:created
 xsi:type="dcterms:W3C-DTF">
   2002-11-28
</dcterms:created>
```

- "created" is a refinement of date
  - refinement relationship is represented in the dcterms schema ("substitutionGroup")

xml:lang attribute

- the language of the *element content*
- expressed using RFC 1766

```
<title xml:lang="x-sil-LLU">
 Na tala 'uria na idulaa diana</title>

<dcterms:alternative xml:lang="en">
 The road to good reading</dcterms:alternative>
```

- no need to declare xml namespace

4. OLAC extensions

- xsi:type - a feature of XML Schema
- … xsi:type="olac:language" …
  - xsi = namespace for XML Schema Instance
  - value = complex type
  - overrides the type declared for the element
  - new type must be validly derived from the overridden type
- optional code attribute
- element content for comments

Example: Language

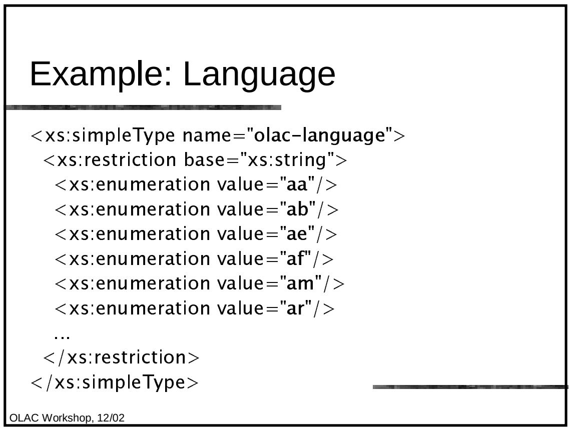
1. <subject>Dschang</subject>

2. Refinement only:
   <subject xsi:type="olac:language">
     Dschang
   </subject>

3. Refinement and encoding scheme:
   <subject xsi:type="olac:language"
     code="x-sil-BAN"/>

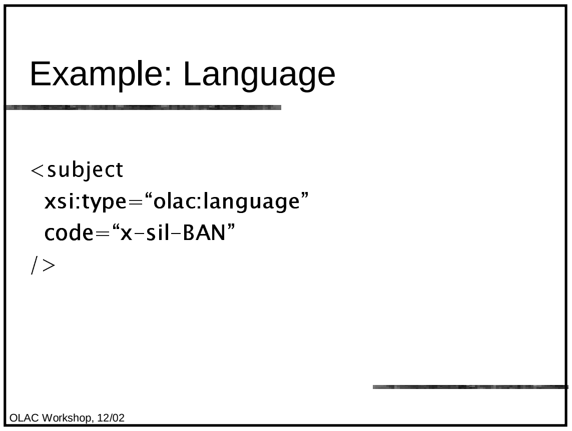
Example: Language

```
<xs:complexType name="language">
  <xs:complexContent mixed="true">
    <xs:extension base="dc:SimpleLiteral">
      <xs:attribute name="code"
        type="olac-language" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```



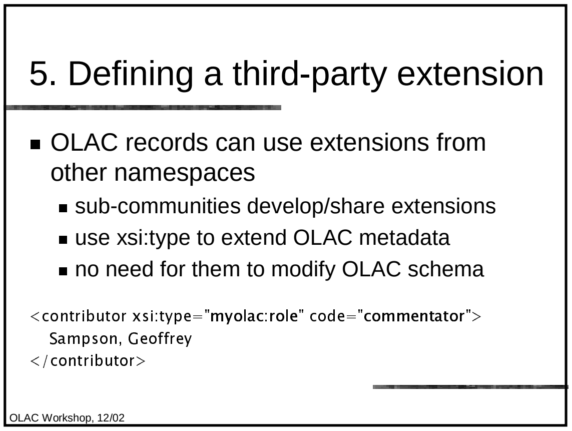
Example: Language

```
<xs:simpleType name="olac-language">
  <xs:restriction base="xs:string">
    <xs:enumeration value="aa"/>
    <xs:enumeration value="ab"/>
    <xs:enumeration value="ae"/>
    <xs:enumeration value="af"/>
    <xs:enumeration value="am"/>
    <xs:enumeration value="ar"/>
    ...
  </xs:restriction>
</xs:simpleType>
```



Example: Language

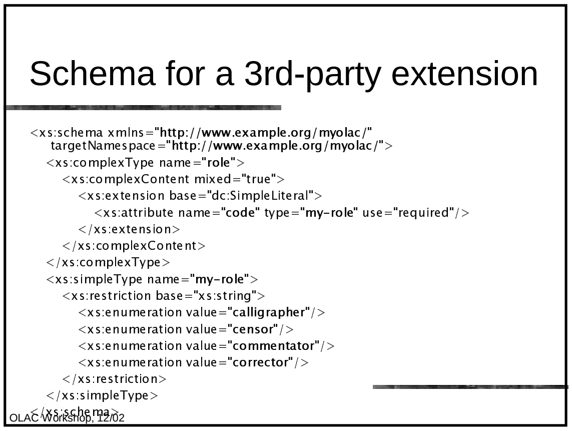
```
<subject
  xsi:type="olac:language"
  code="x-sil-BAN"
/>
```



5. Defining a third-party extension

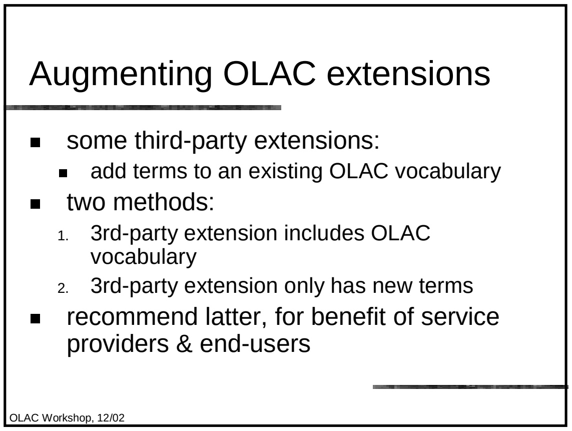
- OLAC records can use extensions from other namespaces
  - sub-communities develop/share extensions
  - use xsi:type to extend OLAC metadata
  - no need for them to modify OLAC schema

```
<contributor xsi:type="myolac:role" code="commentator">
   Sampson, Geoffrey
</contributor>
```



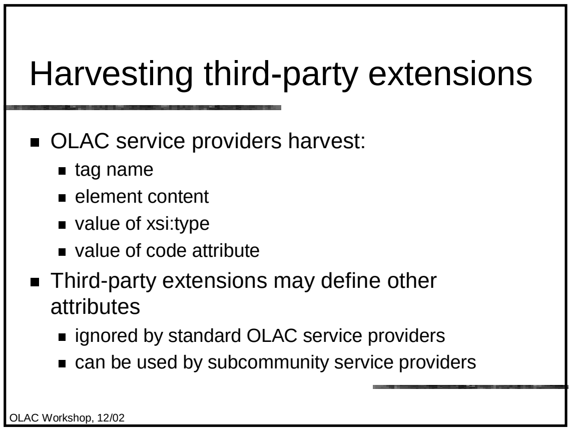
Schema for a 3rd-party extension

```
<xs:schema xmlns="http://www.example.org/myolac/"
    targetNamespace="http://www.example.org/myolac/">
   <xs:complexType name="role">
      <xs:complexContent mixed="true">
         <xs:extension base="dc:SimpleLiteral">
            <xs:attribute name="code" type="my-role" use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:simpleType name="my-role">
      <xs:restriction base="xs:string">
         <xs:enumeration value="calligrapher"/>
         <xs:enumeration value="censor"/>
         <xs:enumeration value="commentator"/>
         <xs:enumeration value="corrector"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```



Augmenting OLAC extensions

- some third-party extensions:
  - add terms to an existing OLAC vocabulary
- two methods:
  1. 3rd-party extension includes OLAC vocabulary
  2. 3rd-party extension only has new terms
- recommend latter, for benefit of service providers & end-users



Harvesting third-party extensions

- OLAC service providers harvest:
  - tag name
  - element content
  - value of xsi:type
  - value of code attribute
- Third-party extensions may define other attributes
  - ignored by standard OLAC service providers
  - can be used by subcommunity service providers

6. Documenting an extension

- All extensions should be documented
  - in human-readable form
  - at a web-accessible location
- The XML schemas for extensions should also contain machine-readable documentation
  - name, version, description, DC element, documentation URL

olac-extension element

```
<olac-extension xmlns="http://www.language-
   archives.org/OLAC/1.0/olac-extension.xsd">
  <shortName>role</shortName>
  <longName>Code for My Specialized Roles</longName>
  <versionDate>2002-08-16</versionDate>
  <description>A hypothetical extension for an individual archive,
   defining specialized roles not available in the OLAC Role
   vocabulary.</description>
  <appliesTo>creator</appliesTo>
  <appliesTo>contributor</appliesTo>
  <extensionDoc>http://www.my.org/roles.html</extensionDoc>
</olac-extension>
```

Summary

- XML format follows DC recommendations
  - new DC qualifiers automatically adopted
  - other communities can use OLAC qualifiers
- Limited change from version 0.4:
  - subject.language becomes
    subject xsi:type="olac:language"
- Flexible: optionality, free-text content
- Extensible: mix in third-party extensions