A Metadata Schema Design Model and Support System Based on an Agile Development Model

Kaori Ochiai[1], Mitsuharu Nagamori[2] and Shigeo Sugimoto[2]

[1] Graduate School of Library, Information and Media Studies, University of Tsukuba

[2] Faculty of Library, Information and Media Science, Research Center for Knowledge Communities, University of Tsukuba

Abstract

A huge amount of metadata is being published from various communities, domains and countries on the Internet. Metadata schema designers need to design metadata schema for their applications considering the interoperability. Linked Open Data (LOD) is the concept and movement to facilitate the use of datasets across communities on the Internet. However, to publish metadata as LOD, metadata schema designers must be experts of LOD. Metadata schema design, which is similar to software design, requires design and evaluation cycles where the iterative prototyping of metadata is useful. Our goal is to help metadata schema designers design a metadata schema in an iterative prototyping process. In this paper, we describe a metadata schema design methodology based on the agile development model. And then, we propose a system to support the proposed methodology including a metadata-editing tool. Finally, we show some metadata schemas as examples.

Keywords: agile development; application profile; metadata; metadata schema design; metadata schema validation

Citation: Ochiai, K., Nagamori, M., & Sugimoto, S. (2014). A Metadata Schema Design Model and Support System Based on an Agile Development Model. In *iConference 2014 Proceedings* (p. 921–927). doi:10.9776/14314

Acknowledgements: We would like to thank Patrick LO and Robert Allen. This work was supported by KAKENHI Grant Numbers 22240023, 23500295.

Contact: ochiai@slis.tsukuba.ac.jp, nagamori@slis.tsukuba.ac.jp, sugimoto@slis.tsukuba.ac.jp

1 Introduction

There is a huge amount of metadata on the Internet and still more is being published by various communities such as governments, research institutions. These datasets are used for different purposes across different communities, domains and countries. Thus, metadata schema designers, who design a metadata schema, need to consider metadata interoperability in their schema design. Linked Open Data (LOD) (Berners-Lee, 2006) is the concept and movement to facilitate the use of datasets on the Internet. However, in LOD, data must be described by RDF (Resource Description Framework), which is a standard model for data interchange on the Internet. Therefore, metadata schema designers need special knowledge and experience of using the Semantic Web to consider a metadata schema, which conforms to the requirement of LOD. In addition, a metadata schema design process needs an iterative process, which is time-consuming work. In this study, we explain a metadata schema design methodology based on an agile development model. Then, we propose a system to support metadata schema design and develop a metadata-editing tool based on the methodology.

2 The Process of Designing Metadata Schema

We divide the metadata schema design process into two steps (Figure 1) based on (Coyle & Baker, 2008; MI3, 2011) and on our experience. We assume that metadata schema designers already have clear functional requirements. In step 1, metadata schema designers define the domain model, metadata attributes and their value constraints. In step 2, metadata schema designers define metadata structure and metadata terms for properties and classes from definitions of step 1. Figure 2 is an example of definitions of step 1 and step 2. Process of step 1 is familiarized with many engineers, because it accords to designing relational database (RDB) schema. While many engineers have knowledge and technique of RDB, few engineers have knowledge and technique of RDF and LOD. Thus, step 2 is generally more difficult than step 1.

Metadata schema designers repeat changes in the metadata schema each time they verify metadata schema by creating and using metadata. Through validation of metadata schema, metadata schema designers often discover problems in their metadata schema, e.g. difficulties to input appropriate values, unsatisfied requirements and so forth. Iterative prototyping of a metadata schema is useful to design a metadata schema and to test whether those schemas meet the requirements. On the other hand, this iterative process is very time-consuming. Basically, metadata schema designers have to input metadata one by one. In addition, the more complex metadata schema become, the more metadata schema designers need a metadata-editing tool to input metadata suitably.

Metadata schema design process is similar to the agile software development, which is one of the software development processes and based on an iterative and incremental development. We adopt the agile software development method to the metadata schema design. We help metadata schema designers carry out metadata schema design development in a process analogous to agile software development.

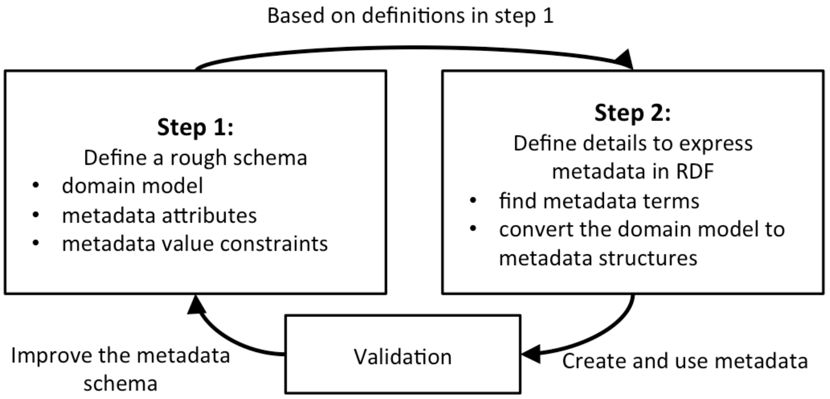
Based on definitions in step 1
Step 1:
Define a rough schema
• domain model
• metadata attributes
• metadata value constraints
Step 2:
Define details to express metadata in RDF
• find metadata terms
• convert the domain model to metadata structures
Improve the metadata schema
Validation
Create and use metadata

Figure 1: The Process of Designing Metadata Schema

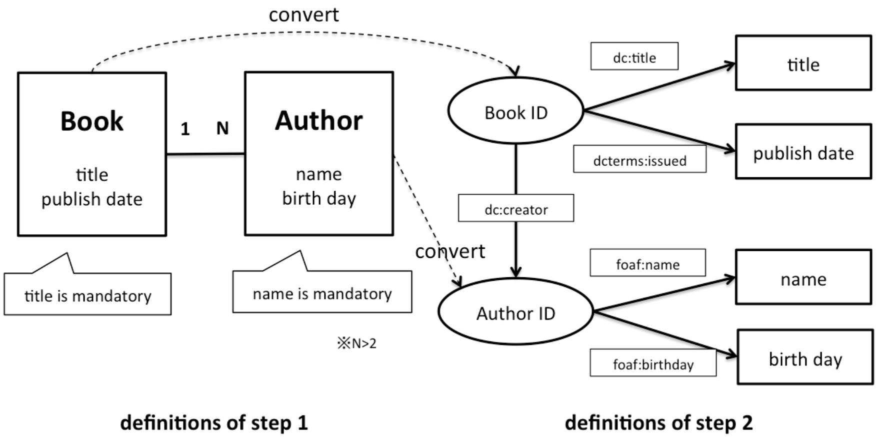

Figure 2: An Example of Definitions of Step 1 and Step 2

3 Support System for Designing Metadata Schema Based on Agile Development Model

The support system enables metadata schema designers to design a metadata schema in the same way as designing a RDB schema along the agile process. Thus, its users are metadata schema designers who are not familiar with designing RDF metadata schema, but RDB schema. We adopt the three ideas to solve issues in previous section.

Metadata schema designers execute step 1 by designing RDB schema for metadata-editing tool.

Using a RDB schema of a metadata-editing tool, the support system helps metadata schema designers execute step 2.

Using a metadata schema, the support system helps metadata schema designers develop a metadata-editing tool.

Based on these ideas, the support system will facilitate the conversion of the RDB schema of the metadata-editing tool into an RDF metadata schema. On the other hand, it will also help with the development of an experimental metadata-editing tool.

Figure 3 shows the iterative cycle of designing metadata schema and the development of the metadata-editing tool. Figure 4 shows the detailed metadata schema design cycle and the role of the support system. The support system provides two kinds of support functions to metadata schema designers:

Support 1. Proposing a draft metadata schema based on the RDB schema of the metadata-editing tool

Support 2. Providing a template which is the base of the metadata-editing tool from the metadata schema

In this approach, metadata schema designers first design a RDB schema of a metadata-editing tool. At this stage, the metadata-editing tool does not have the ability to create RDF metadata but only to input metadata. Metadata schema designers execute step 1 through this task. Second, the support system proposes a draft metadata schema based on the RDB schema of the tool. The support system performs two main tasks to create a draft metadata schema:

Searching terms for properties or classes to express in RDF using table names, column names in the RDB schema.

Converting tables and their relations in the RDB schema to metadata structures

After that, metadata schema designers complete the metadata schema based on a draft. And then the support system provides a template, which is the base of the metadata-editing tool, from the metadata schema. When a metadata-editing tool, which has function to create RDF metadata along the metadata schema, is developed, metadata schema designers have completed one cycle can validate their metadata schema. From result of validation, if the metadata schema needs improvement, metadata schema designers modify the RDB schema of the tool and start next cycle.

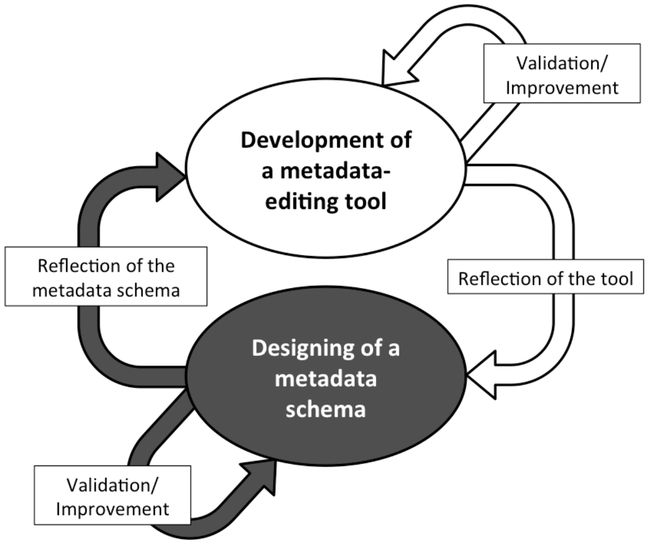
Development of a metadata-editing tool
Validation/ Improvement
Reflection of the metadata schema
Reflection of the tool
Designing of a metadata schema
Validation/ Improvement

Figure 3: The Cycle of Designing Metadata Schema and Development of the Metadata-editing Tool

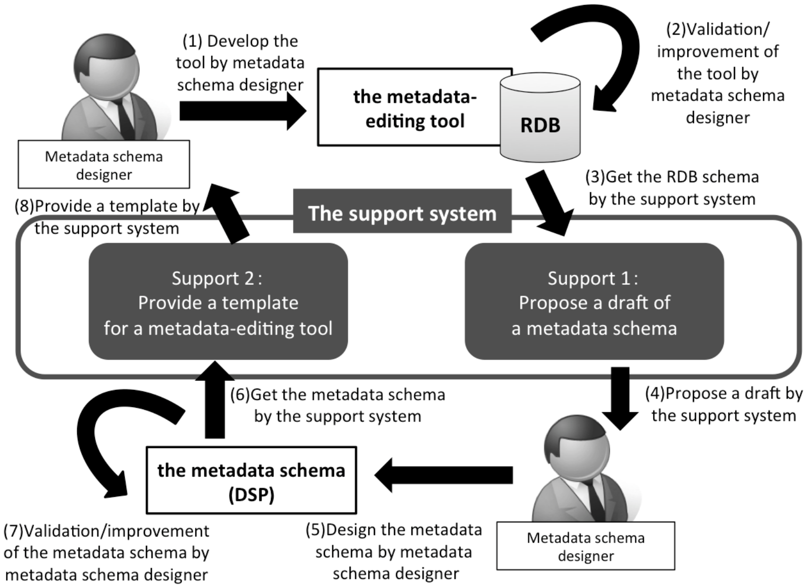
(1) Develop the tool by metadata schema designer
the metadata-editing tool
RDB
(2)Validation/ improvement of the tool by metadata schema designer
Metadata schema designer
(3)Get the RDB schema by the support system
(8)Provide a template by the support system
The support system
Support 2 : Provide a template for a metadata-editing tool
Support 1 : Propose a draft of a metadata schema
(6)Get the metadata schema by the support system
(4)Propose a draft by the support system
the metadata schema (DSP)
(7)Validation/improvement of the metadata schema by metadata schema designer
(5)Design the metadata schema by metadata schema designer
Metadata schema designer

Figure 4: Details of the Metadata Schema Design Cycle

4 Implementation

An iterative process like agile software development seems very time-consuming at first a glance. But recently, some useful support tools help engineers develop software easily and quickly. Ruby on Rails[1] is one of famous web application frameworks. The support system assumes metadata schema designers use Ruby on Rails as a metadata-editing tool.

We develop two Ruby libraries. One is for proposing a draft of metadata schema based on the RDB schema of the metadata-editing tool (support 1), the other is for providing a template for the metadata-editing tool from the metadata schema (support 2). A draft of metadata schema is an incomplete Description Set Profile (DSP) (Nilsson, 2008), which allow metadata schema designers to define detailed constraints of metadata. In a draft DSP, some candidates of metadata terms searched by using the dataset from SPARQL

[1] http://rubyonrails.org/

endpoint in Linked Open Vocabulary[2] are described addition to metadata structures. A template for a metadata-editing tool is the Rails Application Template[3], which enables users to omit some process for development.

5 Examples of Metadata Schema

The support system converts RDB schemas to metadata schemas, while it converts metadata schemas to RDB schemas. This paper explains with a focus on method of conversions from RDB schemas to metadata schemas. The support system judges how convert them from some conditions such as relationships between tables, the number of columns of the table and user's requests.

Figure 5 shows an example of the conversion, which has a relationship of one-to-many. In this case, the number of columns except for primary key and foreign key in the table *Member* is two, i.e. more than one. Therefore the table Member becomes a resource or a blank node and users judge whether is good.

Figure 6 shows an example of the conversion, which has a relationship of one-to-one. In this case, users first judge whether the table *Place* should have a structure or not. Figure 6 is the case that users judge a structure is not needed. Thus the resource *Organization* has two properties, which is for latitude and longitude, instead of the resource *Place*.

Figure 7 show an example of the conversion, which has a relation ship of many-to-many. This is very complex case. The table Writing has no column except for primary key and foreign key. Thus in metadata schema, resources Book and Author have properties to relate to each other.

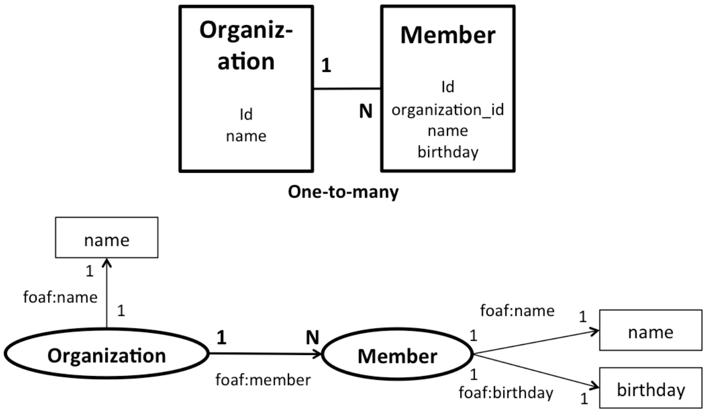

Figure 5: Example 1 of Metadata Schema: One-to-many

[2] http://lov.okfn.org/dataset/lov/

[3] http://guides.rubyonrails.org/rails_application_templates.html

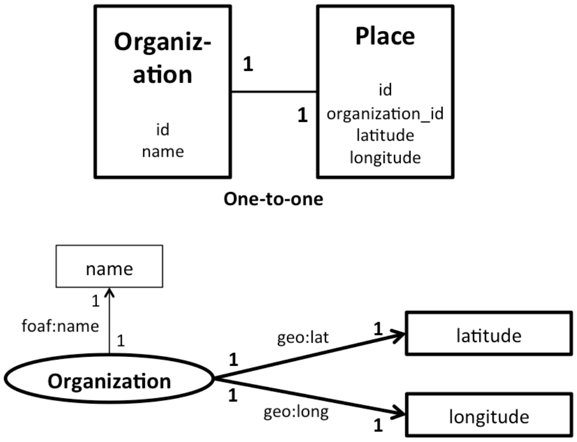

Figure 6: Example 2 of Metadata Schema: One-to-one

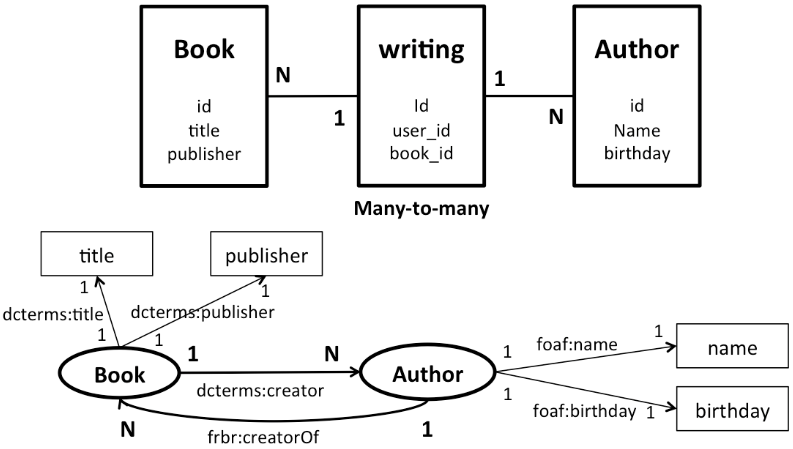

Figure 7: Example 3 of Metadata Schema: Many-to-many

6 Related Work

Malta & Baptista (2013) have proposed a method for the development of DCAP (Me4DCAP). They also adopt the iterative life-cycle development model. However this research does not cover the support for actual development of metadata schema using their method.

Many studies propose a mapping method to convert RDB schema to RDF metadata schema (Hart, Reif & Gall, 2011). But their metadata schemas are almost always expressed in Web Ontology Language (OWL). We describe a metadata schema with DSP, which is more specific about expressing constraints about metadata structure than OWL.

7 Conclusion

This is work in progress, so we will evaluate the support system as next step. For evaluation, we will verify whether we can design metadata schema actually along the conversion pattern we show in this paper.

8 References

Berners-Lee, T. (2006). Design issues: Linked data. Retrieved from http://www.w3.org/DesignIssues/LinkedData.html

Coyle, K., & Baker, T. (2008). Guidelines for Dublin Core application profiles. Retrieved from http://dublincore.org/documents/2009/05/18/profile-guidelines/

Hert, M., Reif, G., & Gall, H. C. (2011). A comparison of RDB-to-RDF mapping languages. In proceedings of the 7th International Conference on Semantic Systems, 35-32. ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2063522

Malta, M., & Baptista, A. A. (2013). A Method for the Development of Dublin Core Application Profile (Me4DCAP V0.2): Detailed Description. Proceedings of the DCMI International Conference on Dublin Core and Metadata Applications 2013, 90-102

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

Nilsson, M. (2008). Description Set Profiles: A constraint language for Dublin Core Application Profiles. Retrieved from http://dublincore.org/documents/dc-dsp/

9 Table of Figures