

Adding Synonyms To A Concept In An Ontology

Herlina Jayadianti, Lukito Edi Nugroho, Paulus Insap Santosa

Department of Electrical Engineering and Information Technology
Universitas Gadjah Mada Yogyakarta, Indonesia
herlinajayadianti@gmail.com, Lukito@ugm.ac.id,
insap@jteti.gadjahmada.edu

Wahyu Widayat

Department Economic Development
Universitas Gadjah Mada
Yogyakarta, Indonesia
wahyu@mep.ugm.ac.id

Carlos Alberta Baptista Sousa Pinto

Department of Information System
Universidade do Minho Guimaraes Portugal
csp@dsi.uminho.pt

Abstract— The purpose of this paper is to provide research based understanding of adding synonyms to a concept in an ontology. We hypothesize that managing synonyms with a good taxonomy and a good integration process are good approaches to organize and share knowledge. This paper can give a discourse to a group of people in different societies that want to share data using different concepts in the same domain. We will generate a common set of terms based on the terms of several different storage devices, used by different societies, in order to make data retrieval independent of the different perceptions and terminologies used by those societies. We use ontologies to represent the particular knowledge of each society and integration techniques to find relations between terms used in those ontologies.

Keywords—*Ontology; Knowledge; Synonyms; Taxonomy*

I. INTRODUCTION

Nowadays many department (community) are thinking how to get more knowledges and metadata by linking more systems in other community. There are great challenges to make all systems organizing knowledge and sharing metadata – to make it easy searched, indexed and used in different context. Metadata is data about data. Metadata describes how and when and by whom a particular set of data was collected, and how the data is formatted. Metadata is essential for understanding information stored in data warehouse and has become increasingly important in Web applications [1]. In this paper we will focus on metadata in specific domain - ‘Poverty’. For some community “Poverty” refers the state of one who lacks a certain amount of material possessions [2], [3]. For other community, “Poverty” refers to the deprivation of basic human needs, which commonly includes food, water, sanitation, clothing, shelter, health care and education (See Fig 1).

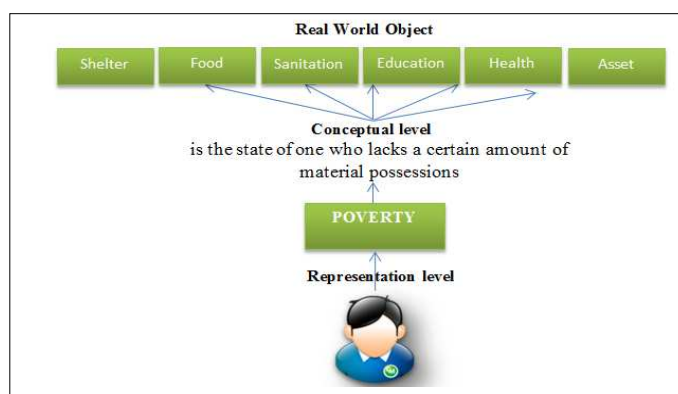


Fig. 1. Terms of Poverty

Regardless of the various definitions of poverty, in this paper we will focus on managing metadata in “Poverty” with many different terms therein.. Example : Some of differences in two ontologies see Table 1.

TABLE I. DIFFERENT TERM IN ONTOLOGY A AND ONTOLOGY B.

Table Column Head		
	Ontology A	Ontology B
Different names of the same concepts	HouseParameter	HouseCondition
Same term for different concepts	Floor : Only material of the floor	Floor : Maximum floor area and material of the floor
Scope	Includes : Material	Includes : Area and Material
Constructs used	Includes defined classes	Only primitive classes
Different modelling conventions	hasLargestFloorArea MadeFrom is an Object Properties	hasMinimumFloorArea , hasConditionOfFloor are Data Properties
Different modelling conventions and level of detail	HouseParameter class broken up to several subclasses : Energy, Floor, Roof, Sanitations	HouseCondition class broken up to subclasses : FloorCondition, RoofCondition and WallCondition.

Ontology Mapping [4], [5] is the process of relating similar

concepts or relations from different sources through some equivalence relation. Mapping allows finding correspondences between the concepts of two ontologies. If two concepts correspond, then they mean the same thing or closely related things. Currently, the mapping process is regarded as a promise to solve the problem between ontologies since it attempts to find correspondences between semantically related entities that belong to different ontologies. It takes as input two ontologies, each consisting of a set of components (classes, instances, properties, rules and axioms). [6] Based on the presented reasons, we believe that ontologies with common terms and common concepts are very important in a metadata sharing process. In this paper we describe an approach of organizing synonyms metadata using a common set of terms derived from several different ontologies. This paper is organized as follows: (1) Introduction; (2) Knowledge management and Implementation of the solution; (3) Conclusions.

II. KNOWLEDGE MANAGEMENT

The following figure shows classes hierarchy in this paper.

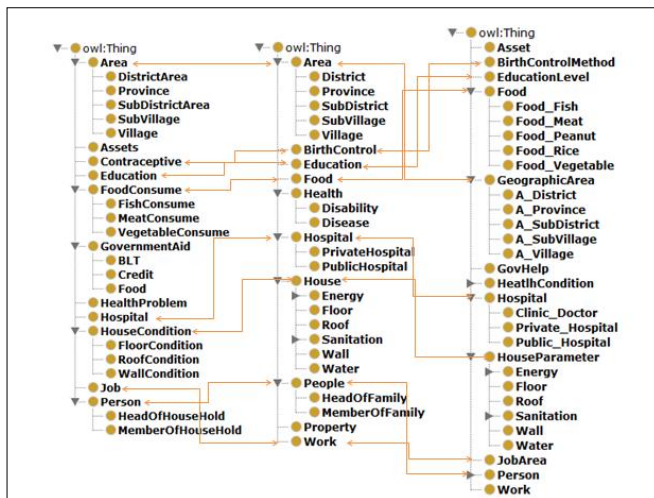


Fig. 2. Classes hirarchy

Bases on Fig.2 ontology UV1 consist of some classes such as Class Person, Class FoodConsume, Class Job, Class Floor and Class Area, each classes are related to each other.

```
?Person :hasRarelyEat ?FoodConsume.
?Person :hasJobPositionAs ?Job.
?Person :hasFloorMaterial ?Floor.
?Person :isLivinginVillage ?Area.
```

hasRarelyEat, hasJobPositionAs, hasFloorMaterial, and isLivinginVillage are some of ObjectProperties that are use in this ontology. The next step, each classes unless Class Person will given values as filters (See Fig 3).

```
?FoodConsume :FoodName ?value1.
?Job :JobName ?value2.
?Floor :TypeOfFloor ?value3.
?Area :hasName ?value4.
```

```
Prefix :
<http://www.semanticweb.org/UV1.owl#>
...
SELECT ?Person ?FoodConsume ?Job
?FloorCondition ?Area
WHERE {
    ?Person :hasRarelyEat
?FoodConsume.
?Person :hasJobPositionAs ?Job.
?Person :hasFloorMaterial
?FloorCondition.
?Person :isLivinginVillage ?Area.
?FoodConsume :FoodName ?value1.
?Job :JobName ?value2.
?FloorCondition :TypeOfFloor ?value3.
?Area :hasName ?value4.
FILTER (?value1 = 'Chicken' && ?value2
='Farmer' && ?value3 = 'Soil' && ?value4
= 'Widodomartani')}
```

Another example : knowledge in Institution B (here we called UV2) refers poor people as a people lack in Food, Job, House (hasLargestFloorAreaMadeFrom) Condition. In Ontology UV2 we build some classes such as Class Person, Class FoodConsume, Class Job, Class Floor and Class GeographicArea. Next step, Class Person will be connected with other classes, such as Class Food, Class JobArea, Class Floor, and Class GeographicArea (See Fig. 3). hasRarelyEat, hasJob, hasHouseFloorMadeFrom, and isLivinginSubDistrict are some of ObjectProperties that are use in this ontology. Furthermore ObjectProperties is used to connect any classes related.

```
?Person :hasFrequentlyEat ?Food.
?Person :hasLargestFloorAreaMadeFrom
?Floor.
?Person :hasjob ?JobArea.
?Person :isLiveinSubDistrict
?GeographicArea.
```

The next step, each class unless Class Person will give values as filters.

```
?Food :NameOfFood ?value1.
?JobArea :JobsArea ?value2.
?Floor :FloorMaterial ?value3.
?GeographicArea :hasCityName ?value4.
```

```
Prefix :
<http://www.semanticweb.org/ontologies/UV
2.owl#>
...
SELECT ?Person ?Food ?JobArea ?Floor
?GeographicArea
WHERE {?Person :hasFrequentlyEat ?Food.
?Person :hasLargestFloorAreaMadeFrom
?Floor.
?Person :hasjob ?JobArea.
?Person :isLiveinSubDistrict
?GeographicArea.
?Food :NameOfFood ?value1.
?JobArea :JobsArea ?value2.
```

```
?Floor :FloorMaterial ?value3.
?GeographicArea :hasCityName ?value4.
FILTER ( ?value1 = 'Chicken' &&
?value2 = 'Government' && ?value3 = 'Cement'
&& ?value4 = 'Moyudan' ) }
```

```
Prefix : <http://www.semanticweb.org/UV1.owl#>
SELECT ?Person ?FoodConsume ?Job ?FloorCondition ?Area

SELECT ?Person ?Food ?JobArea ?Floor ?GeographicArea
Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
```

Fig. 3. Class equivalent between ontology UV1 and Ontology UV2

Fig 3 shows that Class Area in Ontology UV1 \approx Class GeographicArea in Ontology UV2, but Class Area is more general than Class GeographicArea. Class FoodConsume in Ontology UV1 \approx Class Food in Ontology UV2. Class FoodConsume in Ontology UV1 is more specific than Class Food in Ontology UV2. We can see more terms that are equivalent in Figure 4. Instance Adi Sraiono in Ontology UV1 is a HeadOfHouseHold and instance Amat Sahari in Ontology UV2 is a HeadOfFamily.

<ul style="list-style-type: none"> pr-ADI_SRAJONO hasFinancialAbilityForSavingMoney "YES"^^string pr-ADI_SRAJONO hasGender "MALE"^^string pr-ADI_SRAJONO Type HeadOfHousehold pr-ADI_SRAJONO hasMinimumOnePeopleWorkinFamily "YES"^^string pr-ADI_SRAJONO hasDifferentClothesInSubActivity "YES"^^string pr-ADI_SRAJONO hasLivingInSubDistrict KILANGIN pr-ADI_SRAJONO hasJobPositionAs FARMER pr-ADI_SRAJONO hasSurname "ADI SRAJONO"^^string pr-ADI_SRAJONO hasFrequencyOfEatingMeat "NO"^^string pr-ADI_SRAJONO hasLivingSubDistrict NGEMPLAK pr-ADI_SRAJONO hasFrequencyOfEatingTime "YES"^^string pr-ADI_SRAJONO hasMaterialEat CHICKEN pr-ADI_SRAJONO hasMaterialEat COW_MEAT pr-ADI_SRAJONO hasLivingVillage WIDODOMARTANI pr-ADI_SRAJONO hasLivingSubDistrict SLEMAN pr-ADI_SRAJONO hasFloorMaterial floor-SOIL pr-ADI_SRAJONO hasConditionOfWall "GOOD"^^string pr-ADI_SRAJONO hasFinancialAbilityForDoctor "YES"^^string pr-ADI_SRAJONO hasConditionOfRoof "GOOD"^^string pr-ADI_SRAJONO hasMinimum2MealsAday "YES"^^string pr-ADI_SRAJONO hasConditionOfFloor "GOOD"^^string pr-ADI_SRAJONO hasFrequencyOfEatingChicken "NO"^^string pr-ADI_SRAJONO hasLivingName "ADI SRAJONO"^^string pr-ADI_SRAJONO hasMinimumFloorAreaInm2 "NO"^^string 	<ul style="list-style-type: none"> pr-AMAT_SAHARI hasLargestWallMadeFrom WALL-BRICK pr-AMAT_SAHARI hasProperty BICYCLE pr-AMAT_SAHARI hasUsedInDisposal SEPTICTANK pr-AMAT_SAHARI Type HeadOfFamily pr-AMAT_SAHARI isHeadOfFamilyOf man-KLINEH pr-AMAT_SAHARI hasLivingVillage SUNBERBANGUN pr-AMAT_SAHARI hasLivingSubDistrict KEMANGAN pr-AMAT_SAHARI hasHouseCondition "GOOD"^^string pr-AMAT_SAHARI hasLivingSubDistrict MOYUDAN pr-AMAT_SAHARI hasHouseCondition "GOOD"^^string pr-AMAT_SAHARI hasNumberOfRoomArea/Person/m2 18 pr-AMAT_SAHARI hasLivingName "AMAT"^^string pr-AMAT_SAHARI hasAge 81 pr-AMAT_SAHARI hasEnergyUsedForLighting ELECTRICITY pr-AMAT_SAHARI hasLivingName "SAHARI"^^string pr-AMAT_SAHARI hasLargestFloorAreaMadeFrom CEMENT pr-AMAT_SAHARI hasUsedTypeOfToilet FLUSH_TOILET pr-AMAT_SAHARI hasBackground PRIMARY_SCHOOL pr-AMAT_SAHARI Type Person pr-AMAT_SAHARI hasStatus "MARRIED"^^string pr-AMAT_SAHARI hasLargestRoofAreaMadeFrom ROOF-TILE pr-AMAT_SAHARI hasLivingName "AMAT_SAHARI"^^string pr-AMAT_SAHARI hasAbilityToRead "YES"^^string pr-AMAT_SAHARI hasLivingDistrict SLEMAN pr-AMAT_SAHARI hasLivingProvince YOGYAKARTA pr-AMAT_SAHARI hasStatus "MARRIED"^^string pr-AMAT_SAHARI hasLivingName "MARRIED"^^string pr-AMAT_SAHARI hasLivingName "MARRIED"^^string
--	---

Fig.4. Individual example in ontology UV1 and Ontology UV2 as a HeadOfFamily or HeadOfHouseHold

In the next example, we will show the number of poor people in the same district the district Ngemplak. Ontology UV1 use a term Area and Ontology UV2 use a term GeographicArea.

```
Prefix : <http://www.semanticweb.org/UV1.owl#>
...
SELECT ?Person ?Area
WHERE { ?Person :isLivinginSubDistrict
?Area.
?Area :hasName?value.
FILTER ( ?value = 'Ngemplak' ) }
```

```
Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
...
SELECT ?Person ?GeographicArea
WHERE { ?Person :isLiveinSubDistrict
?GeographicArea.
```

```
?GeographicArea :hasCityName
?value.FILTER ( ?value = 'Ngemplak' ) }
```

Term Area in Ontology UV1 \approx term GeographicArea in Ontology UV2. Term GeographicArea is more specific than term Area (See Fig.5 and Fig 6).

```
Prefix : <http://www.semanticweb.org/UV1.owl#>
SELECT ?Person ?Area
SELECT ?Person ?GeographicArea
Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
```

Fig. 5. Class equivalent between ontology UV1 and Ontology UV2

```
Prefix : <http://www.semanticweb.org/UV1.owl#>
WHERE { ?Person :isLivinginSubDistrict ?Area.
WHERE { ?Person :isLiveinSubDistrict ?GeographicArea.
Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
```

Fig. 6. Class equivalent between ontology UV1 and Ontology UV2

```
Prefix : <http://www.semanticweb.org/UV1.owl#>
?Area :hasName?value.
?GeographicArea :hasCityName ?value.
Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
```

Fig. 7. Class equivalent between ontology UV1 and Ontology UV2

Fig. 8 shows the result of query SPARQL in Ontology UV1 and Ontology UV2 based on notation above. SPARQL query in Fig. 10 use the same term Ngemplak as a value.

SPARQL query	SPARQL query																																																		
<pre>Prefix : <http://www.semanticweb.org/UV1.owl#> PREFIX rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl : <http://www.w3.org/2002/07/owl#> PREFIX xsd : <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs : <http://www.w3.org/2000/01/rdf-schema#> SELECT ?Person ?Area WHERE { ?Person :isLivinginSubDistrict ?Area. ?Area :hasName?value. FILTER (?value = 'Ngemplak') }</pre>	<pre>Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#> PREFIX rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl : <http://www.w3.org/2002/07/owl#> PREFIX xsd : <http://www.w3.org/2001/XMLSchema#> PREFIX rdfs : <http://www.w3.org/2000/01/rdf-schema#> SELECT ?Person ?GeographicArea WHERE { ?Person :isLiveinSubDistrict ?GeographicArea. ?GeographicArea :hasCityName?value. FILTER (?value = 'Ngemplak') }</pre>																																																		
<table> <tr> <th>Person</th><th>Area</th></tr> <tr> <td>pr-REPTO_SUHARSONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-MESDIYANTO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SUPARJO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SUMARDI</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SLAMET_SARIFUDIN</td><td>NGEMPLAK</td></tr> <tr> <td>pr-AMAD_TUKIDI</td><td>NGEMPLAK</td></tr> <tr> <td>pr-MUHADI</td><td>NGEMPLAK</td></tr> <tr> <td>pr-KARMIN</td><td>NGEMPLAK</td></tr> <tr> <td>pr-ANTON_HARYONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-ADI_SRAJONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-JARWO_UTOMO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-HARTO_WOYONO</td><td>NGEMPLAK</td></tr> </table>	Person	Area	pr-REPTO_SUHARSONO	NGEMPLAK	pr-MESDIYANTO	NGEMPLAK	pr-SUPARJO	NGEMPLAK	pr-SUMARDI	NGEMPLAK	pr-SLAMET_SARIFUDIN	NGEMPLAK	pr-AMAD_TUKIDI	NGEMPLAK	pr-MUHADI	NGEMPLAK	pr-KARMIN	NGEMPLAK	pr-ANTON_HARYONO	NGEMPLAK	pr-ADI_SRAJONO	NGEMPLAK	pr-JARWO_UTOMO	NGEMPLAK	pr-HARTO_WOYONO	NGEMPLAK	<table> <tr> <th>Person</th><th>GeographicArea</th></tr> <tr> <td>pr-AMAD_TUKIDI</td><td>NGEMPLAK</td></tr> <tr> <td>pr-ISOYONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SUMARDI</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SURJO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SUMARDIWOYO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SUGINO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-REPTO_SUHARSONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-SLAMET_SARIFUDIN</td><td>NGEMPLAK</td></tr> <tr> <td>pr-WARDI_WOYONO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-MESDIYANTO</td><td>NGEMPLAK</td></tr> <tr> <td>pr-ANTON_HARYONO</td><td>NGEMPLAK</td></tr> </table>	Person	GeographicArea	pr-AMAD_TUKIDI	NGEMPLAK	pr-ISOYONO	NGEMPLAK	pr-SUMARDI	NGEMPLAK	pr-SURJO	NGEMPLAK	pr-SUMARDIWOYO	NGEMPLAK	pr-SUGINO	NGEMPLAK	pr-REPTO_SUHARSONO	NGEMPLAK	pr-SLAMET_SARIFUDIN	NGEMPLAK	pr-WARDI_WOYONO	NGEMPLAK	pr-MESDIYANTO	NGEMPLAK	pr-ANTON_HARYONO	NGEMPLAK
Person	Area																																																		
pr-REPTO_SUHARSONO	NGEMPLAK																																																		
pr-MESDIYANTO	NGEMPLAK																																																		
pr-SUPARJO	NGEMPLAK																																																		
pr-SUMARDI	NGEMPLAK																																																		
pr-SLAMET_SARIFUDIN	NGEMPLAK																																																		
pr-AMAD_TUKIDI	NGEMPLAK																																																		
pr-MUHADI	NGEMPLAK																																																		
pr-KARMIN	NGEMPLAK																																																		
pr-ANTON_HARYONO	NGEMPLAK																																																		
pr-ADI_SRAJONO	NGEMPLAK																																																		
pr-JARWO_UTOMO	NGEMPLAK																																																		
pr-HARTO_WOYONO	NGEMPLAK																																																		
Person	GeographicArea																																																		
pr-AMAD_TUKIDI	NGEMPLAK																																																		
pr-ISOYONO	NGEMPLAK																																																		
pr-SUMARDI	NGEMPLAK																																																		
pr-SURJO	NGEMPLAK																																																		
pr-SUMARDIWOYO	NGEMPLAK																																																		
pr-SUGINO	NGEMPLAK																																																		
pr-REPTO_SUHARSONO	NGEMPLAK																																																		
pr-SLAMET_SARIFUDIN	NGEMPLAK																																																		
pr-WARDI_WOYONO	NGEMPLAK																																																		
pr-MESDIYANTO	NGEMPLAK																																																		
pr-ANTON_HARYONO	NGEMPLAK																																																		

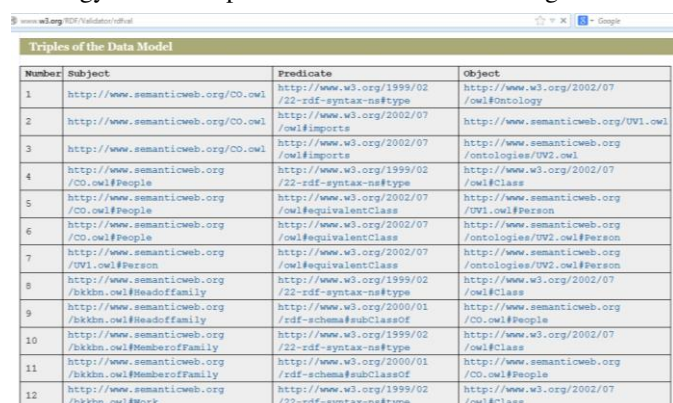
Fig. 8. Class equivalent between ontology UV1 and Ontology UV2

The next step is to import ontology UV1 and ontology UV2 into common ontology CO. Ontology CO in this project have an IRI address: <http://www.semanticweb.org/CO.owl>. Classes in the ontology Ontology UV1 and Ontology UV2 now appear in Ontology CO. Lots of implementations have been done using ontology alignment, but the process is only

carried out mapping between classes in one ontology. In this paper we perform ontology mapping among several ontologies that have been imported into one. Class People from Ontology CO with IRI <http://www.semanticweb.org/CO.owl#People> and Class Person from ontology UV1 with IRI <http://www.semanticweb.org/UV1.owl#Person>. Class Peole from Ontology CO with IRI <http://www.semanticweb.org/CO.owl#People> and Class Person from ontology UV1 with IRI <http://www.semanticweb.org/ontologies/UV2.owl#Person>.

```
<!--
http://www.semanticweb.org/CO.owl#People
-->
<owl:Class rdf:about="&CO;People">
<owl:equivalentClass
rdf:resource="http://www.semanticweb.org/
UV1.owl#Person">
<owl:equivalentClass
rdf:resource="http://www.semanticweb.org/
ontologies/UV2.owl#Person"/> </owl:Class>
```

Next Step is validation in RDF validator. We use RDF validator¹ and converter to validate small snippets of RDF/XML or Notation 3 (including N-Triples and Turtle). The data will be converted and outputted in the other format. RDF Validator and Converter is a tool for parsing RDF Statements and validating them against an RDF Schema. RDF ontology validation process for CO is shown in Fig. 9.



Number	Subject	Predicate	Object
1	http://www.semanticweb.org/CO.owl	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
2	http://www.semanticweb.org/CO.owl	http://www.w3.org/2002/07/owl#imports	http://www.semanticweb.org/UV1.owl
3	http://www.semanticweb.org/CO.owl	http://www.w3.org/2002/07/owl#imports	http://www.semanticweb.org/ontologies/UV2.owl
4	http://www.semanticweb.org/CO.owl#People	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
5	http://www.semanticweb.org/CO.owl#People	http://www.w3.org/2002/07/owl#equivalentClass	http://www.semanticweb.org/UV1.owl#Person
6	http://www.semanticweb.org/CO.owl#People	http://www.w3.org/2002/07/owl#equivalentClass	http://www.semanticweb.org/ontologies/UV2.owl#Person
7	http://www.semanticweb.org/UV1.owl#Person	http://www.w3.org/2002/07/owl#equivalentClass	http://www.semanticweb.org/ontologies/UV2.owl#Person
8	http://www.semanticweb.org/bkkm.owl#HeadoffFamily	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
9	http://www.semanticweb.org/bkkm.owl#HeadoffFamily	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.semanticweb.org/CO.owl#People
10	http://www.semanticweb.org/bkkm.owl#MemberoffFamily	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
11	http://www.semanticweb.org/bkkm.owl#MemberoffFamily	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.semanticweb.org/CO.owl#People
12	http://www.semanticweb.org/bkkm.owl#Work	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class

Fig. 9. RDF Validator and Converter – Testing Ontology UV1 and Ontology UV2 in Ontology CO – Class Person

Some reason why validations are important: (1) Validation is a debugging tool, (2) Validation is a future-proof quality check, (3) Validation eases maintenance, (4) Validation helps teach good practices, and (5) Validation is a sign of professionalism. The parser is a Java application that understands embedded RDF in XML, performs semantic and syntax checking of both RDF Schemata and Metadata

¹<http://www.w3.org/RDF/Validator/>
<http://www.rdfabout.com/demo/validator/validate.xpd>

instances, and validates statements across several RDF/XML namespaces. The results in RDF validator show that the created ontological views correctly reflect the model based on the design of the original relational database or the XML document.

CONCLUSIONS

In this research we try to managing synonym metadata by using ontology integration as a process to create a new ontology (Common Ontology). Using this approach it is possible to share metadata in different conceptualizations, different terminologies, and different meanings between different systems.

ACKNOWLEDGMENT

We would like to acknowledge Universidade do Minho and Universitas Gadjah Mada for the collaboration.

REFERENCES

- [1] D. Park, H.-C. Kwon, and S. Park, "Ontology for semantic representation of marine metadata," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, 2013, p. 66.
- [2] H. Jayadianti, L. E. Nugroho, C. S. Pinto, P. I. Santosa, and W. Widayat, "Solving problem of ambiguity terms using ontology," 2013.
- [3] C. S. Pinto, H. Jayadianti, L. E. Nugroho, P. I. Santosa, and W. Widayat, "Leveraging knowledge from different communities using ontologies," 2013.
- [4] J. Han and M. Kamber, *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.
- [5] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [6] S. K. Kumar and J. A. Harding, "Ontology mapping using description logic and bridging axioms," *Comput. Ind.*, vol. 64, no. 1, pp. 19–28, 2013.