Преглед НЦД 24 (2014), 13–18

Marija Milojević Jevrić, Zoran Ognjanović,
Bojan Marinković
Mathematical Institute
of the Serbian Academy of Sciences and Arts

ONTOLOGY FOR NCD STANDARDS FOR DESCRIPTION OF DIGITAL DOCUMENTS AND COLLECTIONS

Abstract. The development of digital and digitized documents ontology is presented by this paper. The ontology was made in RDF/OWL syntax. Meta-data and all the necessary information was given in the form of an XML schema, and afterwards were made XSD to OWL mappings and necessary relations. The project was developed by use of the Protégé software, with several additional plug-ins such as OnTop. Consistency check was conducted and few queries as a test examples. Graphic interpretation of the ontology is presented as a part of this paper. The testing of possibility for ontology connecting with other systems, such as data base connection is made. Several potentials for upgrade of the ontology are proposed as part of future work, for example the usage of D2RQ or Sesame.

1. Introduction

NCD standards for the description of digitized documents and collection systems represent a complex structure whose elements are constructed with the help of modern information technology. Therefore, it is necessary to know some basic facts such as the Semantic Web and Ontology. The use of electronic resources depends on good search capabilities, and the concept of the Semantic Web has emerged as a result of search for more efficient solution for information retrieval.

Semantic Web enables understanding and reasoning about data which can be found in Web applications [3]. Unfortunately, the formal ontology, needed to express the semantics of data is not generally available. However, the usual scheme of data can help to create ontology. This paper contains translation from XML Schema to OWL, and shows how Web Engineering can benefit from obtained statements as well as the use of the concluder.

Semantic Web is the abstract representation of data on the World Wide Web, based on RDF and other standards. The concept was developed W3C in collaboration with many researchers [1].

Ontology is a formal specification of shared structures of organized knowledge of some domain.

Today, it is increasingly common use of XML for the exchange of ontological knowledge because there are universal parsers. XML is language which helps to define the different languages for the different applications and domains. But it turned out to be a catch, XML is not sufficient for knowledge representation, and it is necessary definition of universal knowledge model that will aside of syntax also provide semantic interoperability.

In order to address this deficiency have been found two standards:

- Resource Description Framework (RDF) for the exchange of knowledge and facts
- RDF Schema (RDFS) to define the structure (a dictionary) ontology.

Current proposal standard ontology language for the Semantic Web: The Web Ontology Language (OWL) is defined based on this semantic framework. Besides OWL other languages should be mentioned that are preceded this ontology language: Ontology Inference Layer (OIL) and the DARPA Agent Markup Language + OIL (DAML + OIL). For practical use and development of ontologies, beside of ontology languages, there are also appropriate tools in existence: Protégé, oiled, OntoEdit, WebOnto etc. [2].

2. XSD to Owl Mappings

Ontology for this project was made in Protégé. Input data used for the development is taken into the XML schema from the XSD document form [4, 5, 6]. It was necessary to determine the ontology's future classes and subclasses. After that, XSD to OWL mapping of all objects from a XML schema was made. On that way the knowledge base was developed for the NCD standards with a variety of upgrading options.

The procedure for mapping the complete XML schema to the ontology is made by mapping of its individual components [2].

According to the instruction, given in [3], each complexType in XML Scheme is mapped into an unique owl:Class. Each element and attribute declaration is mapped to an OWL property. More precisely, elements of simpleType and all attributes are mapped to owl:DataTypeProperty; elements of complexType are mapped to an owl:ObjectProperty. Finally, the schema root element is mapped to an OWL Class of name 'targetNamespace + #Schema'.

Model Groups: Model group definitions and attribute groups' definition of the specialization of complex types for which they contain an element respective attribute declarations. Thus, they are also mapped to OWL classes.

Specialisation: in the object orientated programming, the mechanisms inheritance are central to the modelling tool use scenes to express the "is-a" relationship between classes.

Global Elements: global element and attribute declarations are mapped similarly as local ones. Joined restrictions are added to the class schema.

The future of this work is reflected in the ontology database connecting, data collection that contains individual documents contained in a digital library for search study and the conclusions making.

3. Example

The key elements that constitute the ontology are basic terms, semantic interdependences, simple rules of conclusion and logic for particular topic. Ontologies are mutually different not only by content, but also by its structure, implementation, level of description, conceptual scope, examples and specification language. All ontologies are not built in the same way. There is a large number of programming languages that can be used for the ontology development including general logic programming languages, such as Prolog.

The input data for the development of ontologies were taken in the form of XML schemata, in the other words in the form of XSD document [7]. It was necessary to determine the future classes and subclasses in the ontology. Thereafter, the XSD2OWL mapping of objects from XML schema was made. In the case of XSD2OWL mapping it was necessary to accurately determine the entities, classes, subclasses, object properties, data properties, annotations, and also the relations between them [3]. For example, in the ontology of standards for digital and digitized documents meta-data describing PERSON object (person) is as follows:

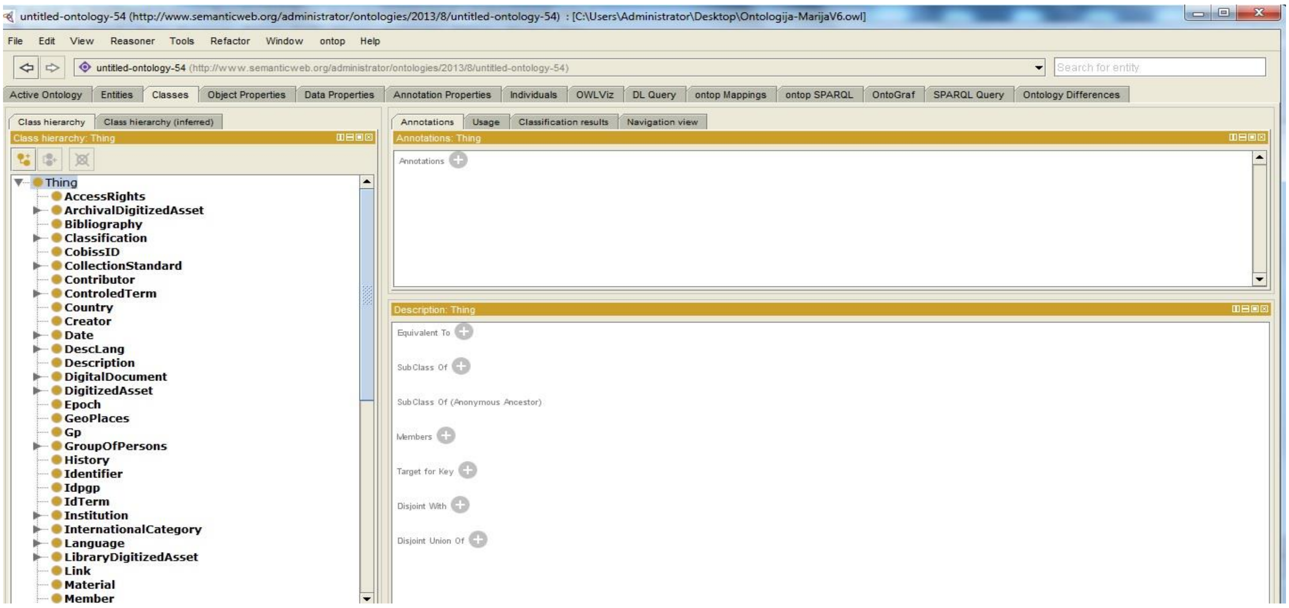
PersonAsset

- Name
  – First Name
  – Middle Name
  – Family Name
- Version of Name
  – First Name
  – Middle Name
  – Family Name
- Nickname
- Day of Birth
- Day of Death
- Sex
  – Type
  – Period
- Biography
  – Biography
  – Language of biography
- Resources related to the person
- Picture(s)
- Note
  – Note
  – Language of note
- Record creation date
- Record creator(s)
- Record owner

In the ontology of standards for digital and digitized documents, object **PERSON** is implemented in the following way: **Person** is represented as a class in the ontology and **Biography**, **DayOfBirth**, **DayOfDeath**, **Nickname**, **PersonID** are adjoined subclasses. Other meta-data takes place in the ontology as a **DataProperty** or **ObjectProperty**. For example, **Creator** becomes **ObjectProperty** named **hasCreator**. The reason for this is the impossibility of ontology classes repetition with the same name . Since the meta-data **Creator** appears in facilities **DigitalDocument**, **DigitizedAsset** and **CollectionStandard**, then **Creator** may not be a subclass but **ObjectProperty**.

Meta-data **PersonID** has a string format which is defined by the DataProperty using the standard property **hasString**. The part of the described example can be shown in the Figure 1. For the ontology development, software package Protégé 4.3 was used. After installing the software Protégé, it was necessary to install additional plug-ins and drivers which are available for the download on [8], [9].

The process of developing an ontology using Protégé can be described in several steps. At the top of the class hierarchy is the class **Thing**. All classes of ontology are always subclasses of **Thing**. Another step is to define primitive classes of the ontology which can be made entering **addsubclass**. Some classes of ontology (as a **Person** for example) have their own subclasses, which are also defined by entering **addsubclass**. Next step is to add object, data and anotation properties. At the top of hierarchy of the object properties is **topObjectProperty**. In regards of the data properties, **topDataProperty** is in the top of hierarchy of all data properties. For example, the domain of the DataProperty **hasGp** is the **Type**.


untitled-ontology-54 (http://www.semanticweb.org/administrator/ontologies/2013/8/untitled-ontology-54) : [C:\Users\Administrator\Desktop\Ontologija-MarijaV6.owl]
File Edit View Reasoner Tools Refactor Window ontop Help
untitled-ontology-54 (http://www.semanticweb.org/administrator/ontologies/2013/8/untitled-ontology-54)
Search for entity
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query ontop Mappings ontop SPARQL OntoGraf SPARQL Query Ontology Differences
Class hierarchy Class hierarchy (inferred)
Class hierarchy: Thing
Annotations Usage Classification results Navigation view
Annotations: Thing
Annotations
Description: Thing
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)
Members
Target for Key
Disjoint With
Disjoint Union Of

Thing
- AccessRights
- ArchivalDigitizedAsset
- Bibliography
- Classification
- CobissID
- CollectionStandard
- Contributor
- ControledTerm
- Country
- Creator
- Date
- DescLang
- Description
- DigitalDocument
- DigitizedAsset
- Epoch
- GeoPlaces
- Gp
- GroupOfPersons
- History
- Identifier
- Idpgp
- IdTerm
- Institution
- InternationalCategory
- Language
- LibraryDigitizedAsset
- Link
- Material
- Member

Figure 1. Some classes in the ontology

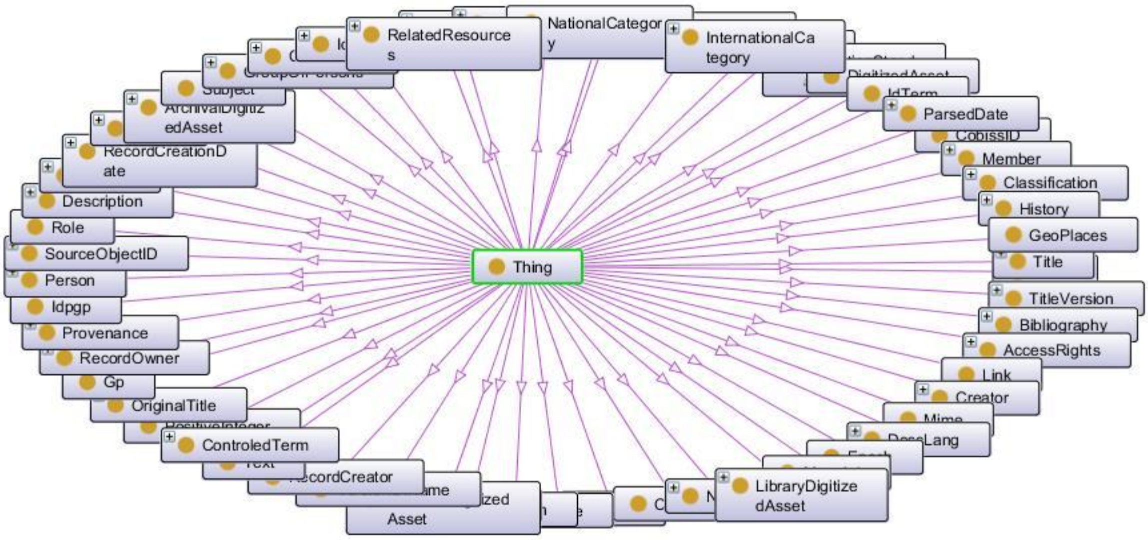


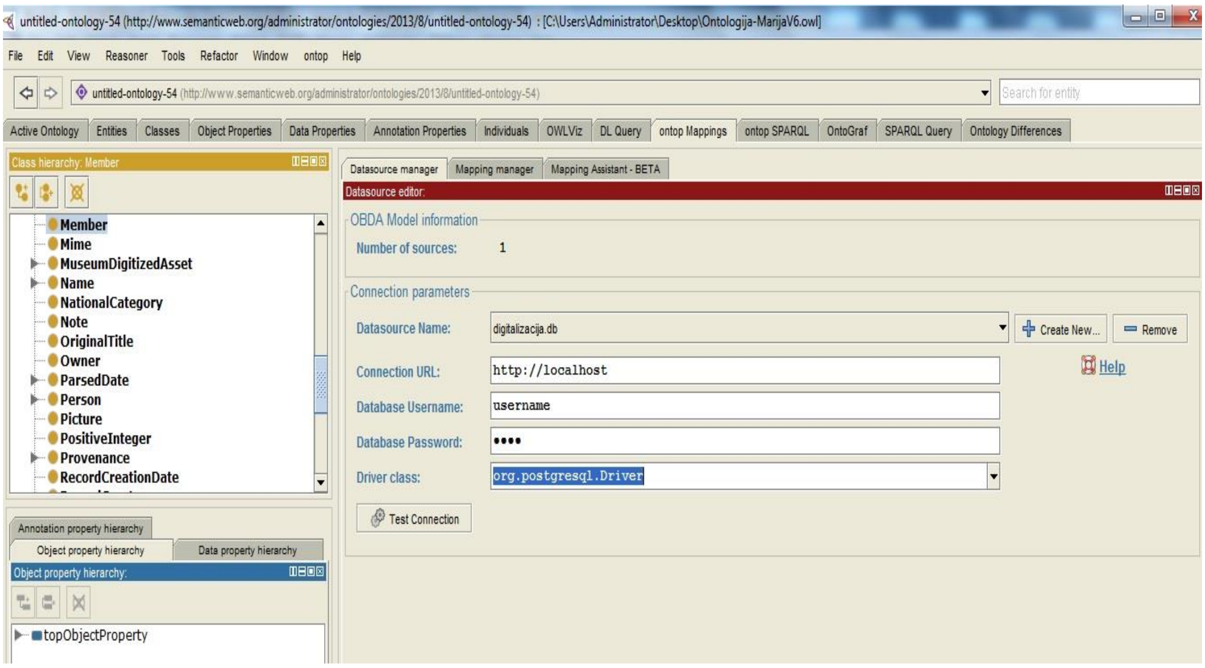
Figure 2. Graphic interpretation of the ontology

To set up queries over the ontology, it is primarily necessary to start **Reasoner** that ensures the consistency. From the drop-down menu in the Protégé, it is possible to select several reasoners such as: Fact++, Hermit or OnTop. After initialization of the **Reasoner**, the queries can be set up using the tab for DL query. The ontology can be illustrated using the graphic display, using tab OntoGraf, which can be seen in Figure 2. A connection between ontology and the database can be made using the tab **OnTop Mappings** [10], as in the Figure 3.

For possible connections of ontology with other systems, such as databases, servers, clients, and creation of their mutual communication, in literature it is possible to find several solutions [11], [12], [13], [14], [15].

One possible solution is OBDA (Ontology Based Data Access) plug-in for Protégé. OBDA plug-in is providing the concept of mapping between ontology and a database after the connection is made [16].

If it is necessary to access the database that stores information in the form of RDF triplets form (subject, predicate, object), then D2RQ language is other possible solution. Data from the database can be accessed by use of the yacht domain ontology, in accordance with the defined mapping [17].



Number of sources: 1

Datasource Name: digitalizacija.db

Connection URL: http://localhost

Database Username: username

Database Password: ••••

Driver class: org.postgresql.Driver

Figure 3. Onto Mappings plug-in for connection between the database and ontology

4. Conclusion and Future Work

Starting from the meta-data obtained in the form of XSD syntax, using the Protégé software, the ontology for description of digital documents and collections by NCD standards was designed. During the design process, it was necessary to obtain classes, subclasses, dataProperty, ObjectProperty, AnotationProperty, and many other relations between the XSD and OWL syntax. Once the ontology was made, the connection with other systems such as the database was possible. For the future, work of storing meta-data of cultural heritage in the Sesame RDF repository is proposed [18], [19], [20].

The main part of this system is a component of the Storage and Inference Layer (SAIL). On top of SAIL there is a functional architecture of the module which is used for the data search, which may be stored in the database. Other functionalities of this module is addition and deletion of statements. To ensure effectiveness and system usage for the clients, Access APIs was used. Sesame enables very easy connection to databases such as MySQL. There are several important plug-in components that would be used in the implementation, including: **OWLIM**, **Bigdata**, **uSeekM**, **phpSesame**.

Acknowledgements

The work presented here was supported by the Serbian Ministry of Education and Science (project III44006).

References

[1] Available at: http://en.wikipedia.org/wiki/Semantic_Web (15.12.2013)
[2] Available at: http://infosys3.elfak.ni.ac.rs/nastava/attach/rusn_lab4/Projekat+INFOSYS.pdf (15.12.2013)
[3] Ferdinand, Matthias, Christian Zirpins, and David Trastour. "Lifting XML schema to OWL" Web Engineering. Springer Berlin Heidelberg, 2004, 354-358.
[4] Available at: http://www.ncd.org.rs/ncd_sr/standards.html (15.12.2013)
[5] Zoran Ognjanović, Tamara Butigan-Vučaj, Bojan Marinković, NCD Recommendation for the National Standard for Describing Digitized Heritage in Serbia, Metadata and Semantics 2009, 45-54
[6] Bojan Marinković, Zoran Ognjanović, Tamara Butigan Vučaj, A distributed implementation of a catalog of digitized cultural collections, Proceedings from NCD, 21 (2012), 19-24
[7] Available at: http://www.ncd.org.rs/ncd_sr/standards/ (15.12.2013)
[8] Available at: http://protege.stanford.edu/download/download.html (15.12.2013)
[9] Available at: http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library (15.12.2013)
[10] Available at: http://ontop.inf.unibz.it/ (15.12.2013)
[11] Available at: http://protegewiki.stanford.edu/wiki/Working_with_the_Database_Backend_in_OWL (15.12.2013)
[12] Available at: https://babbage.inf.unibz.it/trac/obdapublic/wiki/SimpleHelloWorldTutorial (15.12.2013)
[13] Ahmed Maalel,Habib Hadj mabrouk ,Lassad Mejri and Henda Hajjami Ben Ghezela, Development of an Ontology to Assist the Modeling of Accident Scenarii "Application on Railroad Transport", arXiv:1203.1021
[14] Rodriguez-Muro, Mariano, Lina Lubyte, and Diego Calvanese. "Realizing ontology based data access: A plug-in for protégé." Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. IEEE, 2008.
[15] Gali, Anuradha, et al. "From ontology to relational databases." Conceptual Modeling for Advanced Application Domains. Springer Berlin Heidelberg, 2004. 278-289.
[16] Available at: https://babbage.inf.unibz.it/trac/obdapublic/wiki/SimpleHelloWorldTutorial (15.12.2013)
[17] Available at: http://d2rq.org/d2rq-language (15.12.2013)
[18] Sesame framework, Available at: http://www.openrdf.org (15.12.2013)
[19] J. Broekstra, A. Kampman, RDF(S) Manipulation, Storage and Querying using Sesame, 3rd International Semantic Web Conference (ISWC2004), 2004.
[20] Sesame documentation: http://www.openrdf.org/documentation.jsp (15.12.2013)

mmilojevic@mi.sanu.ac.rs
zorano@mi.sanu.ac.rs
bojanm@mi.sanu.ac.rs