SEMANCO Semantic Technologies for Carbon Reduction in Urban Planning

SEMANCO

Deliverable 4.1
Environments for collaborative ontology mapping

Revision:	**9**
Due date:	**2012-05-01 (m8)**
Submission date:	**2012-05-10**
Lead contractor:	**FUNITEC**

Deliverable Administration & Summary

No & name	**D4.1 Environments for collaborative ontology mapping**

Status	Completed	**Due**	m8	**Date**	2012-05-01

Author(s)	Alvaro Sicilia, Joan Pleguezuelos (FUNITEC), German Nemirovskij, Andreas Nolle (HAS)
Editor	Leandro Madrazo (FUNITEC)
DoW	Design of an environment which enables multiple users to collaborate in the definition and maintenance of ontology mapping. Applying standard ontology languages to manage the ontologies. Developing functionalities to upload, retrieve, label and annotate ontologies and their instances. Applying ontology mapping techniques to align ontologies. Implementation of the environment in test cases.
Comments	This report summarizes the work done from month 3 to 8 of the project on the development of collaborative ontology tools. This development has been led by the research group from FUNITEC, with the collaboration of HAS. Alvaro Sicilia (FUNITEC), German Nemirovskij (HAS) and Andreas Nolle (HAS) have worked on the design of the environment. The implementation has been the work of Alvaro Sicilia and Joan Pleguezuelos from FUNITEC.

Document history

V	Date	Author	Description
1	2012-04-25	Alvaro Sicilia (FUNITEC)	First version of the report of the development of the prototype.
2	2012-05-01	German Nemirovskij (HAS)	Review of Sections 2 and 3; addition of Section 3.2
3	2012-05-02	Leandro Madrazo (FUNITEC)	Reviewing and editing of whole document
4	2012-05-02	Alvaro Sicilia (FUNITEC)	Fourth version of the report
5	2012-05-02	German Nemirovskij (HAS)	Review of Section 3
6	2012-05-04	Leandro Madrazo (FUNITEC)	Version submitted to reviewers
7	2012-05-08	Alvaro Sicilia, Leandro Madrazo (FUNITEC)	Inclusion of changes proposed by reviewers
8	2012-05-08	Chris Ennis (UoT)	Editing and proof reading
9	2012-05-09	Leandro Madrazo (FUNITEC)	Final editing and proof reading

Disclaimer

The information in this document is as provided and no guarantee or warranty is given that the information is fit for any particular purpose.

This document reflects the author's views and the Community is not liable for the use that may be made of the information it contains

Table of Contents

EXECUTIVE SUMMARY

This document summarizes the work carried out in Task 4.1 "Environments for collaborative ontology mapping". The outcome of the work done in this task has been the creation of an environment which supports a semantic data integration process. The report contains a description of both the work process and of the prototype environment. This document also introduces the terminology which will be used in the subsequent work to be done in WP4 "Semantic energy information framework".

The report is structured in the following sections:

- 1. Introduction: purpose of the deliverable, contributions of each partner, and relationships of the work done in this task with other work packages.

- 2. Concepts and definitions of the terms used in the various contexts of the project as agreed by all partners.

- 3. Semantic energy information framework: current status of the design and development of the framework. Energy model definition and implementation. Data source relations included in the energy model.

- 4. Semantic integration process: current status of the semantic integration process at this point of the project development. Research community trends and their relations to the SEMANCO's semantic integration process.

- 5. Mapping tools: description of the tools, requirements defined according to the integration process, contrast with the state of the art, and implementation details.

- 6. Validation: verification of the functionality of the tools in a test case using a real data source.

- 7. Conclusions: contributions of the developed tools to the field of ontology design. Role of the tools in the project development and expected impacts on the demonstrations scenarios.

1 INTRODUCTION

1.1 Purpose and target group

The purpose of this deliverable is to report the work of Task 4.1 "Environments for collaborative ontology mapping". The purpose of the tools developed in this task is to help users –domain experts, data owners and ontology engineers– to integrate data sources into the semantic energy information framework (SEIF).

1.2 Contribution of partners

The work done in Task 4.1 has been led by FUNITEC in collaboration with HAS and CIMNE The SEIF –that is, the energy model definition, its relation with data sources, and the integration process– has been defined by FUNITEC and HAS partners.

Mapping tools have been implemented by FUNITEC and validated by HAS and CIMNE.

The report has been written by the FUNITEC with the collaboration of HAS (Section 3.2) and proofread by UoT.

1.3 Relations to other activities in the project

The tools developed in this task have been evaluated in a test case which makes use of the data sources identified in D2.1 "Case study design" and further detailed in by D3.1 "Providing access to distributed energy data repositories". The application of these tools will have a strong impact on the semantic energy model being developed in D4.2 as well as on the guidelines for structuring energy data (D3.2) and on the modelling of contextual data (D3.3).

2 CONCEPTS AND DEFINITIONS

Concerning the development of the semantic framework, the following list of terms is being used in the project:

- **Energy model**: This provides a language by which the interrelations between the different data sources may be understood and interpreted. The energy model is implemented as a global ontology which embraces all the terms which tools can use to gather semantic data from the SEIF.

- **Ontology**: An explicit specification of a conceptualization, according to the definition of Gruber (1993). A set of concepts and its relations which are defined in the form of axioms or properties implementing the RDF triples model. Concepts related to each other by specialization/generalization constitute a taxonomy, which is often seen as the core part of an ontology. Usually, ontologies are formally specified using description logic formalisms and coded in machine-readable languages like OWL.

- **Ontology, Local**: An ontology specifying semantics of the data located in a single data source.

- **Ontology, Global**: The intersection of all local ontologies which is used to facilitate reusing of concepts and mapping of local ontologies. A global ontology can be understood as a neutral reference which results from the union of all local ontologies. Local ontologies are then mapped to this neutral reference (Uschold, 2000).

- **Ontology mapping/matching**: A method of finding correspondences between concepts from different ontologies. This method is part of a more comprehensive process such as ontology merging (the process of bringing together two ontologies which describe the same domain creating a new one which replaces the original ontologies) or ontology integration (the process of interrelating information from diverse sources to create a new ontology from two or more existing ontologies related to diverse domains).

- **Semantic data integration**: Process of adding semantics to heterogeneous (in terms of format and/or structure) data sources with the purpose to unify access to these sources and to facilitate semantically the reach querying of distributed data.

- **SEIF**: Semantic energy information framework which connects heterogeneous (in terms of format and/or structure) data sources, such as energy performance of the building stock at different geographical scales, in order to embody the energy-related knowledge planners needs. The SEIF has three main goals: 1) storing semantic data which is presently off-line and/or heterogeneously structured; 2) providing a bridge between external and internal data sources to facilitate data analysis; and 3) gathering outputs of the data mining processes in order to improve the knowledge stored in it.

- **Use case**: A semi-formal specification of a situated application of a tool or system as well of its application purpose and context to delimit a research problem and/or to state a strategic goal. It describes how actors, tools and data are interrelated in order to fulfil certain goals.

3 SEMANTIC ENERGY INFORMATION FRAMEWORK

The semantic energy information framework (SEIF) is the nexus between the different data sources and tools using the semantically specified data. The SEIF accesses energy data of the domestic and commercial building stock at different geographical scales in order to embody the energy-related knowledge that different stakeholders need.

The SEIF has three main goals:

- Integrating proprietary data which is presently off-line or/and heterogeneously structured into a consistent distributed knowledge base, making the data accessible for information discovery and retrieval purposes.

- Providing a bridge between different domains (city planning and energy provision) and contents (consumption data, pollution sources, simulated energy profiles and benchmarks).

- Gathering outputs generated by the tools developed in the project (WP5) –tools for design evaluation and energy simulation, visualization and modelling at urban scale, and analysis & optimization processes– in order to create a distributed knowledge base.

At the time of writing, the SEIF is under development. Two processes of work have been initiated. The first one is focused on data integration. It begins with the identification of available data sources (Task 2.1), continues with the data analysis and modelling in the Tasks 3.2 and 3.3 and ends with the data migration in Task 3.4. The second process focuses on the usage of the integrated data by the tools. The process begins with Tasks 6.1 and 6.3 with the definition of application scenarios using the tools to be developed in WP5.

3.1 Integrative SEIF architecture

The primary example for semantic interlinking of distributed data is the Linked Open Data Project (Heath & Bizer, 2011). It exploits shared identifiers (URIs) to interlink data distributed over the different sources, whereby the structure of each source is described by a dedicated ontology. In terms of openness and flexibility, this approach works well. It faces two obstacles, however, concerning efficiency and completeness of information retrieval: 1) the structure of single data sources is usually unknown and, therefore, combining data stored in different sources requires discovering all sources where data may be located; and 2) to discover data sources, one needs to interact with multiple endpoints offering a data querying interface (Oren, 2008). Hence, the SEMANCO project takes a centralized approach. Although this approach makes the resulting system less open, it leads to significant improvements regarding efficiency and accuracy.

In contrast to the Linked Open Data architecture, the centralized approach of the SEMANCO project uses a single global ontology as the main reference for all data sources. Data from a single source refer to concepts of the local ontologies univocally mapped to the global ontology. That is, each concept of a local ontology corresponds to one of those in the global ontology. Accordingly, the global ontology is per se the union of elements of local ontologies (Uschold, 2000).

In the SEIF architecture the central single end-point serves are the addresses for the data queries sent by tools developed in WP5. Accordingly, all queries must be formulated in a query language able to express data semantics, for example SPARQL, and must use the vocabulary and the conceptual model of the global ontology. Furthermore, in the proposed SEIF architecture each local source must also expose a SPARQL end-point. It is necessary to

forward queries, or their parts, to the local sources where the searched data are physically stored.

The central end-point receives queries sent by tools and splits them into fragments such that each fragment contains only the names of concepts and properties that are elements of one of local ontologies. These fragments are forwarded to the corresponding local endpoints. Query results sent from a local endpoint to the central one are cumulated and harmonized to ensure the referential integrity and to guarantee the dimensional consistency (e.g. meters, kilometres or miles) and finally forwarded to the tool that initialized the querying process (Figure 1).
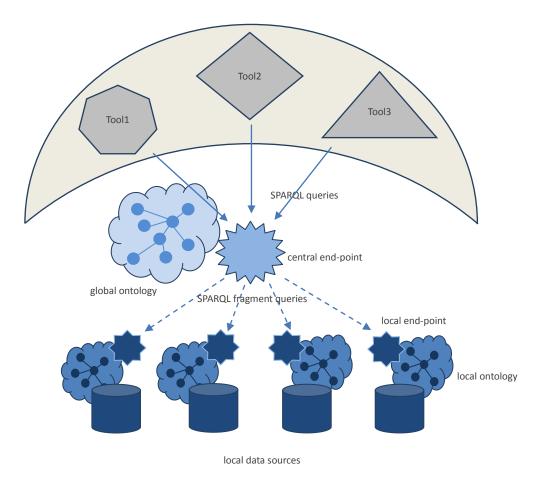

Figure 1.Ontology mapping tools. Inputs and outputs

To support SPARQL end-points, local data sources integrated into the SEIF need to be implemented as Resource Description Framework (RDF) triple stores such as AllegroGraph, Virtuoso or BigOWLIM, where each elementary record is built upon the subject-predicate-object model which is at the core of the RDF. Alternatively, local stores can use a conventional data model. However, in the latter case mediator software is necessary to translate semantic queries formulated in SPARQL to the dedicated query language, for example SQL. Such facility is offered by the D2R server[1] and Ontop[2] framework using the Quest reasoner. In the current stage of the SEMANCO project we leave open the option to use one or another technology for each local store; the final decision will be informed by the evaluation results. However, we note that theory and best practice suggest that, optimally, data is to be stored in relational databases (Poggi et al., 2008).

---

[1] http://d2rq.org/
[2] http://obda.inf.unibz.it/protege-plugin/index.html

As mentioned above, the centralized architecture of SEIF enables higher efficiency and precision of query processing compared to decentralized architectures. However, these advantages can only be realized when the decentralized architecture is supported by an appropriate look up service. The general purpose of such a service is to find local ontologies containing the concepts and properties specified in a SPARQL query currently submitted to the central endpoint. If this knowledge is available then it can be taken into account in the fragmentation of the query. Hence, the fragments sent to the local end-points will not contain superfluous triples that would otherwise increase query processing time and potentially lead to incorrect query results.

Schwarte et al. (2011) have reported successful application of the SPARQL's ASK statement for this purpose. However, sending ASK statements to each local endpoint in real time before query fragmentation increases query processing time. The centralized SEIF architecture advantageously makes available local ontology concepts and properties that can be checked by the central endpoint in processes independent of query execution. The SEMANCO look up service periodically scans local ontologies for all concepts and properties of the global ontology. The scan results are stored in a single data table. Therefore, one single query specifying a concept or property name delivers a list of all local ontologies containing the corresponding concept or property.

3.2 Semanco global ontology

The core component of SEIF, the global ontology, is to be created in Task 4.2 and is currently under development. Here we provide a brief description of the on-going ontology design process. This iterative process consists of generating a series of constructive solutions and their refinements. The first step is the informal specification of a knowledge model that we call the "energy model". This is followed by the formalization and mapping of the data sources identified in Task 2.1 and modelled by Tasks 3.2 and 3.3 onto the currently available ontology version. The mapping process has three purposes: 1) data distributed in diverse sources become semantically rich, that is, they are enriched with metadata; 2) the data are integrated into SEMANCO knowledge base and made available for semantic information retrieval methods; and 3) the SEMANCO global ontology is enhanced and improved by new concepts and relations discovered during the mapping process.

The SEMANCO energy model provides the necessary language to represent the complexity of interrelations between the different data sources identified in D2.1 and modelled in Tasks 3.2 and 3.3. Also, it is a key component of the semantic integration process –carried out with the help of the ontology mapping tools developed in Task 4.1– acting as a global reference which interconnects the data sources.

The terminology used by the semantic energy model is based on energy and environmental international standards. Specifically, it relies on the ISO/IEC CD 1323 (Energy efficiency and renewable energy sources), ISO/DTR 16344 (Common terms, definitions and symbols for the overall energy performance rating and certification of buildings), ISO/CD 16346 (Assessment of overall energy performance of buildings), ISO/DIS 12655 (Presentation of real energy use of buildings), ISO/CD 16343 (Methods for expressing energy performance and for energy certification of buildings), and ISO 50001:2011 (Requirements with guidance for use Energy management systems). The terms selected from these standards to be used in the energy model are being collected and structured by Task 3.2.

As stated at the beginning of this section, the semantic energy model is implemented as a global ontology embracing all the terms which the tools need to interact with the SEIF. The global ontology is based on a top-level ontology, the Suggested Upper Merged Ontology (SUMO) which incorporates terms from philosophical, engineering and linguistic domains

which are inherited by the energy model ontology. The ontology is coded in OWL *DL-Lite$_A$* dialect, recommended by Poggi et al. (2008) due to its good performance in reasoning tasks which require conjunctive queries of large data volumes. This dialect has two mandatory features:

- The domain and range of properties can only be specified for functional data properties;

- The definition of an object property connecting two OWL classes 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 ($\check{\ }$) to express that the class *BuildingGeometry* relates to the class *Building* via the *hasGeometry* property.

The energy model will be fully developed in T4.2.

4 SEMANTIC INTEGRATION PROCESS

The semantic web concept coined by Tim Berners-Lee has been strengthened by the Linked Data movement, which promotes the use of ontologies to connect heterogeneous sources of data handled by domain experts. In order to facilitate the interoperability between the different ontologies, a semantic integration process is needed.

Some authors contend that ontology matching is a plausible solution to the semantic heterogeneity problem (Zhdanova, 2005; Euzenat, 2011). Shvaiko & Euzenat (2008) summarized some of the trends in the ontology matching research field, concluding that users (domain experts) should participate in the matching process since an automatic ontology matching usually do not produce good results. Furthermore, they suggest developing social and collaborative environments to take advantage of the network effect so that each user can improve on what has been done by others. By doing this, the number of errors can be reduced.

In the SEMANCO project, a semantic integration process will be carried out in T3.4 to incorporate data sources into SEIF. This will be the basis for creating the energy model in T4.2. This process will be performed by domain experts, data owners, and ontology engineers. To carry out this work, we need a collaborative environment which fosters the participation of the different actors with different levels of knowledge about energy, urban developments, standards, and ontology design fields. The creation of such an environment has been the purpose of the task reported here.

The integration process includes adding semantic meaning to data which in turn requires knowing the use that will be made of these data in a particular context. The case study approach adopted in the project helps to bring together actors, data and tools. Their interrelationships in a particular context are encapsulated as a use case. This approach enables us to take into consideration the users' needs in the semantic data modelling thus ensuring that integrated data will be used by stakeholders in real contexts (Lannon & Linovski, 2009).

The semantic integration process is divided into three steps:

1. Creation of local ontologies for each data source

2. Integration of local ontologies within the energy model (global ontology)

3. Data source transformation to RDF

Due to the large number of different data sources considered in the project (see D.2.1) it is necessary to automate as many steps as possible of the integration process. Particularly, there are two steps to be automated: 1) the creation of a local ontology from a structured source which is usually stored in a database, XML file or Comma-separated values file; and 2) the translation of structured data to RDF.

Regarding the translation of structured data, a survey published by W3C RDB2RDF[3] incubator group has identified several tools which can automatically carry out this work such as Virtuoso RDF View, D2RQ, R2O, RDBToOnto, or Dartgrid (Sahoo et al., 2009). The survey concluded that there is not a standard method for representing mappings between relational databases and RDF and recommended that, whenever possible, to implement on-demand mapping to access to the last version of the data. In February 2012, the RDB2RDF group has published a R2RML (Relational database to RDF mapping language) recommendation which is being implemented by some projects at this time.

The semantic integration process has been designed *ad hoc* for the SEMANCO project in

---

[3] http://www.w3.org/2001/sw/rdb2rdf

order to include domain knowledge for each case study. From the technological point of view, this process supports the centralized approach adopted of the SEIF, and it is based on the latest community tools.

5 MAPPING TOOLS

This section describes the set of tools for collaborative mapping created in Task 4.1.This set consists of two tools: an OWL mapping extractor; and an ontology mapping collaborative web environment.

These tools have been created to be used in the semantic integration process described in the previous section. They should help users –domain experts, data owners, and ontology engineers– to integrate data in a collaborative way using standard semantic technologies. The tools also automate parts of the semantic integration process.

The requirements and features of the tools are as follows.

1. To generate an OWL ontology from a schema database (almost all data sources exist as databases).

    a. The ontology should be written in *DL-Lite$_A$* format: domain and ranges of object properties are written in axiomatic form.

    b. Some data attributes should be mapped to OWL classes in order to assign dimensions to them (e.g. kilometre, kilojoules/m^2, or tonnes per year).

2. To translate relational data stored in tables and columns to RDF format according the local ontology previously generated. The translation can be static –a dump file– or dynamic based on translating SPARQL queries into SQL.

3. To provide an environment where users can easily map a local ontology to a global ontology according to their knowledge. This environment should:

    a. facilitate users' communication;

    b. support the integration process, tracking the users' activity;

    c. provide an interface for non-ontology engineers.

Before committing ourselves to the development of these tools, we undertook a literature survey of the field.

Several authors have been active in the generation of an OWL ontology from a schema database. The ontology creation is driven by specific rules which include detailed restrictions such as property cardinality (Zhou et al., 2010). RDB2On, Relational.OWL, and DB2OWL tools were examined and it was concluded that, as none of them support *DL-Lite$_A$* dialect as output, the first requirement is not completely fulfilled.

Concerning the translation of relational data to RDF, most of the tools identified by the RDB2RDF group are designed for technicians who are familiar with databases and non-graphical environments. Some of these tools provide a graphical interface such as ODEMapster[4], RDBToOnto[5], or Dartgrid[6] but are not intended for users who are not experts on ontologies.

Finally, with regard to mapping from local ontologies to a global ontology, the tools provided by the community are usually proofs of concept or prototypes which are mostly created for ontology expert users. There is a lack of tools which allow non-expert users to participate in the integration process.

---

[4] http://mayor2.dia.fi.upm.es/oeg/index.php/en/downloads/9-r2o-odempaster
[5] http://www.tao-project.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html
[6] http://www.w3.org/wiki/DartGrid

In conclusion, we have not been able to identify a tool or set of tools which can fulfil the requirements stated above. To fill this gap, we have implemented our own tools to support the semantic integration process being carried out in the project. The tools developed in Task 4.1 are based on D2RQ (Bizer & Cyganiak, 2007) which supports database translation with high performance. This platform dynamically rewrites the SPARQL queries into SQL, is a stable lightweight solution, represents mappings with the easily customizable D2RQ mapping language, and is currently being developed to support R2RML.

The following sections describe the mapping tools within the context of the integration process.

5.1 Using the tools in the integration process

The set of ontology mapping tools we have developed enable users to follow the steps of the integration process which is depicted in Figure 2.The inputs of the ontology mapping tools are the use cases (provided by the users), the current version of the energy model and a data source. The outputs are an improved version of the energy model and a mapping file which is used to transform relational data into RDF.
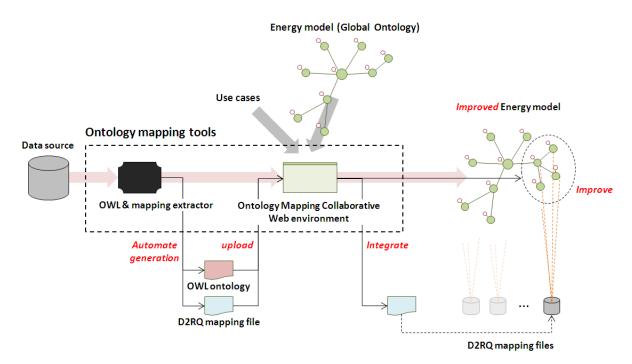

Figure 2.Ontology mapping tools. Inputs and outputs

The integration process is carried out in three steps: generating a neutral ontology automatically, redefining it by domain experts, and transforming the data sources.

First, an OWL ontology is created from the relational schema of a data source. Along the way, a D2RQ mapping file is also generated according to the ontology which is then used to automate the data source transformation (step 3). The OWL ontology is created according the rules found in the data source (Figure3):

- Rule 1: When a single table is found containing columns which are an attribute (e.g. building name, architect, floor plans, or indoor temperature), then the table name will become the OWL class name; the columns containing values that are not associated with dimensions (e.g., *building name*) will turn into data properties, and the columns containing values that are associated with dimensions (e.g. *indoor temperature*

associated with *Celsius degree*) will be converted to new concepts which will be connected to the table class through an object property. This rule is also applicable to CSV or spreadsheet files.

- Rule 2: When two tables are related by a foreign key, then each table becomes an ontology portion following the first rule. Then, an object property is generated connecting both tables.

- Rule 3: When three tables (or more) are related many to many by foreign keys, then the second rule is applied to each table without taking into account the table which acts as a connector between the two other tables. .

At the end of the first step, a neutral OWL ontology is obtained and also a D2RQ mapping file which can transform relational data into RDF based on the OWL ontology classes and properties.
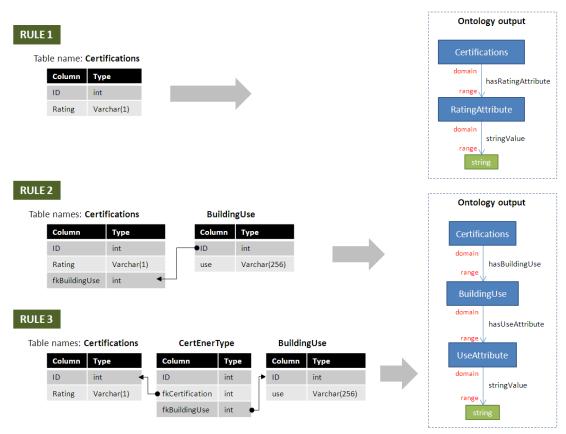

Figure 3. Rules to generate an ontology from a relational data structure

Domains and ranges of the properties between classes (object properties) are implemented with special axioms following *DL-Lite$_A$* recommendation, specifically: *owl:someValuesFrom*. The blue boxes in Figure 4 show the axioms generated for the example of the first rule shown in Figure 3. The blue box in the left defines the domain, and the right box the range of the *hasRatingAttribute* property. The *stringValue* property is not a relation between classes. Therefore, domain and range are modelled applying *rdfs:domain* and *rdfs:range* statements instead of using special axioms (Figure 4, green box).

DL-Lite_A axioms

:Certifications rdf:type owl:Class ;
:RatingAttribute rdf:type owl:Class ;

[rdf:type owl:Restriction ;
rdfs:subClassOf :Certifications;
owl:onProperty :hasRatingAttribute;
owl:someValuesFrom owl:Thing
] .

[rdf:type owl:Restriction ;
rdfs:subClassOf :RatingAttribute ;
owl:onProperty [owl:inverseOf :hasRatingAttribute] ;
owl:someValuesFrom owl:Thing
] .

:stringValue   rdf:type owl:DatatypeProperty ;
                      rdfs:domain :RatingAttribute;
                      rdfs:range xsd:string .

Figure 4. DL-Lite_A axioms expressed in Notation3 (N3)

In the second step of the integration process, domain experts and users redefine the neutral ontology to suit the energy model and the use cases being considered. Each class and property is analyzed and then redefined according the following cases:

- Case 1: Replacement. A local ontology element –class or property– is replaced by an energy model element because both elements are similar. All element name occurrences are renamed by the energy model element name.

- Case 2: Aggregation. When a data source ontology element does not exist in the energy model ontology, then it becomes a new element of the energy model. If the element is a class, then it is needed to define one or more super-classes given by the energy model ontology. In any case, the name of the element can be changed by the user.

The third step is focused on transforming relational data into RDF, which is carried out using D2RQ tool. The mapping file generated in the first step, and redefined in the second step by domain experts is used as a configuration file to execute D2RQ tool.

5.2 OWL & N3 extractor

The purpose of this tool is to extract an OWL ontology file and a D2RQ mapping file from a relational database. The user of the tool can be the owner of the data set. The tool is configured defining the database string connection and driver, username, password and the name of the output ontology. Additionally, the user selects which columns would need units. The tool reads the database structure and, following the three rules described in the previous section, generates an OWL ontology and a D2RQ mapping file.

This tool is developed in Java as a command line program and the configuration parameters are entered in the command line. A help guide can be found in Appendix A of this document.

5.3 Ontology mapping collaborative web environment

The purpose of this environment is to help domain experts and ontology engineers to redefine the data source ontology. Users register a data source in the environment by uploading the files generated by the extractor tool. Afterwards, they can view a list of mappings and their current status: selected to be integrated, target class, type of case applied and the date of the last modification (Figure 5).Users can also comment their actions.
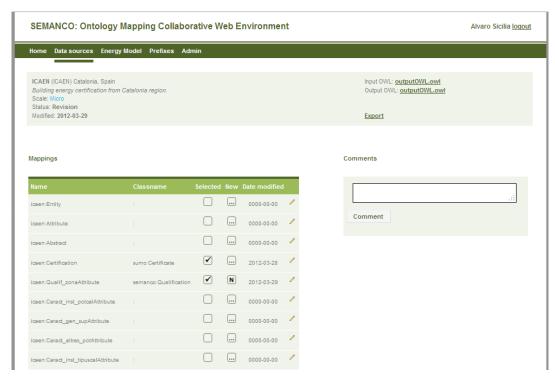
Figure 5.Ontology mapping collaborative web environment. Data sources mappings

As can be seen in Figure 6, users can edit a class mapping by selecting a class of the energy model or providing a new one. In the case of new classes, users need to supply supper-classes to connect them to the energy model. Also, they can modify the data property mappings. In this collaborative environment users can track the activity of their colleagues and also comment their actions.
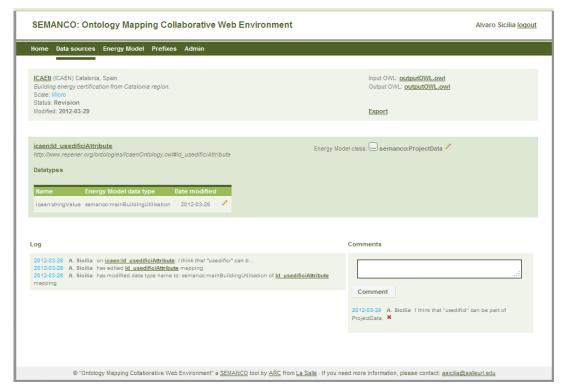

Figure 6.Ontology mapping collaborative web environment. Mapping edition

In order to enhance the interface usability, a graphical ontology representation has been implemented (Figure7). It is based on radial graph visualization where the central node is an ontology class. The edges between nodes are relations between ontology classes. Each circle

hosts the ontology classes which are related to the root class. This kind of representation enables users to explore the classes and properties of a local ontology as well as the global ontology which implements the energy model.

This graphical representation has been packed into an independent module to facilitate its inclusion in different environments such as the ontology explorer which will be implemented in Task 4.3. When the user focuses on a class the module retrieves the annotation properties – *rdfs:label*, *rdfs:comment*, or *rdfs:seeAlso*–and shows them in a window. When a class which is not at the centre is clicked then it becomes the root of the radial graph. The colour of the lines indicates the type of the relation between the classes –aggregation properties and subclass of properties–, and the line weight is proportional to the number of object properties between the classes.

The configuration guide for developers can be found in Appendix C of this document.
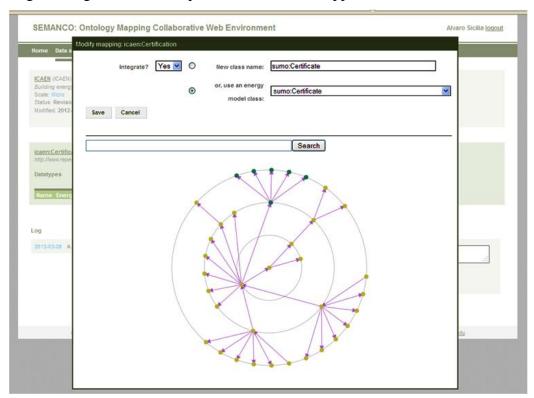

Figure 7. Ontology mapping collaborative web environment

Finally, users can export the work done generating an OWL ontology which contains the performed mappings. Domain and ranges of the properties are generated according of the axioms described above. This ontology is a portion of the energy model. Each time a data source is integrated, the energy model is enhanced with new terms and properties.

The front-end of this environment has been developed using standard languages such as HTML, Javascript and CSS. To speed up the programming and reduce the errors a developing framework for PHP has been used, namely CodeIgniter. MySQL database was selected to store the environment data. Some open source PHP libraries have been used such as ARC[7]to parser RDF files and the JavaScript InfoVis Toolkit[8] to implement the graphical representations.

A help guide can be found in Appendix B of this document.

---

[7] https://github.com/semsol/arc2/wiki
[8] http://thejit.org/

6 VALIDATION

This section describes the test case implemented to validate the tools. The test case has been carried out using real data indentified in D.2.1 for the Manresa case study. The data were provided by ICAEN, an organisation of the Catalan government which collects the energy certificates of newly planned buildings including their simulated performance.

6.1 Description

The ICAEN data source is structured as a single table which contains 35 columns and 1804 rows. Each row is an energy certification of a specific building. The following table includes some of the columns of the table:

Table 1.ICAEN data source structure: column names and descriptions

ID	Id of the energy certification
QUALIF_OBTINGUDA	Energy rating
DATASORCAT	Date when the certification was made
QUALIF_ZONA	Climate zone where the building is located
ID_USEEDIFICI	Building use
ID_LOCALITAT	Name of the city or town where the building is located
CARACT_INST_FONTACS	Energy carrier for domestic hot water
CARACT_ALTRES_ACS	Solar energy contribution to domestic hot water
CARACT_GEN_SUP	Built surface
QUALIF_CONSUM_ANY	Yearly energy consumption
QUALIF_CONSUM_M2	Yearly energy consumption per m^2
QUALIF_EMIS_ANY	CO_2 yearly emissions.
QUALIF_EMIS_M2	CO_2 yearly emissions per m^2

6.2 Mapping process

The mapping process carried out in the test case is as follows.

1. Generation of an OWL ontology from the ICAEN table using the OWL & N3 extractor. The following configuration was used to call the extractor tool:

```
owln3extractor.jar –dbjdbc:mysql://localhost/icaen -driver
com.mysql.jdbc.Driver -u root -p "" -n Icaen -o ICAENtest
```

An excerpt of the output ontology file can be found in Appendix D of this document.

2. The files automatically generated by the extractor are uploaded into the ontology mapping collaborative web environment.

3. The following mappings have been performed:

Table 2.ICAEN data source mappings performed during the test case

Source class	Energy Model class	New
QUALIF_OBTINGUDA	EnergyRating	yes
DATASORCAT	ProjectData	
QUALIF_ZONA	Climat	
ID_USEEDIFICI	MainBuildingUtilisation	
ID_LOCALITAT	BuildingLocation	
CARACT_INST_FONTACS	EnergySource	
CARACT_ALTRES_ACS	SolarContribution	yes
CARACT_GEN_SUP	ConditionalFloorArea	
QUALIF_CONSUM_ANY	YearlyEnergyPerformance	yes
QUALIF_CONSUM_M2	EnergyPerformance	yes
QUALIF_EMIS_ANY	YealryEnergyPerformance	yes
QUALIF_EMIS_M2	EnergyPerformance	yes

New classes have been created during the test case implementation because they were not included in the energy model.

Some source classes were mapped to the same energy model class because this class contains a data type for each mapping.

6.3 Conclusions

The semantic integration of the ICAEN data source was carried out successfully although we came across some difficulties. Once the energy model structure was understood by all users involved, the main issue was to agree on the new terms which were not present in the energy model ontology. This can be solved if the data sources are well documented because the tool has been designed to read annotation properties of classes. However, once the users have become acquainted with the energy model ontology then the integration process becomes more fluid.

Regarding the usability of the tools, searching for a specific class is –at the moment of writing this report– is not as straightforward as was anticipated. More time than expected is required because the energy model ontology is quite large: it is hosting more than 700 classes and 40 data properties.

7 CONCLUSIONS

7.1 Contribution to overall picture

A prototype environment has been designed and implemented to help users from different backgrounds to carry out semantic integration tasks in collaboration. The environment provides a graphical representation to explore the ontologies in an intuitive manner. These mapping tools have been implemented to assist the semantic integration process currently under way in the SEMANCO project. However, the tools we have developed could readily be applied to other projects as well since they have been conceived to address generic problems in the ontology building process and have been created with standardized components. Specifically, the graphical representation of ontologies has been implemented as a module which can be integrated in external web applications based on the CodeIgniter framework.

7.2 Impact on other WPs and Tasks

The work done in this first task of WP 4 establishes the foundations of the SEIF and the tools which will be developed in WP 5.

The tools developed in this task will be applied in Task 4.2 to improve the energy model which will be designed iteratively by integrating data sources. Mapping tools are part of a set of tools (e.g. Ontology editor, ontology explorer) which will help different kinds of users – domain experts, data owners and ontology engineers– to design the energy model.

Furthermore, these tools will be used in Task 3.4 to migrate relational data to RDF. In this task, the mapping files necessary to setup the D2R server will be automatically generated by the tools.

7.3 Contribution to demonstration

The tools described in this document will help users to carry out the data integration process which will solve the interoperability issues between heterogeneous data sources. By doing this, a unified access to data will allow the tools to retrieve data through the SEIF in order to perform analysis and provide final users with qualified data to make decisions.

7.4 Other conclusions and lessons learned

An issue to take into account is the scalability of the energy model ontology. At the time of writing, the ontology contains 700 classes, 76 concerning the energy domain and the remaining coming from SUMO ontology. Each time the semantic integration process is carried out, the energy model will grow requiring more time to be assimilated by users. The success of the application of the tools depends, therefore, on the ability of the users to understand the structure of the energy model as well as having well documented data sources.

8 REFERENCES

Bizer, C. & Cyganiak, R. (2007) D2RQ – Lessons learned. Position paper at the *W3C Workshop on RDF Access to Relational Databases*, Cambridge, October 25-26, 2007.

Euzenat, J. (2011) Semantic technologies and ontology matching for interoperability inside and across buildings. Proceedings of the *2nd CIB Workshop on eeBuildings data models*, Sophia-Antipolis, October 26-28, 2011.

Gruber, T. R. (1993) A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220.

Heath, T.&Bizer, C. (2011) *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers.

Lannon, S. & Linovski, O. (2009) Ontologies for the Classification of Urban Characteristics: Opportunities for Urban Designers. Ontology-Based Models for Improving the Interoperability of 3D Urban Information. Proceedings of the *Final Conference of the COST Action C21 –Towntology: Urban Ontologies for an Improved Communication in Urban Development Projects*, Liège, March 9-10, 2009.

Oren, E. (2008) Sindice.com: A document-oriented lookup index for open linked data. Journal of *Metadata, Semantics and Ontologies*, 3(1), pp.37-52.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008) Linking data to ontologies. Journal on *Data Semantics*, pp.133–173.

Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Jr, T. T., Auer, S., Sequeda, J. &Ezzat, A. (2009) A Survey of Current Approaches for Mapping of Relational Databases to RDF. *W3C RDB2RDF XG Report*, Incubator Group. Available at: http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf (Accessed: 25/4/2012).

Schwarte, A., Haase, P., Hose, K., Schenkel, R., & Schmidt, M. (2011) FedX: Optimization Techniques for Federated Query Processing on Linked Data. Proceedings of the *10th International Semantic Web Conference (ISWC),* Bonn, Germany, October 23-27, 2011.

Shvaiko, P., & Euzenat J. (2008) Ten Challenges for Ontology Matching. Proceedings of the *On the Move to Meaningful Internet Systems (OTM) Conferences 2*, vol. 5332 of Lecture Notes in Computer Sciences, pp. 1164-1182.

Uschold, M. (2000) Creating, Integrating and Maintaining Local and Global Ontologies. Proceedings of the*14th European Conference on Artificial Intelligence (ECAI'00)*, Berlin, Germany, August 20-21, 2000.

Zhdanova, A. (2005) Towards a community-driven ontology matching. Proceedings of the *3rd international conference on knowledge capture K-Cap'05*, Alberta, Canada. October 2-5, 2005.

Zhou, S., Meng, G., Ling, H., & Zhang, G. (2010) Tool for Translating Relational Databases Schema into Ontology for Semantic Web. *2nd International Workshop* on *Education Technology and Computer Science (ETCS),* vol.1, pp.198-201, March 6-7, 2010.

9 APPENDICES

- Appendix A is a help guide for execution of the OWL & N3 extractor tool.

- Appendix B is a help guide for the ontology mapping collaborative web environment.

- Appendix C is a guide for developers to setup the radial graph ontology representation module.

- Appendix D is a part of the OWL ontology file generated in the test case.

9.1 Appendix A

OWL & N3 extractor tool is a java application which is invoked in the command line. When it is called without arguments an argument setup explanation is displayed as in the following code:

```
SEMANCO: OWL & n3 extractor Tool @ 2012
---------------------------------------

owln3extractor.jar -db<databaseURL> -driver <databaseDriver> -u <username>
- p "<password>" -n <ontologyname> -o <outputfiles>

Example:
owln3extractor.jar -dbjdbc:mysql://localhost/icaen -driver
com.mysql.jdbc.Driver -u root -p "test" -n Icaen -o ICAENtest
Password can be empty:
owln3extractor.jar -dbjdbc:mysql://localhost/icaen -driver
com.mysql.jdbc.Driver -u root -p "" -n Icaen -o ICAENtest
------------------------------
```

The tool requires the following arguments:

- -db: database URL according jdbc connection. It should include database manager (e.g. MySQL, SQL Server, Oracle), the name of the server that hosts the database, and also the database name.

- -driver: Driver of the database manager. The library containing the driver should be stored in the same folder as the tool.

- -u: username to query the database.

- -p: password associated to the username between quotes. If there is not a password then the parameter should be left blank as "".

- -n: name of the output ontology.

- -o: name of the output files.

Once the extractor is executed, two files are generated: an OWL file and an N3 mapping file for D2R Server. Figure A1 shows the ontology automatically generated by the extractor tool. Attributes classes are subclasses of the *Attribute* class and the other classes are placed below the *Abstract* class.
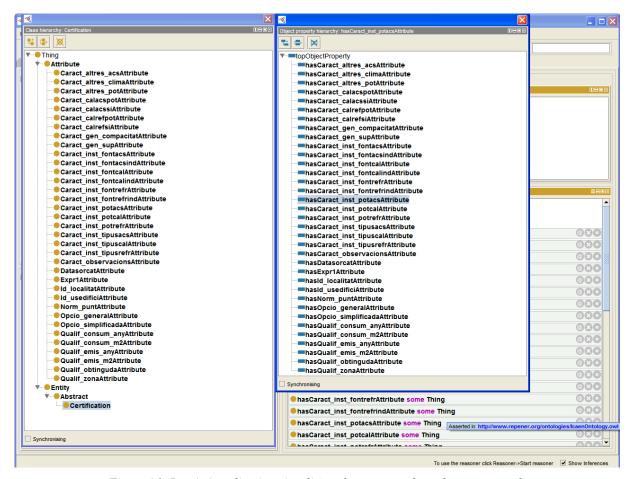
Figure A1. Protégé application visualizing the automated ontology generated

9.2 Appendix B

Ontology mapping collaborative web environment is a multiuser web platform. There are two types of users: regular users and administrators. Table A1 show the actions which each type of user can perform.

Table B1.Ontology mapping collaborative web environment actions related to types of user

Action	Regular user	Administrator
Create/Modify/Delete data source	X	X
Edit a mapping	X	X
Edit a data type	X	X
Add comments	X	X
Export ontology	X	X
Update energy model ontology		X
Manage user accounts		X

9.2.1 Creating/modifying/deleting a data source

In the "Data sources" menu option, click "Create new" data source to go to "add new data source" form (Figure A2). In this form, the user should provide the name, description, scale, location and owner of the data source. Also, the user should upload the files generated by the ontology extractor.

Figure B1. Add a new data source form

To modify a data source, in the "Data sources" menu option, click the "pencil" button. An "edit data source" form will be displayed. This form is equivalent to the "add new data source" form.

To delete a data source, in the "Data sources" menu option, click the "X" button to delete it.

9.2.2 Editing a mapping

Once a data source is selected the mappings are shown (Figure A3). Each mapping has a *Name*, *Classname*, *Selected*, *New* and *Date modified* fields. *Name* is the table name of the source database. *Classname* is the new name of the mapping; it can be an energy model class or a new class. *Selected* indicates whether the mapping will be exported or not. *New* specifies if the mapping reuses an energy model class or if it is a new one. *Date modified* shows the date when the mapping was last edited.
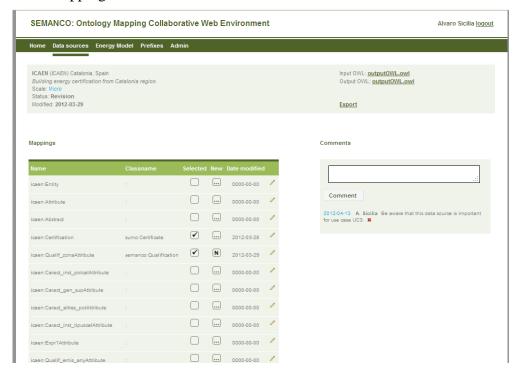

Figure B2. Data source mapping

To edit a mapping, click the "pencil" button to go to "edit mapping" page (Figure A4). Click the "pencil" button to the right of *Energy Model* class label to select an energy model ontology class or to create a new class.

To edit a data type, click the "pencil" button of the *Datatypes* table.

To assign a super-class click the "+" button. Also you can delete a super-class with the "X" button.
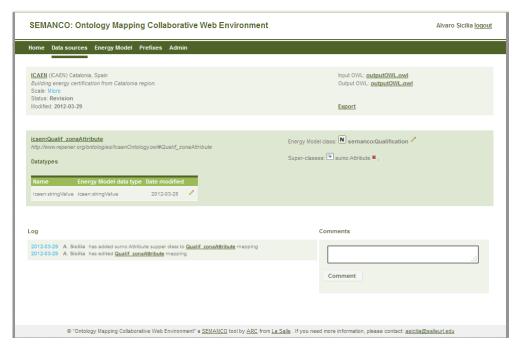

Figure B3. Edit mapping page

9.2.3 Exporting a data source

Select a data source. Click "Export" in the header of the data source and the export process will be executed. Figure A5 shows the log of the export process. Once the process ends, the ontology is downloaded by clicking the link at right of "Output OWL" label in the data source header.
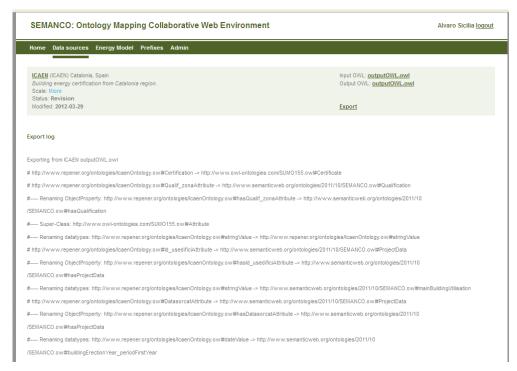

Figure B4. Export output process

9.2.4 Updating energy model ontology

In the "Energy Model" menu option, click "Update energy model" to retrieve the last version of the energy model ontology. Once it is updated it can be used in the mapping process.

9.2.5 Managing user accounts

In the "Admin" menu option users with administration rights can manage the environment accounts. By clicking "Create a new user", the user can create a new user giving first name, last name, email, and password.

9.3 Appendix C

The radial graph ontology representation module has been implemented as a set of controllers, models and views of the CodeIgniter framework. The module can be used in new CodeIgniter web projects or it can be used as a stand-alone web application. The following libraries have been used to implement this module:

- JavaScript InfoVis Toolkit provides a set of tools for creating interactive data visualizations for the Web.

- ARC, a flexible RDF system for semantic web and PHP professionals.

- Jquery, a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

The presentation layer of this module uses AJAX requests to retrieve data from the database and generate the visualization. These requests have several optional parameters to help developers to customize the environment. Table A2 shows the configuration parameters.

Table C1.Configuration parameters of the graphical representation of ontologies

Name	Mandatory	Description
Id	Yes	Identify Rgraph visualization. This id must be unique
Workspace	Yes	Workspace name where ontology is saved in database
selected_class	Yes	Focused class when Rgraph is loaded
annot_onto_div_id	No	DIV id which will contain annotations regarding the ontology
annot_class_div_id	No	DIV id containing annotations about focused class
annot_relat_div_id	No	DIV id which will contain the list of relations of the focused class
control_panel	No	Boolean indicating if buttons to activate / deactivate colors and subClassOf relations are shown
search_div_id	No	DIV id containing the search box to find classes. The class selected box will be focused in Rgraph

The following code is an example of the configuration of the module:

```
$.post("<?php echo site_url("rgraphs/rgraph"); ?>",
{
    id:"test",
    workspace: "aemet",
    selected_class: "http://purl.oclc.org/NET/ssnx/ssn#Accuracy",
    annot_onto_div_id: "annotation_ontology",
    annot_class_div_id: "annotation",
    annot_relat_div_id: "relations",
    control_panel: true,
    search_div_id: "search"
},

function(data){
    $('#rgraph_div').html(data);
}
);
```

9.4 Appendix D

The following code is an extract of the output of the OWL & N3 extractor executed in the test case. The format of the code is RDF/XML:

```
<?xml version="1.0"?>
<rdf:RDFxmlns="http://www.repener.org/ontologies/IcaenOntology.owl#"
xml:base="http://www.repener.org/ontologies/IcaenOntology.owl"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:IcaenOntology="http://www.repener.org/ontologies/IcaenOntology.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<owl:Ontology rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl"/>


        <!--Autogenerated Class Entity -->
        <owl:Class rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Entity"/>

        <!--Autogenerated Class Attribute -->
        <owl:Class rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Attribute"/>

        <!--Autogenerated Class Abstract -->
        <owl:Class rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Abstract">
            <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Entity"/>
        </owl:Class>

        <!--Autogenerated Class Certification -->
        <owl:Class
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Certification">
            <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Abstract"/>
        </owl:Class>

        <!--Autogenerated Class Qualif zonaAttribute -->
        <owl:Class
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Qualif_zonaAttribute">
            <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Attribute"/>
        </owl:Class>

        <owl:DatatypeProperty
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#stringValue">
            <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
            <rdfs:domain
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Qualif zonaAttribute"/>
            <rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        </owl:DatatypeProperty>

        <!-- http://www.repener.org/ontologies/IcaenOntology.owl#hasQualif zonaAttribute -->
<owl:ObjectProperty
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#hasQualif_zonaAttribute"/>
```

```
        <!-- AXIOMS definition in LiteA version -->

        <owl:Class
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Certification">
                <rdfs:subClassOf>
                <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#hasQualif zonaAttribute"/>
                <owl:someValuesFromrdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
        </owl:Restriction>
        </rdfs:subClassOf>
        </owl:Class>

        <owl:Restriction>
                <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Certification"/>
                <owl:onProperty
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#hasQualif_zonaAttribute"/>
                <owl:someValuesFromrdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
        </owl:Restriction>

        <owl:Restriction>
        <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Qualif_zonaAttribute"/>
        <owl:onProperty>
        <rdf:Description>
        <owl:inverseOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#hasQualif_zonaAttribute"/>
        </rdf:Description>
        </owl:onProperty>
        <owl:someValuesFromrdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
        </owl:Restriction>

        <!--Autogenerated Class Caract_inst_potcalAttribute -->
        <owl:Class
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#Caract inst potcalAttribute">
                <rdfs:subClassOf
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Attribute"/>
        </owl:Class>

        <owl:DatatypeProperty
rdf:about="http://www.repener.org/ontologies/IcaenOntology.owl#stringValue">
                <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
                <rdfs:domain
rdf:resource="http://www.repener.org/ontologies/IcaenOntology.owl#Caract_inst_potcalAttribute"
/>
                <rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        </owl:DatatypeProperty>
…
```