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

Tsunagu Honma
Graduate School of Library, Information and Media Studies, University of Tsukuba, Japan
tsuna@slis.tsukuba.ac.jp

Mitsuharu Nagamori
Faculty of Library, Information and Media Science, University of Tsukuba, Japan
nagamori@slis.tsukuba.ac.jp

Shigeo Sugimoto
Faculty of Library, Information and Media Science, University of Tsukuba, Japan
sugimoto@slis.tsukuba.ac.jp

Abstract

A metadata schema which defines constraints about metadata records is a fundamental resource for metadata interoperability. Building interoperable metadata schemas has been a main topic of the Dublin Core since its early days. It is important to make use of existing metadata schemas to develop a new schema in order to minimize newly defined metadata vocabularies, which is how DCMI has developed. In order to improve the usability of existing metadata schemas for developing new schemas, it is important to improve the usability of the publicly available, online information about metadata schemas. This study aimed to develop a technology to help metadata schema designers find useful metadata schemas and use them for the new metadata schema development. Key concepts used in this study are Description Set Profiles (DSP) as a formal basis of metadata schema and Linked Open Data (LOD) as a framework to connect metadata schema resources. In this paper, we discuss a search methodology to find useful metadata terms for a given application domain. We propose to apply two approaches – (1) search metadata terms and description set profiles using resources registered at schema registries and the like, and (2) search metadata terms using metadata instances included in a LOD dataset. We created a search scenario for metadata schema of fictions and applied the methodology to evaluate the methodology. This paper shows the methodology and its evaluation using the scenario after discussion of general requirements analysis and guidelines for metadata schema development.

Keywords: application profiles; metadata schema design; metadata vocabulary selection

1. Introduction

There are a huge number of metadata created and published by various communities on the Web. It is common for Web users to use those metadata for many different purposes, e.g., finding aids, access, evaluation, rating, and so forth. Thus, users often employ different metadata and combine them, which means we use several different metadata sets expressed in different forms, i.e. different languages to express the metadata. In the current Web environment, there is a common understanding of the importance of metadata to link resources and data, which is known as Linked Open Data (LOD)[1]. LOD encourages semantic linking of resources across different communities. It is important for the communities to make information about metadata schema available on the Web in order to enhance the usability of the metadata published by the different communities on the Web.

A large number of Web resources have metadata embedded and are linked to other resources using HTTP-URIs. However, the schemas of those embedded metadata are often implicit or not well defined. We proposed a method for extracting metadata from HTML Documents using Dublin Core Description Set Profiles (DSP) that describe constraints about metadata (Honma, et al., 2012). In that study, we used DSPs manually created from the domain resources, such as a set of news articles on the Web. It is advantageous to reuse existing metadata vocabularies such as

[1] http://linkeddata.org/

the Dublin Core and FOAF to create DSPs in order to enhance metadata interoperability. From this study, we have learned that we need functions to find the appropriate metadata terms and vocabularies (Heath, et al., 2011), and to structure a DSP using the terms and vocabularies in accordance with the domain requirements.

However, it is not realistic to assume that every Web resource is accompanied by a metadata schema definition because the cost of creating schemas is relatively high. In this paper, we propose an approach to support metadata schema creation by helping users find and combine metadata vocabularies to create a schema.

2. Creating Metadata Schema for Metadata Interoperability

When we create a metadata schema, we need to understand the requirements that the metadata has to satisfy, e.g., an underlying data model, controlled vocabularies defined for the metadata, data exchange formats, and so on. The data structure and constraints derived from the requirements should be documented in a (semi-)formal format as a metadata schema so metadata designers can describe well-structured metadata and so users can precisely understand the metadata. As metadata is a crucial component for users to find, select, and access resources on the Internet, metadata and metadata schemas should be implemented in an interoperable format. Resource Description Framework (RDF) and Web Ontology Language (OWL) are well known technologies to make metadata and metadata schemas shareable across communities on the Internet. It is widely recognized that sharing metadata schemas expressed in a widely used standard format is key to enhancing metadata interoperability. However, in practice, it is not easy to develop interoperable metadata schemas because both subject domain knowledge and metadata technology knowledge are required for the development.

The Dublin Core Application Profile (DCAP) is a well-known framework for interoperable metadata schemas. The underlying concept of DCAP is "mixing and matching metadata schemas" (Heery et al., 2000), which means to collect metadata terms from existing vocabularies and combine them to form a new metadata schema. In other words, 'find and combine metadata terms' is the central concept for creating a metadata schema. A straightforward solution to minimize the cost of developing interoperable metadata schema on the Internet is to build a support tool for metadata designers to develop metadata schema based on a well-formed guidelines, using DCAP and LOD technologies.

Dublin Core Application Profiles

The Dublin Core Application Profile (DCAP) framework is used to describe interoperable metadata. DCAP is about combining available vocabularies and creating appropriate metadata schemas for the purposes of each community. The Singapore Framework of DCAP has the following five components (Nilsson, et al., 2008).

1. Functional Requirements: Description of the purpose of the metadata, and what functions communities provide for their own purposes
2. A Domain Model: Expression of the classes of resources and relationships between them
3. Description Set Profiles: Definition of the structural constraints of the metadata records
4. User Guidelines: A set of guidelines for users implementing and using the metadata records
5. A Syntax Encoding Scheme: Syntax for the definition of data in an implemented system and/or for data exchange

The Singapore Framework suggests metadata schema designers should include these components in their metadata schema definitions.

DCMI Description Set Profiles

When we create a new metadata schema, existing schemas provide us with useful information. Those schemas need not be exactly of the same domain but can be reused or customized. For example, when a metadata designer defines a metadata schema (i.e., DSP) for books and another schema for authors of the books, they need to define classes and properties for a book, a person and relationships between them. If the designer knows an available schema for books and another for people, they could reuse and customize the schemas for their service. If they do not know any re-usable schemas, they might want to search for re-usable schemas. The DCMI guideline (Coyle et al., 2009) for creating an application profile mentions that metadata designers should "*scan available RDF vocabularies to see whether the properties needed have already been declared and are available for use*" for metadata interoperability.

3. Related Works

There are several studies of the efficient development of interoperable metadata schema as shown in the following paragraphs. These studies suggest finding and combining vocabularies to build a target schema.

3.1 Guidelines for Reusing Metadata Schemas and Ontologies

(1) Guidelines for Dublin Core Application Profiles (Coyle, et al., 2009)

In 2009, Karen Coyle and Thomas Baker proposed Guidelines for Dublin Core Application Profiles. These guidelines explain a framework for DCAP and give an example process for designing DCAP. The step "*Selecting or Defining Metadata Terms*" in the process shows a method of selecting vocabularies. In this step, metadata schema designers answer questions which make clear the requirements for describing property values. In order to satisfy those requirements, metadata schema designers find and select appropriate vocabularies.

(2) Ontology Development 101 (Noy, et al., 2001)

A process of designing metadata schema is similar to ontology creation. Ontology Development 101 is a well-known guideline for creating an ontology. This guideline showing a process of ontology development includes steps "*Consider reusing existing ontologies*", "*Define the properties of classes - slots*" and "*Define the facets of the slots*". These steps lead us to reuse existing ontologies and define new relationships among instances to form a new ontology. Selecting and combining existing terms and designing ontologies are similar processes in metadata schema design.

(3) Guidelines for Sharing Information about Metadata (MI3, 2011)

The guideline for metadata sharing is created as a part of the metadata schema registry project supported by the Ministry of Internal Affairs and Communications (MIC) of Japan. This guideline is based on three major tasks - selecting existing metadata schemas for re-use, designing application profiles on top of the collected metadata terms, and providing the metadata schema information for people who are involved in metadata schema development but have no in-depth knowledge about sharing a metadata schema and its information on the Web. The guideline strongly recommends re-use of existing schemas in order to enhance metadata interoperability

All of these three guidelines suggest re-using existing schemas. Issues made obvious by these guidelines are,

- How to find and select schemas appropriate for re-use,
- How to define new terms or vocabulary and link them to existing ones,
- How to properly align new terms with existing ones, and

- How to effectively provide the schema for metadata designers who may not have technological knowledge about interoperability of metadata on the Web.

3.2 Functions and Services to Provide Metadata Schema Information on the Web

The most fundamental requirement for sharing metadata schemas on the Internet is to identify every instance using URI, especially HTTP-URI as suggested in LOD. In fact, metadata terms of a vocabulary are often published in the vocabulary namespace specified by an HTTP-URI. Metadata schema designers can get documentation of metadata terms and vocabularies expressed in a formal language, e.g., RDF Schema and Web Ontology Language. As there are a lot of vocabularies for describing metadata which are not known by most metadata schema designers, vocabulary search engines and metadata schema registries have a crucial role for designers, e.g. Linked Open Vocabularies[2], DCMI Metadata Schema Registry[3], Open Metadata Schema Registry[4], and so on. These services accumulate information about vocabularies such as definitions of terms and examples of metadata descriptions, and provide finding aids for these resources.

4. Requirements Analysis Methodologies for Finding and Combining Vocabularies

In order to find metadata terms from one or more vocabularies and to design appropriate combinations of them to build an application profile, it is necessary to clarify the criteria for finding and combining metadata terms. Moreover, it is important to prepare a service by which metadata schema designers can find the information useful for metadata vocabulary searching and re-use. In this section, we discuss the criteria for selecting vocabularies based on the guidelines described in the previous section. Then, we summarize the requirements for supporting the selection of vocabularies.

The Guidelines for DCAP say that metadata schema designers should consider the following questions about vocabularies and terms (Coyle, et al., 2009).

- Is a value free text or a URI?
- Is the format or class of the value given in advance, for example, W3C-DTF for a property to express time-and-date, and Agent class to express a person or an organization?
- Should the value be selected from existing controlled vocabularies?
- Does a simple value suffice the requirement, or does the value have to have a complex structure? For example, author of a book may be a simple text string or a set of values such as family and given names, affiliation, contact address, date of birth, etc.

These questions are essential when defining a value range of a property and an expression scheme of a property value.

Ontology 101 (Noy, et al., 2001) mainly proposes how to create an ontology based on existing ontologies. It mentions a process to extend existing ontologies to satisfy the domain requirements. In this guideline, it is necessary for ontology developers to cope with the following issues.

- Interoperability in the domain has to be taken into consideration when a general concept is brought into the ontology and when a detailed concept is to also broaden its meaning.
- Define cardinality, data type, domain and range of a property in order to make clear, constraints for describing metadata.

---

[2] http://lov.okfn.org/
[3] http://dcmi.kc.tsukuba.ac.jp/dcregistry/
[4] http://metadataregistry.org/

Guidelines for Sharing Information about Metadata (MI3, 2011) addresses the following points for designing interoperable metadata schema.

- Re-use existing schemas used in the same domain or the neighboring domains
- Adopt well known standard vocabularies
- Define relationships between domain-specific metadata terms and well-known terms to help understanding and for simplifying the domain metadata vocabulary.
- Do not deviate from a term definition so as not to make the meaning of a term ambiguous

From all of these guidelines, we have summarized the requirements for selecting vocabulary as follows.

- Reuse metadata vocabularies and schema where possible
- Adopt well known standard vocabularies
- Define relationships between a specific vocabulary and the well known standard vocabularies
- Refer definitions about vocabularies and terms, e.g., domain, range, data type, explanation and so forth
- Provide access to examples of metadata description

Metadata schema designers should have the above points in mind when designing their schemas. 'Find and combine metadata terms' is a common key aspect in these guidelines. As mentioned previously, metadata vocabulary search engines are a key service on the Web to help finding and combining of metadata terms.

5. Finding and Combining Metadata Vocabularies using Metadata Vocabularies and LOD Resources

For the best selection of metadata terms and their combinations, metadata schema designers need access to information about vocabularies, schema and metadata instances. LOD suggests making the metadata schema information open on the Web. Vocabulary search engines help us find and use metadata terms. In this section, we propose a technology to help users find metadata terms information and resources using vocabulary search engines and existing LOD resources.

In the existing guidelines for metadata schema designing, metadata schema designers are recommended to reuse existing vocabularies and/or customize existing DSPs based on the requirements of the metadata and its use in their applications. The Singapore Framework proposes metadata schema designers select existing vocabularies for defining a DSP based on functional requirements and domain models. So metadata schema designers already decide metadata resources, attributes and examples of metadata values when they find and select metadata terms. We propose the following procedure to define a metadata vocabulary for an application, which has two phases – vocabulary search and selection.

(1) Metadata Vocabulary Search

(a) Search by Vocabularies

Metadata schema designers find appropriate vocabularies using vocabulary search engines and schema registries. Vocabulary search engines provide definitions about vocabularies, and schema registries provide usage of the vocabularies in specific domains and applications. A metadata schema designer may use both search engines and registries; for example, when they want to find terms which express "A title of a resource", they search vocabulary search engines and schema registries for terms by keywords, such as "title", "label" and their synonyms. Terms found by these tools are candidate terms to be used in the target DSP.

(b) Search by Metadata Instances

Another effective approach is to look for metadata instances which describe resources in the same subject domain or close to the target domain. A metadata schema designer may extract an element name, i.e., a property term, from metadata instances. For example a book database search using "Harry Potter and the Philosopher's Stone" would return a title text paired with its property term which might mean "title".

(2) Vocabulary Selection

Metadata schema designers have to select appropriate terms from vocabularies found in the search phase. Metadata schema designers are recommended to select well-known vocabularies rather than not-well-known ones from the vocabularies found in the search but it is not straightforward to judge which vocabulary is better-known and which vocabulary is better suited to the application. So, in this paper, we define a better-known vocabulary as one frequently used in metadata instances in existing LOD resources, which can be judged from the resources such as Data Hub and metadata schema registries.

6. Case Study – a case scenario to find metadata terms from LOD resources and some lessons learned

This section shows a case study of finding and combining metadata vocabularies based on the approach shown in the previous section.

Goal

To design a DSP for describing fictions, e.g. Star Wars, Harry Potter, Dragon Ball, which may be a movie, printed book, an electronic book, an anime, or a graphic novel.

Scenario

In this case study, a metadata schema designer tries the following steps.

1. Define functional requirements, resources and attributes
2. Provide examples of metadata values
3. Search terms by vocabularies
4. Search terms by metadata instances
5. Select terms

A metadata schema designer has requirements for their metadata applications. First, they crystalize their functional requirements and what resources and attributes are needed for describing metadata, i.e. Functional Requirements and Domain Models of the Singapore Framework. In this case study, a fiction is chosen because there are a variety of genres and types of manifestations to which conventional bibliographic metadata may not be appropriate. In theory, the functional requirements and domain models should be clarified before the development of the DSP, but, in reality, it is not easy, so that we assume that the designer knows some minimum requirements and the data model sufficiently to create sample queries to search for existing schemas.

Provide Vocabularies and Instances

In advance to the case study, we have collected vocabularies using prefix.cc[5]. Prefix.cc is a web service which provides a dump file including namespace URIs and their prefixes. In order to accumulate definitions about vocabularies, we downloaded that dump file and accessed their namespace URIs using HTTP Redirection.

[5] http://prefix.cc/

- **Vocabularies**: 450 vocabularies (e.g., dcterms, foaf and skos) in RDF Sesame, (note: RDF Sesame is an RDF Triples Store.)
- **Instances**: over 300 million graphs in the LOD Cloud Cache[6]

Search terms for designing metadata schema about fiction works

A metadata schema designer provides metadata values for each metadata attribute before designing the DSP. Table 1 shows metadata attributes and values which the designer provides. In this case study, we suppose the designer has examples of metadata values about three metadata attributes and four fiction works.

TABLE 1: Requirements for designing a metadata schema about Fiction Works

Metadata Attributes	Words for Metadata Attributes	Given Examples of Metadata Values
Title of fiction work	title, label	Star Wars, Harry Potter and the Philosopher's Stone, The Lord of the Rings, Dragon Ball
Creator of fiction work	creator, author, director, writer	George Lucas, J.K. Rowling, John Ronald Reuel Tolkien, Akira Toriyama
Character of fiction work	character, cast	Darth Vader and Luke Skywalker, Harry Potter and Hermione Jean Granger, Frodo Baggins and Gandalf, Son Goku and Begeta

(1) Search terms by vocabularies

As the given requirement is "describing titles of fiction works", a keyword "title" and its synonyms (e.g. "label") are given as a query to a SPARQL endpoint which has definitions of vocabularies. Figure 1 shows an example of a SPARQL query for a list of terms for describing a title of a resource. This query searches all properties and filters out terms which don't have "title" or "label" in the definition of terms.

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

FIG. 1. A SPARQL query for searching terms as a "title" or "label" by a vocabulary search engine

A metadata schema designer also changes the conditions of filters and executes SPARQL queries for finding terms such as "creator" and "character". In this case study, the vocabulary

[6] http://lod.openlinksw.com/

search engine responds 110 terms as "title", 107 terms as "creator", and 86 terms as "character". The following are the example of terms as "title" found by the vocabulary search engine.

- http://purl.org/dc/terms/title
- http://schema.org/jobTitle
- http://schema.org/title
- http://www.cidoc-crm.org/cidoc-crm/P102_has_title
- http://www.loc.gov/mads/rdf/v1#natureOfAffiliation
- http://xmlns.com/foaf/0.1/title

(2) Search terms by metadata instances

The text "Harry Potter and the Philosopher's Stone" is used in the example in Figure 2 to find metadata instances from LOD Cloud Cache. This phrase is chosen because it is the title of a well-known novel and movie and because some communities provide information about fiction works as a LOD resource, e.g., Freebase[7], DBpedia[8] and so forth, so that we can find many metadata instances which include "Harry Potter and the Philosopher's Stone".

```
SELECT distinct ?term
WHERE {
   { ?s ?term "Harry Potter and the Philosopher's Stone" . }
   UNION
   {
      ?s ?term ?o .
      ?o rdfs:label "Harry Potter and the Philosopher's Stone" .
   }
}
ORDER BY ?term
```

FIG. 2. A SPARQL query for searching terms as a title of "Harry Potter and the Philosopher's Stone" by LOD Cloud Cache

In this scenario, the metadata schema designer search requests use the examples of metadata values shown in Table 1 for each metadata attribute. LOD Cloud Cache responds with 50 terms as "title", 105 terms as "creator", and 95 terms as "character". The following are examples of terms as "title" found by the LOD Cloud Cache.

- http://data.linkedmdb.org/resource/movie/performance_film
- http://purl.org/dc/terms/title
- http://sindice.com/hlisting/0.1/itemName
- http://www.w3.org/2006/vcard/ns#fn
- http://xmlns.com/foaf/0.1/made

Select Terms

Table 2 shows the number of terms which a vocabulary search engine and LOD Cloud Cache respond to. In this case study, they select terms from 155 terms for describing "title". It is necessary for the designer to select the appropriate metadata terms they got in the previous steps.

[7] http://www.freebase.com/
[8] http://dbpedia.org/

TABLE 2: Numbers of metadata terms which a metadata schema designer find out

	A) vocabulary-based	B) instance-based	A OR B	A AND B
Title	110 (27 vocabularies)	50 (23 vocabularies)	155	5
creator	107 (26 vocabularies)	105 (26 vocabularies)	212	0
character	86 (18 vocabularies)	95 (33 vocabularies)	181	0

The metadata schema designer found 155 terms to express titles of a fiction work such as Star wars, Harry Potter and so forth. A simple criterion to choose a term is popularity of terms. We calculated the popularity of each candidate metadata term using existing metadata instances in LOD Cloud Cache. We can suggest a set of 'popular terms' from so many terms found. Finally, the metadata schema designer selects a metadata term from the popular terms based on the meaning and their preference.

Another criterion to select terms is confirming classes as domain of metadata terms in definitions and actual metadata instances. Some metadata terms have defined domains using rdfs:domain.

6. Discussion

Based on the case study, we discuss approaches for finding metadata terms and requirements for vocabulary search engines and metadata instances.

Implicit or Unreachable Definitions about Metadata Vocabularies

Calculating the intersection of two approaches for searching terms is useful for narrowing the list of metadata terms. However, in the case study, there are few intersections of the two approaches because some definitions about the metadata vocabularies are implicit or unreachable on the Web, so that we could not get them. In advance to the case study, we accumulated definitions about metadata vocabularies from prefix.cc. Prefix.cc has 950 namespace URIs, but we could only retrieve 450 definitions of vocabularies. We have to solve the following problems about metadata vocabularies.

- Definitions about metadata vocabularies are not created in RDF
- Implicit or non-standard links between namespace URIs and vocabulary definitions
- Several terms do not have enough definition (i.e., lack of label and description) for metadata schema designers

Some definitions about metadata vocabularies are described in HTML, XML Schema and so forth. We accumulated only metadata vocabularies in RDF because we use a RDF triple store, so we drop another files including definitions about metadata vocabularies.

When we access namespace URIs using content negotiation which is a standard way to link a namespace URI and actual RDF files, the access redirected to RDF files. However, Some namespace URIs are not compatible with content negotiation. For example, RSS 1.0 (http://purl.org/rss/1.0/) and Creative Commons (http://creativecommons.org/ns#) navigate from their namespace URIs to specifications in RDF using RDDL[9]. We did not accumulate definitions about those vocabularies.

Several vocabularies (e.g. http://rdf.data-vocabulary.org/#) do not define labels, descriptions, etc. for each term. These terms are unreachable when a metadata schema designer uses vocabulary search engines. Even if he/she accesses those terms, there is no meaningful information for human.

[9] http://www.rddl.org/

Lack of Information about Metadata Vocabularies and Instances

In our case study, we used metadata vocabularies and instances using definitions about metadata vocabularies and LOD Cloud Cache. But we could not find information about metadata vocabularies and instances such as the following.

- Who created and maintains the metadata vocabularies/instances
- When were the metadata vocabularies/instances created
- What domains metadata vocabularies/instances are used in

Metadata schema designers select metadata terms which are well-known in subject domains However, there is poor information about subject domains of metadata terms/instances. It is helpful for metadata schema designers to describe metadata about datasets in Data Catalog Vocabulary (DCAT)[10]. If we describe and use subject domains about metadata instances in DCAT, metadata schema designers search metadata terms by metadata instances in specific subject domains. Also, in order to select metadata terms which are used and maintained continuously, it is important to record the date of creation and modification of metadata terms/instances.

Use Existing Metadata Schema

In our approach, a metadata schema designer finds metadata terms and combines those terms as a DSP from scratch. If existing DSPs are published on the Web, metadata schema designers re-use and customize existing DSPs which are suitable for their functional requirements and domain models. For example, when a metadata schema designer designs a schema for fiction works (e.g. Star Wars), they look up classes of "Star Wars" in LOD Cloud Cache, and search existing DSPs for those classes. MetaBridge (Nagamori, et al., 2011) is one of the metadata schema registries which stores metadata schema described in DSP. A metadata schema designer inputs classes of fiction works, and his registry responds with existing DSPs for fiction works.

7. Conclusion

Find and combine metadata terms is a key aspect in existing guidelines for designing interoperable metadata schemas or ontologies. In this paper, we analyze requirements for finding and combining metadata terms and propose an approach for metadata schema designers to find and combine metadata terms using schema registries and LOD resources. Also, we show a case study and discuss the requirements for vocabulary search engines and metadata instance search engines in order to find metadata terms based on the Singapore Framework.

Johann et al. propose how to support ontology engineers using existing tools (Johann et al., 2013). They analyzed tasks and information which is required for modeling LOD. They mentioned eight functional requirements for supporting environments of ontology development, and showed which requirements the existing tool would cover. They use vocabulary search engines for finding and combining vocabularies. We propose that metadata schema designers use not only vocabulary search engines but also metadata instance search engines with words for metadata attributes and examples of metadata values.

Acknowledgements

The authors are grateful to Liddy Nevile for her help and contributions. This work was supported by KAKENHI Grant Numbers 22240023, 23500295.

[10] http://www.w3.org/TR/vocab-dcat/

References

Coyle, Karen & Thomas Baker. (2009). Guidelines for Dublin Core Application Profiles. Retrieved April 8, 2013, from http://dublincore.org/documents/2009/05/18/profile-guidelines/

Heath, Tom & Christian Bizer. Linked Data: Evolving the Web into a Global Data Space. (2011). Morgan & Claypool.

Heery , Rachel & Manjula Patel. (2000). Application Profiles: Mixing and Matching Metadata Schemas. Ariadne, 25, Retrieved, April 8, 2013, from http://www.ariadne.ac.uk/issue25/app-profiles

Honma, Tsunagu, Mitsuharu Nagamori & Shigeo Sugimoto. (2012). A Model to Support Interpretation of Embedded Metadata without Formal Schema by Linking a Metadata Instance to DCMI Description Set Profiles. Proceedings of the International Conference on Dublin Core and Metadata Application, 2012, 80-89.

Metadata Information Infrastructure Initiative, (2011), Guidelines for Sharing Information about Metadata. Retrieved, April 8, 2013, from http://www.mi3.or.jp/item/A03.pdf

Metadata Information Infrastructure Project. (2010). Web Site of Metadata Information Infrastructure Project. Retrieved, April 8, 2013, from http://meta-proj.jp/.

Nagamori, Mitsuharu, Masahide Kanzaki, Naohisa Torigoshi & Shigeo Sugimoto. (2011). Meta-Bridge: A Development of Metadata Information Infrastructure in Japan. Proceedings of the International Conference on Dublin Core and Metadata Applications, 2011, 63-68.

Nilsson, Mikael, Thomas Baker & Pete Johnston. (2008). The Singapore Framework form Dublin Core Application Profiles. Retrieve April 8, 2013, from http://dublincore.org/documents/2008/01/14/singapore-framework/

Noy, Natalya F. & Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. Retrieved April 8, 2013, from http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

Schaible, Johann, Thomas Gottron, Stefan Scheglmann & Ansgar Scherp. (2013). LOVER: support for modeling data using linked open vocabularies. Proceedings of the Joint EDBT/ICDT 2013 Workshops, 2013, 89-92.