A Metadata Schema Framework for Functional Extension of Metadata Schema Registry

Mitsuharu Nagamori, Shigeo Sugimoto
Graduate School of Library, Information and Media Studies
University of Tsukuba
{nagamori, sugimoto}@slis.tsukuba.ac.jp

Abstract:

The DCMI metadata schema registry has been developed as an authoritative source of DCMI metadata terms. The DCMI registry has an important role to enhance semantic interoperability of the metadata terms. From our experiences in the development of the DCMI registry, we have learned that the registry has large potential to serve as a center of various service functions related to metadata schemas such as metadata editors and search tools. This paper presents a framework of metadata schemas and discusses its application to schema registries for extending the registry functions. The metadata schema framework presented has a layered structure in order to explicitly separate semantic and syntactic features of metadata schemas. This paper presents a few functional extensions of the registry and discusses the extension based on the proposed schema framework.

Keywords:

DCMI Metadata Schema Registry; Metadata Schemas; Layered Model for Matadata Schemas; Metadata Interoperability.

1 Introduction

Interoperability is one of the most crucial topics for the metadata and digital library communities. The most significant part of the efforts of the development of the Dublin Core has been paid to realize semantic interoperability of metadata across domains and cultures. The Dublin Core community has developed and crystallized a few fundamental concepts for metadata interoperability, e.g., the Dumb-down principle for designing qualifiers and the Warwick Framework as a framework to define an application specific metadata schema based on multiple metadata element sets. These concepts are a major contribution of the Dublin Core community to the broader communities in addition to the Dublin Core Metadata Element Set (DCMES).

The authors have been involved in the development of the DCMI registry since 1998. We developed an experimental system, which is called the ULIS Registry in this paper [1]. DCMI has launched the operational DCMI registry [2] to which the ULIS registry contributed in certain crucial aspects, especially in the aspect of multilingual services. The current DCMI registry provides users with search and browsing functions of reference descriptions of the DCMI metadata terms. It has a set of functions to organize and manage the terms in its database. All metadata terms are encoded in RDF Schema. They are provided to human users through user friendly interfaces and to machines through application program interfaces (APIs). The authors have experimentally installed a few sets of non-DCMES metadata terms in the registry in addition to the DCMI metadata terms.

Thus, the DCMI metadata schema registry works as an authoritative dictionary of metadata terms. The authors have found that we can extend the metadata schema registry to provide services related to metadata schemas and software tools for metadata schemas. We have experimentally developed a few functions in order to extend the functionality of the registry at Tsukuba; for example, a cross-schema search function which associates metadata terms across multiple metadata element sets, a element extraction function which extracts common elements among multiple application profiles, and a software generator which produces software tools such as a metadata editor, a metadata search tool and a metadata database management tool.

In parallel to these studies, based on the discussions on metadata schema registries, the authors have developed a conceptual

https://doi.org/10.23106/dcmi.952107646

framework for metadata schemas to enhance interoperability of metadata and metadata schemas. The framework has three layers organized to explicitly separate semantic and syntactic features of the metadata schema. In this paper, we use the layered model to identify the scope of the functional extension of the metadata schema registries. The layered model is advantageous in finding requirements to make metadata schemas interoperable each other.

The rest of this paper is organized as follows. Section 2 describes basic concepts and related works of this study. Section 3 describes the layered model of metadata schema. Section 4 shows a few experimental software tools.

2 Backgrounds

2.1 Basic Concepts of Metadata Schema

This section describes some basic concepts of the model and process of Dublin Core [3].

2.1.1 Metadata Schema

In this paper, we define a metadata schema as a set of description that defines a description scheme of metadata. A metadata schema of an application includes semantic definition of terms used in the schema, structural constraints and data structure definitions, and bindings to physical description syntax such as XML.

For example, Simple Dublin Core gives the definition of the 15-elements as the semantic definition and the structural constraints "every element is optional and repeatable" [4]. The bindings to HTML, XML and RDF are given in separate documents [5][6][7].

In general, a metadata schema consists of the following components,

(1) a set of terms defined to express properties of a resource, e.g., Title, Creator, alternative and so on,

(2) a set of terms which expresses types of property values and/or which are used as a property value, e.g. ISO-8601, DCMI Type Vocabulary, LCSH, and DDC,

(3) a set of rules which defines structural constraints and syntactic features neutral to any implementation specific description scheme, e.g. mandatory levels, repeatability/cardinality, order, and so on, and

(4) a set of binding rules to a specific description language, e.g., XML, HTML and RDF.

The first two components define a name(s) and meaning of every term, which give the semantic basis of the schema. On the other hand, the latter two define syntactic features, which we call abstract and concrete syntax. In a real application environment, a set of guideline statements to create metadata instances in accordance with the application is required. These guidelines are not included in the definition of metadata schema in this paper because they are not included in the formal specification of metadata schema in schema specification languages. (By the same reason, guidelines are excluded from definition of Application Profile in this paper.)

2.1.2 Warwick Framework and Application Profile

Since the Internet is a very diversified environment, it is useless to assume that a single metadata element set will meet the needs of all domains and purposes. It is also impractical to develop metadata sets application by application: the result would be expensive and chaotic, and interoperability would be non-existent. On the other hand, it is desirable for application developers to use established metadata schemas and adopt them in accordance with local requirements. The Warwick Framework, a conceptual model that resulted from the 2nd Dublin Core Workshop in 1996, gave an early expression to the notion of metadata as modular components that may come from more than one metadata schema [8]. In this model, a metadata instance is expressed as a container which contains one or more packages, each of which is expressed in a given metadata schema. The Resource Description Framework (RDF) provided a practical realization of many of the ideas of the Warwick Framework [9]

Application Profiles, which provide a framework to adopt one or more element sets in accordance with an application, could be also caught as a realization of the Warwick Framework. Dublin Core Metadata defines the vocabulary of metadata, i.e., terms and their

https://doi.org/10.23106/dcmi.952107646

meanings, but in general does not specify the encoding or syntactic characteristics. An exception is the feature included in Simple DC that is "Any of the 15 elements is optional and repeatable." Local applications, however, may have domain specific requirements appropriate to a given domain or application:

- Title, Creator and Description might be mandated but others are optional,
- Use only Title, Creator, Description, Date and Language elements,
- Use the 15 elements of Simple DC and some elements from other metadata sets such as the IEEE Learning Object Metadata (IEEE LOM), and so forth.

These requirements can be defined independently of the vocabulary definitions. Description of this application-specific syntactic feature is called an application profile. Any application can have its own application profile, which specifies a set of metadata vocabulary terms used in the application as well as syntactic or structural features of the particular application. Figure 1 shows a model of application profiles. The vocabulary terms could be borrowed from one or more source schemas. More importantly, the application profile could be used to define a mapping between the application's scheme to a global scheme(s), which is crucial for interoperability.

2.1.3 Dumb-down Principle

The Dumb-Down principle gives a guideline for qualification. The Dumb-Down principle suggests that a value of a qualified element has to be consistent as a value of the element without any qualification. For example, assume the following qualified values:

(1) (Element Refinement) Date Accepted: "2004-10-12",

(2) (Encoding Scheme) Language: "en" encoded in RFC 1766, and

(3) (Value Structure) Creator: {name: "Sugimoto, Shigeo", affiliation: "University of Tsukuba", contact: "sugimoto@slis.tsukuba.ac.jp"}

Then, assuming that the qualifications in the above examples, Accepted, RFC 1766 and the component names of the value structure (i.e., name, affiliation and contact) are removed. The values of example 1 and 2, "2004-10-12" and "en" are still consistent with their elements after the removal. However, the value of example 3 {"Sugimoto, Shigeo", "University of Tsukuba", "sugimoto@slis.tsukuba.ac.jp"} causes problems since the second and third values are not valid values of Creator.

Dumbing-down is a crucial function for metadata interoperability in the global community since local communities can extend their schemas in accordance with their requirements, and at the same time they can also keep their metadata interoperable with other metadata communities.

2.1.4 Evolution and Maintenance of Metadata Vocabularies

Any living metadata standard needs its process model to keep the standard updated in accordance with the requirements given to the

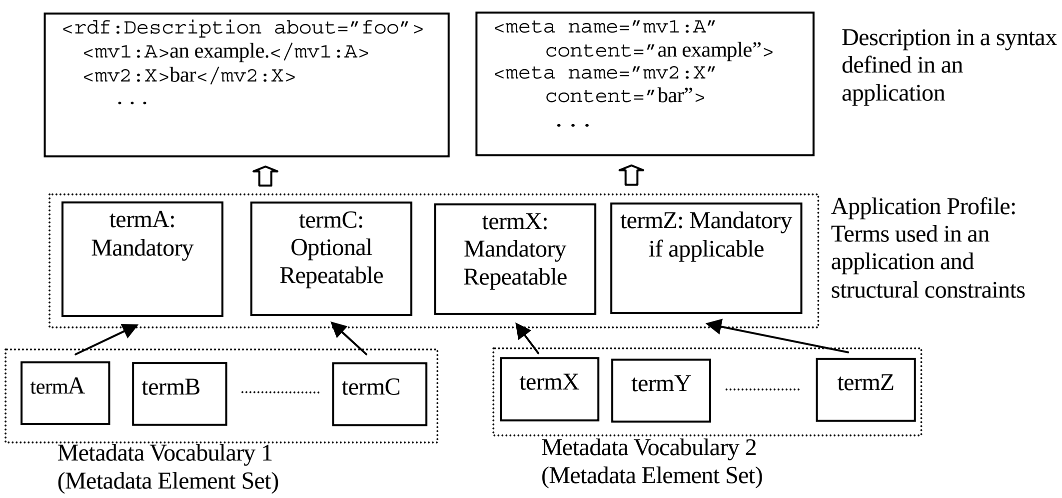

Figure 1 Concept of Application Profile

https://doi.org/10.23106/dcmi.952107646

standard. The first part of this section shows the maintenance process model of metadata terms by DCMI, and the second part shows some discussions on a model for maintenance of metadata terms.

(1) Process Model of DCMI Metadata Terms

To remain relevant in a rapidly evolving Web environment, Dublin Core must be able to grow and evolve in response to user needs. DCMI has therefore instituted a Usage Board and a process model for reviewing proposals for expanding or clarifying the standard. Proposed elements and element refinements that conform to Dublin Core principles are taken into the standard with the status of conforming. To some proposed terms of proven usefulness for resource discovery across domains the Board may assign the status of recommended. Proposals for encoding schemes are reviewed for accuracy and given the status of registered. Once approved, each new term is assigned a Uniform Resource Identifier using one of the official namespace URIs maintained by DCMI. A "namespace policy" defines limits within which the metadata terms maintained by DCMI can evolve or change over time. According to this policy, editorial changes or updates are allowed, but changes of semantics (meaning) are not; new semantics require the creation of new element.

(2) Maintenance Model for Metadata Terms

DCMI metadata terms are stored in the DCMI metadata schema registry and its cooperating registries. The terms are made accessible via the Internet and maintained in the registries. Authoritative reference descriptions of the metadata terms in English are translated into non-English languages for adoption of local communities. By the nature of Dublin Core, this translation of the vocabularies has been and will be done by grassroots volunteers. In addition, a local community can define their own metadata terms, which may or may not be approved as conforming. Therefore, metadata vocabulary maintenance has to be performed in two aspects; one is the authoritative description directly maintained by the Usage Board, and the other is translations in non-English languages. The authoritative description is stable but, on the other hand, a translated description is rather unstable unless it is translated by a local authority.

2.2 Metadata Schema Registry

A goal of metadata schema registries is to make metadata schema understandable both by human and machines and shareable among user communities. Metadata schema registries have gained interests of broad metadata communities because of the strong requirements of interoperability and longevity of metadata and metadata schemas. ISO/IEC11179 describes the standardizing and registering of data elements to make data understandable and shareable. Data element standardization and registration as described in ISO/IEC 11179 allow the creation of a shared

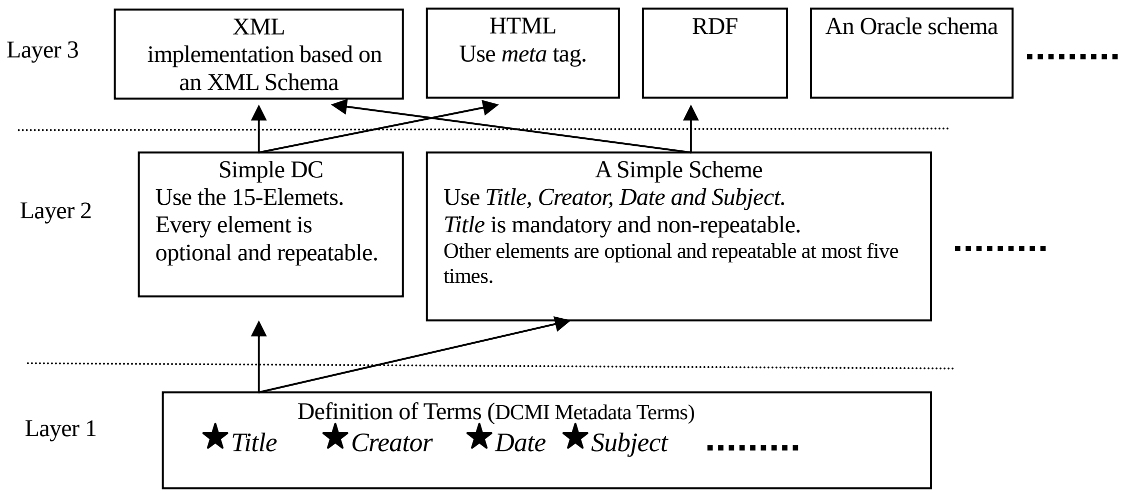

Figure 2 Layered Model of Metadata

https://doi.org/10.23106/dcmi.952107646

data environment. Universal Description Discovery and Integration (UDDI) registries act as reference points for Web Services that allow for common descriptions and discovery of those services, based on XML standards and independent of platform. ISO IEC JTC1 SC32 WG2 has been organizing a series of workshops on metadata registries.

The ULIS registry developed by the authors provides reference descriptions of metadata terms in multiple languages encoded in RDF Schema [1]. We have experimentally stored metadata elements of Internet Public Library Asia (IPL-Asia)[10] and those of the Nippon Cataloging Rules (NCR) in the ULIS registry. The DCMI registry which is in operation provides authoritative reference descriptions of metadata schema, which are internally encoded in RDF Schema and translated in 24 different languages. The reference descriptions are presented in a user friendly form for human users and in RDF Schema for machines. The application program interface is provided based on the Web Services protocols, i.e., SOAP or REST. Description of each metadata term includes the unique name of the term, language dependent labels, definition statement of the term, date(s), type of the term and links to related terms. The descriptions are maintained in accordance with the DCMI terms approved by the Usage Board. The DCMI registry is provided as an open source software for use by broader communities. As of spring 2004, the DCMI registry has been made available in Germany and Tsukuba, Japan in addition to OCLC. The

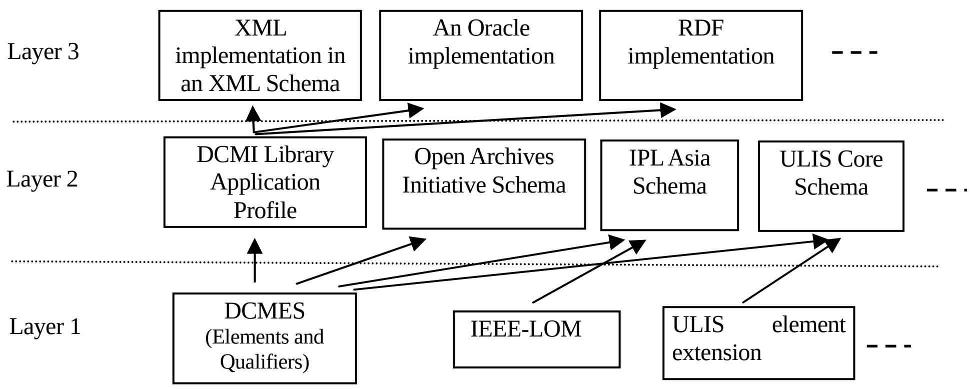

Figure 3 Layered Model of Metadata Schema based on multiple element sets

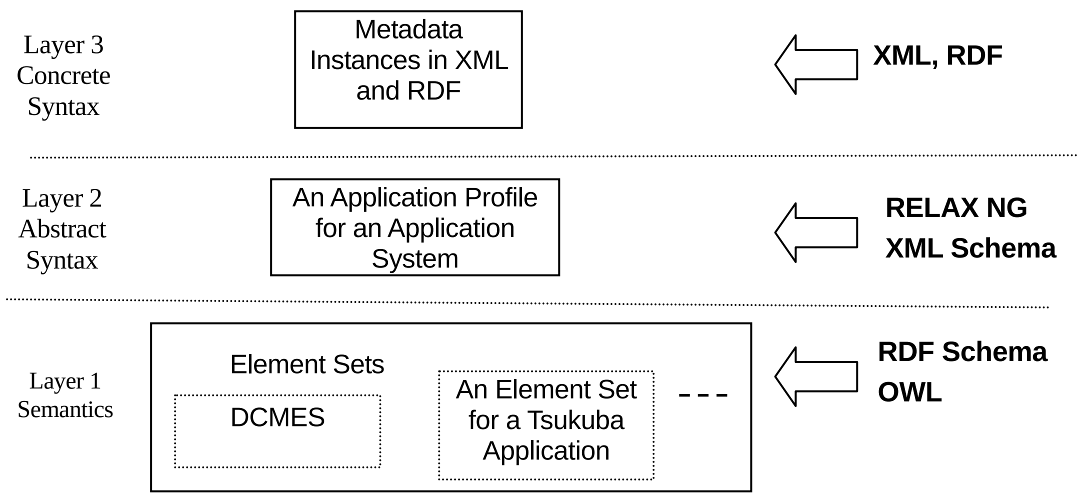

Figure 4 Layers and description schema based on XML

https://doi.org/10.23106/dcmi.952107646

DCMI registry at Tsukuba provides reference descriptions of NCR and IPL-Asia vocabularies which have been transported from the ULIS registry.

2.3 Related Studies and Concepts

The white paper reported by the DELOS Working Group on Registries [11] describes basic concepts of metadata schemas, i.e., metadata vocabulary, layers for metadata interoperability, data models, and so forth. This study is based on the concepts described in this white paper. The layered model discussed in the white paper gives relationship a framework of metadata vocabularies. On the other hand, the layered model presented in the next section gives a logical framework for metadata schemas. This layered model was primarily introduced to separate syntactic and semantic features of metadata schema descriptions in order to clarify relationships among constructs of metadata schemas and to help cross-schema mappings for metadata interoperability [12] and [13].

This study is primarily based on the XML technologies for metadata and metadata schema. In the experimental studies shown in section 4, we have used Relax NG, RDF, RDF Schema, DAML+OIL, OWL, and so forth. This is also based on the conventions of the schema description of Dublin Core in RDF Schema.

3 A Layered Model for Metadata Schema

This section first defines the layered model of metadata schemas, and then discusses requirements analysis of interoperability based on the model.

3.1 A Layered Model of Metadata Schema

As introduced in section 2.1.1, a metadata schema includes semantic and syntactic components. These components can be organized into layers as follows.

Layer 1 - Semantic Definition Layer (semantics layer or ontology layer): Definition of terms used in the schema. In other words, definition of metadata vocabulary, i.e. metadata element set. In general, two types of metadata terms are included in the metadata vocabulary ? property vocabulary and value vocabulary[11]. A property vocabulary, or in other words element vocabulary, is a set of property terms, for example, elements and element refinement qualifiers of DCMES. A value vocabulary is a set of value terms, for example, encoding schemes of DCMES. Definition of each term should primarily include a primary name and its meaning. Thus, a vocabulary definition gives the semantic basis of a metadata schema.

Layer 2 - Structural Constraints Definition Layer (abstract syntax layer): Definition of syntactic features which does not depend on any particular implementation scheme. A set of terms used in the schema and structural constraints applied to each term should be included in a definition. Application profiles are given in this layer. The structural constraints would include composition, ordering, mandatory levels, repeatability and cardinality, and specification of controlled vocabularies used in a metadata element. In other words, this layer defines application profiles in implementation neutral syntax.

Layer 3 - Implementation Dependent Syntax Definition Layer (concrete syntax layer): Definition of syntax of metadata in an implementation; for example, metadata description syntax in HTML, XML, RDF or in a specific database management system such as Oracle and MySQL.

Figure 2 illustrates a layered model which is based on a single element set, i.e. DCMI Metadata Terms. Simple Dublin Core, which specifies "use the 15 elements of Dublin Core where every element is optional and repeatable." Figure 3 shows a layered model which is based on multiple metadata element sets.

An application schema developer would provide guidelines for creating metadata in addition to their schema. The guidelines can be documented in the layers 2 and/or 3 in accordance with the implementation specificity; for example, the DCMI Library Application Profile includes some general guidelines in implementation neutral level, which should be associated to the layer 2.

A metadata term defined in the layer 1

https://doi.org/10.23106/dcmi.952107646

can be defined in an ontology specification language such as RDF Schema and OWL. Structural constraints in the layer 2 can be defined in a syntax description scheme such as DTD, RELAX NG and XML Schema. Figure 4 illustrates relationships between the layers and description scheme based on XML which are adopted at Tsukuba for realizing software tools described in the next section.

3.2 A Simple Requirements Analysis for Metadata Interoperability

The layered model helps us understand requirements to realize functions for cross-repository retrieval. The following paragraphs show requirements analysis cases for retrieval across metadata repositories.

Case 1: Repositories A and B have the same metadata schema in all layers. Metadata instances of both repositories are interoperable as they are.

Case 2: Metadata schemas of A and B are the same in the layers 1 and 2. This case needs common implementation syntax but conversion from original physical syntax to the common syntax should be straightforward.

Case 3: Metadata schemas of A and B use the same vocabularies defined in layer 1 but syntactic features in the higher layers are different. This case needs extraction of commonly used metadata terms and definition of a set of metadata terms as an interoperability set. Detailed discussion on the structural constrains is given in the next section.

Case 4: Metadata Schema of A and B partly share vocabularies in layer 1. This case needs extraction of a common set of terms and definition of an interoperability set. For the extraction, dumb-down function could be applied.

Case 5: Metadata schema A and B have no common vocabulary. This case needs definition of crosswalks between A's and B's vocabularies for creating a common set of metadata terms and a common syntax of metadata instances.

In the requirements analysis above, metadata vocabulary gives the basis for metadata interoperability. Formal definition scheme of metadata vocabulary should be used to create descriptions of metadata term definitions that have to be both machine and human understandable. RDF Schema has been used by the DCMI metadata schema registry as a formal vocabulary description scheme. In RDF Schema description, every metadata term is given a unique identifier which works as its primary name. A term definition could include one or more secondary names and related information as well. The primary name ? typically a URI ? is defined to uniquely identify the term. On the other hand, since secondary names are given as a human-friendly label, the secondary names could be translated, for example, Title element of Dublin Core could have labels in English, German, Japanese, and so forth. The primary names are used in the formal specification of metadata schemas to identify metadata terms and other constructs and to define relationships between them.

3.3 Discussion on Structural Constraints for Interoperability Requirements Analysis

Structural constraints are classified into the following types:

(1) A composite value composed of named sub-elements. For example, a person name composed of a first-name, a given-name, an affiliation, and a contact address.

(2) A composite value composed of ordered or unordered sequence of component values. For example, an ordered list of component values whose minimum and maximum lengths are 5 and 10, respectively.

(3) Mandatory levels, e.g., optional, recommended, mandatory if applicable, mandatory.

(4) A set of value types of an element adopted in the application, which should be a proper set of the value types of the element defined in the layer 1.

(5) Ordering constraints, i.e., descending or ascending order of values or significance of values, e.g. list of authors.

In general, mapping of metadata structures between different schemas needs structural transformation case-by-case basis. It is possible to define a generic function for the

https://doi.org/10.23106/dcmi.952107646

transformation, e.g., a function to structurally dumb-down a composite value into a simple value which conforms to the schemas, and a function to extract elements which are common among the schemas being cross-used. On the other hand, some information could be lost during this transformation.

4 Extending the Functions of Metadata Schema Registry: Experimental Studies

4.1 Functional Extension of Metadata Schema Registries

The primary function of the DCMI Metadata Schema Registry is to provide reference descriptions of DCMI Metadata Terms both for human users and machines. The metadata terms are instances defined in the layer 1. On the other hand, metadata schema registries can provide metadata schema components defined in the higher layers such as application profiles. In addition, certain types of generic functions such as dumb-down and crosswalk can be provided by the registry for users. Thus, metadata schema registries have potential to provide wider range of services based on the metadata schemas. We have experimentally developed a few functions to evaluate feasibility of functional extension of the metadata schema registry. The functions presented below are to be incorporated with the basic functions of the metadata schema registry. The functions are software tools to support information access across metadata schemas, a software generator based on metadata schemas, and a support tool for developing and maintaining metadata vocabularies.

4.2 Experimental Study 1: Metadata Vocabulary Centered Tools for Resource Access

We developed a crosswalk function which connects multiple metadata element sets and a cross-schema search function [14]. We defined an element set called abstract element set (AES) as a hub set to associate related elements with each other. In this experiment, we chose DCMES and NCR as the base vocabularies to define the hub set and defined a set of association rules between the base vocabulary terms and the hub vocabulary terms. The vocabularies and the association rules were encoded in RDF Schema and DAML+OIL. The association between the base vocabularies and the hub vocabulary was done manually. Figure 5 shows a description of DAML+OIL description which defines relationship between metadata terms.

```
<rdfs:Property rdf:resource="http://purl.org/dc/elements/1.1/#Alternative">
      <rdfs:label>Alternative</rdfs:label>
      <daml:samePropertyAs rdf:resource="&aesns;#Alternative"/>
      <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/#Title"/>
      <rdfs:comment>Any form of the title used as a substitute or alternative to
                        the formal title of the resource.</rdfs:comment>
      <rdfs:comment>This qualifier can include Title abbreviations as well as
                  translations.</rdfs:domment>
</rdfs:Property>
<rdfs:Property rdf:resource="&iplns;#TITLE">
      <rdfs:label>TITLE</rdfs:label>
      <daml:samePropertyAs rdf:resource="&aesns;#Title"/>
      <rdfs:comment>A name given to the resource.</rdfs:comment>
      <rdfs:comment>Typically, a Title will be a name by which the resource
            is formally known.</rdfs:comment>
</rdfs:Property>
```

Figure 5 Definition of Title and Alternative elements of DCMES with relationship specification

dc:altrenative is defined as a sub-property of dc:title. dc:title and dc:alternative are associated with aes:title and aes:alternative as a same property, respectively.

https://doi.org/10.23106/dcmi.952107646

In the ULIS metadata schema registry, each metadata term was assigned multiple labels and descriptions expressed in non-English languages. In the IPL-Asia, every subject term were given labels in multiple languages, i.e., Chinese, Japanese, Korean and English. In parallel to this multiple labels of a single term, we gave a subject term multiple labels that are defined based on the ages of the audience. These 1:n association between a subject term and its labels are useful to develop user interfaces in accordance with the user languages and user ages.

The metadata schema registry is a natural place to store the association rules and the multi-label definitions and provide them to the users. These descriptions are associated with the layer 1.

4.3 Experimental Study 2: A Metadata Schema Driven Software Tool Generator

From our experiences in developing software tools for metadata applications, we have learned that basic software tools such as a metadata editor and a search tool can be (semi-)automatically derived from metadata schemas. Based on this idea, we have been developing an experimental software tool generator for metadata application systems, which uses schema descriptions of metadata vocabularies and application profiles [15][16]. This experimental system has a set of built-in primitive functions, e.g., to load/store texts from/to a database, to search text in a database, and so on. This system produces a software tool from a set of XML documents, which is named Application System Description (ASD). An ASD of an application software tool is composed of the following four elements.

- Element Syntax Definition (ESD): definition of syntactical features of the application metadata schema tailored to define application tool specific metadata syntax.
- User Interface Definition (UID): definition of logical structures of user interfaces of the application software tool.
- System Interface Definition (SID): definition of flow of data to built-in functions prepared for the application.
- Association Definition: association description of a ESD, UID and SID for the application.

Figure 6 shows an overview of the generation process. The generator reads an ASD and definitions of metadata vocabularies. A set of XML texts are created from the UID and SID using syntactic constraints defined in the ESD. The XML texts have interfaces to call the built-in functions.

This software generator uses metadata schema descriptions given in the layers 1 and 2. The metadata instances handled in the application software tools automatically conform to the syntactic definition of the layer 3. Since user interfaces are derived from a metadata schema which includes class definitions of domain and range of a metadata

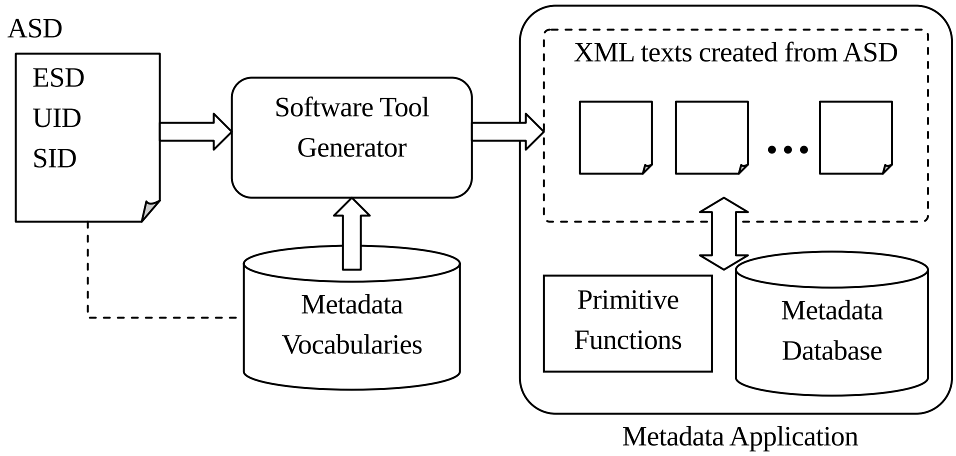

Figure 6 An overview of the generation process

https://doi.org/10.23106/dcmi.952107646

element, we can choose user interface widgets and built-in functions for the element in accordance with the class definitions.

5 Concluding Remarks

We have been involved in the research and development of the metadata schema registry since 1998. During these years, the process model and the data model of the Dublin Core have been established. The layered model shown in this paper was inspired by the DELOS Registry WG white paper.

The two experimental systems shown above are rather straightforward extensions of the metadata schema registry. We have found that the separation of syntactic and semantic features is useful to understand the functionality of the extended functions.

From these studies presented in this paper and other related studies, we have learned the following lessons:

- A metadata schema registry can serve not only as an authoritative information source of metadata schemas but also as a center which provides software tools defined in association with the schemas.
- It is crucial to organize a network of collaborating metadata schema registries in order to share not only globally approved metadata schemas but also locally developed schemas.
- We need to establish a process model for long-term maintenance of metadata schemas which should be able to manage life cycle of metadata schemas across languages. We need an automated function to collect revisions of authoritative descriptions and translations and that to manage an authoritative revision history.
- We need to develop a process model for enhancing interoperability and reusability of metadata schemas across communities. XML-based ontology technologies seem to be useful to develop the process model.

References

1. Nagamori, M., et al., A Multilingual Metadata Schema Registry Based on RDF Schema, Proceedings of DC-2001, pp.209-212, Tokyo, 2001.
2. The Dublin Core Metadata Registry, http://dublincore.org/dcregistry/
3. Sugimoto, S., et al., Models and Tools ? Metadata-Centered Projects at Tsukuba and Lessons Learned for Interoperability, ICADL'04, India, 2004.
4. Dublin Core Metadata Element Set, http://purl.org/dc/elements/1.1/
5. Hyper Text Markup Language, http://www.w3.org/MarkUp/
6. Extensible Markup Language, http://www.w3.org/XML/
7. Resource Description Framework, http://www.w3.org/RDF/
8. Lagoze, C., The Warwick Framework ? A Container Architecture for Diverse Sets of Metadata, D-Lib Magazine, July/August, 1996, http://www.dlib.org/dlib/july96/lagoze/07lagoze.html
9. Resource Description Framework, http://www.w3.org/RDF/
10. Lee, W., et al., A Subject gateway in Multiple Languages: a Prototype Development and Lessons Learned, Proceedings of DC-2003, pp.59-66, Seattle, 2003.
11. Baker, T., et al., Principles of Metadata Registries, http://delos-noe.iei.pi.cnr.it/activities/standardizationforum/Registries.pdf
12. Sugimoto, S., Metadata Schemas, Models and Tools - Metadata-Centered Projects at Tsukuba and Lessons Learned for Interoperability, Proceedings of ICDL'04, India, 2004.2, pp.690-699
13. Sugimoto, S., et al., Developing Community-Oriented Metadata Vocabularies: Some Case Studies, Proceedings of DLKC'04, 2004.3, pp. 128-135
14. Morozumi, A., A Cross-Referencing Metadata Search Engine Based on the Resource Description Framework, Master Thesis, University of Library and Information Science, Japan, 2003.
15. Syn, S., A Software Development Support System for Metadata Application based on Metadata Schemas, Master Thesis, University of Library and Information Science, Japan, 2002.
16. Kato, H., Construction of system generator based on the application system description system on metadata schema,

https://doi.org/10.23106/dcmi.952107646

Master Thesis, University of Library and Information Science, Japan, 2004.

https://doi.org/10.23106/dcmi.952107646