Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology

Mariana Damova, Atanas Kiryakov, Kiril Simov, Svetoslav Petrov

[1] Ontotext AD, Tsarigradsko Chosse 135, Sofia 1784, Bulgaria
{ mariana.damova,naso,kivs,svetoslav.petrov}@ontotext.com

Abstract. Linking Open Data (LOD) facilitates the emergence of a web of linked data by publishing and interlinking open data on the web in RDF. One can explore linked data across servers by following the links in the graph. The LOD cloud has 203 datasets and more than 14 billion RDF triples (http://lod-cloud.net). This paper describes an approach to access these data by means of a single ontology, matched to the schemata describing several of the most common LOD datasets. They are presented in a reason-able view - FactForge (http://factforge.net) - the biggest and most heterogeneous body of factual knowledge on which inference is performed. Techniques of (a) making matching rules with "ontology expressions", (b) adding new instances with inference rules, and (c) extending the upper level ontology with classes and properties are employed. They succeed to align ontologies designed according to different principles and displaying conceptual and structural mismatches.

Keywords: Linked Open Data, FactForge, PROTON, ontology matching, upper level ontology, semantic web, RDF, dataset, DBPedia, Freebase, Geonames.

1 Introduction

Linking Open Data (LOD) initiative [1] aims to facilitate the emergence of a web of *linked data* by means of publishing and interlinking open data on the web in RDF. One can explore linked data across servers by following the links in the graph in a manner similar to the way the HTML web is navigated. LOD cloud's (figure 1) constantly increasing volume has a wealth of information which is of more than 14 billion RDF triples coming from a vast variety of data sources - 203 datasets. They are highly heterogeneous covering different subject domains with contribution from companies, government and public sector projects, as well as from individual Web enthusiasts. Accessing this wealth of data and making use of their full potential is still problematic. Linked data poses issues with respect to different dimensions: (a) open-world assumption of WWW data, combined with high complexity of reasoning even with OWL Lite, (b) some datasets are not suitable for reasoning, (c) publishing OWL datasets without accounting for its formal semantics. Linked data are generally unreliable as no consistency can be guaranteed. They are highly heterogeneous and hard to query. One way of accessing them is by using *reason-able views* [7] - an

approach for reasoning and management of linked data. A *reason-able view (RAV)* is an assembly of independent datasets, which can be used as a single body of knowledge with respect to reasoning and query evaluation. FactForge is such a reason-able view of the web of data.

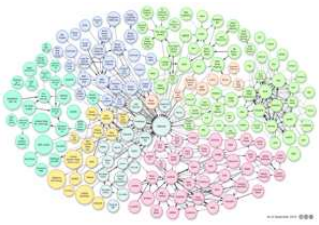

Fig. 1. Linking Open Data cloud (LOD), [9].

It gathers 8 datasets from the LOD cloud - general knowledge (DBPedia, Freebase, UMBEL, CIA World Factbook, MusicBrainz), linguistic knowledge (Wordnet, Lingvoj), geographical knowledge (Geonames). FactForge is the biggest and most heterogeneous body of factual knowledge on which inference has been performed. It comprises an overall of 1.4 billion loaded statements, 2.2 billion stored statements and 10 billion retrievable statements. FactForge is developed as an evaluation case in the European research project LarKC [8] and is used as a testbed for different large scale reasoning experiments like WebPIE [11]. It is available as a free public service at http://factforge.net**,** offering the following access facilities: (a) incremental URI auto-suggest; (b) one-node-at-a-time exploration through Forest and tabulator linked data browsers; (c) RDF Search: retrieve ranked list of URIs by keywords; (d) SPARQL end-point. One can compose SPARQL queries with predicates from multiple datasets, as shown in figure 2.

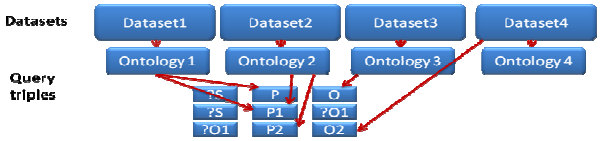

Fig. 2. SPARQL query construction for FactForge datasets.

For example, the query

```
SELECT * WHERE
          {
              ?Person dbp-ont:birthPlace ?BirthPlace ;
                      rdf:type opencyc:Entertainer ;
              ?BirthPlace geo-ont:parentFeature dbpedia:Germany .
          }
```

connects 4 datasets – DBPedia, OpenCyc, Geonames, and RDF. This powerful method to access the data from the LOD cloud has the drawback that one has to be familiar with all schemata and predicates of all datasets in FactForge in order to formulate the queries. It is even more difficult to automate the access to FactForge data and use the SPARQL end point in algorithms because of its heterogeneity. That

is why we envisaged a simplified way to access the data by providing an intermediary layer - a single ontology, as shown in figure 3. To do this, we chose to align the separate schemata of FactForge with the upper-level ontology – PROTON (the Base upper-level ontology (BULO)) [14].

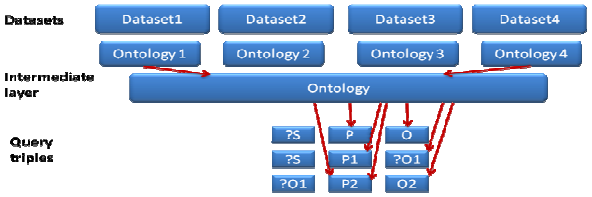

Fig. 3. SPARQL query construction for FactForge datasets in the proposed approach.

The unified access point to FactForge using a single ontology as an interface to connect to all datasets in FactForge is designed to provide an easier and simpler access to the wealth of data, higher degree of interoperability and better integration of the datasets in FactForge. It allows obtaining information from many datasets via one single ontology schema. This unified access point has important applications such as semantic search and annotation using the entities from FactForge, semantic browse and navigation, querying FactForge in natural language, and many others. It should be clear however that the upper-level ontology does not cover the full diversity of the data in the datasets. Still, for specific fine-grained queries the original data schemata and ontologies should be used.

Thus, the main objective of our project was to build a foundational ontology to explore FactForge with a balanced class hierarchy and consistent three to four levels of depth. This implied extending PROTON to obtain optimal coverage of the rich data in FactForge. In addition, the structural and conceptual differences between PROTON and the schemata organizing the datasets of FactForge like DBPedia inspired the introduction of a method for extending FactForge datasets with new instances. So, the matching model of PROTON with FactForge schemata consists in a series of iterations of enrichments at conceptual and at data levels.

2 Approaches to Matching Ontologies

Ontology matching is a key interoperability enabler for the Semantic Web, as well as a useful tactic in some classical data integration tasks. It refers to the activity of finding or discovering relationships or correspondences between entities of different ontologies or ontology modules. Matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate. Distinct methods are employed to perform ontology matching. There are syntactic and semantic matching systems [3]. In the syntactic matching the relations are computed between labels at nodes, and they are evaluated as [0, 1]. In the semantic matching the relations are computed between concepts at nodes, and they are evaluated as set theoretic relations. The semantic matching discovers semantic relationships across distinct and autonomous generic

structures and recognizes relationships between matched entities, such as equivalence, subsumption, disjointness and intersection. When integrating two models, substantial difficulties may arise in transforming information from one model to the other in a heterogeneous context. Harmonising semantics is one approach for model integration by formal mapping between two domains. In this approach reference ontology is built to provide the link between the two models [3]. Except for the types of relationships that are matched between the ontologies, distinctions are made in the way the two initial ontologies are accessed. Thus, there are bidirectional and unidirectional matching methods. The bidirectional method ensures access to the two ontologies from the two ontologies, whereas the unidirectional method ensures access from one to the other ontology only [3]. Another difference in the matching methods is in the way the matching is done. There is manual and automated matching. Automated mapping is suitable for simple ontologies and simple matching tasks, where the exact accuracy of the matching is not of highest importance. In automatic matching structures that are being matched are labeled with natural language typically using WordNet. This is the vocabulary mapping. It consists in comparing Classes, Properties and Instances of two ontologies in a relation one to one. Automated matching competitions are carried out for several years now with tracks on different evaluation parameters [2], [4]. The benchmark track is run on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided. The best result on this track of the 2009 matching competition is F-measure of 80% [4]. Extensive surveys of automated ontology matching methods can be found in [12], [13]. The main drawback of automated ontology matching systems is that they cannot cope with ontological heterogeneity. The fact is ignored that the classes and the properties may be described in different unrelated ontologies, thus the algorithms cannot discover hidden relationships that hold between unrelated entities. Mapping by hand is considered difficult, time consuming and too long, but it derives the most accurate results. Manual mapping is suitable when maximum quality of mapping is seeked for a small quantity of concepts.

Our adopted approach is unidirectional semantic manual alignment of PROTON and the ontologies of the selected datasets of FactForge.

3 The Data

This section describes the data on which the matching in our approach is being performed, e.g. PROTON (the Base upper-level ontology (BULO) [14]) and DBPedia, Freebase and Geonames of FactForge. They are ontologies built according to different design principles. PROTON is built according to the OntoClean method [5], [6] where, for example, type and role are distinguished. It consists in evaluating the ontology concepts according to Meta properties and checking them according to predefined constraints helping to discover taxonomic errors. Using the OntoClean methodology one can discover confusions between concepts and individuals,

confusions in levels of abstraction, e.g. object-level and meta-level, constraints violations, different degrees of generality.

The ontologies of FactForge datasets are made according to different methodologies. The ontologies of DBPedia and Geonames are data-driven. They provide structure and semantics to a large amount of entities in a shallow structure, but are however very different: DBPedia ontology includes many ad hoc predicates which appear in only one or several statements reflecting the variety of knowledge included in it. Geonames ontology has a concise conceptualization organized in very few well structured concepts and instances.

The upper level ontology – PROTON – is one side of the alignment process. An upper ontology is a model of the common objects that are applicable across a wide range of domains. It contains generic concepts that can serve as a domain independent foundation of other more specific ontologies. PROTON is built with a basic subsumption hierarchy comprising about 250 classes and 100 properties which provide coverage of most of the upper-level concepts necessary for semantic annotation, indexing, and retrieval.

DBPedia (http://dbpedia.org) is an RDFized version of Wikipedia. It is a collection of the structured information of Wikipedia, contained in its Infoboxes, represented in RDF and published on the Web. DBPedia ontology counts 24 first level concepts of very different degree of generality ranging from the philosophical concept of "event" through "person" and "place" to very specific concepts like "beverage", "drug", "protein". Not all of DBPedia is comprised in the existing ontology. Many of the properties from the infoboxes are described separately as stand alone properties which pertain to ontological dimensions, but are not modelled in the ontology. Nevertheless some of these concepts are used in our alignment.

Freebase (http://freebase.com) is a large collaborative knowledge base, an online collection of structured data harvested from many sources, including individual wiki contribution. Freebase contains data from Wikipedia, Chemoz, NNDB, MusicBrainz and individually contributed data from its users. It has 5 million topics and no defined ontology. The entities described in this knowledge base are in structured predicate names, which reflect a hidden class hierarchy. Freebase has an overall of 19632 predicates with a structure of the predicate name in which the left most word denotes the subject domain of the property; the middle word denotes a class which is the domain of the property denoted by the last right most word, e.g.

government.legislative_session.date_ended
celebrities.romantic_relationship.end_date

Geonames (http://geonames.org) is a geographic database that covers 6 million of the most significant geographical features on Earth. It contains over 8 million geographical names and consists of 7 million unique features whereof 2.6 million populated places and 2.8 million alternate names. All features are categorized into one out of nine feature classes and further subcategorized into one out of 645 feature codes. Geonames is integrating geographical data such as names of places in various

languages, elevation, population and others from various sources. All lat/long coordinates are in WGS84 (World Geodetic System 1984).

4 The Methodology

The project of building an intermediary layer between the heterogeneous data of FactForge and the end user requires matching of ontologies built according to different methods, e.g. data-driven ontologies and an upper-level ontology. This implies a translation from the one method to the other method. Further, the heterogeneity of the data in FactForge prompts the building of a unidirectional matching scheme, e.g. making FactForge accessible through PROTON predicates and entities, but not vice versa - PROTON through FactForge predicates and entities. The alignment was performed manually as the most suitable approach to find the correspondences of the small amount of upper-level concepts.

Our approach summarizes a method of matching ontologies with different methodological background – data-driven ontologies and an upper level ontology. The upper level ontology (PROTON) was chosen to be the basis for the mapping decisions, e.g. the representations of the other ontologies were translated into its model by (a) making matching rules with "ontology expressions", (b) adding new instances with inference rules, and (c) extending the upper level ontology with classes and properties.

Thus, the adopted matching method includes:

- mapping of the concepts from PROTON to the concepts described in the datasets of FactForge, more precisely DBPedia, Freebase, Geonames
- assigning subsumption relations between entities and properties from FactForge to PROTON
- extending PROTON with classes and properties to obtain mapping at a conceptual level with FactForge
- using OWL class and property construction capabilities to represent classes and properties from FactForge and map them to PROTON classes
- extending FactForge with instances to account for the conceptual representations of the matching

The matching of the concepts and properties between DBPedia and PROTON and between Geonames and PROTON took place based on comparing the definitions of the concepts and their use. Respecting the commitment for unidirectional matching we have designed the rules with subsumption relations from FactForge to PROTON, as shown in the example below:

```
(a)  dbp:Place
            rdfs:subClassOf ptop:Location .

(b)  geo-ont:parentFeature
            rdfs:subPropertyOf ptop:subRegionOf .
```

But first, the upper level ontology PROTON was extended with new classes and properties. This was done after analyzing the content of the available data in DBPedia and Geonames with a result - a list of classes and properties which are represented within the data, and analyzing the structure of the current version of PROTON with respect to the new classes and properties. We obtained a classification of the new classes and properties using inheritance from already existing classes to the new ones. We have also used properties assigned to the new classes in order to structure them in a better way. Thus, we built a new version of PROTON with more classes and properties. Adding a new class or a new property in PROTON followed specific. A new class was added when the instances in FactForge formed a distinguishable group for which there was no concept description in PROTON. For example, DBPedia has instances for Fictional Characters, like Harry Potter, which are classified as Persons, the class FictionalCharacter was introduced in PROTON as a subclass of Person. A generic criterion for adding a new class to PROTON is the compliance with the principle of completeness of the ontology. This happens when for a given concept there are subconcepts represented in the ontology, but siblings of these concepts are missing. For example, if car and bicycle are subclasses of vehicle, but motorcycle is not, then we add motorcycle into the ontology.

To match Freebase predicates to PROTON the class construction capabilities of OWL have been used, to bind Freebase properties into classes and then match them to PROTON concepts as shown in the example (c) below:

```
(c)pfb:Location
              rdf:type owl:Restriction ;
              owl:onProperty
       <http://rdf.freebase.com/ns/type.object.type> ;
              owl:hasValue
      <http://rdf.freebase.com/ns/location.location> ;
              rdfs:subClassOf ptop:Location .
```

Here a class pfb:Location is created which is restricted to a Freebase type Location.

Another aligning method used is expression mapping. It consists in construction of classes on the basis of one of the ontologies, and mapping them to classes, or expressions of the other ontology, satisfying a relation of type many to many. For example, PROTON has a class Person and a class Profession. The subclasses of Person are Man and Woman and the subclasses of Profession are different professions, e.g. Architect, Teacher, etc. In DBPedia, Person is represented with the profession he exercises. Architect is a subclass of the class Person. Here we see a structural and conceptual difference between the PROTON model and the DBPedia

model with this respect. To perform the alignment we have adapted the DBPedia model to PROTON's model in the mapping rule, as shown in figure 3.

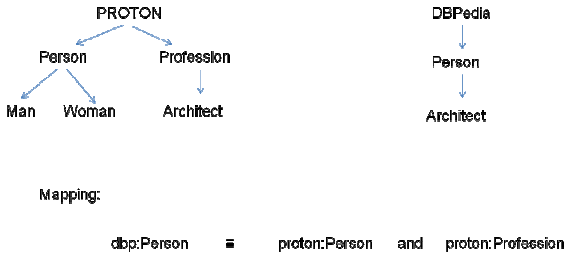

Fig. 3. Mapping of concepts in ontologies designed according to different principles (PROTON, DBPedia).

Technically, the mapping rule looks like this:

```
(d)   dbp:Architect
            rdfs:subClassOf
                      [ rdf:type owl:Restriction ;
                                owl:onProperty
                                        pupp:hasProfession ;
                                owl:hasValue p-ext:Architect
                      ] .
```

The professions are modeled as instances of the class Profession in PROTON, and the single entity of DBPedia is matched to an expression in PROTON which restricts the property hasProfession to the value of the profession of interest.

The method of expression matching is not universally applicable as described above. In some cases the expressions require a reference to instances which are not included in the datasets of FactForge. This triggered the next adopted aligning method - extending the dataset of FactForge with the necessary instances, ensuring their availability to cover the entire model of the chosen basic ontology - PROTON.

FactForge is loaded into BigOWLIM, the most scalable OWL engine (http://www.ontotext.com/owlim/) supporting light-weight and high-performance reasoning with inference based on OWL Horst. BigOwlim allows the definition of custom semantics via special rules and axiomatic triples which are exploited in the process of full materialisation performed during loading. This last mechanism was used to extend FactForge with new instances by adding inference rules to the built-in ruleset. The inference rules provide the insights on what triples have to be added into the repository. They are resolved at the time of loading of the datasets into the semantic repository. For example, the inference rule (e) below:

```
(e)    p  <rdf:type> <dbp-ont:PrimeMinister>
      ----------------------------------------
```

```
p <ptop:hasPosition> j
j <pupp:hasTitle> <p-ext:PrimeMinister>
```

translates the DBPedia representation of someone holding a position of a Prime minister into PROTON representation. In DBPedia this is done with a type relation, whereas in PROTON this is a complex relation between a person holding a position with the title of Prime minister.

The translation of a single type relation in DBPedia can require more complexe representations, such as the ones given in example (f). Here the Freebase predicate `government.us_president` is represented as a person who holds a position in the US with the title president.

```
(f)
  a  <fb:type.object.type> <fb:government.us_president>
  ----------------------------------------------------
  a <rdf:type> <ptop:Person>
  a <ptop:hasPosition> y
  y <ptop:withinOrganization> <dbpedia:United_States>
  y <pupp:hasTitle> <p-ext:President>
```

Except for making the process of querying heterogeneous datasets easier, using one upper level ontology as an entry point to such data has another advantage. It allows to obtain information from many datasets via one single query. For example, one PROTON predicate covers three data driven predicates, e.g. PROTON `locatedIn` takes Freebase `time.event.locations`, and DBPedia `place` and `location`, as shown in the example (g) below.

```
(g)
dbp:place
    rdfs:subPropertyOf ptop:locatedIn .

dbp-prop:location
    rdfs:subPropertyOf ptop:locatedIn .

<http://rdf.freebase.com/ns/time.event.locations>
    rdfs:subPropertyOf ptop:locatedIn .
```

This makes the exploration of FactForge richer and simpler, as a query with the single PROTON predicate will retrieve information with the three other predicates from the two different datasets.

5 Results and Statistics

The outcomes of this work can be summarized as follows: (1) a new layer of unified semantic knowledge over FactForge was created by matching PROTON to FactForge schemata (2) we produced an original approach to providing similar layers to other datasets; (3) and developed a new version of PROTON ontology, which will be used in other projects. The extension of PROTON was governed by two main principles: (1) to provide coverage for the available data; and (2) to reflect the best approaches in the design of ontologies such as OntoClean methodology [5]. Table 1 shows statistics about the datasets of FactForge before and after the matching rules have been added to the semantic repository with full materialization performed. The alignment brought close to 800 million more statements and 50 million new entities available for exploration, while the matching rules cover 554 mapped classes and 103 mapped properties. The biggest number of mapped classes comes from the mapping of PROTON to Geonames' feature codes (368). As far as PROTON enrichment is concerned, 166 new classes and 73 new properties have been introduced. They cover the classes which were identified during the analysis of the instance data in FactForge and their ontologies as described in section 4.

	FactForge Initial State	FactForge with Alignments	Difference
Number of Statements	1,782,541,506	2,630,453,334	847,911,828
Number of ExplicitStatements	1,143,317,531	1,942,349,578	799,032,047
Number of Entities	354,635,159	404,798,593	50,163,434

Table 1. Statistics of FactForge

The adopted method was tested on 27 evaluation SPARQL queries selected to cover different domains, e.g. public administration, military conflicts, art and entertainment, business, medicine and to use multiple datasets from FactForge. Table 2 presents an example of an evaluation query. It is about cities around the world which have "Modigliani art works". This query is considered the ultimate test for the Semantic Web [10]. To our knowledge FactForge is the only engine capable of passing this test. The right column of the table gives the query written with PROTON predicates only. It is simpler and more intuitive than the FactForge standard one as the mapping has put all FactForge location predicates into one PROTON predicate. The number of results returned with PROTON query and with FactForge standard query are the same, presented in a slightly different way. This proves the validity of the approach.

FactForge – Standard

```
PREFIX fb: <http://rdf.freebase.com/ns/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbp-prop: <http://dbpedia.org/property/>
PREFIX dbp-ont: <http://dbpedia.org/ontology/>
PREFIX umbel-sc: <http://umbel.org/umbel/sc/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ot: <http://www.ontotext.com/>

SELECT DISTINCT ?painting_l ?owner_l ?city_fb_con ?city_db_loc
?city_db_cit

WHERE {
        ?p fb:visual_art.artwork.artist
                              dbpedia:Amedeo_Modigliani ;
           fb:visual_art.artwork.owners [
              fb:visual_art.artwork_owner_relationship.owner
                                                  ?ow ] ;
           ot:preferredLabel ?painting_l.
        ?ow ot:preferredLabel ?owner_l .
```

FactForge - PROTON

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ot: <http://www.ontotext.com/>
PREFIX ptop: <http://proton.semanticweb.org/protont#>
PREFIX ploc: <http://proton.semanticweb.org/protonl#>
PREFIX p-ext: <http://proton.semanticweb.org/protonue#>

SELECT DISTINCT ?painting ?owner ?city

WHERE {
        ?p p-ext:author dbpedia:Amedeo_Modigliani ;
           p-ext:ownership [ ptop:isOwnedBy ?ow ] ;
           ot:preferredLabel ?painting .
        ?ow ot:preferredLabel ?owner .
        ?ow ptop:locatedIn [  rdf:type ploc:City ;
                              ot:preferredLabel ?city].
    }
```

```
        OPTIONAL { ?ow fb:location.location.containedby
                   [ ot:preferredLabel ?city_fb_con ] }
         OPTIONAL { ?ow dbp-prop:location ?loc.
                    ?loc rdf:type umbel-sc:City ;
                         ot:preferredLabel ?city_db_loc }
         OPTIONAL { ?ow dbp-ont:city [ ot:preferredLabel
                                       ?city_db_cit ] }
}
```

Table 2. Modigliani Test Query

In cases where several FactForge predicates are matched to a single PROTON predicate, like the location predicates mentioned earlier in the paper, the PROTON queries return more results than FactForge – Standard queries. Thus, the advantages of the approach to have a single access point to the Linked Open Data (LOD) cloud are twofold: they provide access by simpler queries and they provide leveraged query results.

5 Future work

We envision in the future building a two level intermediary layer to access FactForge and then LOD cloud mapping PROTON to UMBEL (http://www.umbel.org/documentation.html) – "a lightweight subject concept reference structure for the Web" with about 20 000 subject concepts based on OpenCyc (http://www.cyc.com/opencyc/). We intend to cover more datasets from the LOD cloud, and to experiment with the balance between the data from the LOD and FactForge datasets and the ontological schemata describing them.

References

1. Christian Bizer, Tom Heath, Tim Berners-Lee. Linked Data – The Story so Far. In: Heath, T., Hepp, M. and Bizer, C. (eds.) Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS), http://linkeddata.org/docs/ijswis-special-issue, (2009).
2. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondˇrej Šváb-Zamazal, and Vojtˇech Sváte, Results of the Ontology Alignment Evaluation Initiative 2008, (2008).
3. Mohamed El-Mekawy and Anders Östman. Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CITYGML. In Proceedings of the 3rd ISDE Digital Earth Summit 12-14 June, 2010, Nessebar, Bulgaria, (2010).
4. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Šváb-Zamazal, V. Svátek, C. Trojahn dos Santos, G. Vouros, S. Wang: Results of the Ontology Alignment Evaluation Initiative 2009. In Proceedings of the 4th Ontology Matching Workshop at ISWC, (2009).
5. Nicola Guarino and Christopher Welty. "Evaluating Ontological Decisions with OntoClean." Communications of the ACM, 45(2): 61-65 (2002).

6. Michael Grüninger and Mark Fox. Methodology for the Design and Evaluation of Ontologies. Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing (1995).
7. Atanas Kiryakov, Damyan Ognyanoff, Ruslan Velkov, Zdravko Tashev, Ivan Peikov. LDSR: Materialized Reason-able View to the Web of Linked Data. In: Proceedings of OWLED 2009. Chantilly, USA, 23-24 October 2009 (2009).
8. LarKC project. The Large Knowledge Collider - http://www.larkc.eu/, (2010).
9. Linking Open Cloud Diagramme, as of September 2010, http://lod-cloud.net/.
10. Richard MacManus. The Modigliani Test: The Semantic Web's Tipping Point. http://www.readwriteweb.com/archives/the_modigliani_test_semantic_web_tipping_point.php (2010).
11. Vassil Momchev, Matthias Assel, Alexey Cheptsov, Barry Bishop, Luka Bradesko, Christoph Fuchs, Georgina Gallizo, Spyros Kotoulas, Gaston Tagni D5.5.3 Report on platform validation and recommendation for next version, LarKC EU-IST-2008-215535, (2010).
12. P. Shvaiko, J. Euzenat: Ten Challenges for Ontology Matching. In Proceedings of ODBASE, (2008).
13. P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005.
14. Ivan Terziev, Atanas Kiryakov, Dimitar Manov. „D.1.8.1 Base upper-level ontology (BULO) Guidance" Deliverable of EU-IST Project IST – 2003 – 506826 SEKT (2005).