Find and Combine Vocabularies to Design Metadata Application Profiles Using Schema Registries and LOD Resources

Tsunagu Honma†  Mitsuharu Nagamori‡  Shigeo Sugimoto‡

† Graduate School of Library, Information and Media Studies, University of Tsukuba
‡ Faculty of Library, Information and Media Studies, University of Tsukuba
Goal: To Support the Finding of Metadata Schemas

• Reusing existing schemas for designing new schemas

• Finding and narrowing existing schemas using two requirements from the schema designer
  – Names of Metadata Attributes
  – Examples of Metadata Values
Metadata Schema Design

• When Designing schema for creative works such as movies, audio books and novels ...
Metadata Schema Design

- A metadata schema designer defines some names of metadata attributes

http://en.wikipedia.org/wiki/Harry_Potter_and_the_Philosopher's_Stone
**Approach A:** Finding Existing Schemas by Names of Metadata Attributes

- A metadata schema designer finds and reuses existing schemas by names of attributes.

![Diagram showing metadata schema designer finding existing schemas through schema registries like LOV](image)
Approach A: Finding Existing Schemas by Names of Metadata Attributes

Barriers

- A metadata schema designer finds and reuses existing schema by names of attributes

- Machine-readable metadata schemas are often implicit
  - It is difficult to find metadata schemas

- A huge number of metadata schemas exist
  - Even if schemas are explicit, it would be hard for a designer to look through the huge amount of existing schemas
A Designer has
Examples of Metadata Values

• A metadata schema designer also has **examples of metadata values**

<table>
<thead>
<tr>
<th>Names</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>J. K. Rowling</td>
</tr>
<tr>
<td>Illustrators</td>
<td>Thomas Taylor (UK)</td>
</tr>
<tr>
<td>Genre</td>
<td>Fantasy</td>
</tr>
<tr>
<td>Publishers</td>
<td>Bloomsbury (UK)</td>
</tr>
<tr>
<td>Released</td>
<td>26 June 1997 (UK)</td>
</tr>
<tr>
<td></td>
<td>Arthur A. Levine/Scholastic (US)</td>
</tr>
<tr>
<td></td>
<td>1 September 1998 (US)</td>
</tr>
</tbody>
</table>
Metadata Instances are Explicit

• On the other hand, a large amount of metadata instances have been made explicit as LOD datasets
**Approach B**: Finding Existing Schemas by using Examples of Metadata Values

- A metadata schema designer can find and reuse existing instances using examples.
Approaches for Finding Metadata Schema

A. By names of Metadata Attributes
   1. Register metadata schemas/vocabularies
   2. Find metadata terms using names of metadata attributes
   3. Detect schema/vocabularies by the namespaces of metadata terms

B. By Examples of Metadata Values
   1. Register metadata instances
   2. Find metadata terms using examples of metadata values
   3. Detect schema/vocabularies by the namespaces of metadata terms
An Example of Approach A: Finding Terms for a Title

```
SELECT distinct ?term
WHERE {
  { ?term rdfs:subPropertyOf* rdf:Property . }
  UNION
  {
    ?c rdfs:subPropertyOf* rdf:Property .
  }
  FILTER isIRI(?term) .
  FILTER REGEX(?o, ".*(title|label|name).*", "i") .
}
ORDER BY ?term
```
An Example of Approach A: Finding Terms for a Title

```sql
SELECT distinct ?term
WHERE {
  { ?term rdfs:subPropertyOf* rdf:Property . } 
  UNION 
  { ?c rdfs:subPropertyOf* rdf:Property . 
    ?term rdf:type ?c . }
FILTER isIRI(?term) .
FILTER REGEX(?o, ".*(title|label|name).*", "i") .
ORDER BY ?term
```

A name given to the resource.
An Example of Approach B:
Finding Terms for “George Lucas”

```
SELECT distinct ?term
WHERE {
  { ?s ?term "George Lucas" . }
  UNION
  {
    ?o rdfs:label "George Lucas" .
  }
}
ORDER BY ?term
```
An Example of Approach B: Finding Terms for “George Lucas”

```sql
SELECT distinct ?term
WHERE {
  { ?s ?term "George Lucas" . }
  UNION
  {
    ?o rdfs:label "George Lucas" .
  }
}
ORDER BY ?term
```

![Diagram showing the relationships between dc:creator and rdfs:label properties with George Lucas as a node.](image)
Case Study: Designing Fiction Metadata

- Designing a schema for creative works such as movies, audio books and novels
Preparation

• Vocabularies
  – source: prefix.cc
  – 450 vocabularies from prefix.cc
  – 5181 properties

• Instances
  – source: LOD Cloud Cache
  – 2 million graphs
  – 51 billion triples
# Designer Requirements

<table>
<thead>
<tr>
<th>Metadata Attributes</th>
<th>Names of Metadata Attributes</th>
<th>Given Examples of Metadata Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title of a Fiction Work</td>
<td>Title, Label, Name</td>
<td>Star Wars, Harry Potter and the Philosopher’s Stone, The Lord of the Rings, Dragon Ball</td>
</tr>
<tr>
<td>Creator of a Fiction Works</td>
<td>Creator, Author, Director, Writer, Maker, Composer, Contributor</td>
<td>George Lucas, J.K. Rowling, John Ronald Reuel Tolkien, Akira Toriyama</td>
</tr>
<tr>
<td>Character of a Fiction Works</td>
<td>Character, Cast</td>
<td>Darth Vader and Luke Skywalker, Harry Potter and Hermione Jean Granger, Frodo Baggins and Gandalf, Son Goku and Begeta</td>
</tr>
</tbody>
</table>
Result: Finding Schema using Designer Requirements

<table>
<thead>
<tr>
<th></th>
<th>A) by names of metadata attributes</th>
<th>B) by metadata instances</th>
<th>A OR B</th>
<th>A AND B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>468 terms (84 vocabularies)</td>
<td>51 terms (29 vocabularies)</td>
<td>510</td>
<td>9</td>
</tr>
<tr>
<td>Creator</td>
<td>120 terms (44 vocabularies)</td>
<td>102 terms (29 vocabularies)</td>
<td>220</td>
<td>2</td>
</tr>
<tr>
<td>Character</td>
<td>91 terms (25 vocabularies)</td>
<td>104 terms (40 vocabularies)</td>
<td>195</td>
<td>0</td>
</tr>
</tbody>
</table>
Result: Finding Schema using a Designer Requirements (LOV Aggregator)

<table>
<thead>
<tr>
<th></th>
<th>A) by names of metadata attributes</th>
<th>B) by metadata instances</th>
<th>A OR B</th>
<th>A AND B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>436 terms (69 vocabularies)</td>
<td>51 terms (29 vocabularies)</td>
<td>478</td>
<td>9</td>
</tr>
<tr>
<td>Creator</td>
<td>215 terms (42 vocabularies)</td>
<td>102 terms (29 vocabularies)</td>
<td>315</td>
<td>2</td>
</tr>
<tr>
<td>Character</td>
<td>127 terms (23 vocabularies)</td>
<td>104 terms (40 vocabularies)</td>
<td>231</td>
<td>0</td>
</tr>
</tbody>
</table>
Result 1: Approach A is not enough to find schema for a title

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Terms</th>
<th>Vocabularies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>468</td>
<td>84</td>
</tr>
<tr>
<td>Creator</td>
<td>120</td>
<td>44</td>
</tr>
<tr>
<td>Character</td>
<td>91</td>
<td>25</td>
</tr>
</tbody>
</table>

*Names of Metadata Attributes*

- Title, Label, Name
- Creator, Author, Director, Writer, Maker, Composer, Contributor
- Character, Cast
A part of results: Terms for a Title

Approach A (found 468 terms including 384 terms not using in LOD Cloud Cache)

- http://langegger.at/xlwrap/vocab#fileName
- http://purl.org/dc/elements/1.1/rights
- http://spinrdf.org/sp#graphNameNode
- http://www.cidoc-crm.org/cidoc-crm/P102_has_title
- http://www.loc.gov/mads/rdf/v1#code
Result 2: Approach B made good results to find schema for a title

<table>
<thead>
<tr>
<th></th>
<th>B) by metadata instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>51 terms (29 vocabularies)</td>
</tr>
<tr>
<td>Creator</td>
<td>102 terms (29 vocabularies)</td>
</tr>
<tr>
<td>Character</td>
<td>104 terms (40 vocabularies)</td>
</tr>
</tbody>
</table>

Given Examples of Metadata Values:

- Star Wars,
  Harry Potter and the Philosopher’s Stone,
  The Lord of the Rings,
  Dragon Ball

- George Lucas,
  J.K. Rowling
  John Ronald Reuel Tolkien,
  Akira Toriyama

- Darth Vader and Luke Skywalker,
  Harry Potter and Hermione Jean Granger,
  Frodo Baggins and Gandalf,
  Son Goku and Begeta
A part of results: Terms for Title

Approach B (51 terms including 28 undefined terms)

- http://climb.dataincubator.org/vocabs/climb/route
- http://data.linkedmdb.org/resource/movie/film_series_name
- http://ogp.me/ns#title
- http://purl.org/ontology/po/episode
- http://xmlns.com/foaf/0.1/primaryTopic
- http://rdf.freebase.com/ns/type.object.name
- http://www.w3.org/2006/vcard/ns#organization-name
- http://xmlns.com/foaf/0.1/maker
Terms for a Creator

Approach A (120 terms including 89 terms not using in LOD Cloud Cache)
- http://ndl.go.jp/dcndl/terms/seriesCreator
- http://schema.org/creator
- http://xmlns.com/foaf/0.1/publications
- http://www.w3.org/2011/http#authority

Approach B (102 terms including 62 undefined terms)
- http://data.linkedmdb.org/resource/movie/director_name
- http://dbpedia.org/ontology/author
- http://dbpedia.org/property/author
- http://purl.org/ontology/po/participant
- http://rdf.basekb.com/ns/m.03k0ppw
- http://schema.org/name
- http://yago-knowledge.org/resource/isMarriedTo
- http://www.w3.org/2006/vcard/ns#fn
Terms for Character

Approach A  (91 terms including 82 terms not using in LOD Cloud Cache)

- http://purl.org/ontology/cco/core#characteristic
- http://schema.org/actor
- http://www.semanticdesktop.org/ontologies/2007/01/19/nie#characterSet
- http://www.w3.org/2000/10/swap/string#replace
- http://www.w3.org/2008/content#characterEncoding

Group 91 terms by Vocabularies: 25 Domains: 34
Result 3: There is a few Intersection of both approaches

<table>
<thead>
<tr>
<th>For Title</th>
<th>For Creator</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://ogp.me/ns#title">http://ogp.me/ns#title</a></td>
<td><a href="http://purl.org/dc/terms/contributor">http://purl.org/dc/terms/contributor</a></td>
</tr>
<tr>
<td><a href="http://purl.org/dc/elements/1.1/title">http://purl.org/dc/elements/1.1/title</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://purl.org/dc/terms/title">http://purl.org/dc/terms/title</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://purl.org/stuff/rev#title">http://purl.org/stuff/rev#title</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://schema.org/name">http://schema.org/name</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.w3.org/2000/01/rdf-schema#label">http://www.w3.org/2000/01/rdf-schema#label</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.w3.org/2004/02/skos/core#prefLabel">http://www.w3.org/2004/02/skos/core#prefLabel</a></td>
<td></td>
</tr>
<tr>
<td><a href="http://xmlns.com/foaf/0.1/name">http://xmlns.com/foaf/0.1/name</a></td>
<td></td>
</tr>
</tbody>
</table>
Lessons Learned

1. Broken links in metadata vocabularies and instances
2. Standardize ways of publishing metadata vocabularies
3. Publish information about LOD datasets
4. Lack of definitions of metadata terms
5. Less use of registered terms in instances
Lessons Learned 1: Broken links in metadata vocabularies and instances

• A lot of metadata vocabularies and instances are inaccessible
  – moved or deleted resources
  – server unavailable
  – shut down hosting services
Lessons Learned 2: Standardize ways of publishing metadata vocabularies

• Ideal:
  – When a designer accesses a namespace URI (with HTTP Content-Negotiation), the designer should get a definition of the metadata vocabulary in RDF

• Reality:
  – In some cases, at the namespace URI, there is an HTML file including a link to an RDF file
  – Various ways of navigating from a namespace URI to a file
    • Embed RDF into HTML (RDFa, Microdata, GRDDL)
    • Include links to RDF (RDDL, <link rel="alternate" />, <a href="..." />)
Lessons Learned 3: Publish information about LOD datasets

• Some information is useful for a designer to evaluate interoperability, domain specificity and freshness
  – Created/modified date
  – number of outbound links from same/other subject domains

• The Datahub only publishes information about some datasets
Lessons Learned 4: Lack of definitions of metadata terms

• For approach A, it is important to define details about metadata terms
• Defining Description Set Profiles (DSP) which express concrete usages of vocabularies

<table>
<thead>
<tr>
<th>label</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc:title</td>
<td>Title The name given to the resource</td>
</tr>
<tr>
<td>DSP for fiction works</td>
<td>Main Title The main title of a fiction work.</td>
</tr>
</tbody>
</table>
Lessons Learned 5:
Less use of registered terms in instances

• Some metadata vocabularies are defined but not used
• And some metadata instances use metadata terms which are undefined
To improve our Approaches: Requests for Designers

• Define names for metadata attributes and resource classes
  – Designers should prepare words which adequately express the features of attributes
  – If designers define names for resource classes, those names are also useful to narrow schemas

• Define examples of metadata values
  – Designers should prepare proper nouns as examples which may appear only for specific metadata attributes
To improve our Approaches: Requests for Data Publishers

• Publish datasets with metadata
  – created/updated date, number of outbound links to same/other subject domains
  – this information is useful for designers to evaluate schema interoperability, domain specificity and freshness

• Publish datasets with machine-readable schema
  – DSP define concrete usage of metadata vocabularies
  – we are going to use DSP with definitions about metadata vocabularies to find schema by names of metadata attributes
Conclusion

• Goal:
  – To support the finding of metadata schema by reusing existing schema

• Approaches:
  – A: By names of metadata attributes
  – B: By examples of metadata values

• Results:
  – It is hard to narrow schemas using only names of metadata attributes
  – Designers should prepare proper nouns as examples which may appear only for a specific metadata attribute
Future Works

• Maintain Registry for Metadata Vocabularies and Metadata Instances
  – Metadata Vocabularies and Instances are updated, deleted and become inaccessible
  – In LOD Cloud Cache, there are many instances and terms which already cannot be accessed

• More Information about Metadata Instances
  – Creators, Subjects, Users, Created Date, ...
    • What kind of subject domain is each metadata term often used
    • When the metadata terms are often used
Structure of an Application Profile: Singapore Framework

Functional Requirements → Domain Model → Description Set Profile (DSP) → Syntax Guidelines and Data Formats

Community Domain Model uses RDF Schema

Metadata Vocabularies uses DCMI Abstract Model

DCMI Guidelines built on RDF

Domain Model built on RDF Schema

Application Profile built on Foundation Standards
Define Structural Constraints

Functional Requirements built on Domain Model built on Description Set Profile (DSP) built on Syntax Guidelines and Data Formats

Domain Model uses Metadata Vocabularies uses DCMI Abstract Model built on DCMI Syntax Guidelines

Description Set Profile (DSP) uses RDF Schema built on RDF

Usage Guidelines annotate

Each Metadata Attribute are assigned Metadata Terms

Foundation Standards

Domain Standards built on Application Profile built on Application Profile

DCMI Syntax Guidelines built on RDF

DCMI Abstract Model built on RDF

RDF Schema built on RDF

Metadata Vocabularies uses Community Domain Model

Applicaton Profile built on Functional Requirements

Abstract Model built on RDF

Usage Guidelines

Structural Constraints

Structure of an Application Profile: Singapore Framework
Requirements which a Metadata Schema Designer has

- Names of Metadata Attributes
  - title, author, published date, ...

- Examples of Metadata Values
  - “Star Wars”, “George Lucas”, “1977”, ...
Select Metadata Terms to build DSP

• There are a lot of existing metadata terms

• A schema designer select those terms for metadata interoperability
Select Metadata Terms to build DSP

Barriers

- There are a lot of existing metadata terms

- No metadata terms dictionary
  - It is difficult to find and compare metadata terms with those definitions

- A huge number of metadata terms
  - Even if we make a dictionary, a designer could not look all the huge terms
Goal: Support Finding Metadata Terms

• Access and Accumulate Definitions about Existing Metadata Terms

• Narrow metadata terms as candidates using requirements which a schema designer has
Register Definitions about Metadata Terms

• Lists of Metadata Vocabularies
  – Prefix.cc
  – Linked Open Vocabularies (LOV)

• Access namespace URIs and register definitions in an RDF Repository
  – HTTP Content-Negotiation
  – RDFa, Microdata, GRDDL, RDDL
A Process of Schema Designing

• 既存のガイドラインにスキーマ作成のおおまかな手順が書かれている
  －Guidelines of DCAP, Ontology Development 101, メタ基盤報告書など

• 目的を設定する
• 記述対象とその関連を定義する
• 記述項目に制約を与える（タームの決定など）