SOLVING PROBLEMS OF DATA HETEROGENEITY, SEMANTIC HETEROGENEITY AND DATA INEQUALITY – AN APPROACH USING ONTOLOGIES

Jayadianti, Herlina, Universidade do Minho, Azurem, Guimaraes, Portugal, herlinajayadianti@gmail.com

Pinto, Carlos Sousa, Universidade do Minho, Azurem, Guimaraes, Portugal, csp@dsi.uminho.pt

Nugroho, Lukito Edi, Universitas Gadjah Mada, Yogyakarta, Indonesia, lukito@mti.ugm.ac.id

Santosa, Paulus Insap, Universitas Gadjah Mada, Yogyakarta, Indonesia, insap@mti.ugm.ac.id

Abstract

Knowledge is people's personal map and people's personal model of the world. Knowledge acquisition involves complex cognitive processes such as perception, communication, and reasoning. According to the knowledge differences, then it is possible for people have a different perception to attain awareness or understand the environment or reality. This paper provides a case study where there is a group of people in different communities managing data using different perceptions, different concepts, different terms (terminologies), and different semantics to represent the same reality. Perceptions are converted into data, and then saved into separate storage devices that are not connected to each other. Each user – belonging to different communities - use different terminologies in collecting data and as a consequence they also get different results of that exercise. It is not a problem if the different results are used for each community, the problem occur if people need to take data from another communities, sharing, collaborating and using it to get a bigger solution. In this paper we present an approach to generate a common set of terms based on the terms of several and different storage devices, used by different communities, in order to make data retrieval independent of the different perceptions and terminologies used by those communities. We use ontologies to represent the knowledge and discuss the use of mapping and integration techniques to find correspondences between the concepts used in those ontologies. We discuss too how to derive a common ontology to be used by all the communities. We can find in literature several documents about the theoretical concepts and techniques that can be used to solve the described problem. However, in this paper we are presenting a real implementation of a system using those concepts.

Keywords: Knowledge; Perception; Terminology; Ontology; Common Ontology; Mapping.

1 INTRODUCTION

Researchers in the fields of databases and information integration have produced a large body of research to facilitate interoperability between different systems (Noy, 2004). Those studies range from techniques for matching and mapping database schemas to mechanisms to answer questions using different data sources. Using ontologies is one of the possible approaches to implement matching and mapping processes. Based on research of Noy (Noy, 2004), ontologies are the study of other disciplines related to data and semantic heterogeneity in structured knowledge. While there are many definitions of what an ontology is (Gruber, 1993, 1995), the common thread in these definitions is that an ontology is some formal description of a domain of discourse, intended for sharing among different applications, using different data, having different semantics and expressed in a language that can be used for reasoning (Noy, 2004). Ontologies have gained popularity in the AI community as a mean for establishing explicit formal vocabulary to share between applications (Noy, 2004). One of the main problems related to the use of different representations of a reality, done by different communities, is the fact that those communities have different perceptions about that reality and, as a consequence, we can identify a problem of data and semantic heterogeneity. Using ontologies is not the only way to solve the problem of heterogeneity. Despite many advantages in using it, it has not been able to overcome the referred problem of data and semantic heterogeneity. We need to map ontologies in order to make compatible the different terminologies (sets of terms). While having some common ground, either within an application area or for some high-level general concepts, this could alleviate the problem of data and semantic heterogeneity (Noy, 2004). Based on the presented reasons, we believe that mapping ontologies is the right way to solve the problem of data and semantic heterogeneity.

2 FUNDAMENTAL CONCEPTS

The brain links all these things together into a giant network of ideas, memories, predictions, and beliefs. Everything is inter-connected in the brain. Computers are not artificial brains. They do not understand what they are processing, and can not make independent decisions based upon what we tell them. There are two sources that the brain uses to build knowledge – data and information. In the research underlying in this paper we use reality about poverty as a case study, to demonstrate the correctness of our approach. Figure 1 shows the relation between knowledge and ontologies.

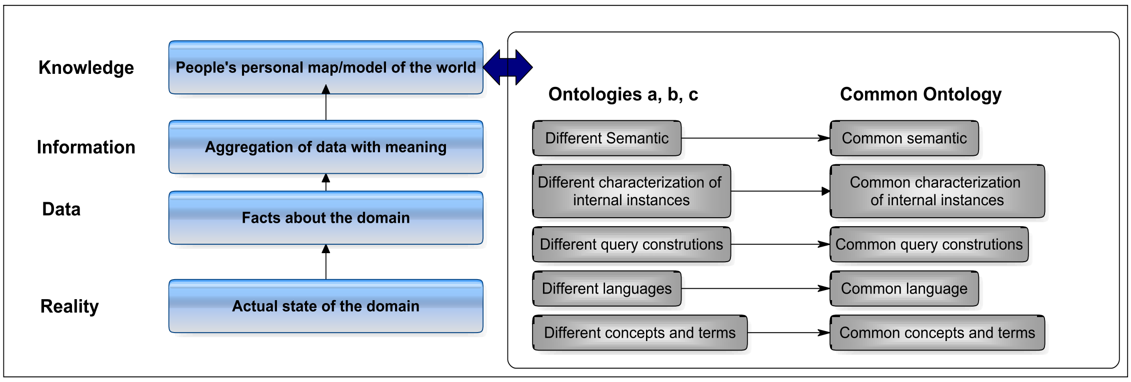

Figure 1. Managing Knowledge with Ontologies

At the level "Reality" we represent the actual state of a particular domain. At this level we can find lots of data. Data are facts in the context of a domain of discourse. At the next level, establishing relationships between data, it is possible to derive information and expand it beyond the limits of understanding of each person. Knowledge is obtained by adding experience, reflection and reasoning to information. If different information is discussed by people, it is easy to understand each other about what is inside their minds, either by argue or communicate. But what happens if that differences exist at the machine level? We need to combine information so that machines can "think" and understand the concepts we can find inside human brains. To do that, we can use ontologies to represent data and information of the several communities.

In the right side of Figure 1 we present a representation of this. Ontology is some formal description of a domain of discourse. However, ontology is not enough to make computers understand what is necessary. Scattered ontologies should then be incorporated and integrated into a new ontology, a common ontology. Ontology integration is one way to solve the problem of data, information, and semantic heterogeneity. Semantic heterogeneity on naming includes problems with synonyms (same concept with different terms) and homonyms (same term with different meaning). Semantic heterogeneity occurs when the same reality is modeled by two or more diferrent people or systems

(Bouzeghoub, 2004). In our research, we decided to use mapping process to find the similarities and correspondences between terms of the ontologies. Mapping works with logical axioms, typically expressing logical equivalences or inclusions among ontology terms. The integration of ontologies creates a new ontology by reusing other available ontologies through assembling, extending, or specializing operations (Xue, 2010). The goal of ontology integration is to derive a more general domain ontology (common ontology) from other several ontologies.

Every person has their own knowledge. They can justify everything based on their thoughts, perceptions and conceptualizations. An ontology is a specification of a conceptualization (Gruber, 1993). Conceptualization is an abstraction of the external world inside an individual mind. It can be used to construct one or several concepts and also to interpret some reality in a conceptual way. As referred before, discussing and sharing is one way to make the same perceptions between humans. If the differences of perceptions are happening between machines we need a common conceptualization and a process to do that. To represent conceptualizations between machines we can use ontologies. Different conceptualizations are specified by different ontologies. We can use a process that maps the terms of one ontology into the terms of another one.

There are many definitions of what an ontology is (Gruber, 1993, 1995), the common thread in these definitions is that an ontology is some formal description of a domain of discourse, intended for sharing among different applications, using different data, having different semantics and expressed in a language that can be used for reasoning (Noy, 2004). Ontology consists of classes, data properties, object properties, and instances.

A class or concept is a logic description, and it can be defined intentionally in terms of a description that specifies the properties that objects must satisfy to belong to the class. These descriptions are expressed using a language that allows the construction of composite descriptions, including restrictions on the binary relationships connecting objects. A class can also be defined extensionally by enumerating its instances. Classes are the basis of knowledge representation in ontologies and represent concepts. Data properties and object properties are related and operate among the various objects populating the ontology. A property is a directed relation that specifies class characteristics.

Instances or individuals are objects which cannot be divided without losing their structural and functional characteristics.

3 AN APPROACH TO SOLVE THE PROBLEM

In this section, we describe the problem we are trying to solve and an approach to solve it. There is a reality; the reality is the state of a particular domain as it is. People have their own knowledge, and it is independent to each other then it is possible that they have different opinions, use different sets of data and have diverse perceptions about the same reality.

Figure 2 represents several groups of people (communities) that faced reality with different perceptions (Perception_1, Perception_2, and Perception_n). Perceptions are converted into data and saved into separate storage devices that are not connected. The storages (db1,db2, and dbn) contain different data, different concepts, different terms, and different semantics. It depends on people in the group who look at reality (policy makers) and people who create and store data (users that use technology). Users who deal with computers has a very important role in controlling and changing the terminology and semantic of the data. Each group (community) uses technology to find data. It is very difficult for those different groups to get similar results and the problem happen if people need to use data from another group in order to share, collaborate and use it to get a bigger solution.

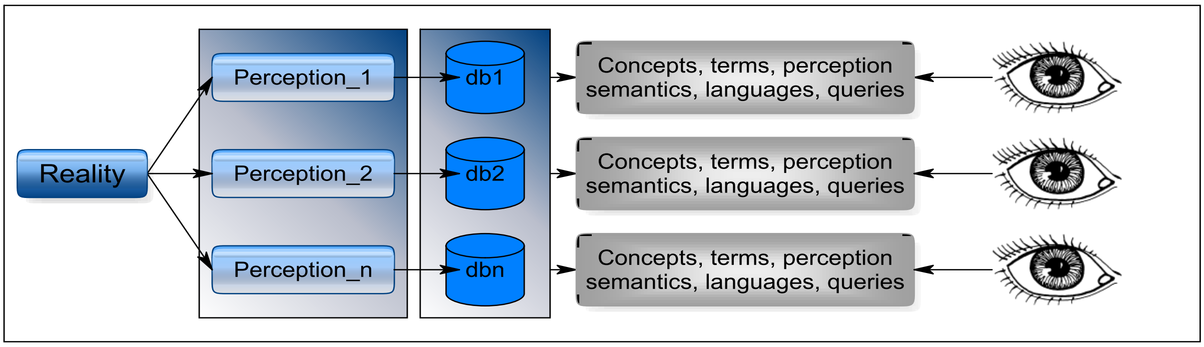

Figure 2. The Problem of different perceptions

The solution presented in this paper is based on different knowledge about the same reality based on different perceptions and uses a mechanism that works with a set of common concepts, common terms, common semantics, common languages, and a set of common queries (See Figure 3). Users in each community still can use their different terms, concepts, and perception for querying the system. According to the proposed solution, we aim to get similar answers from such a common layer that acts like an interface between the different systems and the users.

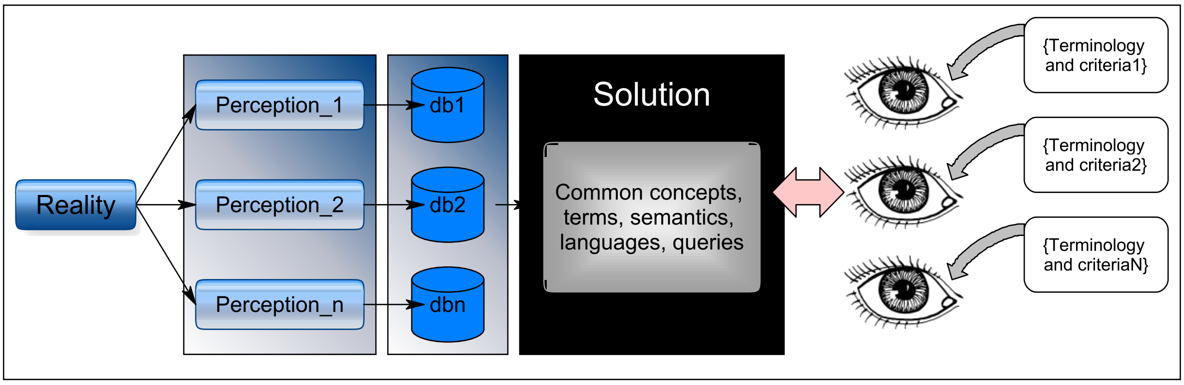

Figure 3. The Towards a Solution of Different Perceptions

4 IMPLEMENTING THE SOLUTION

4.1 Ontology Integration

Ontology integration is one way to solve the problem of semantic heterogeneity and it can be done using several approaches. For example, merging, matching or mapping. The integration of ontologies creates a new ontology by reusing other available ontologies through assembling, extending, or specializing operations. Using integration the source ontologies and the resultant ontology can have different amounts of information (Xue, 2010). Ontology integration process implies several steps. According to Noy (Noy et al., 2003) there are some specific challenges in ontology integration process:

- Finding similarities and differences between ontologies in an automatic and semi-automatic way;
- Defining mappings between ontologies;
- Developing an ontology integration architecture;
- Composing mappings across different ontologies;
- Representing uncertainty and imprecision in mappings.

Particularly, in ontology integration, some tasks should be performed to eliminate differences and conflicts between those ontologies (Noy et al., 2003). Ontology integration is used to find similarities and differences between ontologies. Based on the fundamental concepts above and on the aspects showed in Figure 3 the solution for solve the problem is ontology integration (see Figure 4). The goal of ontology integration is to derive a more general domain ontology (common ontology) from several other ontologies in the same domain, into a consistent unit. The domain of both the integrated and the resulting ontologies is the same.

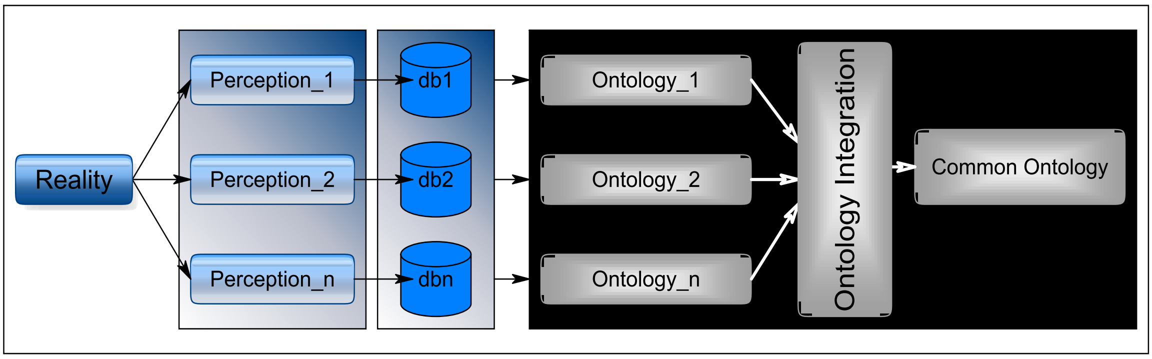

Figure 4. Different perception of poverty problem

4.2 Ontology mapping

A single ontology is not enough to support the tasks envisaged by a distributed environment. Multiple ontologies need to be accessed from several applications or systems (Kalfoglou & Schorlemmer, 2003). Ontology mapping is required for combining distributed and heterogeneous ontologies (Choi, Song, & Han, 2006). Based on Choi (Choi et al., 2006) ontology mapping is classified into the following three categories: (1) Mapping between an integrated global ontology and local ontologies. In this case, ontology mapping is used to map a concept found in one ontology into a view or a query over other ontologies; (2) Mapping between local ontologies. In this case, ontology mapping is the process that allows us to transform the source ontologies entities into a new ontology, using semantic relations. The source and target ontologies are semantically related at a conceptual level; (3) mapping by ontology merging and alignment. In this case, ontology mapping establishes correspondence among source (local) ontologies to be merged or aligned, and determines the set of overlapping concepts (synonyms) or unique concepts of those sources. This mapping process identifies similarities and conflicts between the various source ontologies to be merged or aligned.

4.3 Ontology mapping between local ontologies

Based on Choi (Choi et al., 2006), this category of mapping provides interoperability for highly dynamic, open and distributed environments and can be used for mediation between distributed data in such environments. This kind of mapping is more appropriate and flexible for scaling up to the web than mappings between an integrated global ontology and local ontologies. In this case, the mapping process enables ontologies to be contextualized because it keeps their contents local. It can provide interoperability between local ontologies when different local ontologies cannot be integrated or merged because of mutual inconsistency of their information.

Two ontologies can’t be integrated or merged as a single ontology if those two ontologies contain mutually inconsistent concepts (Choi et al., 2006). However, the two ontologies can be mapped using bridge rules which are the basic notion about the definition of context mappings. A mapping process between two ontologies is a set of bridge rules using the following operators: $\equiv$, $\cong$, $\neq$, $\subseteq$, $\supseteq$, $\cap$, $\cup$, $\forall$, $\exists$, * (related) and $\perp$ (unrelated). For example A is more general than B ($A \supseteq B$), A is less general than B ($A \subseteq B$), A is similar to B ($A \cong B$) and A is not equal to B ($A \neq B$).

4.4 Example Case Study

To demonstrate the capabilities of the described mechanisms we implemented a mapping process between local ontologies using data about poverty. Poverty is not the focus of our research. We just use that case as a scenario that allows us to demonstrate our approach. We combine different existing terminologies about the same reality (poverty in this case) 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. We use Indonesia as the country for the example because in that country there are several institutions in charge of dealing with poverty data, generating problems due to differences in the criteria used by them to make their surveys, even considering that the semantics of these different criteria are the same.

For example, let's imagine that there are two institutions A and B that are responsible for collecting data on poverty. Each institution has a different system and use different terms to describe the same domain.

Example:

if *ConsumptionOfFood* $\cong$ *FoodConsumption*

then *FoodConsumption* $\cong$ *ConsumptionOf Food*

if *Food* $\cong$ *Meal and Meal* $\cong$ *Diet*

then *Food* $\cong$ *Diet*

Possibilities that could happen are the similarity or difference of each term. As an example: The probability of *People* $\cong$ *Person* is similar to the probability of *People* $\neq$ *Person*. To be similar *(*$\cong$*)* or not equal *(*$\neq$*)* depend on several factors, such as the interpretation of the technical staff, the needs of the system itself, and last but not least the domain/area that we are talking about. One term has always a strong relationship with the domain.

Another example:

HeadOfFamily $\cong$ *HouseHolder* ***or*** *HeadOfFamily* $\neq$ *HouseHolder*

"HeadOfFamily" is a part of Family, and *"HeadOfFamily"* is also subclass of the class *"Person"*. *"HeadOfFamily"* means that he/she has a very important role in the Family, and he/she is the leader of the family, can have a job (or not), and have an income (or not). When we refer "income", we can consider also several terms with the same meaning, for example *Salary, Wage, or Money.*

But, since we can identify different sets of terms (terminologies) about the same reality (poverty, in this case), that are appointed by governmental departments to calculate the levels of poverty, we get a problem. The solution to this problem is in the machine, not in humans. If the difference in poverty terminologies occurs among users, it is easy to find common perception, but if it occurs between machines, it would require an intelligent system that can understand the differences in terminology of each institution who works in the same domain.

In our research, we implement an importing process on the source ontologies (UVs) into a target ontology, which we call a common ontology (CO). CO consists of common terms. Common term is a common word recognized and used with the same meaning by different sets of people. CO is expected to overcome the differences that exist in the UVs. CO contains terms equated with each term in the source UVs. Figure 5 shows the relationship scheme between terms in the source UVs (Ontology UV1, Ontology UV2, and Ontology UV3) and the common terms in the CO. In this figure, CO terms are indicated by black dots. Each ontology is a model (representation) of the same domain. It is inevitable to get considerable heterogeneity of data, and how to create a mutual understanding of the semantic system is the main goal of the research we are doing.

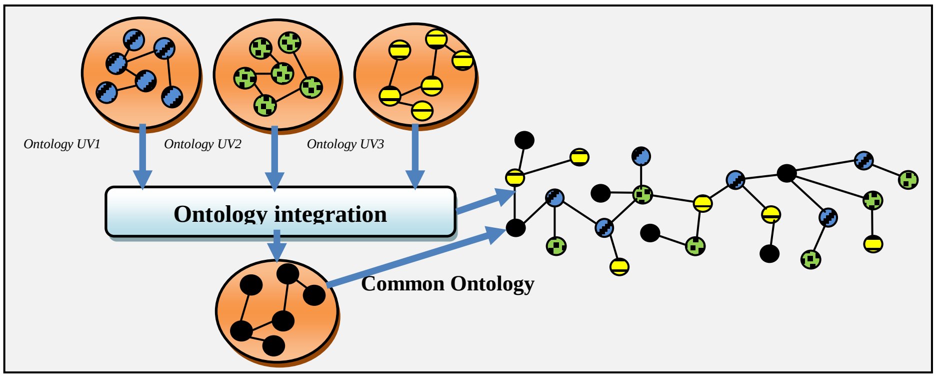

Figure 5. The importing process between UV's and CO

4.5 Integration between Classes

In our work, we have created three ontologies, UV1, UV2 and UV3, consisting of classes such as:

UV1= {Area, Assets, Contraceptive, Education, FoodConsume, GovernmentAid, HealthCenter, HealthProblem, HouseCondition, Job, Person}

UV2= {Assets, BirthControlMethod, EducationLevel, Food, GeographisArea, GovHelp, HealthCondition, Hospital, HouseParameter, JobArea, Person}

UV3= {Education, Family, FoodConsuming, G_Area, Health, HousingParameter}

As already mentioned, CO consists of terms that can be recognize and are used by a great number of people. In this case we selected the terms based on the frequency of the use of those terms by Google and Swoogle[1] search engines (see Table 1). The results provided by Google and Swoogle are different mainly due to the number of documents that are available in each of the systems. Google provide more documents than Swoogle. Currently, Swoogle only indexes some metadata elements about Semantic Web documents[2]. In table 1 we show the terms widely and commonly used by the authors of the documents available on the Web.

*Data was taken on 20 March 2012

Search String	Number of result*		Search String	Number of result*	
	Google (in millions)	Swoogle		Google (in millions)	Swoogle
'**Area**'	6.060	1.747	'Health_Condition'	396	170
'Geographic_Area'	52	366	'Health_Problem'	2.000	89
'G_Area'	2	0	'**People**'	12.750	1.820
'Location'	7.280	20.375	'Person'	3.440	16.318
'Contraceptive'	17	35	'Family'	6.590	2.208
'Birth_Control_Method'	41	0	'**Property**'	3.050	169.425
'**Birth_Control**'	637	3	'Asset'	359	212
'Contraceptive Methods'	0,9	0	'**Work**'	6.910	1.701
'**Education**'	3.080	1680	'Job'	3.910	387
'Education_Level'	784	11	'Job Area'	2.860	0
'Education_Background'	259	0	'Health_Center'	1.890	11
'**Food**'	3.730	3.164	'**Hospital**'	1.190	287
'Food_Consume'	93	4	'HouseParameter'	27	0
'Meal'	347	222	'HouseCondition'	188	0
'Food_Consuming'	93	0	'**House**'	5.520	425
'**Health**'	4.590	690			

Table 1. Search results for some terms about poverty using Google and Swoogle search engines

[1] http://swoogle.umbc.edu/
Swoogle was the first search engine dedicated to online semantic data. Its development was partially supported by DARPA and NFS (National Science Foundation).
[2] http://swoogle.umbc.edu/index.php?option=com_swoogle_manual&manual=faq

Using the referred criterion, CO consists of classes such as: Area, BirthControl, Education, Food, Health, Hospital, House, People, Property,Work}. We can conclude that Area (UV1) ≅ GeographicArea (UV2) ≅ G_Area (UV3) to Area (CO).

4.6 Integration between Properties

In this work we created a relation/link between individuals *(Object properties)* and link individuals to data values *(Datatype properties).*

Example:
There is a *class Person* and a *datatype property hasAccessToInformation. Class Person* includes *an individual x* and *Datatype Property hasAccesToInformation* have a value *Newspaper.*
If x is a Person that has access to a newspaper, we can say the same thing using a different syntax:

hasAccessToInformation(x,Newspaper).

Other examples relating the same individual *x* to other d*atatype properties* are:

hasSalary(x, 5000)
hasFinancialaAbilityToGoToDoctor(x, yes)
hasUseContraceptive(x, implant)
hasTotalfamilyMember(x, 4)
hasMarriageStatus(x, Married)
hasFamilyID ≅ ID(x, 1236620)
has Age (x, 45)
hasConditionOfFloor(x, good)
hasMinimum2mealsADay(x, No)

4.7 Integration between Individuals

Example:

- *Individuals belonging to the class Person of ontology UV1= {Arif Ndaru Winarto, Anton Haryono, Ashadi Suwarno, Kasinem, Isdiyon, Amat Sahari, Budi Raharjo, Eko Handoko}*
- *Individuals in the class Person belonging to ontology UV2={Amat Sahari, Budi Raharjo, Herlina Jayadianti, Sugeng, Sri Hartati, Wahyuni, Budiarti, Lalawedo, Hartono, Bambang}*

Let's consider that A and B are two private or governmental agencies. Agency A is using ontology UV1 and agency B is using ontology UV2. Each agency conducts a survey of poverty but the data obtained is different because each agency uses different criteria in looking at poverty. Using its selection criteria, the agency A selects *n* number of candidates and agency B selects m candidates but they selected only two common candidates (See Figure 6 and Figure 7).

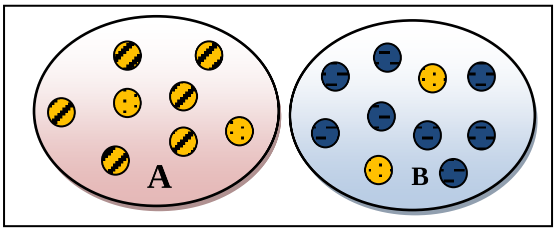

Figure 6. Venn Diagram I

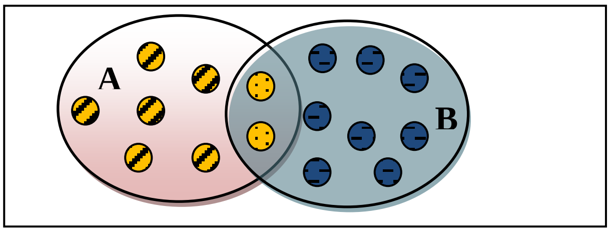

Figure 7. Venn Diagram II

Ideally the set of candidates selected by A and B should be the same! This fits the problem discussed before, that A and B are different agencies working on the same domain, but using different criteria to classify poor people. Our aims are to proove that the usage of a commun ontology will erase or significantly reduces the differences of interpretation described above.

5 USED TECHNOLOGY

5.1 Ontology Web Language (OWL)

Ontology web Language (OWL) is a language for create ontologies for the web. OWL was designed for processing information and was designed to provide a common way to process the content of web information. But, there are many different kinds of languages proposed as ontology languages. These languages have ranged from very powerful languages in which just about anything can be said, such as higher-order logics, through less expressive languages in which only certain kinds of things can be said, such as Description Logics, down to very simple languages, such as simple generalization taxonomies. OWL can be used to build most kinds of ontologies, but it is not as expressive as higher-order or even first-order logic, and thus certain kinds of ontologies cannot be built in OWL. in particular, OWL is ill-suited to create and reason with an ontology for OWL itself (Patel-Schneider, 2004).

5.2 Description Logic (DL)

DL is a universal query language that allows queries on ontologies using logics. Protege 4[3] is an OWL ontology development environment[4] that integrates a plug-in implementing DL query language. The query language supported by that plug-in is based on the Manchester OWL syntax, a user-friendly syntax for OWL DL that is fundamentally based on collecting all information about a particular class, property, or individual into a single construct.

Example 1

"Head of family with no income and also without financial ability to buy meat or to eat, even with a working member in the family"

Query 1 in UV1
```
Person and hasSalary value "NO"^^string and  hasMinimumOnePeopleWorkinFamily value
"YES"^^string and MinimumEatMeatOnceinWeek value "NO"^^string
```
UV1 Query Answer
```
Arif Ndaru Winarto, Anton Haryono, Ashadi Suwarno, Kasinem, Isdiyono
```

Query 1 UV2
```
HeadOfHouseHolder and  hasFrequentlyEaten value "VEGETABLE"^^string and and
hasSalary some int [<=0]
```
UV2 Query Answer
```
Amat Sahari, Anton Haryono, Ashadi Suwarno, Isdiyono, Arif Ndaru Winarto, Kasinem,
Budi Raharjo, Herlina Jayadianti
```

Query CO
```
(People or FamilyMember) and hasSalary "0"^^string RarelyEat “MEAT”^^string
```
CO Query Answer
```
Arif Ndaru Winarto, Anton Haryono, Ashadi Suwarno, Kasinem, Isdiyono, Amat Sahari,
Budi Raharjo, Herlina Jayadianti
```

Example 2

"Households that living in marginal or fragile environments and without access to clean water or sanitation"

---

[3]Protégé is a free, open source ontology editor and a knowledge acquisition system being developed at Stanford University in collaboration with the University of Manchester.Like Eclipse, Protégé is a framework for which various other projects suggest plugins. This application is written in Java and heavily uses Swing to create the rather complex user interface. http://protege.stanford.edu/

[4] http://protegewiki.stanford.edu/wiki/Protege4GettingStarted

Query 1 UV1

```
Person and and (hasConditionOfWall value "BAD"^^string or hasConditionOfFloor value
"BAD"^^string or hasConditionOfRoof value "BAD"^^string)and (TypeOfToiletUsed
“river”^^String or TypeOfFinalDisposalUsed “river”) and TakeWaterResourceFrom
“well”.
```

UV1 Query Answer

Arif Ndaru Winarto, Anton Haryono, Ashadi Suwarno, Kasinem, Isdiyono

Query 2 UV2

```
HeadOfFamily and (hasLargestFloorAreaMadeFrom value CEMENT or
hasLargestFloorAreaMadeFrom value SOIL) and (hasLargestRoofAreaMadeFrom value
ROOF_TILE or hasLargestRoofAreaMadeFrom value ASBESTOS) and(hasLargestWallMadeFrom
value WOOD or hasLargestWallMadeFrom value BAMBOO) and(hasUsedFinalDisposal value
SEPTICTANK or hasUsedFinalDisposal value RIVER) and(hasUsedTypeOfToilet value SOIL)
and(UsingWaterResourcesFrom value SPRING or UsingWaterResourcesFrom value
UNPROTECTED_WELL)
```

UV2 Query Answer

Amat Sahari, Anton Haryono, Ashadi Suwarno, Isdiyono, Arif Ndaru Winarto, Kasinem, Budi Raharjo, Herlina Jayadianti

Query CO

```
People and (UseWaterResourceFrom value SPRING or UseWaterResourceFrom value
UNPROTECTED_WELL) and (TypeOfFinalDisposalUsed value RIVER or
TypeOfFinalDisposalUsed value SOIL)
```

CO Query Answer

Arif Ndaru Winarto, Anton Haryono, Ashadi Suwarno, Kasinem, Isdiyono, Amat Sahari, Budi Raharjo, Herlina Jayadianti

DataProperty *TakeWaterResourceFrom* in UV1 and DataProperty *UsingWaterResourceFrom* in UV2 are equivalent with *UseWaterResourceFrom.* This relation also connects to the same individual of Class person “Isdiyono” and individual of class Water “Spring”.
TakeWaterResourceFrom (Isdiyono,Spring); *UsingWaterResourceFrom(Isdiyono, Spring); UseWaterResourceFrom(Isdiyono ,Spring)*.
People who use water resources from “spring” are considered to be poorer than the people who use water resource from “tap” and people who have a private toilet at home are certainly more capable than the people who do not have a toilet and have to go to the river to bath.

In OWL:

```
<!--http://www.semanticweb.org/ontologies/CO.owl#ISDIYONO-->
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/ontologies/CO.owl#ISDIYONO"> <rdf:type
rdf:resource="http://www.semanticweb.org/ontologies/CO.owl#HeadOfFamily"/>
<NumberOfChildren rdf:datatype="&xsd;integer">3</NumberOfChildren> <ID
rdf:datatype="&xsd;string">1202110030502087</ID>
<NumberOfShelter rdf:datatype="&xsd;integer">7</NumberOfShelter>
<Job rdf:datatype="&xsd;string">farmer</Job>
<FullName rdf:datatype="&xsd;string">ISDIYONO</FullName>
<Sex rdf:datatype="&xsd;string">MALE</Sex>
<MarriageStatus rdf:datatype="&xsd;string">MARRIAGE</MarriageStatus>
<TypeOfFinalDisposalUsed
rdf:resource="http://www.semanticweb.org/ontologies/CO.owl#RIVER"/>
<UseWaterResourceFrom
rdf:resource="http://www.semanticweb.org/ontologies/CO.owl#SPRING"/>
</owl:NamedIndividual>

< TypeOfFinalDisposalUsed
rdf:resource="http://www.semanticweb.org/ontologies/UV1.owl#RIVER"/>
< TakeWaterResourceFrom
rdf:resource="http://www.semanticweb.org/ontologies/UVI.owl#well"/>
</owl:NamedIndividual>
```

```
<hasUsedFinalDisposal
rdf:resource="http://www.semanticweb.org/ontologies/UV2.owl#RIVER"/>
<UsingWaterResourcesFrom
rdf:resource="http://www.semanticweb.org/ontologies/UV2.owl#SPRING"/>
</owl:NamedIndividual>
```

Testing was done using Query DL and Hermit 1.3.6 reasoner in Protégé 4.

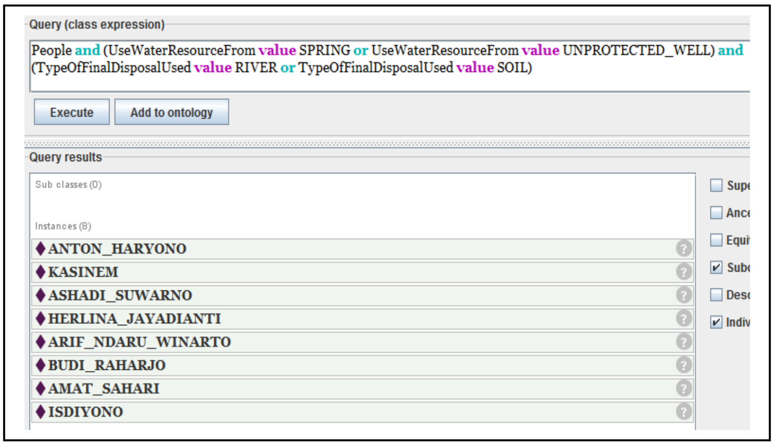

Figure 8. Query DL with Hermit 1.3.6 Reasoner

5.3 JENA

Jena[5] is a Java application programming interface that is available as an open-source[6]. Jena was developed to satisfy two goals (McBride, 2001):

- To provide an API that was easier for the programmer to use than alternative implementations;
- To be conformant to the RDF specifications.

5.4 SPARQL

SPARQL can be used to express queries across diverse data sources whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying and also supports extensible values for testing and constraining queries. The results of SPARQL queries can be either values or RDF graphs. Essentially, SPARQL is a graph-matching query language (Pérez, Arenas, & Gutierrez, 2006).

6 PROPOSED ARCHITECTURE

As a scenario, let’s consider several institutions, each of them using different lists of questions to make a survey about poverty. All of them use different sets of terms to create their conceptualizations about that domain and as we saw before, it is expectable to get different perceptions of that reality, named Perception1, Perception2 and PerceptionN. We can represent the knowledge of each perception using ontologies. Using Java Server Page (JSP) available in the user web browser the system can deal with query processing. The controller part of JSP applications and JENA SPARQL query will generate code that will be used to conduct the search on the knowledge base stored in the form of OWL/RDF files. OWL/RDF files are generated using the Protégé tools. Query results from the OWL/RDF files will be returned to the server and then displayed to the user's web browser.

[5] http://incubator.apache.org/jena/
[6] www.hpl.hp.com/semweb/jena-top.html.

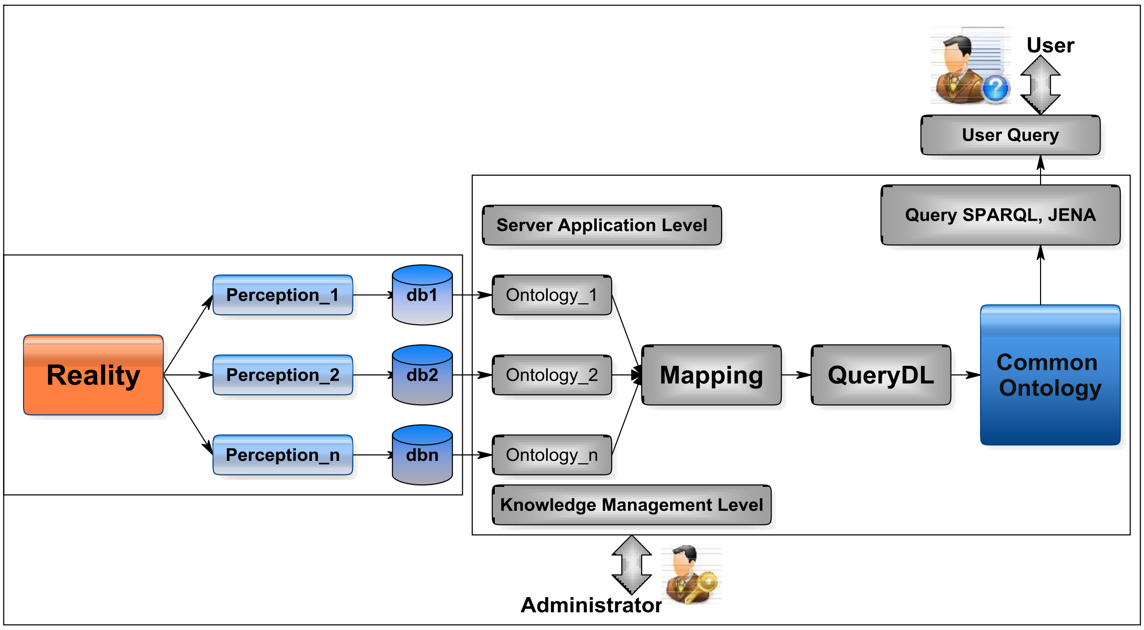

Figure 9. System Architecture

7 CONCLUSION

Different communities have different perceptions and use different sets of terms (terminologies) to represent the same reality. This leads us to a problem related to data heterogeneity. We can use ontologies to implement the several perceptions that can occur. Using a common set of terms based on the terms of the original ontologies we can construct a new ontology, a common ontology, that serves all the communities. To generate that common ontology we can use mapping and integrating mechanisms to map and integrate the original ontologies. In the research underling this paper we conclude that "mapping" is one of the best approaches to solve the problem of data heterogeneity.

Acknowledgment

We would like to acknowledge the support of the Erasmus Mundus EuroAsia program for the research foundation of this research, and also to acknowledge Universidade do Minho and Universitas Gadjah Mada for the collaboration.

References

Bouzeghoub, M. (2004). Semantics of a networked world: semantics for grid databases: First International IFIP Conference, ICSNW 2004, Paris, France, June 17-19, 2004: revised selected papers (Vol. 1). Springer.

Choi, N., Song, I. Y., & Han, H. (2006). A survey on ontology mapping. ACM Sigmod Record, 35(3), 34–41.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199–220.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. International journal of human computer studies, 43(5), 907–928.

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. The knowledge engineering review, 18(01), 1–31.

McBride, B. (2001). Jena: Implementing the rdf model and syntax specification. Semantic Web Workshop (Vol. 40).

Noy, N. F. (2004). Semantic integration: a survey of ontology-based approaches. ACM Sigmod Record, 33(4), 65–70.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protégé-2000: An Open-Source Ontology-Development and Knowledge-Acquisition Environment: AMIA 2003 Open Source Expo. AMIA Annual Symposium Proceedings (Vol. 2003, p. 953).

Patel-Schneider, P. F. (2004). What is OWL (and why should I care). Ninth International Conference on the Principles of Knowledge Representation and Reasoning. Whistler, Canada.

Pérez, J., Arenas, M., & Gutierrez, C. (2006). Semantics and Complexity of SPARQL. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, et al. (Eds.), The Semantic Web - ISWC 2006 (Vol. 4273, pp. 30–43). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://www.springerlink.com/content/p14356kpx6631057/

Xue, Y. (2010). Ontological View-driven Semantic Integration in Open Environments. The University of Western Ontario.