A Crosswalk in the Park?

Converting from MARC 21 to Linked Art at Yale University

Semantic Web in Libraries | November 28, 2022

Martin Lovell (Yale University Library)
Timothy A. Thompson (@timathom@indieweb.social)
Outline

1. Yale, LUX, and Linked Art
2. MARC 21 to JSON-LD: Defining the Crosswalk
3. MARC 21 to JSON-LD: Implementing the Crosswalk
LUX: Illuminating the Collections of Yale’s Museums, Libraries, and Archives via Usable Linked Open Data

Timothy Thompson
Librarian for Applied Metadata Research
timothy.thompson@yale.edu
@timathom

Robert Sanderson
Director for Cultural Heritage Metadata
robert.sanderson@yale.edu
@azaroth42

https://www.youtube.com/watch?v=C4IAJH0s1gY
Defining the Crosswalk: DITA

- Darwin Information Typing Architecture (DITA)
- Maintained by the OASIS standards organization.
- Modular (XML) framework for technical writing and documentation.
- Based on the idea of topics.
- Topics are organized using map documents.
- Topics and content can be reused and interlinked.

https://www.dita-ot.org
Defining the Crosswalk: DITA in oXygen
Defining the Crosswalk: Structure

- LUX Top-Level Entities
  - Record-Level Entities
  - Related Entities
    - Concepts
    - Groups
    - People
    - Places

- Descriptive Content
  - Identifiers
  - Names and Labels
  - Notes and Statements
  - Dates
"Supertype" Taxonomy Specifications

**Source data**
- name: Collages
- sampleBibs: 7647390
- fieldSpec:
  - ldr[6]
  - 006[0]
  - 007[0]
  - 007[1]
  - 300a
  - 655a

**Processing steps and output**

```json
{
"classified_as": [
{
"id": "http://vocab.getty.edu/aat/300033963",
"type": "Type",
"_label": "Collages",
"classified_as": [
{
"id": "http://vocab.getty.edu/aat/300435443",
"type": "Type",
"_label": "Type of Object"
}
]
}
}
```

# Collages mapping

**conditions:**
- **OR:**
  - # Two-dimensional nonprojectable graphic
    - ldr[6]
    - k
  - # Two-dimensional nonprojectable graphic
    - 006[0]
    - k
- **AND:**
  - **OR:**
    - **AND:**
      - # Nonprojected graphic
        - 007[0]
        - k
      - # Collage
        - 007[1]
        - c
    - lower-case(300a) contains 'collage'
    - lower-case(655a) contains 'collage'
Narrative Specifications (Places from fixed field 008)

**Introduction**

008[15-17] contains a two- or three-letter code representing a country- or state-level place entity.

The code represents a place of publication, production, execution, or sometimes location (in the case of manuscript holdings).

**Step**

1. Generate and store the top-level place resource, identified by an IRI.

   i. Normalize whitespace to test for null values and eliminate trailing whitespace after two-letter codes.

   ii. Match the two- or three-letter code against the Library of Congress's MARC List for Countries, which is available as a tab-delimited file, and add the corresponding URI as an equivalent reference, as shown below.

   iii. Use the place name from the Library of Congress file as a key to match against and merge with equivalent place entities.

   For example, references to "France" should point to the same entity IRI, regardless of the data source in MARC. 008 858723s1984 fr a b 00180 fre d and 650 0 $a Opera $z France. should both result in a link to the same place entity representing France.

**Substeps**

**Step result**

```json
{
   "@context": "https://linked.art/ns/v1/linked-art.json",
   "id": "https://lux.collections.yale.edu/data/place/3d6bc9fa-ca9c-4fa1-bd0e-d25e93f4d1e5",
   "type": "Place",
   "_label": "France",
   "identified_by": [
   {
   "type": "Name",
   "content": "France",
   "classified_as": [
   }
```
Technical Implementation
MARC Holdings and Bibliographic → Linked Art

Typical Crosswalk

- Holdings
- Bibliographic

Transformation that *Simplifies*
Converting Data into *Single Record*
Driven by Configuration Files

Config JSON

Record
Easy to ingest into SOLR and use facets
MARC Holdings and Bibliographic → Linked Art
MARC Holdings and Bibliographic → Linked Art

- Holdings
- HumanMadeObject
  - carries
  - LinguisticObject
- Bibliographic
MARC Holdings and Bibliographic → Linked Art

Holdings

Bibliographic

HumanMadeObject, DigitalObject

carries, digitally_caries, shows, digitally_shows, member_of

LinguisticObject, Set, VisualItem
MARC Holdings and Bibliographic → Linked Art

- Holdings
- Bibliographic

HumanMadeObject, DigitalObject

carries, digitally_carries, shows, digitally_shows

LinguisticObject, Set, Visual Item

Identifiers

Person

Place

Type

Group

LinguisticObject
Processing the Data
Processing the Data
Processing the Data

Holding

Transformers
7 total holdings
transformers

Bibs

Transformers
73 total bib
transformers
Processing the Data

- **Holding**
  - Transformers
    - 7 total holdings
      - transformers

- **Bibs**
  - Transformers
    - 73 total bib transformers
      - 32 for supertypes
Processing the Data

Holding

TopLevelEntity
JSON
Transformers
7 total holdings
transformers

Bibs

TopLevelEntity
JSON
Transformers
73 total bib
transformers
Processing the Data

Holding

TopLevelEntity
JSON

Transformers
7 total holdings
transformers

TopLevelEntity
JSON

Bibs

TopLevelEntity
JSON

Transformers
73 total bib
transformers

TopLevelEntity
JSON

TopLevelEntity
JSON
Processing the Data

Diagram showing the relationship between different entities and JSON objects.
Storing the Entities
Storing the Entities

TopLevelEntity

JSON

SAVE

Postgres Database
Storing the Entities

TopLevelEntity

JSON

Process IDs

Postgres Database
Storing the Entities

- TopLevelEntity
  - JSON
- Process IDs
- Relationships
- Postgres Database
Storing the Entities

TopLevelEntity

JSON

Postgres Database

Process IDs

Relationships

Activity Stream
URL Schema
URL Schema

https://linked-art.library.yale.edu/ils/lib/13635257
URL Schema

https://linked-art.library.yale.edu/ils/lib/13635257

https://linked-art.library.yale.edu/node/fc7e1846-aa0f-4d07-879a-e19e85cb9a8a
URL Schema
Incoming Relationships

<table>
<thead>
<tr>
<th>JSON</th>
<th>Raw Data</th>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>Copy</td>
<td>Collapse All</td>
</tr>
</tbody>
</table>

| page: | 0 |
| total: | 2 |
| totalPages: | 1 |

**results:**

0:

- **subject:** "https://linked-art.library.yale.edu/node/7cf270c2-8168-4e6f-a271-a4663f77756"
- **relationship:** ".created_by.part.carried_out_by"
- **object:** "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
- **_subject_label:** "Family of liars [electronic resource] : The prequel to we were liars"

1:

- **subject:** "https://linked-art.library.yale.edu/node/7da036b8-80c7-46ab-a20a-cb916aac723d"
- **relationship:** ".created_by.part.carried_out_by"
- **object:** "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
- **_subject_label:** "French braid [electronic resource] : A novel"
Incoming Relationships

JSON

```
page: 0
total: 2
totalPages: 1

results:

0:
  subject: "https://linked-art.library.yale.edu/node/7cf270c2-8168-4e6f-a271-a46e63f77756"
  relationship: "created_by_part.carried_out_by"
  object: "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
  _subject_label: "Family of liars [electronic resource] : The prequel to we were liars"

1:
  subject: "https://linked-art.library.yale.edu/node/7da0360b-80c7-46ab-a20a-cb916aac723d"
  relationship: "created_by_part.carried_out_by"
  object: "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
```
## Incoming Relationships

<table>
<thead>
<tr>
<th>subject.cf:</th>
<th>[..]</th>
</tr>
</thead>
<tbody>
<tr>
<td>created_by:</td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td>type:</td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td>part:</td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td>carried_out_by:</td>
<td></td>
</tr>
<tr>
<td>0:</td>
<td></td>
</tr>
<tr>
<td>id:</td>
<td>&quot;<a href="https://linked-art-test.library.yale.edu/node/4102fbd6-3f59-44dd-ba5c-58fba37b8e0d">https://linked-art-test.library.yale.edu/node/4102fbd6-3f59-44dd-ba5c-58fba37b8e0d</a>&quot;</td>
</tr>
<tr>
<td>type:</td>
<td>&quot;Person&quot;</td>
</tr>
<tr>
<td>_label:</td>
<td>&quot;Dion, Joan&quot;</td>
</tr>
<tr>
<td>classified_as:</td>
<td></td>
</tr>
<tr>
<td>0:</td>
<td></td>
</tr>
<tr>
<td>id:</td>
<td>&quot;<a href="https://linked-art-test.library.yale.edu/node/7a13c013-ee4b-41dd-a721-c4c2742767f6">https://linked-art-test.library.yale.edu/node/7a13c013-ee4b-41dd-a721-c4c2742767f6</a>&quot;</td>
</tr>
<tr>
<td>type:</td>
<td>&quot;Type&quot;</td>
</tr>
<tr>
<td>_label:</td>
<td>&quot;Creator&quot;</td>
</tr>
<tr>
<td>1:</td>
<td></td>
</tr>
<tr>
<td>type:</td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td>label:</td>
<td>&quot;Creation&quot;</td>
</tr>
<tr>
<td>carried_out_by:</td>
<td></td>
</tr>
<tr>
<td>0:</td>
<td></td>
</tr>
<tr>
<td>id:</td>
<td>&quot;<a href="https://linked-art-test.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a736ab0">https://linked-art-test.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a736ab0</a>&quot;</td>
</tr>
<tr>
<td>type:</td>
<td>&quot;Person&quot;</td>
</tr>
<tr>
<td>_label:</td>
<td>&quot;Farr, Kimberly&quot;</td>
</tr>
<tr>
<td>classified_as:</td>
<td></td>
</tr>
<tr>
<td>0:</td>
<td></td>
</tr>
</tbody>
</table>
Normalizing and Merging
Normalization and Merging

String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel(source: "its-d10", type: "Person", normalizedValue);
if (personEntity == null) {
Normalization and Merging

```java
String normalizedValue = TransformerHelper.stringNormalization(value, includePunctuation: "() ");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
}```
Normalization and Merging

String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()");  
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel(source: "ils-bib", type: "Person", normalizedValue);

if (personEntity == null) {
Normalization and Merging

String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()");
TopLevelEntity personEntity = linkedDataService.findBYSourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
    personEntity = new TopLevelEntity( source: "ils-bib", type: "Person", normalizedValue);
    linkedDataService.save(personEntity);
    Node personAgentNode = linkedDataService.createNodeWithTypeAndLabel( type: "Person", value);
    personAgentNode.setId(personEntity.getId());
    if (equivalent != null) {...}
    Node identifyNode = linkedDataService.getNodeFactory().primaryName(value);
    linkedDataService.createRelationship(personAgentNode, relationship: "identified_by", identifyNode);
    linkedDataService.updateEntityJson(personEntity, personAgentNode);
}
Node creationNode = bibNode.findOrCreateChildNode( type: "Creation", relationship: "created_by");
Node partNode = linkedDataService.createNodeWithTypeAndLabel( type: "Creation", label: "Creation");
linkedDataService.createRelationship(creationNode, relationship: "part", partNode);
Normalization and Merging

```java
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
    personEntity = new TopLevelEntity( source: "ils-bib", type: "Person", normalizedValue);
    linkedDataService.save(personEntity);
    Node personAgentNode = linkedDataService.createNodeWithTypeAndLabel( type: "Person", value);
    personAgentNode.setId(personEntity.getId());
    if (equivalent != null) {
        Node identifyNode = linkedDataService.getNodeFactory().primaryName(value);
        linkedDataService.createRelationship(personAgentNode, relationship: "identified_by", identifyNode);
        linkedDataService.updateEntityJson(personEntity, personAgentNode);
    }
}
Node creationNode = bibNode.findOrCreateChildNode( type: "Creation", relationship: "created_by");
Node partNode = linkedDataService.createNodeWithTypeAndLabel( type: "Creation", label: "Creation");
linkedDataService.createRelationship(creationNode, relationship: "part", partNode);
```
Database Schemas
Database Diagram

<table>
<thead>
<tr>
<th>Database Diagram</th>
</tr>
</thead>
</table>

1. **top_level_entity**
   - id: bigint
   - create_date_time: timestamp
   - update_date_time: timestamp
   - version: bigint
   - aspace_uri: varchar(255)
   - external_id: varchar(255)
   - ils_uri: varchar(255)
   - json: text
   - label: text
   - loaded: boolean
   - source: varchar(255)
   - source_modification_date: timestamp
   - source_version_number: bigint
   - type: varchar(255)

2. **activity_stream_entry**
   - id: bigint
   - activity_stream: varchar(255)
   - create_date_time: timestamp
   - document_type: varchar(255)
   - end_time: timestamp
   - type: varchar(255)
   - uuid: varchar(255)

3. **record_id**
   - id: bigint
   - aspace_uri: varchar(255)
   - ils_uri: varchar(255)
   - label: text
   - source: varchar(255)
   - original_id: varchar(255)

4. **top_level_entity_metadata**
   - id: bigint
   - external_id: varchar(255)
   - label: text
   - metadata_type: varchar(255)
   - subject_id: varchar(255)

5. **top_level_entity_relationship**
   - id: bigint
   - object_id: varchar(255)
   - relationship: varchar(255)
   - subject_id: varchar(255)
Original Database Plan: Triple based

- Relationships
- subject
- object
- Entities
  - Every JSON Object
  - No JSON stored in the Database
Original Database Plan: Triple based

Worked until we had about 10% of the data in the system. We had to find another solution so we could have a faster turn around.
Simplified TopLevelEntity Design

TopLevelEntities
JSON stored in the database

Worked for our full set of data. Reingest is about 48 hours.
Simplified TopLevelEntity Design

TopLevelEntities
JSON stored in the database

Relationships
Stored as paths

Worked for our full set of data. Reingest is about 48 hours.
Some Statistics
Stats:

12,799,915 Holdings Records

11,962,919 Bibliographic Records

2M+ Archives Space Records

50,006,415 TopLevelEntities

193,784,739 TopLevelEntityRelationships

24,104,513 TopLevelEntityMetadata

50,133,873 ActivityStreamEntries
Thank you!