ISSN: 2442-6571

Adding synonyms to concepts in ontology to solve the problem of semantic heterogeneity

Herlina Jayadianti [a,1,*], Lukito Edi Nugroho[a,2], Paulus Insap Santosa [a,3], Wahyu Widayat [b,4]

[a] *Electrical Engineering and Information Technology, Gadjah Mada University, Yogyakarta, Indonesia*
[b] *Economic Development, Gadjah Mada University, Yogyakarta,Indonesia*
[1] *Herlinajayadianti@gnail.com*; [2] lukito.nugroho@gmail.com;* [3] Insap@jteti.gadjahmada.edu,[4]wahyu@mep.ugm.ac.id

ARTICLE INFO

Article history:
Received April 7, 2015
Revised June 5, 2015
Accepted July 1, 2015

Keywords:
Ontology
Knowledge
Agreement
Synonyms
Poverty

A B S T R A C T

Nowadays many department (community) are thinking how to get more knowledges and metadata by linking more systems from 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. In this paper we will focus on metadata in specific domain - 'Poverty'2. Regardless of the various definitions of poverty, in this paper we will focus on managing metadata in "Poverty" with many different terms therein. Ontology Mapping 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). Based on the presented reasons, we believe that ontologies with common terms and common concepts are very important in a metadata sharing process.

I. Introduction

Synonyms are words with the same or similar meanings. Words that are synonyms are said to be synonymous, and the state of being a synonym is called synonymy. An example of synonyms are the words "*begin*" and "*commence*". Likewise, for term a"*long time*" or an "*extended time*", "*long*" and "*extended*", "*buy*" and "*purchase*", "*big*" and "*large*", "*quickly*" and "*speedily*", *"hospital"* and *"infirmary"*, "*on*" and "*upon*" become synonyms.

Other example is term *"poverty"*. For some people that already have a knowledge and information about poverty can easily say that poverty is the state of one who lacks a certain amount of material possessions or money. For other people, *"Poverty"* refers to the deprivation of basic human needs, which commonly includes food, water, sanitation, clothing, shelter, health care and education. Regardless of the various definitions of poverty, this paper will focus only on terms related to the reality of *"Poverty"*. Figure 1 shows that *"Poverty"* related with many things (terms) such as *"Shelter"*, *"Food"*, *"Sanitation"*, *"Health"*, and *"Asset"*.

It can be said that every reality will be represented by a variety of terms associated with it (Fig. 1). Ontology is one of the solution for the semantic integration problem [1], [2]. The *ontology integration and ontology interoperability* can be applied by discovering semantic correspondences [3] among a set of formal ontologies and (sometimes) creating a more complete ontology given that multiple source ontologies are available. Some of the most representative ontology definitions are: (1) Ontology as an explicit specification of a conceptualisation [4] , (2) Ontology is a formal description of entities, properties, relations, constraints and behaviours [5], and ontology is a formal explicit description of concepts in a discourse domain, where properties of each concept describe several characteristics, attributes of concepts and attributes' constraints [6] . It can be concluded that the issue of synonyms become the focus in this paper. The use of terms in each system is very dependent on each programmer, it does not become a problem until the system A and system B will be integrated. The main problem is how to equate the terms in both systems. As an example in the case study *"Poverty"*. System A and system B refers term *"Poverty"* as the deprivation of basic human needs. System A uses the term *"Money"*, *"Food"*, *"Water"*, *"Sanitation"*, *"Asset"*, *"Clothing"*, and *"Shelter"*. System B *uses the term "Salary"*, *"Feed"*, *"Water"*, *"Sanitation"*, *"Property"*, *"Clothing"*, and *"House"*. Synonyms are words with the same or similar meanings. An example of synonyms are the words "*Money"* and "*Salary" and "Wage"*. Likewise, for the term an "*Asset"* or an "*Property"*, "*Food"* and "*Feed"*, "*House"* and "*Shelter"*, "*People"* and "*Person"* become synonyms. This paper is organized as follows: (1) Introduction; (2) Knowledge management and Implementation of the solution; (3) Conclusions.

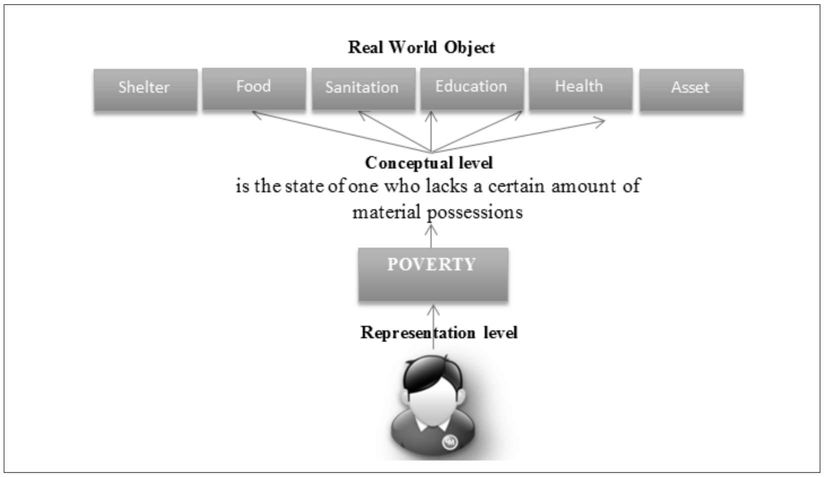

Fig. 1. Reality of Poverty – Representation level

II. Knowledge Management and The Implementation of the Solution

A. Architecture system

An architecture system is a sets of developing steps for designing system. Fig. 2 shows an architecture process of the system : Block 1 understanding of the realities of a domain, Block 2 building ontologies, Block 3 intergating ontolgies with mapping techniques, Block 4 illustrates this in an interface. The first block (Block 1) in an arsitecture system. Reality (actual state of the domain) support the emergence of data (fact about the domain). Information can be considered as an aggregation of data (processed data) which makes decision making easier. Information has usually got some meaning and purpose. Knowledge is derived from information in the same way information is derived from data. Knowlege is ussulay appear based on learning, thinking and proper understanding of the problem area. Block 1 shows the reality from domain x, reality can produces a variety of perceptions such as perception a, perception b, and perception c of a reality, next step perceptions stored in a data base (db1, db2, and db3) in each system. Each end user make a query based on the knowledge that they have. In accordance with the objectives of this dissertation, the first step that must be done is to manage the data into knowledge. Table in a data base managed into classes in knowledge base ontology .The process is done manually. Furthermore, the problem faced after ontology is made. Ontology a, ontology b, and ontology c consists of different name/term of classes, different name/term data properties, different name/term object properties and different name/term instances wich depends on the data source in the data base.

Example : (Poverty Domain)
Ontology a from institution A: *"Normally all family members have **meal** two or more times a day"*.
Ontology b from Institution B: "*Minimum two times per day the family have **food**"*
Meal and *food* have the same meaning, as well as *suit* and *clothes* or *clinic* and *hospital.* To be similar *(≅)* or not equal *(≠)* depend on several factors, such as the programmer's interpretation, the

needs of the system itself, and the domain/area. each term has always a strong relationship with the domain.

Block 2 and block 3 carried out mapping classes, properties, and instances between ontologies. The target is to combined different existing terminologies about the same reality used by different communities in order to get a common set of terms that can be transparently used by those communities, while maintaining the original terms in the data sources . A single ontology is no longer enough to support the tasks envisaged by a distributed environment. Multiple ontologies need to be accessed from several applications or systems. Ontology mapping is required for combining distributed and heterogeneous ontologies.

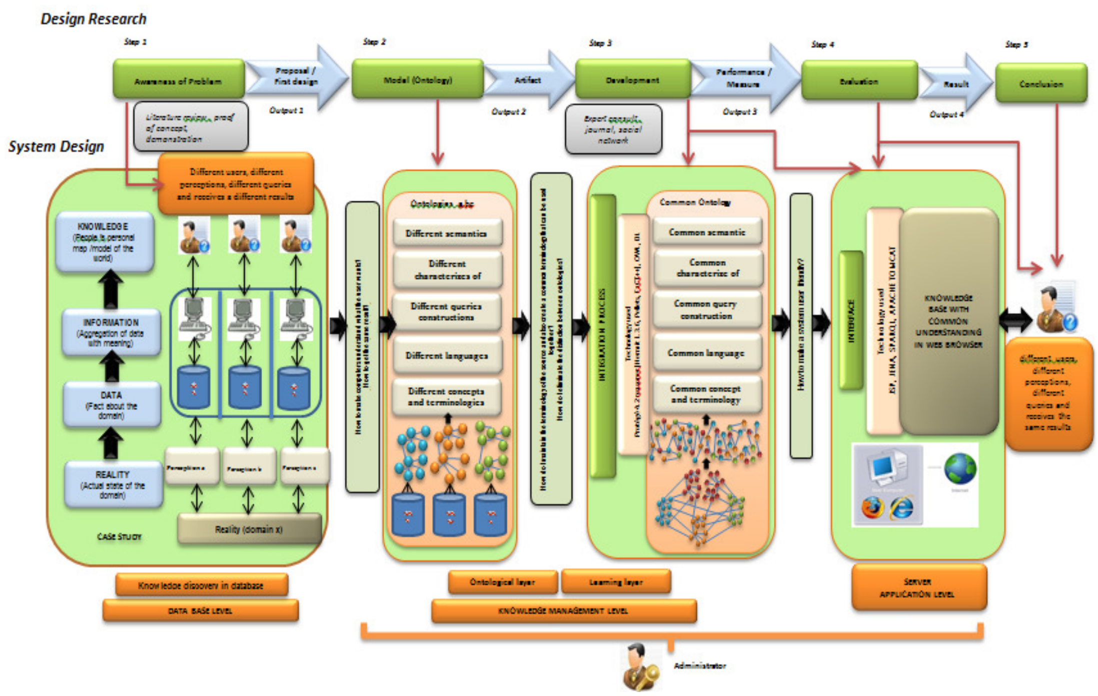

Fig. 2. Proposed Model

B. *Example of SPARQL query*

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.

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. 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.

?Person :hasJobPositionAs ?Job ≈ ?Person :hasjob ?JobArea
hasJobPositionAs ≈ hasjob
Job ≈ JobArea

Another example, Class Area in Ontology UV1 ≈ Class GeographicArea in Ontology UV2, but Class Area is more general than Class GeographicArea. Class FoodConsume in Ontology UV1 ≈ Class Food in Ontology UV2. ClassFoodConsume in Ontology UV1 is more specific than Class Food in Ontology UV2 (Fig. 3 and Fig. 4).

```
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

```
Prefix : <http://www.semanticweb.org/UV1.owl#>

SELECT ?Person ?Area

SELECT ?Person ?GeographicArea

Prefix : <http://www.semanticweb.org/ontologies/UV2.owl#>
```

Fig. 4. Class equivalent between ontology UV1 and Ontology UV2

III. Testing Ontology Model

Testing and evaluation of the results performed on a ontology model prototype which has been built in SPARQL testing (expert) where the testers who understand SPARQL language can perform manual testing by input SPARQL query. Here's an example of testing prototype application of ontology model for poverty case study. Testing covers 11 classes in Ontology UV1 (Area, Assets,Contraceptive, Education, FoodConsume, GovermentAid, HealthProblem, Hospital, HouseCondition, Job, Person) , 12 Classes in Ontology UV2 (Asset, BirthControlMethod, EducationLevel, Food, GeographicArea, GovHelp, HealthCondition, Hospital, HouseParameter, JobArea, Person, Work), are being integrated with classes in Ontology CO (People, Property, House, Work, Area, BirthControl, Health, Education, Hospital,Food). Testing also covers 67 Data Properties, 22 Object Properties, consist of more than 600 intances from Ontology UV1. Ontology UV2 42 Data Properties, 43 Object Properties, consist of more than 450 intances. Ontology CO covers 33 classes, 84 Object Properties, 121 Data Properties, and consist of more than 1500 instances (Table 1).

Table 1. Ontology Poverty Domain

	Poverty Domain Ontology UV1	***Poverty Domain Ontology UV2***	***Poverty Domain Ontology CO***
Class	11	12	33
Data Properties	67	42	84
ObjectProperties	22	43	121
Instances	>600	>450	>1500

Example question:

Class : Area, GeographicArea, Location, Person (HeadOfHouseHold, MemberOfHold), Person (HeadOfFamily, MemberOfFamily) , People (HeadOfFamily, MemberOfFamily)
Property :Living in, Live in
Instance : Ngemplak

Vol. 1, No. 2, July 2015, pp. 84-89

Q: Who is the Person that live in Ngemplak?

Validation = True	
UV1	Head of house hold that living in area "Ngemplak"
UV2	Head of family live in geographic area "Ngemplak"
CO	Head of family (OR head of house hold OR head of family) that living in (OR live in) location (OR Area OR geographic Area) "Ngemplak"

Class : Person (HeadOfHouseHold, MemberOfHold), Person (HeadOfFamily, MemberOfFamily) , People (HeadOfFamily, MemberOfFamily), Assets, Asset, Property
Property : have
Instance : television, TV, Televison

Q: Who is the Person that have Television (OR TV OR television)?

Validation = True	
UV1	Head of house hold have Assets "Television"
UV2	Head of family have Assset "tv"
CO	Head of family (OR head of house hold OR head of family) have Property (OR Assets OR Asset) "Televison", "tv", "television"

Next Step is validation in RDF validator. We use RDF validator (http://www.w3.org/RDF/Validator/ and http://www.rdfabout.com/demo/validator/validate.xpd) 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. 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 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.

IV. Conclusions

Domain poverty chosen because the problem of poverty is a very complex problems. In this paper we build models and ontology representing the n institutions to visualize the synonym problem. SPARQL queries performed to test the equivalent search. Furthermore, RDF validator is used to check online whether RDF / XML are valid. We propose a model for supporting the heterogeneity and exchange of different perceptions, that support or reject possible correspondence. Each actor can decide, according to its preferences. The proposed framework considers perceptions and terminologist are based on ontology semantics. This proposed model relies on formal perceptions or common perceptions from each ontologist creator. Moreovere in future work we intend to improve and automate the model with modul negottiation (agreement) process to enable actor to reach an agreement on a mapping when they differ in their ordering of terminologist types. Another interesting topic for future work would be to investigate how to argue about the whole perceptions. These perceptions could occur when a global similarity measure (agrrement) between the whole ontologies is applied. In this paper improvement measurement is done by testing of all components from integrated ontologies. Improvement better measurement would be a challenge in future work. In the future, the improvement of measurement will be done by involving ontology agreement

Acknowledgment

We would like to acknowledge Universitas Gadjah Mada for the support.

Vol. 1, No. 2, July 2015, pp. 84-89

References

[1] F. Hakimpour and A. Geppert, "Resolving semantic heterogeneity in schema integration," in *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, 2001, pp. 297–308.

[2] F. Hakimpour and S. Timpf, "Using ontologies for resolution of semantic heterogeneity in GIS," in *4th AGILE Conference on Geographic Information Science*, 2001.

[3] Y. Xue, H. Ghenniwa, and W. Shen, "Ontological view-driven semantic integration in collaborative networks," *Leveraging Knowl. Innov. Collab. Netw.*, pp. 311–318, 2009.

[4] T. Gruber, "What is an Ontology," *Encycl. Database Syst.*, vol. 1, 2008.

[5] M. Grüninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," 1995.

[6] N. F. Noy, D. L. McGuinness, and others, *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, 2001.