Data Integration Driven Ontology Design, Case Study Smart City

Data Integration Driven Ontology Design, Case Study Smart City

German Nemirovski, Andreas Nolle
Albstadt-Sigmaringen University of Applied Sciences
Albstadt, Germany
nemirovskij@hs-albsig.de
nolle@hs-albsig.de

Álvaro Sicilia
ARC Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Barcelona, Spain
asicilia@salle.url.edu

Ilaria Ballarini, Vincenzo Corado
Department of Energy (DENERG)
Politecnico di Torino
Torino, Italy
vincenzo.corrado@polito.it
illaria.ballarini@polito.it

ABSTRACT
Methods to design of formal ontologies have been in focus of research since the early nineties when their importance and conceivable practical application in engineering sciences had been understood. However, often significant customization of generic methodologies is required when they are applied in tangible scenarios. In this paper, we present a methodology for ontology design developed in the context of data integration. In this scenario, a targeting ontology is applied as a mediator for distinct schemas of individual data sources and, furthermore, as a reference schema for federated data queries. The methodology has been used and evaluated in a case study aiming at integration of buildings' energy and carbon emission related data. We claim that we have made the design process much more efficient and that there is a high potential to reuse the methodology.

Categories and Subject Descriptors
D.3.3 [**Information Storage and Retrieval**]: Systems and Software – *distributed systems*.

General Terms
Performance, Design, Standardization

Keywords
Ontology Design, Ontology Mapping, Description Logic, DL-Lite family, Data integration, Semantic Web.

1. INTRODUCTION
In the last decade, the paradigm of Semantic Web has gained lots of new ideas through approaches that focus on data integration and semantic interoperability. The cloud of Linked Opened Data has been growing rapidly and become one of the central components of Semantic Web. According to W3C, in 2011; it included over 31 billion RDF triples, stored in over 295 data sources[1]. The utmost advantage of federation of distributed data through interlinking using RDF triples is expected in areas where the heterogeneity of data builds a critical obstacle for its processing. This is when:

- large volumes of data have been stored in data sources supporting different data models,
- data describing characteristics of similar items has been generated using different standardization systems,
- measures characterizing equal physical quantities have been specified using different units of measurement, for example, following standards adopted in different countries.

The Smart City cluster clearly features all of these properties. Approaches like "sustainable low-carbon city" use statistic data for energy consumption and CO_2 emission of buildings that has been collected over many years in municipalities, energy and development companies, architecture offices and standardization organizations. The data stock is basically managed by relational database systems using wide diversity of data models. Taking this into concern, properties of ontologies specifying data semantics become crucial for the integration of this data into the Semantic Web environment.

In this paper we present a methodology for ontology design based on a series of document templates, tools and specifications. This methodology focuses on the requirements emerging in the context of data integration. Its application and effectiveness is shown in examples originated from the SEMANCO project[2] targeting the development of tools and data integration for the needs of the Smart City cluster.

A case study is highlighted in section 2 as an example of the variety of decisions that can be made in ontology design. Section 3 presents related work. Sections 4 to 8 illustrate details of the methodology. In section 9 we present the most important results and conclusions.

2. CASE STUDY WEATHER DATA
SEMANCO ontology has been developed as a mediator for integration of buildings' technical and statistical data, distributed in a set of heterogeneously structured data sources. Similar ideas of ontology driven data integration can be found in Calvanese [6] and Wang [26]. All data sources use relational schema. The

WIMS'13, June 12-14, 2013 Madrid, Spain

[1] http://lod-cloud.net/state/
[2] http://www.semanco-project.eu

ontology should help to interlink this data according to its semantics, facilitate federated querying for the entire data stock and enable semantic interoperability of tools to operate on these data. Thereby relation between the ontology and the integrated sources can be expressed in terms of [18]: a so-called global ontology is defined as a union of elements of local ones, representing the schemas of sources being integrated.

Let us illustrate the desired solution with an example. Given a data source 1 containing city names, weather station names and distances between cities (Table 1). The source 2 contains names of weather stations, temperature values measured at these stations and the dates when they were measured (Table 2).

Table 1. Data source 1

City	Weather station	Distance
Terrassa	Viladecavalls	7
Terrassa	Granollers	25
Manresa	Pont de Vilomara	4
Manresa	Torre d'en Roca	16
Manresa	Ajuntament de Navarcles	15

Table 2. Data source 2

Weather station	Temperature	Date
Viladecavalls	32,3	08.08.12
Granollers	34.5	08.08.12
Pont de Vilomara	38.2	08.08.12
Torre d'en Roca	37.0	08.08.12
Ajuntament de Navarcles	33.2	08.08.12

Let us suppose that a user requests the temperature values for cities measured at particular dates. The expected result of this query will be the following:

32.3 Terrassa 08.08.12 (1)

38.2 Manresa 08.08.12

To generate these results we have to know which temperature measures are related to particular cities. This information is not contained in the data directly. Nevertheless, a human agent after a short consideration of the data will be able to conclude that weather stations that are close enough to particular cities (For example, less than 10 km apart) can deliver the temperature values of these cities. This simple semantic implication; logical for humans; needs to be specified for the purposes of automated data retrieval, explicitly.

One option is to code the semantics in a query. A SPARQL query returning these results can look like this:

```
SELECT ?temp ?city ?date                                    (2)
WHERE {
      _:ws hasTemperatureMeasure ?tm.
      _:ws relatedTo ?city.
      _:ws distancedBy ?dist
      ?tm hasValue ?temp
      ?tm hasDate ?date
      FILTER (?dist < 10)
}
```

After this query is analyzed by a federated query processor, its parts are sent to particular sources.Afterwards, the results of subqueries are aggregated as shown in [9]. Yet, the same results could be targeted by a muchsimpler query:

```
SELECT ?temp ?city ?date                                    (3)
WHERE {
      ?city hasTemperatureMeasure _:tm
      _:tm hasValue ?temp.
      _:tm hasDate  ?date
}
```

However, in this case, if the semantic described above is missing in the query, we have to specify it somewhere else, e.g. in a TBox. The role inclusion in line seven of the code below contains one part of the information missing in the query. Namely, it connects theconcepts City and TemperatureMeasure (the connection is missing in the data sources).

```
∃hasTemperatureMeasure ⊑ City                               (4)
∃hasTemperatureMeasure ¯⊑TemperatureMeasure
∃closestTo ⊑ City
∃closestTo ¯ ⊑ WeatherStation
∃measuredTemperature ⊑ WeatherStation
∃measuredTemperature ¯ ⊑ TemperatureMeasure
closestTo ◦ measuredTemperature ⊑ hasTemperatureMeasure
∃hasDate ⊑ TemperatureMeasure
Range(hasDate) ≡ rdf:date
∃hasValue ⊑ TemperatureMeasure
Range(hasValue ¯) ≡  rdf:decimal
```

If the query and the TBox are specified as shown above, another part of the semantic is still missing: neither TBox nor the Query specify the rule for identification of the closest weather station to a city. Such a rule can be specified in a mapping of the corresponding data source, for example:

```
?ws closestTo ?city →
SELECT weatherStation from DS1 ds1_a WHERE
city='Manresa' and distance=(select
min(distance) from DS1 ds1_b where
ds1_b.distance < 10 and
ds1_b.city=ds1_a.city);
```

Such mappings are supported by tools for publishing of relational databases into a Semantic Web context. These tools rewrite SPARQL queries into SQL format and transform the query results to RDF triples. One of the most popular tools of this sort is D2R Server [3] another perspective mapping tool is Quest [22]. The mapping shown above couldlook in the D2R syntax as follows:

Data source 1: (5)

```
map:ds1_city  a d2rq:ClassMap;
      d2rq:dataStorage map:database;
      d2rq:uriPattern"city/@@ds1.city@@";
      d2rq:class :City.

map:ds1_wheatherstation a d2rq:ClassMap;
      d2rq:dataStorage map:database;
      d2rq:uriPattern  "station/@@ds1.weatherstation@@";
      d2rq:class :WeatherStation.

map:ds1_cityhasweatherstation a d2rq:PropertyBridge;
      d2rq:belongsToClassMap map:ds1_wheatherstation;
      d2rq:property :closestTo;
      d2rq:uriPattern "city/@@ds1. city@@";
      d2rq:condition "ds1.distance < 10".
```

Data source 2: (6)

```
map:ds2_weatherstation a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "station/@@ds2.weatherstation@@";
        d2rq:class :WeatherStation.

map:ds2_temperature a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "tempmeasure/@@ds2.temperature@@";
        d2rq:class :TemperatureMeasure.

map:ds2_temperaturevalue a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:ds2_temperature;
        d2rq:property  :hasValue;
        d2rq:column "ds2.temperature".

map:ds2_temperaturedate a  d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:ds2_temperature;
        d2rq:property  :hasDate;
        d2rq:column "ds2.date".

map:ds2_weatherstationtemperature a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:ds2_weatherstation;
        d2rq:property :measuredTemperature;
        d2rq:uriPattern "tempmeasure/@@ds2.temperature@@".
```

Question is: Which one of these two alternatives is better? Is there a third one? The choice between alternative designs is not only a question of designers' taste. It may have consequences for business processes, for example it could influence their performance or completeness and soundness of the query results.

In the following sections we will present instruments that determine ontology design decisions at different stages of a project, targeting data integration and semantic interoperability of tools.

3. RELATED WORK

The design of formally specified ontologies has been object of research since the early 1990s. Important work in this context was published by Gruber [14] and Uschold and King [25]. The former work is one of the most quotedin the field of semantic web. Its author defines the properties of ontological knowledge representation with relation to the requirements of engineering sciences. The approach of Uschold and King addresses the design process of ontologies, specifying four phases: identifying ontology purposes, building the ontology, evaluating and documenting. The ontology building phase is subdivided into three steps: 1) ontology capture, 2) ontology coding and 3) integration of existing ontologies. This approach has been further elaborated, for instance in Fernandes [12]. A survey of up-to-date methodologies for ontological design can be found in [10]. It became evident that the methodology per se is not enough. It should be supported by design patterns, document templates, tools or platforms, guiding developers along the methodology steps and making complex design tasks, easier. This requirement led to the development of ontology tools such as Protégé [17], WebODE [1] and OntoEdit [24]. Recent comparative studies of such tools are provided in Khondoker [16] and in Kapoor and Sharma [15]. Fonou-Dombeu and Magda Huisman [13] provide an interesting case study for ontology design.

Furthermore an important aspect of the design methodology is the selection of the formalism, a set of rules and constructors for the ontology specification. The right selection of the formalism usually determines the compromise between the expressive power of the ontology and the processing efficiency of the knowledge represented by the ontology. For example, the Description Logic that is mostly used as the formal basis for the ontology specification comprises a family of formal languages. Some of them like $\mathcal{SOIN}$(D), $\mathcal{SROIQ}$(D), or DL-LiteR have been used as basis for different OWL dialects, i.e. OWL DL, OWL 2 and OWL QL respectively[3].

Further approaches related to particular aspects of the proposed methodology are referred to in following sections.

4. METHODOLOGY

From the example provided in section 2 we have learned that the semantics of data can be expressed as a union of elements (concepts, roles and axioms) expressed by an ontology TBox and data source mappings. Furthermore, in [21] this issue is discussed more formally. It is shown that TBox and mappings generally supplement each other. However, they may have unnecessary overlaps. Moreover, as shown above, TBox and mappings specifications should be designed with respect to the required queries. Vice versa, as shown in [7], ontology design determines the efficiency of conjunctive queries, as in the case of queries aiming at retrieval of data properties of individuals (instances of ontology concepts).

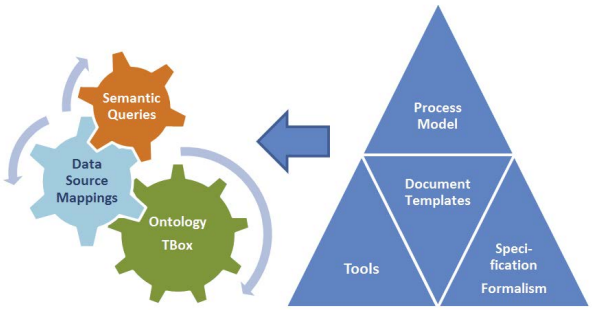

Figure 1. Dependencies between data integration items and parts of the methodology for ontology design

Cross dependencies between queries, TBox definition and mappings increase complexity of the ontology design process.. Such dependencies can be easily overlooked by designers. This can lead to severe consequences while a query is processed, like incompleteness of query results or problems with its answer time. The proposed methodology addresses this issue by taking these dependencies into account. As shown in Figure 1 it combines four components: i) an integrated process model for ontology design and data integration ii) a set of document templates supporting designers in every phase of the design/integration process, iii) a set of tools for implementation of TBox and data source mappings exploiting iv) a specification formalism adapted for requirements on data integration.

We argue that the proposed methodology helps to make complex design decisions, for example to decide where to specify parts of query semantics, as described in selection 2.

[3] http://www.w3.org/TR/owl2-overview

5. PROCESS MODEL

The ontology design process is divided into three phases: i) vocabulary building; comprising use cases specification, building of an initial vocabulary and informal mapping of data sources' vocabularies, ii) implementation; that implicates TBox coding and integration of data sources with the help of of formal mappings, and iv) evaluation implying the usage of informal specification of final vocabulary and of use cases generated at the beginning of the process. In doing soeach of the following phases takes as input the specifications developed in the previous one (Figure 2).

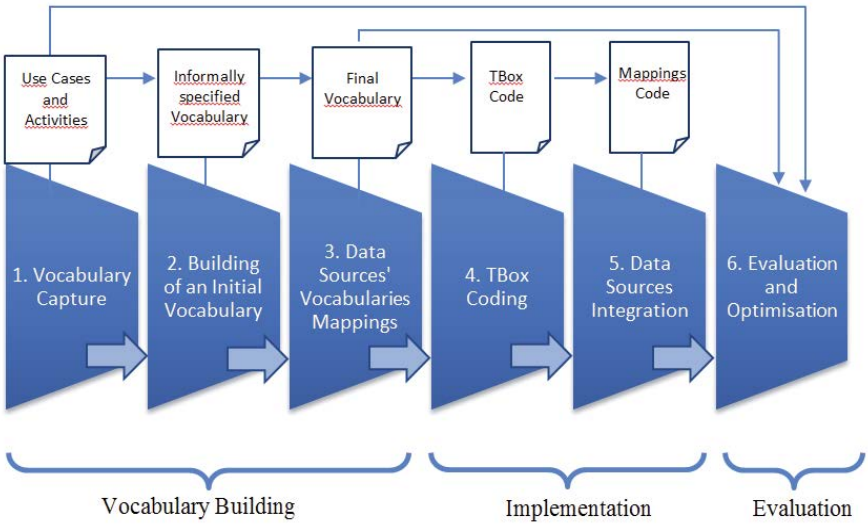

Figure 2. Process model

Subdivision of the design process into phases was initially proposed by Uschold and King [19]. Authors defined three phases 1) ontology capture: for instance definition, naming and description of ontology concepts, roles and relations between them; 2) ontology coding: for example specifying the classes and roles using one of the formal languages, for instance OWL; 3) integration of existing ontologies into business processes and tools. This approach has been elaborated thoroughly, in further research work adding some new details like iterations [18] or new phases like scoping, evaluation and documentation [15].

. The most important difference between the proposed model and the aforementioned approach is its specialization on data integration. This issue is explicitly addressed by steps 3 and 5. Coming back to the example from section 2, the proposed methodology used already in step 3 would help to identify the conflict between the information required by the user (the temperatures ofcities) and the information available in the data sources (temperatures are not associated with cities but with the weather stations that have measured them). Furthermore, in step 5, the design that solves this conflict would be developed. Bringing the query, the TBox and the data source mapping in correspondence with each other is an example of a design that resolves this conflict.

As this will be shown; in the vocabulary building phase the design decisions are supported by document templates and in the implementation phase by a formalism designed to fulfill requirements of data integration, as well as by tools for ontology design and data source mapping.

6. VOCABULARY BUILDING

The vocabulary building phase is divided into three steps which increasingly capture knowledge from the context where the ontology is going to be used and the data sources to be integrated.

6.1 Vocabulary Capture

As mentioned above, we consider query design as an important part of the ontology design. Furthermore, queries are formulated by users or by tools controlled by users. Hence, for understanding the nature of potential queries it is important to take into consideration the users' perspective.

In the proposed methodology, this task is solved by the use case specifications generated at the beginning of ontology design process. Each use case specification contains a set of activities interconnected by flow lines, determining their sequences. An activity can occur in multiple use cases, so that a network of activities emerges, as shown below. Such specifications help to understand the users' requirements, the needs for data, its semantics, the vocabulary and the desired level of values aggregation. Starting the ontology design process with the specification of the users' perspective is not new: [12] describe an approach of goal modeling which is close to the one presented in this paper. Yet the goal modeling serves to prepare the so called "competency questions", also referred to in [22]. However as long as integration of data sources and information retrieval is focused on, the use cases and activity specifications provide an ideal basis for the formulation of semantic queries. As shown in table 3, an activity description contains a field for specification of all data related to this activity. On the contrary, "competency questions" only appear to be a good instrument for concepts capturing and less appropriate for query design.

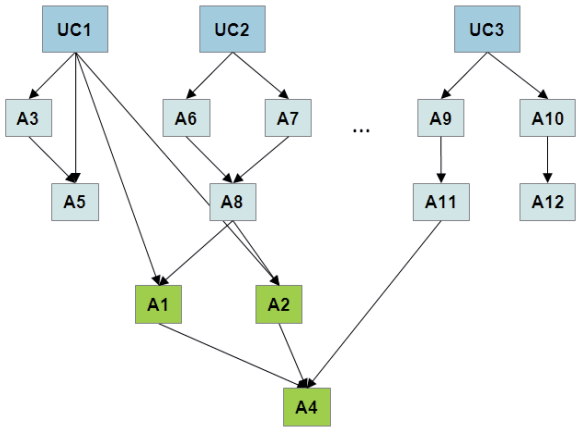

Figure 3. Relationships between activities and use cases

Table 3. An activity description (short version) generated using activity design template which is a part of the proposed methodology

Acronym	A9
Goal	Determination of characteristics of urban environment
Urban Scale	Messo –Macro (urban area)
Process scale	Operational
Actors	• The municipality (councillors of urban planning, housing, environment and countryside, …) (stakeholder) • Urban Planners, from public authorities or from private companies • Public company of social housing • Owner/promoter of the building • Neighbours association (stakeholder)
Related national/local policy framework	• National energy code and national technical building construction code (CTE, and RITE) • Nation , regional and local urban planning regulations
Issues to be addressed	▪ Volumetric information of the buildings conforming the urban area (to obtain profile of shadows) ▪ Geography of the Area ▪ Location and volume of other urban elements o Climatic information (Horizontal radiation, wind speed, relative humidity, external temperature)
Input Data	

Name	**Description**	**Domain**	**Format**
Vector Maps from Manresa GIS	Polygon map showing 3D geometry (buildings footprint, perimeter and height) of the buildings of the urban area	Geography, Manresa GIS	Rdf
GIS maps with topographic information	Topographic information of the urban area and surroundings	Geography, Manresa GIS	Rdf
Horizontal radiation	Amount of $W \cdot h/m^2$	Climatic	
Wind speed	Speed of the wind in m/s at the nearest weather station	Climatic	
Relative humidity	Relative humidity at the nearest weather station	Climatic	
Air temperature	Outside Temperature at the nearest weather station	Climatic	

Possible semantic queries related to "Air temperature", referred to the last data entry in the table above, are shown in section 2 of this document. However, no information about the available data is accessible, in this step. For this reason, it is still not clear how the term of "nearest weather station" can be interpreted. Therefore, the query design probably would look similar to (3) at this stage. The query (2) can be formulated only after the available data would have been analyzed.

6.2 Building of an Initial Vocabulary

The second step of the building vocabulary phase is focused on the constitution of an initial vocabulary (Figure 2). The names of the data items in the activity specifications are integrated into the vocabulary from standardization systems, taxonomies of terms or data models, well known in the Smart City context. The correct terminology, the definitions of data names and the relationships among concepts are based on technical standards (For instance, EN ISO 13786[4], EN 15193[5], EN 15251[6] and NREL/TP-550-38600[7]) and on scientific literature. These references also provide the symbols and the units of the defined quantities, if applicable. The emerging initial vocabulary includes terms and the relations between them. The initial vocabulary is specified in the form of an excel table using the corresponding template. One extraction of such vocabulary is shown in table 4. In this table the name of a relation connecting two terms is written left to these terms. The tree structure of the table determines the other term that is connected by the relation, e.g. *Air Temperature* is a *Climatic_Parameter*.

On the one hand, the table shown in table 4 is an important intermediate step towards TBox design. It effectively prepares TBox coding using a formal specification language such as (4), shown in section 2. On the other hand, the completeness of the vocabulary within the use cases originated from the smart city context is guaranteed by the involvement of data specified in the activity description, as the one shown in table 4. For example, the term Air Temperature is part of the vocabulary specification twice, once as a climatic parameter and once as a value measured by a weather station. The resulting vocabulary is subdivided in categories, such as building use, climate, and building geometry. Each of these categories contains numerous data names identified in diverse activity descriptions.

[4] Thermal performance of building components. Dynamic thermal characteristics. Calculation methods.

[5] Energy performance of buildings.

[6] Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics.

[7] Standard Definitions of Building Geometry for Energy Evaluation.

Table 4. An activity description (short version) generated using activity design template

Name/Acronym				*Description*	*Reference*	*Type of data*	*Unit*
Climate				climatic data	-	-	-
has	**Climatic_Parameter**			climatic parameter	-	-	-
	is **Air_Temperature**			**the temperature of external air**	**EN ISO 15927-1**	**real**	**°C**
	Is **Solar_Irradiance**			radiation power per area generated by the reception of solar radiation on a plane	EN ISO 15927-1*	real	W/m^2
		has **Solar_Irradiance_Type**		type of solar irradiance	-	string	-
			Is **Direct_Solar_Irradiance**	irradiance generated by the reception of solar radiation on a plane from a conical angle which surrounds concentrically the apparent solar disk	EN ISO 15927-1*	string	-
			Is **Diffuse_Solar_Irradiance**	irradiance generated by the reception of scattered solar radiation from the full sky hemisphere on a plane, with the exception of that solid angle which is used to measure the direct solar irradiance	EN ISO 15927-1*	string	-
			Is **Global_Solar_Irradiance**	irradiance generated by reception of solar radiation from the full hemisphere on a plane	EN ISO 15927-1*	String	-
...							
Stationary_Artefact					-	-	-
is	**Weather_Station**				-	-	-
	measuredTemperature	**Air_Temperature**	**the temperature of external air**	**EN ISO 15927-1**	**Climate**	**°C**	

6.3 Data Sources' Vocabularies Mappings

In the last step of the vocabulary building phase, Data Sources' Vocabularies Mappings (Figure 2), the names of the data items, used in sources to be integrated, are mapped on the initial vocabulary; as shown in table 4. In the case of relational databases, the fields of a table will be mapped to the terms of the vocabulary. This is done by mapping tables as the one shown in table 5.

Table 5. An activity description (short version)

Data source	Data name (in the Data source)	Data name (in the vocabulary)	Data category (in the vocabulary)
Cataluña Building Data BuildingParametersNONDomestic	average set pint temperature	Air_Temperature	Building
Cataluña Building Data BuildingParametersNONDomestic	USE	Building_Use	Building
Cataluña Building Data BuildingParametersNONDomestic	DATE	Year_Of_Construction	Building
Cataluña Building Data BuildingParametersNONDomestic	Orientation main façade	Main_Orientation	Building
Cataluña Building Data BuildingParametersNONDomestic	Orientation main façade: East	**MISSING**	Building
Cataluña Building Data BuildingParametersNONDomestic	Orientation main façade: West	**MISSING**	Building

In the data source analyzed in this table, the vocabulary term *Air Temperature* was identified under the name of *average set point temperature*. The corresponding table elementserves as an instruction for the following coding of the mapping files. However, not all of the data fields, in the considered document, could be mapped unambiguously (see missing correspondences in table 5). Now, designers are facing three alternative options: to change the initial vocabulary; to implement non-trivial mappings like (5) or to specify complex queries like (2).

7. IMPLEMENTATION

7.1 TBox Coding

The proposed methodology is exploiting the *DL-Lite*$_A$ formalism for the ontology coding and design. The main reason for the use of *DL-Lite*$_A$ was its special features designed w.r.t the requirements of data integration [20]. Furthermore *DL-Lite*$_A$ serves as a basis for the OWL QL profile of OWL 2, designed for the purpose of data accessing/management[8].

As stated in [18], the most important features of *DL-Lite*$_A$ are the following: 1) domain and range of properties can be specified only for functional data properties; and 2) definition of an object property connecting two OWL classes with each other, has to be modelled by means of axioms and not by specifying the property's domain and range. For example, two following axioms in DL notation use subsumption ($\sqsubseteq$), existence quantification ($\exists$) and inversion ($^-$) to express that the class BuildingGeometry relates to the class Building via the *hasGeometry* property.

$$\exists \text{hasExternal\_Temperature} \sqsubseteq \text{Building}$$

$$\exists \text{hasExternal\_Temperature}^- \sqsubseteq \text{External\_Temperature}$$

Although domains and ranges of properties are not explicitly specified in the code, if an ontology specification is valid, they can be inferred by reasoner software to be visualized by the user. In this context, using conventional ontology editors like Protégé is time consuming and prone to errors, if used for coding of numerous axioms.

[8] http://www.w3.org/TR/owl2-profiles/

The ontology editor developed in the SEMANCO project provides an instrument to generate a set of axioms defining a relation between two concepts only by a mouse click in the context menu. Besides that, the ontology editor facilitates on the fly inferring of properties' domains and ranges, and enables simultaneous representation of subsumptions' taxonomy with the properties graph (Figure 4). These three features make this editor (to our knowledge) a unique tool for editing *DL-Lite*$_A$ ontologies.

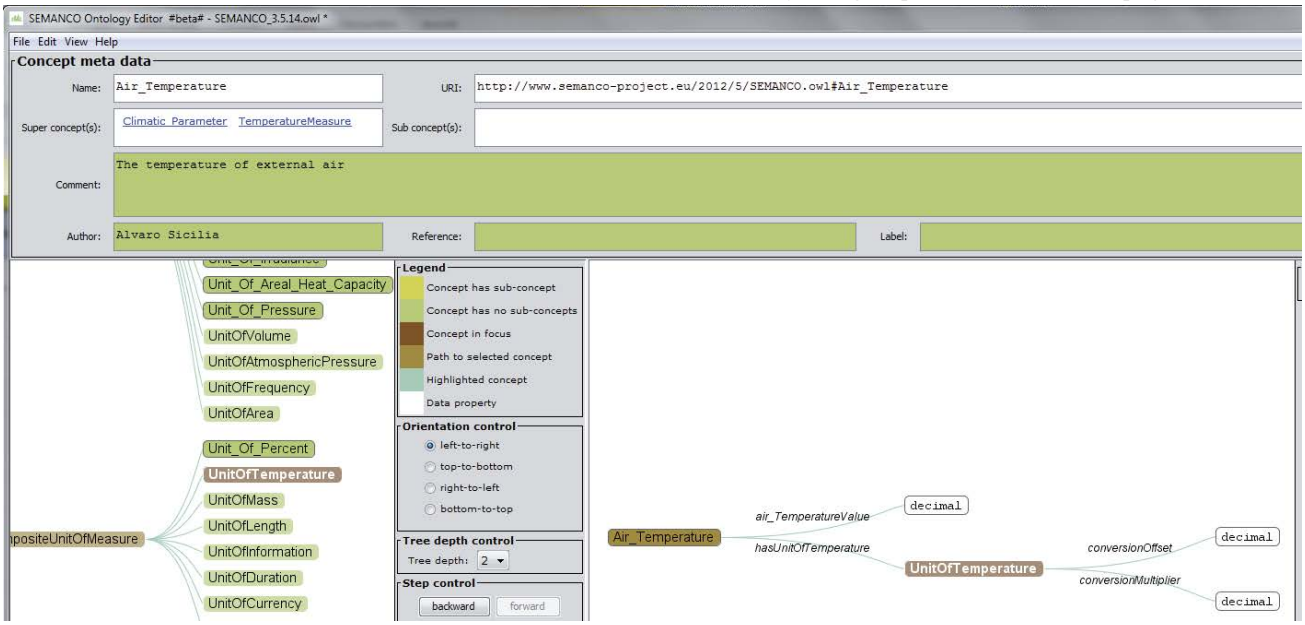

Figure 4. Ontology Editor presents the ontology graph using on the fly inferring

It is important to notice that the selection of a specific formalism like *DL-Lite*$_A$ immediately determines (restricts and simplifies) the TBox design. Returning to the questions formulated in section 2, when *DL-Lite*$_A$ is used, line seven in (4) cannot be specified as follows. The constructor for roles chaining is not a part of this DL language.

$$\text{closestTo} \circ \text{measuredTemperature} \sqsubseteq \text{hasTemperatureMeasure}$$

Consequentially, corresponding semantics should be specified somewhere else outside of TBox, e.g. in the query or in the data source mapping. The desirable effect can be achieved by replacing the query (3) through the following:

```
SELECT ?temp ?city ?date                                    (7)
WHERE {
      ?city closestTo _:ws.
      _:ws measuredTemperature _:tm.
      _:tm hasValue ?temp.
      _:tm hasDate  ?date
}
```

Hence the selection of *DL-Lite*$_A$ formalism determines not only specification of TBox but also the form of semantic queries and/or mappings. The last statement is not illustrated here due to lack of space.

7.2 Mapping Data Sources

This step uses the outputs generated by the previous two steps (Figure 2) to transform the contents of the data sources into RDF resources. To do so, the mappings established in the step Data sources' vocabularies mappings (step 3) are coded as relations between a relational database and the target ontology TBox created in step 4. These mappings are usually implemented with declarative mapping languages which offer rich expressive features to bring the rigid relational schemas to real cases. The prime example is the RDB to RDF Mapping Language (R2RML)[9] which became a W3C recommendation in September, 2012 and it is currently being implemented in several projects. However, other languages can be used for the same purpose, e.g.R_2O [2] and D2RQ [4].

Two environments were developed within the SEMANCO project to help the data sources mapping processes based on D2RQ language: a) the OWL mapping extractor with the purpose of extracting an OWL ontology file and a D2RQ mapping file, reading the structure of a relational database; b) the ontology mapping collaborative web environment that provides a graphical interface to assist non ontology experts to implement the mappings (Figure 5).

The extractor tool uses a configuration file –written in Turtle[10] syntax– to extract the structure of the database. The default tool's behavior is to map each table and column of the database as a class. This can be customized by removing statements or modifying the attributes of the configuration file. The outputs of the extractor tool are an OWL and a D2RQ mapping files like in cases (5) and (6).

8. EVALUATION

After a comparative analysis, we have adapted some ideas related to ontology evaluation described in Gómez-Pérez, [8], Obrst [19], Gangemi [5], and Nemirovskij [18]. In particular w.r.t data integration as the purpose of ontology design, the proposed methodology comprises evaluation of the following three

[9] http://www.w3.org/TR/r2rml/

[10] http://www.w3.org/TR/turtle/

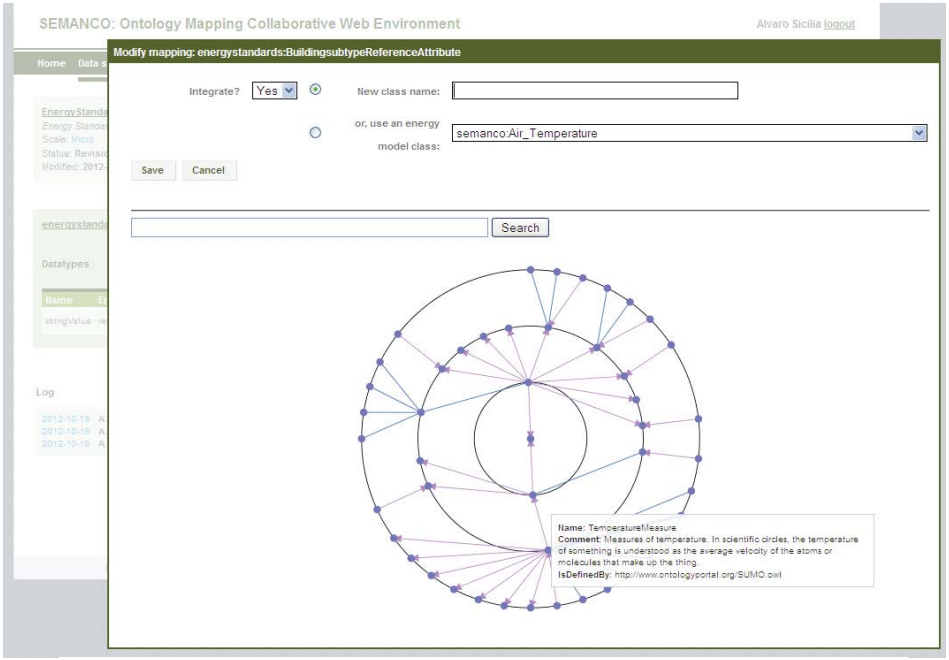

Figure 5. Ontology mapping web environment: ontology graphical representation

properties of the resulting ontology, corresponding to three data integration items (Figure 1):

- TBox Intelligibility: the ability of actors that use the ontology to understand the ontology structure.
- Mappings compliance: correspondence of mappings with the TBox
- Computational efficiency: the ability of the ontology to support conjunctive querying on high efficiency level, i.e., with a comparatively short response time.

TBox Intelligibility: especially as a consequence of frequent vocabulary mappings in step 3, there is a risk that the initial structure of the vocabulary designed in step 2 changes significantly and its semantics get unintentionally altered. For the purposes of intelligibility testing, independent testers are asked to find concepts by navigating along the TBox graph. The navigation is done using the editor described in section 7.1. The evaluation is carried out by two independent groups of users, for example of computer science students, and experts in the field of building energy. Each tester is offered a list of terms to find in the ontology. The average score of each group is measured, compared to the shortest navigation path. Our experiments have shown average scores of 97.30%, and 91.20% for each group correspondingly.

Mappings compliance: as stated in [21], a new TBox emerges as a result of a data source mapping. The goal of this evaluation strategy is to make such a TBox explicit and to compare it with the target specified in step 4. This is done by generating an OWL code, out of mapping files. The task is carried out by the mapping environment described in 7.2. As mentioned in section 2, TBoxes generated from mappings should be subset of the target. On the other hand such TBoxes haveto contain concepts and properties used in basic graph patterns of queries, e.g. lines 2, 3 and 4 of (3) or lines 2, 3, 4 and 5 of (7). If the query (3) is in use w.r.t. target TBox (4), at least one of the mapping TBoxes should contain the concepts *City, TemperatureMeasure*, *Date*, and properties *hasTemperatureMeasure*, *hasValue* and *hasDate.* Alternatively, these elements should be inferable w.r.t. entailment regimes [11], for instance, if a query contains a basic graph pattern "?city a City", it is sufficient if a mapping TBox contains a concept *Village* and the target TBox contains the subsumption *Village* $\sqsubseteq$ *City*. If this is not the case, the query results cannot be considered complete. Therefore mappings, TBox or queries should be altered.

Computational efficiency: in the focus of this method is the evaluation of query processing. All queries to be evaluated are designed w.r.t. use cases and activity description, specified in step 1 of the design process. Alternative design approaches can be compared toeach other, directly. The following table illustrates the method by comparing processing of the query (2) specified in section 2 and the query (7) shown in section 7.1 and using mappings (5) and (6). The query (2) uses slightly simpler mappings. Five measures have been made for each query. While there is no difference w.r.t. completeness (the right column); the second query constantly shows better time performance.

Table 6. Query performance evaluation

Query ID	Time (in minutes, seconds, and milliseconds,)	Records retrieved
(2)	1:33:45.384	16566
(2)	1:31:08.581	16566
(2)	1:32:23.737	16566
(2)	1:30:35.088	16566
(2)	1:31:36:434	16566
(3)	1:17:30.026	16566
(3)	1:17:17.816	16566
(3)	1:17:33.300	16566
(3)	1:17:46.940	16566
(3)	1:17:27.311	16566

An obvious explanation for this is that the mathematical comparison "datasource1.distance < 10" is specified in the mapping is carried out by native methods of a data source that perform better than ones specified in a SPARQL query "FILTER (?dist > 10)" and consequently, running on RDF data.

9. CONCLUSIONS

In this paper we have described a methodology for ontology design addressing the needs of approaches using ontologies for data integration. We have shown that in this case the design process apart from the ontology TBox has to target semantic queries and mapping of data source. The methodology includes four components: a process model, a set of document templates, a specification formalism *DL-Lite*$_A$ and a set of tools for the simplification of the coding.

To our knowledge the methodology is unique. There are a few approaches addressing ontology design, the most relevant ones are mentioned in this paper. However, none of the existing methodologies put the data integration into focus. Hence, these approaches basically target the development of a TBox and in some cases of an ABox, but do not address query and mapping design.

The efficiency of the approach as a whole, and of its components as well, has been proved by its application. The complete approach has been applied in the SEMANCO project. Within the first 18 months of project time, 592 TBox concepts and 468 relations in *DL-Lite*$_A$ style have been implemented with 3459 axioms, 244 corresponding mappings have been done and 25 queries have been tested.

Furthermore, the ontology editor and the mapping tool presented in this paper have been designed to address generic problems of data integration. During the last year, previous versions of ontology editor and of the mapping tool have been applied in other projects concentrated on data integration issues. This is the case of RÉPENER. It is estimated that around 71 TBox concepts, 100 relations using 858 axioms in *DL-Lite*$_A$ style have been developed, using these tools. Moreover, the high level of standardization and modularization of the code – the code has been developed using Jena[11] and CodeIgniter[12] frameworks - simplify the customization of tools and their reuse for alternative purposes.

Acknowledgement

The main contribution of this work has been developed under SEMANCO project, which is being carried out with the support of the Seventh Framework Programme "ICT for Energy Systems" 2011–2014, under the grant agreement no. 287534.

10. REFERENCES

[1] Arpírez, J.C., Corcho, O., Fernández-López, M. Gómez-Pérez, A. 2001. WebODE: a scalable workbench for ontological engineering. In *Proceedings K-CAP '01 Proceedings of the 1st international conference on Knowledge capture*. 6-13, ACM New York, USA.

[2] Barrasa, J., Corcho, O. and Gómez-Pérez, A. 2004. R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language. In *Proceedings of the Second International Workshop on Semantic Web and Databases (SWDB 2004)*. Springer, 1069–1070.

[3] Bizer, C. and Cyganiak, R. 2006. D2R Server - Publishing Relational Databases on the Semantic Web. In *Proceedings of 5th International Semantic Web Conference (ISWC'06)*.

[4] Bizer, C. and Cyganiak, R., 2007. D2RQ – Lessons learned. Position paper at *the W3C Workshop on RDF Access to* Relational Databases. Cambridge, USA.

[5] Gangemi, Catenacci, A.C., Ciaramita, M. and Lehmann, J. 2005. Ontology evaluation and validation: an integrated formal model for the quality diagnostic task. *Technical report*, Laboratory of Applied Ontologies-CNR, Italy.

[6] Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D. and Rosati, R. 1998, Description Logic Framework for Information Integration. In *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning* (KR-98). Italy.

[7] Calvanese, D, De Giacomo, G., Lembo, D., Lenzerini, M. and Rosati, R.. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 39(3), 385–429.

[8] Gómez-Pérez, A. 2004. Ontology evaluation, *Handbook on Ontologies,* Staab, S. and Studer, R. Eds., (1st ed.), Chapter 13, 251-274, Springer.

[9] Görlitz, O. and Staab, S. 2011. Federated Data Management and QueryOptimization for Linked Open Data, In *New Directions in Web Data Management,* A. Vakali & L.C. Jain Eds.: 1, SCI 331, 109–137.

[10] Contreras, J. and Martínez-Comeche, J. 2008. Ontologías: ontologías y recuperación de información.DOI= http://www.sedic.es/gt_normalizacion_tutorial_ontologias.pdf

[11] Glimm, B. 2011. Using SPARQL with RDFS and OWL Entailment regomes. In Reasoning Web 2011, volume 6848 of Lecture Notes in Computer Science, 137-201.

[12] Fernandes, B.C.B., Guizzardi, R.S.S. and Guizzardi, G. 2011. Using Goal Modeling to Capture Competency Questions in Ontology-based Systems. *Journal of Information and Data Management*, Vol 2, No 3, 527-540.

[13] Fonou-Dombeu, J.V. and Huisman, M. 2011. Combining Ontology Development Methodologies and Semantic Web Platforms for E-government Domain Ontology Development. *International Journal of Web & Semantic Technology* (IJWesT) Vol.2, No.2.

[14] Gruber, T. 1995. Towards principles for the design of ontologies used for knowledge sharing. In *International Journal of Human Computer Studies,* Vol. 43 (5-6), 907-928.

[15] Kapoor, B. and Sharma, S. 2010. A Comparative Study Ontology Building Tools for Semantic Web Applications. *International journal of Web & Semantic Technology (IJWesT)* Vol.1, Num.3.

[16] Khondoker, M.R., Mueller, P. 2010. Comparing Ontology Development Tools Based on an Online Survey. In *Proceedings of the World Congress on Engineering 2010 Vol I WCE 2010*, London, U.K.

[17] Knublauch, H., Fergerson, R.W., Noy, N. F. and Musen, M. A. 2004. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *Proceedings of the Third International Semantic Web Conference*, Lecture Notes in Computer Science, Hiroshima, Japan, November 7-11,. 229-243.

[18] Nemirovski, G., Sicilia, A., Galán, F., Massetti, M. and Madrazo, L. 2012. Ontological Representation of Knowledge Related to Building Energy-efficiency. In *Proceedings of the Sixth International Conference on Advacnes in Semantic Processing*, Barcelona.

[19] Obrst, L. Ceusters, W. Mani, I., Ray, S. and Smith, B. 2007. The evaluationof ontologies, In Revolutionizing Knowledge Discovery in the Life Sciences, C.J.O. Baker, and K.-H. Cheung, Eds., Chapter 7, 139- 158, Springer.

[20] Poggi, A. Lembo, D., Calvanese, D, De Giacomo, G. Lenzerini, M and Rosati, R. 2008. Linking data to ontologies. *Jounal on Data Semantics*, 133–173.

[21] Rodríguez-Muro, M, and Calvanese, D. 2012. High Performance Query Answering over DL-Lite Ontologies. In *Proceedings of 13th International Conference on Principles of Knowledge Representation and Reasoning*, Rom.

[22] Rodríguez-Muro, M, and Calvanese, D. 2012. Quest, a System for Ontology Based Data Access. *In OWLED 2012*.

[23] Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E. and Gangemi., A.. 2012. *Ontology Engineering in a Networked World.* Berlin: Springer.

[11] http://jena.apache.org/

[12] http://ellislab.com/codeigniter

[24] Sure, Y., Erdmann, M. Angele, J., Staab, S Studer, R. and Wenke, D. 2002. OntoEdit: Collaborative Ontology Development for the SemanticWeb, In *Proceedings of Fisrt International Semantic Web Conference (ISWC 2002)*. Horrocks and Hendler Eds. Vol. 2342 of LNCS, 221–235, Springer-Verlag Berlin.

[25] Uschold, M. and King, M. 1995. Towards methodology for building ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, Canada.

[26] Wang, J., Lu, J., Zhang, Y., Miao, Z. and Zhou, B. 2009. Integrating Heterogeneous Data Source Using Ontology. *JOURNAL OF SOFTWARE*, VOL. 4, NO. 8.