Project ICT 287534
Start: 2011-09-01
Duration: 36 months
Co-funded by the European Commission within the 7th Framework Programme

SEMANCO Semantic Tools for Carbon Reduction in Urban Planning

SEMANCO

Deliverable 4.5
Semantic Energy Information Framework

Revision: 7
Due date: 2013-08-31 (m24)
Submission date: 2013-10-29
Lead contractor: FUNITEC

Dissemination level		
PU	Public	X
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Deliverable Administration & Summary					
No & name	**D4.5 Semantic Energy Information Framework**				
Status	Final	**Due**	m24	**Date**	2013-10-29
Author(s)	Andreas Nolle (HAS), German Nemirovski (HAS), Álvaro Sicilia (FUNITEC)				
Editor (s)					
DoW	Deliverable 4.5 is the outcome of Task 4.5 *Semantic energy information framework integration*. The object of this task is the integration of the components previously developed in T4.1, T4.2, T4.3, T4.4, and T3.4 facilitating access to semantic sources and their interrelations. Implementation of the infrastructure needed for high demand performance. Component testing and integration testing to ensure a minimum level of quality. The purpose of the deliverable is to make SEIF components work together. This should be demonstrated through a set of live examples to test the capabilities of the framework including: data management, mapping processes, data exploration, and interoperability with external tools.				
Comments					

Document history

V	Date	Author	Description
1	2013-08-29	German Nemirovski (HAS), Andreas Nolle (HAS)	Document structure and Sections 2 and 3 written.
2	2013-09-20	German Nemirovski (HAS)	Review of previous version, writing Conclusion and Introduction
3	2013-10-02	Andreas Nolle (HAS)	Review of previous version
4	2013-10-06	German Nemirovski (HAS)	Review and editing
5	2013-10-08	Leandro Madrazo (FUNITEC), Álvaro Sicilia (FUNITEC), Andreas Nolle (HAS), German Nemirovski (HAS)	Review and editing
6	2013-10-09	Leandro Madrazo (FUNITEC), German Nemirovski (HAS), Andreas Nolle (HAS)	Review and editing. This version has been sent to the internal reviewer, Martin Carpenter (UoT)
7	2013-10-10	Martin Carpenter (UoT)	Proof-reading

Table of Contents

EXECUTIVE SUMMARY

Deliverable 4.5 *Semantic Energy Information Framework*, developed within Work Package 4 having the same title, summarizes the work done and the results achieved in Task 4.5 *Semantic Energy Information Framework Integration*. This work is based upon and went beyond the results achieved in the previously completed Tasks 4.1, 4.2 and 4.3. The purpose of those tasks was i) developing an energy model, an ontology serving as a mediator for integration of single data sources (Task 4.2); ii) creating mapping tools (Task 4.1) and iii) creating visualisation tools (Task 4.3). The Semantic Energy Information Framework (SEIF) developed in Task 4.5 is a key technological component of the WP 4 and one of the central components of the entire SEMANCO project. The main goal of the SEIF is to facilitate the energy assessment and analysis tools integrated within the SEMANCO platform the access to the distributed data sources that hold the data they require.

We believe that the application of the SEIF will lead to a substantial increase in the volume of data used in urban planning processes. Moreover, the SEIF will allow the use of data, such as numerous Linked Open Data sources, that has not been previously usable in urban planning processes. We believe that this integration will lead to a qualitative improvement in the standard of decision-making processes in the planning of energy efficient cities.

Furthermore, for users and tools that already retrieve data from multiple sources, the SEIF provides a substantial simplification of the queries required for data retrieval. It allows users to abstract from the specifics of the data access and data schemas used by particular individual sources. All queries are formulated in standard SPARQL using terms drawn from a vocabulary with which domain experts (urban planners, architects, communal development managers, etc.) are familiar with.

Single data sources are made accessible by their vocabularies/data models/schemas mapping onto the energy model. When a federated query which is formulated in terms of the energy model is processed, the SEIF rewrites it to express its semantics in all possible ways and looks up in the index directory to identify relevant data sources for each component of the query.

The SEIF is made of the following components:

- A federation engine (Section 3);
- Mapping tools (Deliverable 4.1 (Sicilia, Deliverable 4.1: Environments for collaborative ontology mapping, 2012);
- A semantic data explorer (Deliverable 4.3 (Wolters, Nemirovski, Pleguezuelos, & Sicilia, 2013));
- A set of requirements in the energy model (Section 4).

The document is structured as follows: after the introduction in Section 1, it follows Section 2 which describes the requirements for the SEIF and a survey of related work. In Section 2.3 we introduce in particular the system architecture of the SEIF. The federation engine implementing entailment regimes for rewriting of SPARQL queries and an improved R-Tree-based index is described in Section 3. Section 4 focuses on the issues related to the integration of single sources into the SEIF repository. We conclude the results of our work in the Section 5.

As a result of the work carried out in Work Package 4 we have developed a working prototype of a complex piece of software containing thousands of our own code lines and reusing sophisticated software components developed by third parties, e.g. Quest reasoner

(Rodriguez-Muro & Calvanese, Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access, 2012). Our next steps will be the improvement or adaptation of an existing query execution plan as well as the optimization of the all around federated querying process. Furthermore further Linked Open Data sources will be identified and integrated into the SEIF repository.

The current version of the SEIF will be compared with other systems aiming at similar goals (Section 2.2). We plan to use existing or to develop our own benchmarks for doing this.

1 INTRODUCTION

1.1 Purpose and target group

The purpose of this deliverable is to report on the work undertaken in Task 4.5 *Semantic energy information framework*.

The SEIF is the central technological component of the SEMANCO project. Its main goal is to facilitate tools used by stakeholders in urban planning process the access to data held in distributed data sources. These tools are those created in WP 5 *Integrated tools* as well as standard/legacy tools originally developed for other or more general purposes and used in the urban planning process. Examples of latter tools are URSOS, 3DMaps and RapidMiner.

The SEIF's functionality has been designed to ensure the principle of data access interoperability, that is, tools/users should be able to access data distributed in different sources. In particular, they should be able to query these sources without knowing the technical details behind data access, source content and data schema/model supported by each source.

The SEIF is a compound framework consisting of the following components:

- A federation engine, that is to say, a service developed to process SPARQL queries generated by clients (applications/tools), that is, to identify related sources that may contain required data, to adapt queries to a form appropriate for each source, to elaborate an individual execution plan for each query, to forward queries to data sources according to an execution plan and to merge answers generated by different sources and to deliver the results to the clients. This federation engine is described in Section 3 of this document;
- Mapping tools described in the Deliverable 4.1 *Environment collaborative ontology mapping* (Sicilia, Deliverable 4.1: Environments for collaborative ontology mapping, 2012) for on the fly conversion of relational data to RDF format and by these means integration of relational sources to the SEIF repository connected as shown in Deliverable 3.4 *Ontology repository with migrated data* (Sicilia, Deliverable 3.4: Ontology repository with migrated data, 2013);
- A semantic data explorer, a software for navigation over the distributed data space and visualisation of single data items; this tool is described in the Deliverable 4.3 *User interfaces for knowledge representation* (Wolters, Nemirovski, Pleguezuelos, & Sicilia, 2013);
- A set of requirements for the development of an energy model, an ontology used as a mediation schema for the repositories to be federated (see Section 4 of this document);
- The energy model and the methodology for its development are exhaustively described in the deliverable 4.2 (Nemirovskij & Sicilia, 2013).

Since some of the SEIF components have been described in Deliverables 4.1 (Sicilia, Deliverable 4.1: Environments for collaborative ontology mapping, 2012), 4.2 (Nemirovskij & Sicilia, 2013), 4.3 (Wolters, Nemirovski, Pleguezuelos, & Sicilia, 2013) and 3.4 (Sicilia, Deliverable 3.4: Ontology repository with migrated data, 2013), in this document we will not include their detailed description. An extended description will be given only for the overall SEIF architecture and primarily for the SEMANCO federation engine that has not been mentioned in any previous deliverables.

The main target group of the work carried out in Task 4.5 are the developers of the tools in Work Package 5, tools that operate on the data accessible over SEIF. Another target group of this deliverable is the ontology engineers and data experts since they are supposed to design data queries and interrogate the SEIF in the phase when the project results will be applied, e.g. in the time after the three years of the SEMANCO project development will be expired.

1.2 Contribution of partners

The software components of SEIF -such as the SEMANCO federation engine, mapping tools and mappings or the semantic data explorer- have been developed by FUNITEC and HAS. A semi-formal specification of the energy model was developed by POLITO and coded in formal languages (RDFS/OWL) by FUNITEC and HAS. CIMNE, FORUM, UOT and RAMBOLL have participated in the integration of single data sources. These partners also contributed to the development of the energy model and mappings between data. A proof-reading of the final document was done by UoT.

1.3 Relations to other activities in the project

Since the SEIF facilitates tools and users in accessing data distributed in multiple sources, it becomes one of the central technological components of the project SEMANCO. It is difficult to find a work package whose activities would not be related to the semantic framework.

WP3 *Energy data modeling* has facilitated the data structures to be used in the design of the energy model (Task 3.2 *Structuring available data according to energy standards* and Task 3.3 *Structuring contextual data according to standards*) and sources containing data related to the issues of urban planning, CO_2 emission of buildings and their energy consumption (Task 3.1 *Providing access to distributed energy data repositories*). Furthermore, the interconnection of data sources to the ontology repository, i.e. a knowledge base accessible over the federation engine, has been carried out in Task 3.4 *Ontology Repository and Data migration to OWL format.*

As the name of the work package 4 says, all its tasks are dedicated to the development of the SEIF. In the Task 4.1 *Environments for collaborative ontology mapping* has been developed the tools for mapping of relational sources.

Furthermore, the tools developed in WP5 *Integrated tools* and included in the integrated platform use the SEIF endpoint to retrieve data from the data sources. Thus, the data generated by those tools would be included as a new source of data in the SEIF.

2 SEMANTIC ENERGY INFORMATION FRAMEWORK (SEIF)

2.1 SEIF requirements

The main purpose of the SEIF is to facilitate querying of numerous data sources, whereby clients (applications or tools) should be able to submit queries to a single endpoint that in turn would enable high level of interoperability for these clients. In particular such an endpoint should fulfill a common standard for a query language and use its own logic to hide the complexity of particular subtasks of federated querying processing from the clients. These subtasks are:

- Identification of relevant data sources,
- Query rewriting for each single source,
- Execution logic that determines the sequence of query parts and,
- Resolution of dependencies in the data retrieved from different sources (e.g. deleting redundant data records).

For the sake of interoperability, the clients that submit queries are required neither to specify the physical location of targeted data (e.g. an address of a special data source) nor to take into account any information about the access methods or local data schema implemented in a particular source.

Furthermore, the analysis of available data sources carried out in WP3 has shown, on the one hand, that most of them use a relational schema. On the other hand, the idea to take into account open data accessible over the Linked Open Data (LOD) cloud (LinkingOpenData - W3C Wiki, 2013), most of which (68,14%) are accessible over a SPARQL interface (Bizer, Jentzsch, & Cyganiak, State of the LOD Cloud - Version 0.3, 2013) caused the requirement on the SEIF to operate with both data models: the relational one and the RDF model, i.e. subject-predicate-object.

In this context, the first questions that had to be answered were: is there an available technology that fulfills these requirements on federated querying? Also should such technology be developed? The latter option raised another question: are there technologies available that can be reused and combined with each other?

The analysis of related technologies and our conclusions are presented in the following section. The remainder of the document is based on our publication presented in the International Workshop on Description Logics 2013 (Nolle & Nemirovski, 2013).

2.2 Background and related technologies

From our analyses of existing related technologies we have concluded that, there are two basic approaches of data federation which have been developed in recent times: systems for federation of Linked Open Data exposing SPARQL services and ontology-based data access (OBDA).

A survey of the first approach is given in Görlitz & Staab (Görlitz & Staab, Federated Data Management and Query Optimization for Linked Open Data, 2011). These authors identify three main paradigms for the design of a linked data infrastructure: query-based search, peer-to-peer architecture and the federation architecture. Query-based search and peer-to-peer architecture have less relevance for the purpose of our project. Hence we will skip their description. The federation architecture, aims at analyzing and processing a query initially formulated by the client in order to i) identify query parts related to each single distributed

sources, ii) modify the original query in order to adapt it to each source, iii) forward the modified queries to the corresponding sources and iv) federate the query results delivered out by each source. Obviously this behavior corresponds with the requirements initially formulated in the introduction to this section.

In federative systems, data is physically stored in local sources. At the central end point this data is represented by metadata, indices and/or data statistics. These items are used to ensure completeness of query results, on the one hand, and, on the other hand, to increase the efficiency of query processing. Some recent implementations of this approach are FedX (Schwarte, Haase, Hose, Schenkel, & Schmidt, FedX: a federation layer for distributed query processing on linked open data, 2011; Schwarte, Haase, Hose, Schenkel, & Schmidt, FedX: Optimization techniques for federated query processing on linked data, 2011), SemWIQ (Langegger, Wöß, & Blöchl, 2008), DARQ (Quilitz & Leser, 2008), SQUIN (Hartig, Bizer, & Freytag, 2009), UniStore (Karnstedt, Sattler, & Hauswirth, 2012), SPLENDID (Görlitz & Staab, SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions, 2011), ANAPSID (Acosta, Vidal, Lampo, Castillo, & Ruckhaus, 2011), Avalanche (Basca & Bernstein, 2010) or AliBaba (AliBaba, 2013). Apart from these systems, federated queries can be processed by endpoints implementing SPARQL version 1.1 that in comparison to the former version, SPARQL 1.0, has been extended by a construct for specification of federated queries (SPARQL 1.1 Federated Query, 2013). However, SPARQL 1.1 queries contain an explicit specification of the sources locations to be queried. Therefore, the knowledge of the location of data stays in the competency of the client which formulates a query. This fact decreases the interoperability of clients and forces users to think in terms which are unusual in their particular domain and hence may lead to errors. Furthermore, the well-known obstacle of systems listed above, results from the fact that LOD data does not refer to a single but to multiple vocabularies, e.g. DBPedia (DBpedia, 2013), FOAF (The Friend of a Friend (FOAF) project, 2013) or YAGO (YAGO2s: A High-Quality Knowledge Base, 2013), that in common case could have been designed independently and therefore, in theory, may contain inconsistencies and contradictions.

In comparison to the federation architecture, that operates with multiple ontologies, Ontology-based Data Access (OBDA), the second approach mentioned above, aims at creating a single conceptual representation of the domain of discourse in terms of a formal ontology and, in a later stage, mapping this ontology onto the data layer. Semantic queries formulated by clients refer to the ontology and hence, to the native vocabulary of the domain of discourse. In contrast, the data layer typically implements the relational schema. Semantic queries referring to the ontology and formulated in SPARQL are translated in runtime into SQL which is understandable by the data sources comprising the data layer. The OBDA approach has been implemented in platforms like –ontop– (Rodriguez-Muro & Calvanese, High Performance Query Answering over DL-Lite Ontologies, 2012) and MASTRO-I (Calvanese, et al., MASTRO-I: Efficient integration of relational data through DL ontologies, 2007; Calvanese, et al., The MASTRO system for ontology-based data access, 2011). These platforms facilitate highly efficient query evaluation through a syntactically restricted ontology language. They enable reasoning for complete query results with reduced complexity as well as highly optimized techniques for query rewriting/modification. In contrast, other RDB2RDF tools simply focus on the real-time conversion of relational data to RDF. These tools commonly have weak performance in query answering especially for data sets of significant size. To these tools count apart from the most popular D2R server (Bizer & Seaborne, D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs, 2004; Bizer & Cyganiak, D2R Server – Publishing Relational Databases on the Semantic Web, 2006), Virtuoso RDF views (Mapping Relational Data to RDF with Virtuoso's RDF Views, 2013),

Triplify (Auer, Dietzold, Lehmann, Hellmann, & Aumueller, 2009) or Revelytix Spyder (Spyder, 2013). Extended but not complete lists of other systems can be found in (Implementations - RDB2RDF, 2013; Links and Resources, 2013). (Rodriguez-Muro & Calvanese, High Performance Query Answering over DL-Lite Ontologies, 2012; Rodriguez-Muro & Calvanese, Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access, 2012). However, an important restriction of OBDA platforms, as well as of the entire RDB2RDF paradigm, is that they focus on accessing relational data only. These platforms do not take into account data stored in other formats, for example in RDF triple stores. Therefore, they do not allow combining RDF data with data stored in relational sources for query answering. (Rodriguez-Muro & Calvanese, High Performance Query Answering over DL-Lite Ontologies, 2012; Rodriguez-Muro & Calvanese, Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access, 2012; Poggi, et al., 2008; Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, 2007; Artale, Calvanese, Kontchakov, & Zakharyaschev, 2009)

Moreover, in order to achieve complete results on SPARQL queries, not only the explicitly specified data and its relations have to be taken into account but also the knowledge that can be inferred by reasoning on the RDF graph. The definition of such extended interpretation on query evaluation is part of the proposed recommendation of SPARQL 1.1 and is called entailment regimes. (SPARQL 1.1 Entailment Regimes, 2013; Glimm, 2011)

After having analyzed these two main approaches to data federation, we came to the conclusion that the Semantic Energy Information Framework to be developed in this project should take some of the advantageous features of each approach and avoid some of their most significant disadvantages. Similar to the OBDA approach, SEIF should use the conceptual view for the domain of discourse formally specified as ontology. The central ontology, however, should be linked or mapped onto all vocabularies referred by the data stored in each single data source integrated into the federated infrastructure, similarly to the techniques used in the federation platforms for LOD. The usage of the central ontology should facilitate reasoning applied centrally to each query launched by the client. This strategy makes the query results independent of the reasoning (or its absence) provided by the integrated sources.

We have identified six approaches that partially fulfill these requirements:

1) Vidal et al. (Vidal, de Macêdo, Pinheiro, Casanova, & Porto, 2011) uses explicit rule definitions to map elements of the domain specific ontology to the central one. Yet, due to the formalism used for the rule specifications, the application of entailment regimes is not considered fully.
2) Correndo et al. (Correndo, Salvadores, Millard, Glaser, & Shadbolt, 2010) uses RDF to express rewriting rules and enables query rewriting for each single source. This approach does not require a central ontology. Instead it enables translations among ontologies referred in the sources to be queried.
3) LOQUS (Jain, Verma, Yeh, Hitzler, & Sheth, 2010) provides a conceptual view (central ontology) to the distributed sources. LOQUS rewrites the original query into a set of source specific ones by using links and mappings specified in the central ontology, but does not take into account entailment regimes.
4) An enhancement of LOQUS called ALOQUS, introduced by Joshi et al. (Joshi, et al., 2012) facilitates the querying of linked data based on ontology alignments without the need to have detailed information about the data sources. ALOQUS additionally supports an automatic mapping between ontologies to get the alignments but also omits reasoning steps.

5) Li & Heflin (Li & Heflin, Using Reformulation Trees to Optimize Queries over Distributed Heterogeneous Sources, 2010; Li, A Federated Query Answering System for Semantic Web Data., 2013) uses a reasoner to produce results related to a query and therefore ensure completeness. However since the reasoner is located centrally and is invoked at the end of the query execution process all of the data selected has to be entirely loaded in the reasoner.
6) Makris et al. (Makris, Gioldasis, Bikakis, & Christodoulakis, SPARQL Rewriting for Query Mediation over Mapped Ontologies, Tech. rep., 2010; Makris, Gioldasis, Bikakis, & Christodoulakis, Ontology Mapping and SPARQL Rewriting for Querying Federated RDF Data Sources, 2010) employs ontology mappings to rewrite original queries in terms of the target ontologies in order to access federated sources and is therefore the closest one to our approach.

To the best of our knowledge, none of these approaches addresses at the same time the issues of complexity of query answering, on the one hand, and, on the other hand, high efficiency of reasoning. This conclusion led us to develop a new framework which would not have such limitations. An important decision in this concern was to use in SEIF the ideas described by Calvanese et al. (Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, 2007; Artale, Calvanese, Kontchakov, & Zakharyaschev, 2009), who proposed the description logics family DL-Lite with the aim to achieve high reasoning efficiency. They have shown that reasoning tasks for DL-Lite are solvable in PTIME and query answering is in LOGSPACE (more precisely even in AC^0), each in size of the TBox and ABox, respectively. DL-Lite provides important prerequisites for combination of efficient reasoning and query answering with maximum expressiveness of a DL language. In contrast, reasoning on more expressive DLs, such as $\mathcal{SHOIN}$, is in worst-case EXPTIME-hard and query answering co-NP-hard in the size of the ABox (data complexity). Since the OWL 2 QL profile (OWL 2 Web Ontology Language Profiles (Second Edition), 2013) bases upon the DL-Lite family, we decided to use a subset of OWL 2 QL according to $\text{DL-Lite}_{\mathcal{A}}$ specification as the language of the central ontology. By using $\text{DL-Lite}_{\mathcal{A}}$ we are able to integrate OBDA tools like –ontop– and its integrated reasoner Quest (Rodriguez-Muro & Calvanese, High Performance Query Answering over DL-Lite Ontologies, 2012; Quest, 2013; Rodriguez-Muro & Calvanese, Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access, 2012) to ensure efficient reasoning and accessing of relational data sources.

2.3 System architecture of the SEIF

The proposed architecture has been developed to respond to the requirements formulated in Section 2.1 and refined afterwards following the extensive analysis of existing related technologies. Thus, the specifications for the SEIF architecture would be the following: i) efficient processing of conjunctive federated queries, ii) ability for clients to submit queries independent from access methods and data schema implemented in the integrated sources, and iii) ability to federate data stores of two types, RDF triple stores and relational data bases. As stated above, one of the key features of the federative approach is the use of a single central ontology, i.e. energy model described in Deliverable 4.2 (Nemirovskij & Sicilia, 2013) and representing the vocabulary and knowledge structure of the domain of discourse and, at the same time, serving as a mediator for the integrated sources. All queries formulated by clients and targeting data distributed in several integrated sources can only refer to the energy model. The system architecture of the SEMANCO federation engine is outlined in Figure 1.

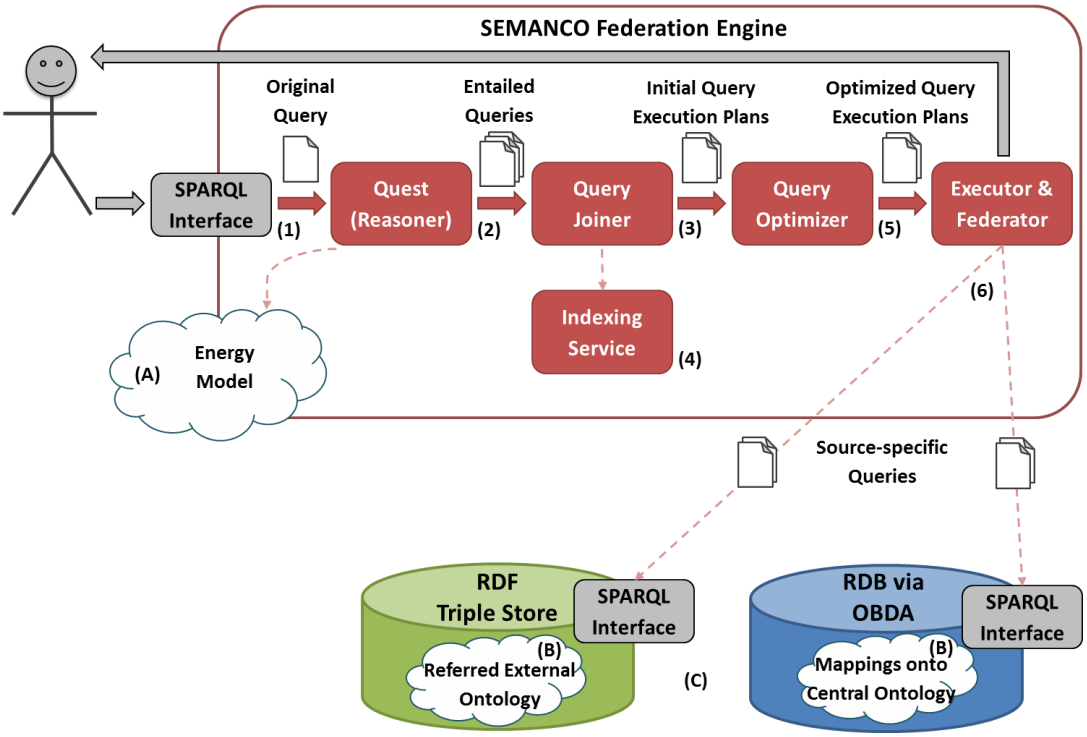


Figure 1. System Architecture of the Semantic Energy Information Framework

The central component of the Semantic Information Framework is the federation engine. It is responsible for the processing of queries submitted by clients, interaction with the integrated data repositories (C – in Figure 1) and federating the data retrieved from these sources to the consistent query answer. The SEMANCO federation engine consists of the following components: SPARQL Interface (1), Quest Reasoner (2), Query Joiner (3), Indexing Service (4), Query Optimizer (5) and Executor & Federator (6). The roles of these components are explained in Section 3 that describes processing of a query. Furthermore, the SEIF provides the location for the storage of the energy model (A – in Figure 1), which is basically a TBox that encompasses apart from native items defined in its own namespaces, i.e. concepts, roles and axioms, also linked or mapped items semantically specified in other ontologies (B – in Figure 1), e.g. those used in the LOD sources.

Integrated sources (C – in Figure 1) can be of two types: triple stores such as Virtuoso (OpenLink Virtuoso Universal Server, 2013) and relational databases. They are integrated by means of SPARQL services for data querying exposed by each source. Relational databases are converted to the RDF format by –ontop– software (–ontop–, 2013). This tool developed at the University of Bolzano is applied locally for each source to map relational data to items of the central ontology. The mapping is done on the fly, i.e. in the time when a query is being processed. For a detailed description of –ontop– see Rodríguez et al. (Rodriguez-Muro & Calvanese, High Performance Query Answering over DL-Lite Ontologies, 2012; Rodriguez-Muro & Calvanese, Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access, 2012; –ontop–, 2013; Quest, 2013).

3 COMPONENTS OF THE FEDERATION ENGINE

3.1 Query processing flow

The query process starts when a query formulated by an external application or by a user is launched at the central SPARQL interface of the SEMANCO federation engine (1 – in Figure 1). Elements of this query, e.g. basic graph patterns (BGP) - which resembles the basic RDF structure (object-predicate-subject) whereby one or more of the items can be a variable - refer to items of the energy model and elements of other vocabularies linked or mapped onto the energy model as described in Section 2.3. In the first step, the initial query is rewritten (2 – in Figure 1) by means of entailment regimes, whereby the semantics of the energy model as well as of the linked external ontologies are taken into account. The application of entailment regimes (described in Section 3.2) results in multiple queries reflecting all possible ways of expressing the same statement using the knowledge derived from the vocabulary that is expressed by the energy model. The disjunction of the evaluation results of these queries by each data source delivers a query result which is complete insofar it encompasses their related vocabularies and the data available from all integrated sources.

Yet, not all BGPs contained in these queries are relevant to all integrated sources. Therefore not all queries, more precisely its patterns generated by application of entailment regimes, can be answered by each source. To avoid sending queries which give no results, all generated queries are assessed by an index look-up procedure (part of 4 – in Figure 1). The query joiner (3 – in Figure 1) applies a look-up procedure for every single BGP to determine the sources which contain data related to each BGP. The index also provides an estimated size of results that would be returned by each data source for each BGP (see Section 3.3 for a detailed description of indexing and index look-up procedure). Using estimated sizes and dependencies among patterns of each query, an optimized query execution plan is generated (5 – in Figure 1). After results to all preselected queries executed by preselected sources according to the previously generated execution plan are retrieved and merged, the federated query answer is returned to the client launched the initial query (6 – in Figure 1).

3.2 Query rewriting

Knowledge implicated in the persistently stored RDF triples can be made explicit (inferred) and taken into account while performing query evaluation. If the implicit knowledge is not processed then there is a high probability that the results of the query will not be complete.

There are three different techniques for knowledge inference using the implemented entailment regime. The first technique is to materialize all of the inferred triples of an ABox. This technique is also called forward-chaining. The alternative technology, called backward-chaining, focuses on query rewriting in runtime. The original query is reformulated using alternative expressions that can be implicated by the TBox. The result of this process is a set of queries asking the same in terms of the semantics but using different expressions. The third option is a hybrid combination of both approaches. These techniques (query rewriting and materialization of inferred triples) are further described by Glimm (Glimm, 2011). Though query evaluation using forward-chaining can be more efficient because e.g. no further reasoning is required in the runtime (Kiryakov & Damova, 2011), it is hardly compatible with the federation approach. Since most of linked (open) data sources grant read-only access to the clients, an external location for storing the inferred triples has to be found. Furthermore, all changes of the original data have to be reflected by the materialized data to keep it up to date. Therefore, for data sources that are often modified, materialization

becomes a challenging task. In contrast to this, backward-chaining is more flexible insofar it takes into account real time data updates and does not require any additional location to store materialized data.

Since the SEIF supports backward chaining, all entailed knowledge is taken into account in the patterns of the rewritten SPARQL queries. Therefore the system does not have to rely on the entailment facilities of single data sources to assure the completeness of the query results. Even if local SPARQL services do not implement entailment regimes at all, that is to say, they do not support any reasoning, the query results - thanks to the entailment regimes based query rewriting done centrally - will be complete in terms of the energy model. Furthermore, even though a SPARQL service of a single source implements entailment regimes, such a source can only infer knowledge based on the vocabularies (ontologies) referred by the corresponding source. If the energy model specifying the domain of discourse is not among these vocabularies, the completeness of query results in terms of the energy model cannot be guaranteed on the basis of only local reasoning.

To implement the query rewriting we have used the source code of the Quest reasoner, which was provided to us by its developers. We have extended the code using a converter that transforms the generated datalog program into appropriated SPARQL queries. Since Quest contains various implementations for query rewriting we have chosen the Tree Witness Rewriter implemented by Kontchakov et al. (Kontchakov, Lutz, Toman, Wolter, & Zakharyaschev, 2011; Kikot, Kontchakov, Podolskii, & Zakharyaschev, 2012; Kikot, Kontchakov, & Zakharyaschev, On (In)Tractability of OBDA with OWL2QL, 2011), which is the most efficient and accurate among all available implementations.

Let us illustrate the query rewriting with an example. Let us assume, for instance, that the energy model (i.e. TBox) contains the following statements (namespace specifications are omitted):

```
ResidentialBuilding ⊑ Building                                          (1)
CommercialBuilding ⊑ Building
∃hasBuilding_Floor_Area ⊑ Building
∃hasBuilding_Floor_Area⁻ ⊑ Building_Floor_Area
∃hasValue ⊑ Building_Floor_Area
Range(hasValue) ≡ rdf:decimal
∃hasResBuildingFloorArea ⊑ ResidentialBuilding
∃hasResBuildingFloorArea⁻ ⊑ ResidentialFloorArea
ResidentialFloorArea ⊑ Building_Floor_Area
hasResBuildingFloorArea ⊑ hasBuilding_Floor_Area
```

The distributed sources $DS_A(2)$ and $DS_B(3)$ contain the following data (ABox):

```
ResidentialBuilding(Hundertwasserhaus)                                  (2)
ResidentialFloorArea(Hundertwasserhaus3356)
:Hundertwasserhaus :hasResBuildingFloorArea :Hundertwasserhaus3356
:Hundertwasserhaus3356 :hasValue 3356.0^^xsd:decimal
:Casa_Mila :hasResBuildingFloorArea :Casa_Mila1000

:Casa_Mila1000 :hasValue 1000^^xsd:decimal
CommercialBuilding(Tanzende_Tuerme)                                     (3)
Building_Floor_Area(Tanzende_Tuerme33357)
:Tanzende_Tuerme :hasBuilding_Floor_Area :Tanzende_Tuerme33357
:Tanzende_Tuerme33357 :hasValue 33357.0^^xsd:decimal
```

To get all buildings the user only has to define the following query:

```
SELECT ?building { ?building a :Building . }                            (4)
```

After query rewriting we get the following datalog program:

```
q(building) :- hasResBuildingFloorArea(building,_)                      (5)
q(building) :- Building(building)
```

```
q(building) :- CommercialBuilding(building)
q(building) :- hasBuilding_Floor_Area(building,_)
q(building) :- ResidentialBuilding(building)
```

This program can be rewritten accordingly as SPARQL queries that have to be federated. To get all floor area values of all buildings the user only has to write the following query:

```
SELECT ?building ?buildingFloorArea {                                    (6)
 ?building :hasBuilding_Floor_Area _:bfa .
 _:bfa :hasValue ?buildingFloorArea . }
```

It is important to notice that in a case when entailment regimes are neither implemented centrally nor locally, it will be the task of the client to formulate the query in a way that assures that all related RDF graph patterns are taken into account. For this purpose, the client has to be aware of all formal vocabulary structures that the query refers to, and the semantics that can be applied to these vocabularies. Consequently, the complexity of queries and of client applications would increase significantly. For instance, the query for selecting all floor area values of all buildings (6) would look like this:

```
SELECT ?building ?buildingFloorArea {                                    (7)
  { ?building :hasBuilding_Floor_Area _:bfa . }
 UNION
  { ?building :hasResBuildingFloorArea _:bfa . }
 _:bfa :hasValue ?buildingFloorArea . }
```

3.2.1 Inference rules implemented in the quest reasoner

In this section we provide the list of formally specified inference rules implemented in the SEMANCO Federation Engine which result from applying the Quest reasoner. These rules are a combination of the basic RDF/RDFS semantics and the semantics implemented by the DL-Lite family. The RDF(S) entailment rules can be also found in (Chekol, Euzenat, Genevès, & Layaïda, 2012).

The following description is provided in (Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, 2007):

Let I be an inclusion assertion that is applicable to the atom g. Then, $gr(g, I)$ is the atom defined as follows:

- If $g = A(x)$ and $I = A_1 \sqsubseteq A$, then $gr(g, I) = A_1(x)$;
- If $g = A(x)$ and $I = \exists P \sqsubseteq A$, then $gr(g, I) = P(x, \_)$;
- If $g = A(x)$ and $I = \exists P^- \sqsubseteq A$, then $gr(g, I) = P(\_, x)$;
- If $g = P(x, \_)$ and $I = A \sqsubseteq \exists P$, then $gr(g, I) = A(x)$;
- If $g = P(x, \_)$ and $I = \exists P_1 \sqsubseteq \exists P$, then $gr(g, I) = P_1(x, \_)$;
- If $g = P(x, \_)$ and $I = \exists P_1^- \sqsubseteq \exists P$, then $gr(g, I) = P_1(\_, x)$;
- If $g = P(\_, x)$ and $I = A \sqsubseteq \exists P^-$, then $gr(g, I) = A(x)$;
- If $g = P(\_, x)$ and $I = \exists P_1 \sqsubseteq \exists P^-$, then $gr(g, I) = P_1(x, \_)$;
- If $g = P(\_, x)$ and $I = \exists P_1^- \sqsubseteq \exists P^-$, then $gr(g, I) = P_1(x, \_)$;
- If $g = P(x_1, x_2)$ and either $I = P_1 \sqsubseteq P$ or $I = P_1^- \sqsubseteq P^-$, then $gr(g, I) = P_1(x_1, x_2)$;
- If $g = P(x_1, x_2)$ and either $I = P_1 \sqsubseteq P^-$ or $P_1^- \sqsubseteq P$, then $gr(g, I) = P_1(x_1, x_2)$.

3.3 Indexing look-up service

Since a query may address complex relations across several data sources, it is not unusual that a single source is only able to answer a part of the query. However, the complete query

cannot be forwarded to a source that is not able to deliver results for each query pattern and hence very probably would return an empty result set. Instead, the SEMANCO federation engine has to identify relations between data sources and query patterns in order to generate sub-queries which can be answered by single sources and forward the sub-queries to relevant sources. To select all sources relevant to each single part of a query an index catalog and an index look-up service is required.

For a survey of the state of the art for indexing methods we refer to the work of Görlitz & Staab (Görlitz & Staab, Federated Data Management and Query Optimization for Linked Open Data, 2011). Here we only describe the indexing approach we have developed for the SEIF. Our index is an extension of the approach known under the name QTree developed by Harth et al. (Harth, et al., 2010). The QTree index is based on the R-Tree data structure (Guttman, 1984) and is able to summarize RDF data in so called Minimum Bounding Boxes (MBBs). MBBs are approximation cubes of three-dimensional points built with the hash values for each triple element (subject, predicate and object). The query result estimation for single SPARQL operations like joins is done by performing the same operation on the MBBs. Afterwards, these results are used to select data sources related to particular BGPs of a query and to determine the sequence of sub-query processing (query execution plans). Yet, the QTree approach has some limitations, for example inaccurate granularity of MBBs, and therefore is only able to process simple queries for small sets of data. Due to this reason QTree has been adapted by Prasser et al. (Prasser, Kemper, & Kuhn, 2012) who have implemented PARTree, a more efficient indexing able to work with complex queries on the basis of large RDF data sources.

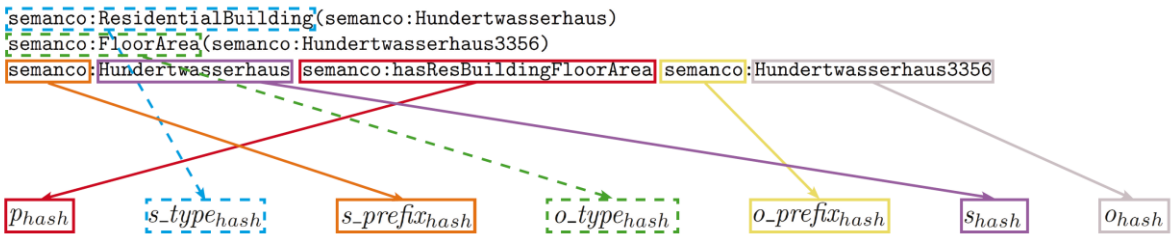


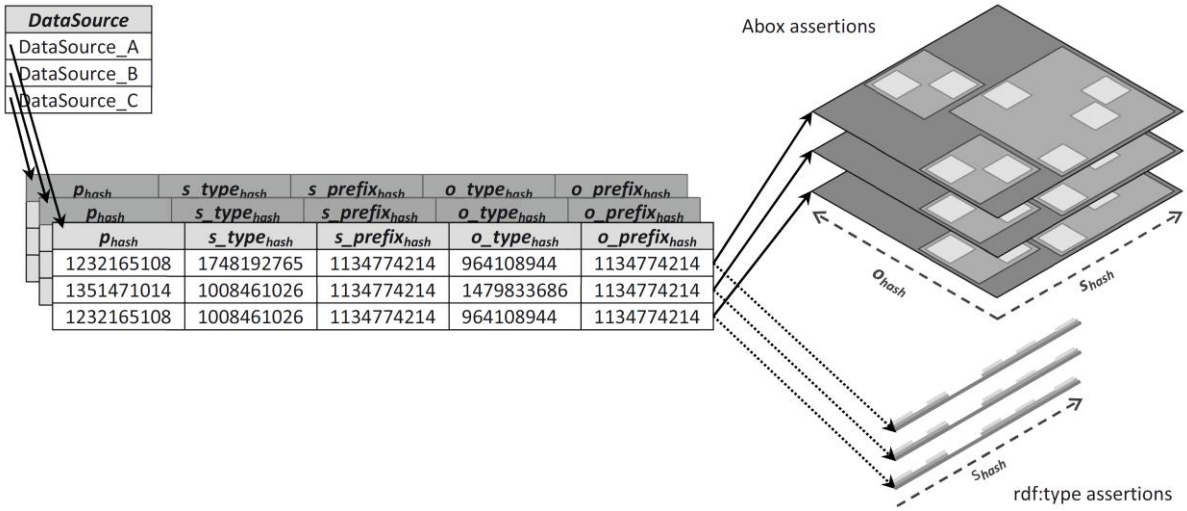
Figure 2. Graphical Representation of the Index Creation

Similar to PARTree, our approach indexes each single data source independently but in contrast to PARTree it works with a more fine-grained partitioning of RDF triples. A systematic representation of the described index architecture is shown in Figure 3. The index partitions are identified by the hash value of the predicate's URI (p_{hash}), the hash value of the prefix - more precisely the namespace - of the subject (s_prefix_{hash}) and object (o_prefix_{hash}) and additionally by the type of the subject (s_type_{hash}) and object (o_type_{hash}). A graphical representation of the index creation is shown in Figure 2. In this context the type of a particular ABox individual is defined as the URI of the TBox concept connected to the individual by the `rdf:type` relation. For the last triple of the assertion below the type of the subject `Hundertwasserhaus` would be `ResidentialBuilding`:

```
ResidentialBuilding(Hundertwasserhaus)                                    (8)
FloorArea(Hundertwasserhaus3356)
:Hundertwasserhaus:hasResBuildingFloorArea:Hundertwasserhaus3356
```

If no type of an individual is available the assigned hash value for s_type_{hash} or o_type_{hash} is 0. Since the `rdf:type` for TBox elements does not exist, this would be also the case for the objects in type assertions like the first two statements of the example above (8). Using this fine-grained segmentation for triple partitions leads to more accurate results as compared to the original QTree as well as to the PARTree and, therefore, to a more exact selection of data source relevant for each BGP being evaluated. For type assertions like, for example, the first

one shown in (8), the described segmentation results in an one-dimensional index structure, because the type URIs of the predicate and the object as well as the namespaces for the subject are unique in each partition.



p_{hash}	s_type_{hash}	s_prefix_{hash}	o_type_{hash}	o_prefix_{hash}
1232165108	1748192765	1134774214	964108944	1134774214
1351471014	1008461026	1134774214	1479833686	1134774214
1232165108	1008461026	1134774214	964108944	1134774214

Figure 3. Systematical Representation of the Index Structure

Invoking the index look-up service for a BGP results in selections of MBBs for each source containing RDF triples fitting to this pattern. For this purpose, the corresponding hash values are calculated for all pattern elements defined by an URI and all found MBBs of each source are returned. For example two BGBs that are connected with the same variable at the object position in the first BGP and at the subject position in the second BGP, estimated results can be calculated by joining the spatial results (MBBs). A graphical representation of this example is given in Figure 4.

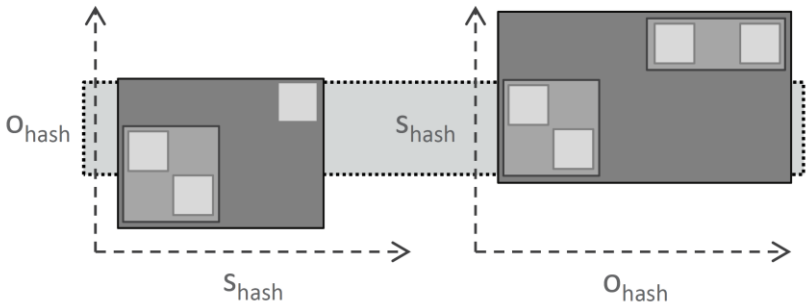


Figure 4. Systematical Representation of a Spatial Join of two BGPs

Besides partitioning, another main difference to PARTree and QTree is that the index of SEIF is not held in main memory but instead stored in a spatial database. Because of this, we benefit from the optimized features of spatial database systems to manage spatial objects, to perform spatial operations like joins and to achieve high scalability of the index. To fill the index with summarized data of each data source we crawl the space of integrated sources by means of SPARQL queries specially constructed for this purpose.

4 REPOSITORY AND MAPPING PROCESS

4.1 SEIF repository and integration of data sources

Data sources being integrated (C – in Figure 1) are required to expose a SPARQL service in order to be able to answer SPARQL queries sent by the federation engine, which in turn receives and rewrites SPARQL queries from urban planning or data mining tools that act as clients.

With these purposes, relational data sources are integrated into the virtual SEIF repository by their mapping onto the SEIF vocabulary (i.e. energy model) as it is described in the Deliverable 3.4 (Sicilia, Deliverable 3.4: Ontology repository with migrated data, 2013). In order to implement the required SPARQL service such a mapping is carried out locally using mapping tools described in Deliverable 4.1 (Sicilia, Deliverable 4.1: Environments for collaborative ontology mapping, 2012). As was stated in this deliverable, the D2R server (Bizer & Cyganiak, 2007) is a key component of the repository architecture. In the meantime, besides the D2R server we also applied the –ontop– framework for selected relational data sources. Both systems are designed to translate SPARQL queries to SQL queries required by the relational databases and return answers in RDF format. In both cases, this is done by transforming the relational data into RDF format. To achieve this purpose both systems apply so-called mappings. However, in case of D2R server these mappings are coded in D2RQ language. In contrast, –ontop– follows the R2RML recommendation issued by W3C (R2RML: RDB to RDF Mapping Language, 2013). In comparison to D2R server the most important advantage of the –ontop– framework is the efficiency of query processing. Thanks to its support for the DL-Lite formalism implemented by the OWL 2 QL profile (OWL 2 Web Ontology Language Profiles (Second Edition), 2013) the –ontop– framework is able to process queries much faster than D2R server.

In contrast to the relational data sources, the integration of RDF triple stores into the SEMANCO federated environment is accomplished centrally (Figure 5). It is enough to interlink concepts of the energy model to the terms of the vocabularies referred by the integrated sources. The interlinking of concepts is made by means of RDFS or OWL properties. For example, by equating the energy model concept `ResidentialBuilding` with an external ontology concept `DwellingHouse` (`ResidentialBuilding owl:sameAs DwellingHouse`), or by subsumption of an external concept `DwellingHouse` (`DwellingHouse rdfs:subClassOf ResidentialBuilding`).

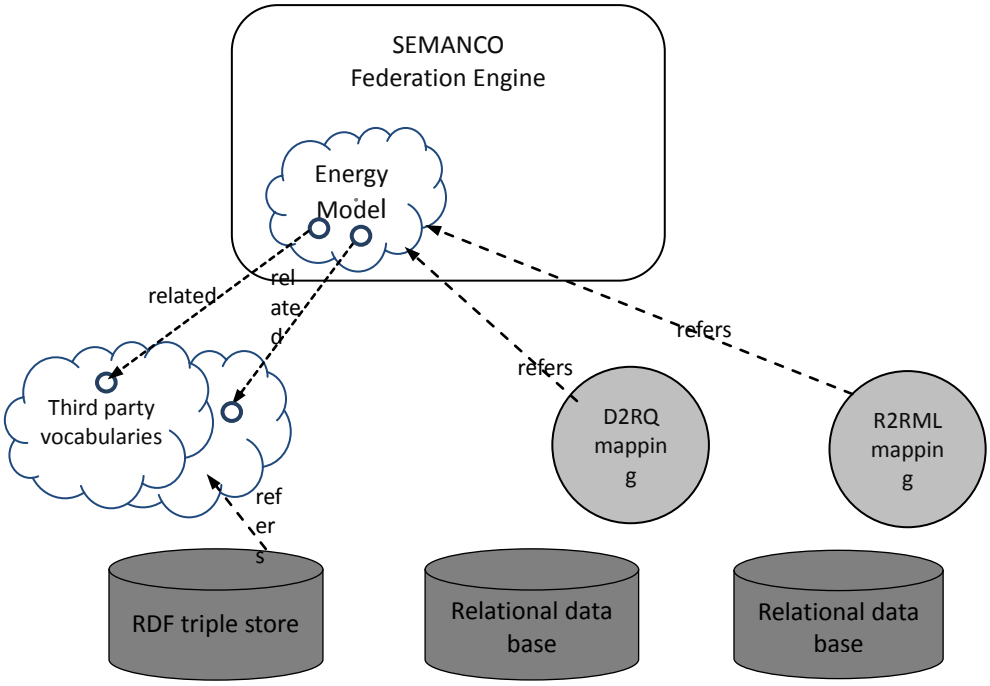


Figure 5. Integration of Data Sources by Interlinking of their Vocabularies

If one of two interlinks mentioned above occurs, then the following query is launched by a tool asking for all individuals of the concept `ResidentialBuilding`

```
SELECT ?building                                                    (9)
 { ?building a ResidentialBuilding }
```

will be supplemented through the federation engine by the following one:

```
SELECT ?building {                                                  (10)
  { ?building a ResidentialBuilding }

 UNION
  { ?building a DwellingHouse }}
```

Sequentially, after the evaluation of both BGPs of query (9) with respect to the index directory, only for the second will BGP relevant sources be identified. Therefore, the second BGP of query (9) will be sent to these data sources but the first BGP won't.

4.2 Requirements on the energy model

Techniques for mapping data source vocabularies onto the terms of the energy model are part of the ontology design methodology described in the Deliverable 4.2 *Design of a semantic energy model* (Nemirovskij & Sicilia, 2013),in Madrazo et al. 2013 (Madrazo, Sicilia, & Nemirovski, 2013) and Nemirovski et al. (Nemirovski, Nolle, Sicilia, Ballarini, & Corado, 2013). We do not consider this methodology as an integral part of SEIF. Rather the requirements on the energy model should be seen as a part of the SEIF. These requirements are:

- Formality: an energy model specification should be an ontology specified using the DL-Lite$_{\mathcal{A}}$ formalism;
- Completeness: an energy model should be complete in terms of the vocabularies of data sources to be federated. This means that the energy model should comprise a union of all terms contained in all vocabularies referred by the sources being federated. If this requirement is not fulfilled then federation querying may still work. However the clients, that is, the urban planning tools, will not be able to query the

complete data stock available in the federated sources;
Though, clients may also use other vocabularies referred by single sources, and the corresponding queries may be executed successfully, the important advantage of the SEIF, to facilitate users to express query in consistent vocabulary of the domain of discourse, will be lost in this case;

- Consistency: an energy model should not contain expressions that contradict with each other or with the logical consequences that can be inferred out of expressions it contains.

In case of SEMANCO the use of the methodology for the ontology design described in Deliverable 4.2 *Design of a semantic energy model* (Nemirovskij & Sicilia, 2013) should be seen as a step towards generating an ontology that fulfills these requirements. Hence, the methodology can be seen as an important complement of the SEIF.

It is important to notice that the requirements mentioned above are formulated for the ontology TBox only, that is, basically the energy model[1]. Such requirements for the (distributed) ABox do not make sense because of two reasons: i) the ABox is changing dynamically, these changes result from real business processes and can hardly be predicted or restricted; ii) "real" data as a rule is both incomplete and inconsistent. Therefore the formulation of requirements listed above would crucially reduce the number of sources that can potentially be integrated. Rather than formulating strict requirements on the distributed data, the federation approach should be flexible enough to facilitate querying of arbitrary incomplete and inconsistent sources. However, in praxis these "features" of the sources often lead to unsatisfying results from users' queries.

[1] In fact the energy model contains a limited number of individuals that are applied centrally. To such individuals for instance belong items required for conversion of units of measures. For example temperature measures can be expressed in Celsius, Fahrenheit or Kelvin units. The values describing their relations to each other are stored centrally (not in the integrated "local" sources but as components of the energy model) as attributes of individuals.

5 CONCLUSIONS

5.1 Contribution to overall picture

The Semantic Energy Information Framework (SEIF) developed in Task 4.5 and described in this deliverable is the main technological component of the WP 4 and a key component of the entire SEMANCO project. It facilitates the tools developed in the WP 5 as well as those developed by third parties launching federated queries addressing data sources integrated into the SEIF repository.

The basic effect of the SEIF application consists in the substantial simplification of the query formulation. SEIF allows the user to abstract away from the particularities of the data access and data structure used by single data sources. Instead she merely has to "tell in SPARQL" to SEIF which data in terms of the energy model she requires. The identification of relevant sources and reformulation of queries to the language and format "understandable" for each source is the task of SEIF. The user / tool retrieves the data in homogeneous format unified units of measure and univocally structured with regard to the energy model. No data transformations are required to process the data according to the business logic of the corresponding task.

On the other hand, SEIF facilitates the access of tools to the complete data stock distributed over all sources integrated into the SEIF repository. In other words, the SEIF repository can be compared to a single data Warehouse.

The integration of data sources into the SEIF repository is done by mapping local sources' schemas onto a central ontology, called the energy model, which has been developed in Task 4.2. When launched, queries formulated in terms of the energy model are rewritten by applying inference rules that belong to so-called entailment regimes. After evaluating basic graph patterns (BGP) - elementary query parts - by means of the index directory, selected BGPs are forwarded to the sources that may contain data relevant for these BGPs.

The interlinking of data sources using the semantics of data and with federated query processing facilitates the interoperability of tools with regard to the data they use to operate. Namely, even though targeted data can be distributed over numerous sources, users of the tools formulate data queries as if they would be sent to a single data source. The queries formulated in standard SPARQL 1.1 query language neither contain technical details, references to the schemas of single sources nor are written in a syntax specific to the access methods used by single sources. Besides being aligned with the SPARQL standard the only requirement on the queries is that they have to use the vocabulary of the energy model.

Several novel approaches have been integrated within the SEIF architecture. They are primarily:

- A federation engine carrying query rewriting basing on the entailment regimes (Section 3.2);
- A look-up service using modified R-tree algorithm (Section 3.3);
- A methodology for the development of the energy model and the integration of data sources (Nemirovskij & Sicilia, 2013).

5.2 Impact on other WPs and tasks

The immediate impact of the work carried out in WP 4, and especially the SEIF presented in this report, is fundamental for the tools developed WP 5 to operate with multiple data sources.

5.3 Contribution to demonstrations

The demonstration of the benefit that the SEIF has for stakeholders is not an easy task. To do so, it is necessary to show how data federated from different sources is used in the business processes, that is, in the assessment and analysis of the energy performance of urban areas. More precisely, it should be shown how difficult would be to aggregate these data without using the SEIF and how poor the analysis, simulation or calculation results would be if the SEIF would not have been used.

A simpler (technical) demonstration would enable to show that tools integrated into the SEMANCO platform can – by means of SEIF – operate with the data obtained from multiple sources. This can be demonstrated for instance using the tools developed in the Task 5.2.

5.4 Other conclusions and lessons learned

We recognize that the work done so far can be only considered as prototyping. The results achieved in particular the developed framework (SEIF) need to be compared to other systems aiming at similar goals (Section 2.2), for instance by means of a benchmark for the efficiency evaluation of query processing.

On the other hand, the development of the SEIF cannot be considered completed yet. Our next steps will be the improvement of our own or adaptation of existing approaches for query execution plan generation, its optimization as well as further improvement of the indexing system. We hope to fulfill the later objective using approximation techniques and including incremental index enhancements.

Furthermore additional efforts will be required for identifying and integrating Linked Open Data sources into the SEIF repository.

6 GLOSSARY

Ontology

According to the definition of Gruber, an ontology is an explicit specification of a conceptualization. It is a set of concepts and their 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.

OWL

The OWL Web Ontology Language is designed to be used by computer applications that need to process the content of information instead of just presenting it to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between them. This representation of terms and their interrelationships is called an ontology. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing an additional vocabulary along with a formal semantics.

Federated Query Processing

Federated Query Processing is a mechanism to carry out aggregations, joins or conjunctions of data distributed over multiple data sources. Usually, a so-called federated query contains a part that can be answered by multiple sources. In a common case, it is not necessary that the sources are explicitly referenced in the query. However, such references are not prohibited. It is a task of a federation engine to analyse a query, identify the locations of relevant data partition or rewrite the query, to forward it to relevant data sources and to merge the query answers generated by these sources into a single answer to the initial query.

Description Logics

Description Logics (DLs) is a family of knowledge representation (KR) formalisms. DLs comprise decidable fragments of first-order logic with attractive and well-understood computational properties. Since KR systems basically aim at answering queries of a user in reasonable time, the DLs-bases reasoning procedures focus on generation of decisions i.e. should always terminate. Whereas investigating the computational complexity of a given DL with decidable inference problems is an important issue.

SPARQL

SPARQL is a computer language used to make queries into databases stored in RDF format. RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources; regardless the data is stored natively as RDF or viewed as RDF via middleware. SPARQL has capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. It also supports extensible value testing and constraining queries by source RDF graph. The

results of SPARQL queries can be results sets or RDF graphs.

Index

An Index is a data structure used for fast identification of physical location of data during a data retrieval process, e.g. querying. In the case of SEMANCO federation engine, such location is specified by an address of a corresponding SPARQL end-point.

An index provides a basis for a quick lookup service whose task is to identify data sources related to a query being processed. If data in the indexed data space is changing then an index has to be updated in order to assure the adequatness of the lookup.

Ontology Mapping

Ontology mapping is a method to find correspondences between concepts from different ontologies. The mapping or matching operation retrieves alignment for two ontologies. An alignment is a set of mapping elements, which its formal description is a 5-uple: *‹id, e, e´, n, R›* where:

- *id* is a unique identifier of the mapping;
- *e* and *e´* are entities that belong to the ontologies (classes, instances, properties…);
- *n* is the confidence measure holding for the correspondence between the entities; and
- *R* is the relation (equivalence, more general, more specific, mismatch, overlap) holding between the entities.

The mapping can be injective or bijective. In an injective mapping the entities of an ontology are mapped to entities of the other ontology. A bijective mapping works in both ways. For example, an entity of ontology A can be expressed in terms of the ontology B and the other way around.

7 REFERENCES

Acosta, M., Vidal, M.-E., Lampo, T., Castillo, J., & Ruckhaus, E. (2011). ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In *The Semantic Web – ISWC 2011, Lecture Notes in Computer Science Volume 7031* (pp. 18-34). Springer-Verlag.

AliBaba. (2013). In *http://www.openrdf.org/alibaba.jsp.* (Retrieved October 10, 2013).

Artale, A., Calvanese, D., Kontchakov, R., & Zakharyaschev, M. (2009). The DL-Lite family and relations. In *Journal of Artificial Intelligence Research 36(1)* (pp. 1-69).

Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., & Aumueller, D. (2009). Triplify: light-weight linked data publication from relational databases. In *Proceedings of the 18th international conference on World wide web* (pp. 621-630). ACM.

Basca, C., & Bernstein, A. (2010). Avalanche: Putting the Spirit of the Web back into Semantic Web Querying. In *Scalable Semantic Web Knowledge Base Systems (SSWS 2010)* (pp. 64-79). CEUR Workshop Proceedings.

Bizer, C., & Cyganiak, R. (2006). D2R Server – Publishing Relational Databases on the Semantic Web. In *5th international Semantic Web conference* (p. 26).

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

Bizer, C., & Seaborne, A. (2004). D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. In *Proceedings of the 3rd international semantic web conference (ISWC2004)* (p. 26).

Bizer, C., Jentzsch, A., & Cyganiak, R. (2013). State of the LOD Cloud - Version 0.3. In *http://lod-cloud.net/state/.* (Retreived October 10, 2013).

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The DL-Lite family. In *Journal of Automated Reasoning* (pp. 385-429). Springer.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., & Rosati, R. (2007). MASTRO-I: Efficient integration of relational data through DL ontologies. In *Proceedings of the 20th International Workshop on Description Logics* (pp. 227-234). CEUR Electronic Workshop Proceedings.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodríguez-Muro, M., . . . Savo, D. F. (2011). The MASTRO system for ontology-based data access. In *Semantic Web, Volume 2, Number 1 / 2011* (pp. 43-53). IOS Press.

Chekol, M. W., Euzenat, J., Genevès, P., & Layaïda, N. (2012). SPARQL query containment under RDFS entailment regime. In *Automated Reasoning* (pp. 134-148). Springer Verlag.

Correndo, G., Salvadores, M., Millard, I., Glaser, H., & Shadbolt, N. (2010). SPARQL query rewriting for implementing data integration over linked data. In *Proceedings of the 2010 EDBT/ICDT Workshops.* New York: ACM.

DBpedia. (2013). In *http://dbpedia.org* . (retrieved October 10, 2013).

Glimm, B. (2011). Using SPARQL with RDFS and OWL entailment. In *Reasoning Web. Semantic Technologies for the Web of Data* (pp. 137-201). Springer Verlag.

Görlitz, O., & Staab, S. (2011). Federated Data Management and Query Optimization for Linked Open Data. In A. Vakali, & L. C. Jain, *New Directions in Web Data Management 1, Studies in Computational Intelligence Volume 331* (pp. 109-137). Berlin / Heidelberg: Springer.

Görlitz, O., & Staab, S. (2011). SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In *Proceedings of the 2nd International Workshop on Consuming Linked Data.* Bonn, Germany.

Guttman, A. (1984). R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (pp. 47-57). ACM.

Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.-U., & Umbrich, J. (2010). Data summaries for on-demand queries over linked data. In *Proceedings of the 19th international conference on World wide web* (pp. 411-420). ACM.

Hartig, O., Bizer, C., & Freytag, J.-C. (2009). Executing SPARQL queries over the web of linked data. In *The Semantic Web - ISWC 2009, Lecture Notes in Computer Science Volume 5823* (pp. 293-309). Springer-Verlag.

Implementations - RDB2RDF. (2013). In *http://www.w3.org/2001/sw/rdb2rdf/wiki/Implementations.*

Jain, P., Verma, K., Yeh, P. Z., Hitzler, P., & Sheth, A. P. (2010). LOQUS: Linked Open Data SPARQL Querying System. In *Tech. rep., Kno. e. sis Center.* Wright State University, Dayton, Ohio.

Joshi, A. K., Jain, P., Hitzler, P., Yeh, P. Z., Verma, K., Sheth, A. P., & Damova, M. (2012). Alignment-Based Querying of Linked Open Data. In *On the Move to Meaningful Internet Systems: OTM 2012, Lecture Notes in Computer Science Volume 7566* (pp. 807-824). Springer-Verlag.

Karnstedt, M., Sattler, K.-U., & Hauswirth, M. (2012). Scalable distributed indexing and query processing over Linked Data. In *Web Semantics: Science, Services and Agents on the World Wide Web* (pp. 3-32). ELSEVIER.

Kikot, S., Kontchakov, R., & Zakharyaschev, M. (2011). On (In)Tractability of OBDA with OWL2QL. In *The 24th Int. Workshop on Description Logics* (pp. 224-234). CEUR-WS.

Kikot, S., Kontchakov, R., Podolskii, V., & Zakharyaschev, M. (2012). Long Rewritings, Short Rewritings. In *The 2012 Int. Workshop on Description Logics* (pp. 235-245). CEUR-WS.org.

Kiryakov, A., & Damova, M. (2011). Storing the Semantic Web: Repositories. In *Handbook of Semantic Web Technologies* (pp. 231-297). Springer.

Kontchakov, R., Lutz, C., Toman, D., Wolter, F., & Zakharyaschev, M. (2011). The Combined Approach to Ontology-Based Data Access. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Three* (pp. 2656-2661). AAAI Press.

Langegger, A., Wöß, W., & Blöchl, M. (2008). A semantic web middleware for virtual data integration on the web. In *The Semantic Web: Research and Applications* (pp. 493–507). Springer-Verlag.

Li, Y. (2013). A Federated Query Answering System for Semantic Web Data. In *Theses and Dissertations.* Lehigh University.

Li, Y., & Heflin, J. (2010). Using Reformulation Trees to Optimize Queries over Distributed Heterogeneous Sources. In *The Semantic Web – ISWC 2010, Lecture Notes in Computer Science Volume 6496* (pp. 502-517). Springer-Verlag.

LinkingOpenData - W3C Wiki. (2013). In *http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.*

Links and Resources. (2013). In *http://d2rq.org/resources.* (Retrieved October 10, 2013).

Madrazo, L., Sicilia, Á., & Nemirovski, G. (2013). Shared Vocabularies to Support the Creation of Energy Urban Systems Models. In *ICT for sustainable places.* Nice, France.

Makris, K., Gioldasis, N., Bikakis, N., & Christodoulakis, S. (2010). Ontology Mapping and SPARQL Rewriting for Querying Federated RDF Data Sources. In *On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science Volume 6427* (pp. 1108-1117). Springer-Verlag.

Makris, K., Gioldasis, N., Bikakis, N., & Christodoulakis, S. (2010). SPARQL Rewriting for Query Mediation over Mapped Ontologies, Tech. rep. In *http://www.music.tuc.gr/reports/SPARQLREWRITING.PDF.* Technical University of Crete.

Mapping Relational Data to RDF with Virtuoso's RDF Views. (2013). In *http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html.* (Retrieved October 10, 2013).

Nemirovski, G., Nolle, A., Sicilia, Á., Ballarini, I., & Corado, V. (2013). Data integration driven ontology design, case study smart city. *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics.*

Nemirovskij, G., & Sicilia, Á. (2013). Deliverable 4.2: Semantic Energy Model. In *SEMANCO*. http://semanco-project.eu/index_htm_files/SEMANCO_D4.2_20130321.pdf. (Retrieved October 10, 2013).

Nolle, A., & Nemirovski, G. (2013). ELITE: An Entailment-based Federated Query Engine for Complete and Transparent Semantic Data Integration. In *Proceedings of the 26th International Workshop on Description Logics.*

–ontop–. (2013). In *http://ontop.inf.unibz.it/.* (Retrieved October 10, 2013).

OpenLink Virtuoso Universal Server. (2013). In *http://virtuoso.openlinksw.com/.* (Retrieved October 10, 2013).

OWL 2 Web Ontology Language Profiles (Second Edition). (2013). In *http://www.w3.org/TR/owl2-profiles.* (Retrieved October 10, 2013).

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. In *Journal on Data Semantics, Volume 10* (pp. 133-173). Springer-Verlag.

Prasser, F., Kemper, A., & Kuhn, K. A. (2012). Efficient distributed query processing for autonomous RDF databases. In *Proceedings of the 15th International Conference on Extending Database Technology* (pp. 372-383). ACM.

Quest. (2013). In *http://ontop.inf.unibz.it/?page_id=7.* (Retrieved October 10, 2013).

Quilitz, B., & Leser, U. (2008). Querying distributed RDF data sources with SPARQL. In *The Semantic Web: Research and Applications* (pp. 524–538). Springer-Verlag.

R2RML: RDB to RDF Mapping Language. (2013). In *http://www.w3.org/TR/r2rml/.* (Retrieved October 10, 2013).

Rodriguez-Muro, M., & Calvanese, D. (2012). High Performance Query Answering over DL-Lite Ontologies. In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning* (pp. 308-318).

Rodriguez-Muro, M., & Calvanese, D. (2012). Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access. In *Proc. of the 9th Int. Workshop on OWL: Experiences and Directions (OWLED 2012).*

Schwarte, A., Haase, P., Hose, K., Schenkel, R., & Schmidt, M. (2011). FedX: a federation layer for distributed query processing on linked open data. In *The Semanic Web: Research and Applications* (pp. 481–486). Springer-Verlag.

Schwarte, A., Haase, P., Hose, K., Schenkel, R., & Schmidt, M. (2011). FedX: Optimization techniques for federated query processing on linked data. In *The Semantic Web–ISWC 2011* (pp. 601–616). Springer-Verlag.

Sicilia, Á. (2012). Deliverable 4.1: Environments for collaborative ontology mapping. In *SEMANCO*. http://semanco-project.eu/index_htm_files/SEMANCO_D4.1_20120629.pdf. (Retrieved October 10, 2013).

Sicilia, Á. (2013). Deliverable 3.4: Ontology repository with migrated data. In *SEMANCO.* http://semanco-project.eu/index_htm_files/SEMANCO_D3.4_20130430.pdf. (Retrieved October 10, 2013).

SPARQL 1.1 Entailment Regimes. (2013). In *http://www.w3.org/TR/sparql11-entailment.* (Retrieved October 10, 2013).

SPARQL 1.1 Federated Query. (2013). In *http://www.w3.org/TR/sparql11-federated-query.* (Retrieved October 10, 2013).

Spyder. (2013). In *http://www.revelytix.com/content/spyder.* (Retrieved October 10, 2013).

The Friend of a Friend (FOAF) project. (2013). In *http://www.foaf-project.org.* (Retrieved October 10, 2013).

Vidal, V. M., de Macêdo, J. A., Pinheiro, J. C., Casanova, M. A., & Porto, F. (2011). Query Processing in a Mediator Based Framework for Linked Data Integration. In *International Journal of Business Data Communications and Networking (IJBDCN)* (pp. 29-47). IGI Global.

Wolters, M., Nemirovski, G., Pleguezuelos, J., & Sicilia, Á. (2013). Deliverable 4.3: User interfaces for domain experts interacting with SEIF. In *SEMANCO.* http://semanco-project.eu/index_htm_files/SEMANCO_D4.3_20130430.pdf. (Retrieved October 10, 2013).

YAGO2s: A High-Quality Knowledge Base. (2013). In *http://mpii.de/yago.* (Retrieved October 10, 2013).