RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable 1.2.1

Initial data integration

Editor:	Maurice Grinberg, Ontotext
Author(s):	Maurice Grinberg, Ontotext; Mariana Damova, Ontotext; Atanas Kiryakov, Ontotext
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)[1]	Public (PU)
Contractual Delivery Date:	March 2011
Actual Delivery Date:	March 2011
Suggested Readers:	Research staff working on the data collection and management; developers working on use cases.
Version:	1.2

[1] Please indicate the dissemination level using one of the following codes:
• **PU =** Public • **PP =** Restricted to other programme 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) • **Restreint UE =** Classified with the classification level "Restreint UE" according to Commission Decision 2001/844 and amendments • **Confidentiel UE =** Classified with the mention of the classification level "Confidentiel UE" according to Commission Decision 2001/844 and amendments • **Secret UE =** Classified with the mention of the classification level "Secret UE" according to Commission Decision 2001/844 and amendments

Disclaimer

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP1 Data collection and management
Document Title:	D1.2.1 - Initial data integration
Editor (Name, Affiliation)	Maurice Grinberg, Ontotext AD
Work package Leader (Name, affiliation)	Atanas Kiryakov, Ontotext AD
Estimation of PM spent on the deliverable:	5.9

Executive Summary

The work within WP1 (M1-M6) started with building an initial version of the data management concept for the project, identification of the various data types to be managed within the project (news, blogs and collaborative wiki publications, tweets, enterprise communication, ontologies, linked data and other factual knowledge), including their structure and dynamics (e.g. update in time). A questionnaire was prepared to make the elicitation of requirements for data collection and management more concrete and precise. More details, at this stage of data requirement collection, can be found in D1.3.1 and will be commented in an informal deliverable in M12 when more data are gathered.

This deliverable provides brief descriptions of the Ontotext work on implementing a working method for data integration. This method is partly based on the Reference Knowledge Stack (RKS) which includes PROTON, UMBEL, OpenCyc, and FactForge. So, the ontology alignment work needed for the implementation of RKS is briefly presented together with the details of its implementation. This is the case with the mappings from PROTON to UMBEL and to the specific schemata and ontologies of DBPedia, Geonames and Freebase.

This deliverable begins with presenting the idea of the Reference Knowledge Stack (RKS), its advantages and basic building components. RKS is seen as a vehicle to navigate the wealth of integrated data from different sources, which are interlinked and fully materialized. The Reference Knowledge Stack (RKS) is described as diversity tolerant, and facilitating the access to the data by providing easier ways of query formulation.

Further, the single building components of the RKS are outlined in sequence. FactForge, as a reason-able view of a subset of the Linking Open Data cloud is presented as the major source of data. Intermediary layers of schemata, upper-level ontologies ensure their interconnectedness. This is done with a series of mappings at schema level between the upper-level ontologies PROTON and UMBEL and the ontologies of the datasets integrated in FactForge, such as DBPedia. The methods of mapping PROTON to DBPedia, Geonames and Freebase and the methods of mapping PROTON to UMBEL and UMBEL to DBPedia are described in detail. This is an example of mapping ontologies of different kinds. Except for schema level mapping, the work presented in this deliverable concerns mapping at instance level. More precisely, this is the linkage between UMBEL reference concepts and Wikipedia articles via its categories, which is also part of the description of the mapping methods presented in this deliverable. The process of mapping is bound to extension of the underlying ontologies. Quantitative information about the number of mapping rules and the resulting size of the ontologies complete the description of the mapping methods. The interlinked ontologies present the core of the RKS.

The issue of preserving diversity is discussed at length emphasizing the point that the usage of reference data does not impact the diversity of the data, but rather facilitates their access and optimizes their usability by merely ensuring pathways to their interconnectedness.

The RKS is implemented in a major update of FactForge. This upgraded service allows for formulating complex queries using only PROTON or only UMBEL predicates and obtaining results from a variety of datasets, such as DBPedia, Freebase, OpenCyc, New York Times, etc. for a single query. Each entity from the obtained results can be further explored through the forest of structured and interlinked RDF data.

Table of Contents

Abbreviations

LOD the Linking Open Data project is a W3C SWEO Community project and is an initiative for publishing “linked data”(http://linkeddata.org/); more details are provided in Section 1.1.

RDF Resource description framework, a basic specification determining the data model of the Semantic Web (http://www.w3.org/RDF/).

SPARQL a query language for RDF (http://www.w3.org/TR/rdf-sparql-query/).

Definitions

This deliverable assumes prior knowledge of the basic semantic web standards, namely RDF (http://www.w3.org/RDF/), RDFS (http://www.w3.org/TR/rdf-schema/), and OWL (http://www.w3.org/TR/owl-features/). More detailed descriptions of the ontologies and datasets listed bellow have been given in deliverable D1.1.1, [8].

DBPedia	RDF dataset derived from Wikipedia, aiming to provide as complete as possible coverage of the factual knowledge that can be extracted with high precision from there. DBPedia is one of the most central LOD datasets (http://dbpedia.org).
Linked data	Linked data represents a set of principles for publishing of structured data they can be explored and navigated in a manner analogous to the HTML WWW. The linked data concept is an enabling factor for the realization of the Semantic Web as a global web of structured data around the Linking Open Data initiative. The notion of "linked data" is defined by Tim Berners-Lee in (http://www.w3.org/DesignIssues/LinkedData.html) and prescribes that data should be published on the WWW as RDF graphs. It is viewed as a method for sharing and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF;
OpenCyc	OpenCyc is the open source version of the Cyc technology, the world's largest and most complete general knowledge base and commonsense reasoning engine;
PROTON	An upper-level schema ontology which defines about 542 classes and 183 properties relevant for entity classification, description and relation across multiple domains;
Reason-able view	Reason-able views represent an approach for reasoning and management of linked data. It can be obtained by grouping selected datasets and ontologies in a compound dataset, clean-up, post-processing and enriching the datasets if necessary for each new version of the dataset. The compound dataset is loaded in a single semantic repository.
Reference master data	Reference data shared over a number of systems, [24], [25].
Reference Knowledge Stack	a data organisation approach combining several types of ontologies and datasets, that can be used together as master reference data. (See section 1.2 for further details.)
UMBEL	an upper-level ontology defining about 28 thousand concepts (classes and predicates) derived from OpenCyc and clustered into 33 Super Types, [28]. UMBEL is equipped also with a mapping of the DBPedia entities with respect to the classes from OpenCyc. Description is provided in section 3.

1 Introduction

The work within WP1 started with building an initial version of the data management concept for the project, which was required to structure the activities on initial data collection. Some initial data collection took place, reported in D1.1.1, [8]. The central result was the proposal of a concept for data organisation and integration, which allows for easy management and access to the data, while at the same time respecting the diversity and the dynamicity of the different types and pieces of data relevant to the project and its use cases. A data organisation approach was put forward based on the so-called Reference Knowledge Stack (RKS), [8], including: PROTON, UMBEL, OpenCyc and FactForge.

A semantic annotation prototype for analysis of unstructured content in English and its annotation based on the RKS, was planned to be reported here, but it is related to work in progress in WP2 and will be reported on in deliverable D1.2.2.

In the present deliverable, the actual implementation of RKS, which has been completed during the period M03-M06, is described in some detail. The mapping between PROTON and Linking Open Data (LOD), and more specifically PROTON to UMBEL, are the core of RKS and are presented in (Section 2). Section 3 describes the updated version of UMBEL – 1.0 (mapping Wikipedia and DBPedia to UMBEL and UMBEL to PROTON).

The remainder of this section provides a brief description of RKS for self-sufficiency of this deliverable together with a discussion of its capabilities and limitations of accounting for the diversity of the data which can be seen as a fundament to the data organisation within the project.

1.1 Linking Open Data

LOD is a W3C SWEO community project which provides sets of referenceable, semantically interlinked resources with defined meaning, [2]. The central dataset of the LOD is DBPedia, [14]. Because of the many mappings between other LOD datasets and DBPedia, the latter serves as a sort of hub in the LOD graph assuring a certain level of connectivity. Currently, DBPedia version 3.6 consists of about 672 million explicit statements, of which 286 million are extracted from the English version of Wikipedia. It contains descriptions of 3.5 million things, of which 1.7 million are classified in the DBPedia's own ontology. DBpedia includes also 6.5 million links to external datasets, which means almost 2 links per described thing.

LOD is rapidly growing – as of September 2010 it contains more than 200 datasets, with total volume above 25 billion statements, interlinked with 395 million statements as illustrated on Figure 1 (the figure presents an older picture of the dataset map as the new one is too detailed to be readable in this format).

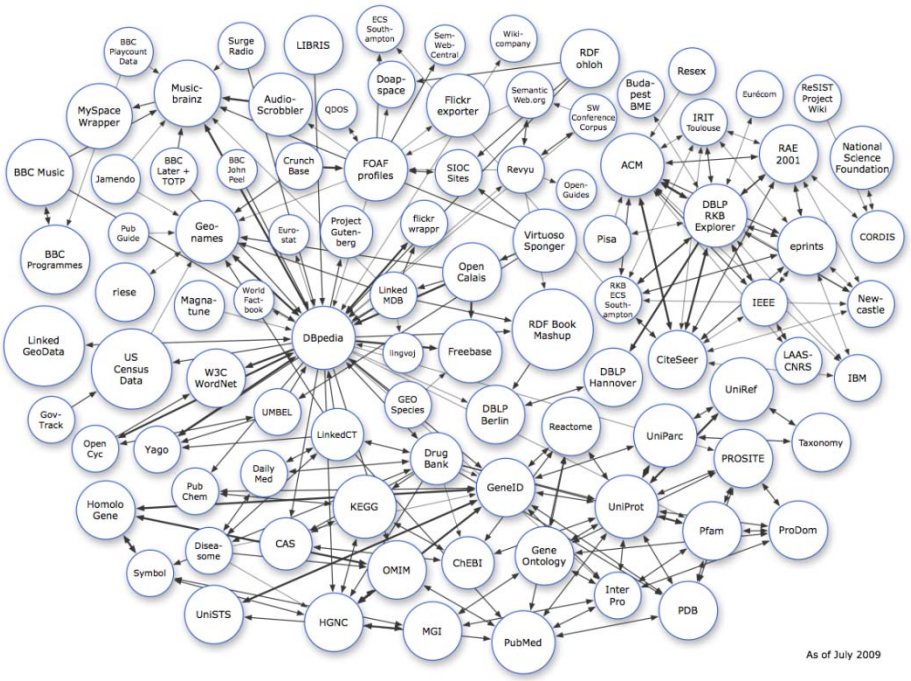

Figure 1. Map of the Datasets in Linking Open Data (LOD) Project, [11].

It should be noted that most of the factual knowledge, in terms of attribute-value pairs and relations between objects in DBPedia, is derived from the so-called Info-boxes of Wikipedia. As a result, DBPedia is characteristic for the fact that about 100 thousand different properties (attribute or relation types) are used with little degree of reuse and modelling consent. Often one and the same type of relationship (e.g. employment) is represented in tens or hundreds of different ways, which makes querying DBPedia with structured query languages such as SPARQL, very challenging and unrewarding exercise. For instance, two predicates “employer” belong to two classifications – one of DBPedia ontology, and the other of DBPedia properties; a number of professions and positions are DBPedia ontology predicates linking “companies” or “artefacts” to “people” or “companies,” e.g. “engineer,” “executiveProducer,” “designer,” “developer,” “manager,” etc. This can be leveraged by mapping DBPedia predicates to a well-structured upper-level ontology as advocated in D1.1.1, [8]. While making SPARQL queries much more efficient and concise, this approach does not affect the diversity of data per se. These questions are discussed in some detail bellow.

1.2 Reference Knowledge Stack

As discussed at length in D1.1.1, [8], mapping the schemata of several datasets to a small upper-level ontology considerably facilitates multiple retrieval scenarios. On the other hand, the data management needs within RENDER require a more comprehensive approach that considers both schema- and instance-level reference structures. It also requires higher granularity which would preserve the diversity of the data, i.e. more extensive reference ontologies and datasets. Thus a trade-off is needed, which would provide a solution for efficient data access via a relatively small upper-level ontology and would ensure minimal loss of diversity specific information.

To meet the requirements for query efficiency and higher accessibility of the data, Ontotext, in cooperation with Structured Dynamics LLC developed the concept for the so-called “Reference Knowledge Stack” (RKS), which includes:

- PROTON – an upper-level ontology, 542 entity classes and 183 properties;
- UMBEL – 28,000 concepts extracted from OpenCyc and mapped to DBPedia instances (see section 3 for details about the updated version 1.0);

- OpenCyc – the largest and most comprehensive hand-crafted knowledge base, including 1.6 million statements;
- FactForge, combining the above with a "refined" version of DBPedia and the original versions Freebase, Geonames and a few other LOD datasets, e.g. Wordnet, CIA World Factbook, Lingvoj, MusicBrainz, RDF from Zitgist, New York Times in a compound dataset containing a couple of billon explicit statements and implicit facts.

Figure 2 represents the mappings that are available or under development between the different elements of the reference knowledge stack.

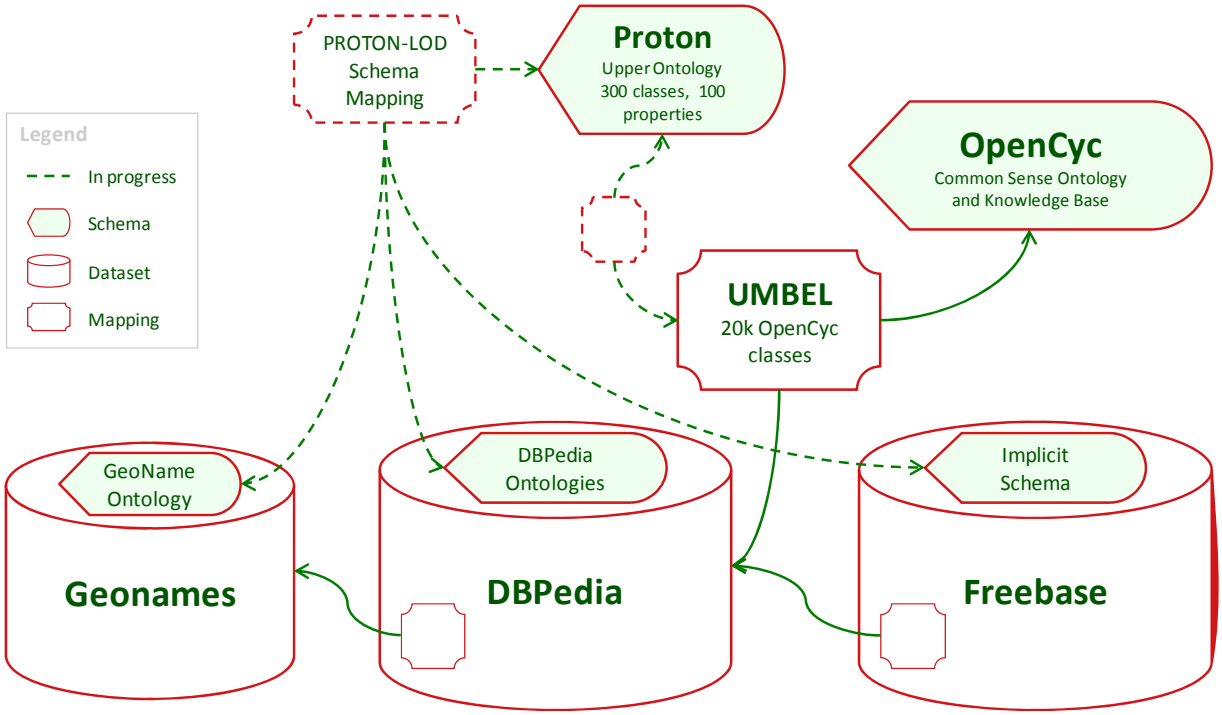

Figure 2. Reference Knowledge Stack.

Table 1 describes the major features of the datasets and their intended usage.

Table 1. Reference Knowledge Stack Elements[2].

Dataset	**Size** (approx.)	**Reusable Schema-level Vocabulary**	**Reusable Instance-level vocabulary**	**Reliable formal semantics**
PROTON	500+ concepts	+		+
UMBEL	27,917 classes	+		+

[2] Different terms are used in the table to refer to the size of the single elements of the Reference Knowledge Stack. This is due to the fact that the different publishers use different words for refering to the entities in their knowledge bases. This difference is preserved in table 1. The terms in its second column correspond to the original terms used by each data producer.

OpenCyc	2 million assertions	+	+	+
DBPedia	700 million assertions		+	
Freebase	500 million assertions			+
Geonames	100 million statements	+		+

The Reference Knowledge Stack is implemented as a reason-able view, discussed in deliverable D.1.1.1. *Reason-able views* [8], [10] represent an approach for reasoning with and management of linked data defined at Ontotext currently presented in two use cases FactForge [11], [17] and Linked Life Data [18]. It is an assembly of independent datasets, which can be used as a single body of knowledge with respect to reasoning and query evaluation. Each reason-able view is aiming at lowering the cost and the risks of using specific linked data datasets for specific purposes.

We will illustrate a sample usage of such a reference knowledge stack with a sample query of FactForge, which aims to retrieve "the most popular entertainers born in Germany".

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbp-ont: <http://dbpedia.org/ontology/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX opencyc: <http://sw.opencyc.org/2008/06/10/concept/en/>
PREFIX geo-ont: <http://www.geonames.org/ontology#>
PREFIX om: http://www.ontotext.com/owlim/

SELECT * WHERE {
     ?Person dbp-ont:birthPlace ?BirthPlace ;
             rdf:type opencyc:Entertainer ;
             om:hasRDFRank ?RR .
     ?BirthPlace geo-ont:parentFeature dbpedia:Germany .
} ORDER BY DESC(?RR) LIMIT 100
```

The definition of this query involves: schema- and instance-level vocabulary from DBPedia; schema-level vocabulary of OpenCyc [19] and schema-level vocabulary from Geonames [20]. A system predicate of the BigOWLIM semantic repository, where FactForge's compound dataset is loaded, is used to retrieve the so-called RDFRank of the corresponding node, which represents an equivalent of Google's PageRank computed on top of the RDF graph. This rank is used as a measure for "popularity". It is interesting to note that this query returns unexpected results – on top of the result set is Friedrich Nietzsche. He qualifies as an entertainer because the little known fact that he was not only philosopher, but also virtuoso piano player, was available in the MusicBrainz dataset [21]. Although MusicBrainz vocabulary is not used in the query, this fact was considered because of the mappings between MusicBrainz classes and the OpenCyc classes which are established through UMBEL and the mappings between instance level identifiers between MusicBrainz and DBPedia.

Technically, the Reference Knowledge Stack is now available as the next version of the FactForge reason-able view, which would allow the different components to be used together or in separation. Further, the interlinking at schema and instance level of the components of the Reference Knowledge Stack gives the way to querying the reason-able views by using the predicates of only one schema. Thus, the query about locations of Modigliani artwork, discussed in deliverable D1.1.1, [8], which would typically have to use predicates from several datasets, can be expressed only with predicates from the upper ontology PROTON, and deliver the same query results in a more concise way.

Most of the mappings within the reference knowledge stack are developed semi-automatically, using various machine learning techniques. More details on those mappings will be provided in Section 2.

1.3 Diversity Tolerant Design of RKS

We remind here the forest exploration metaphor, used in D1.1.1, to stress the importance of reference data for data management. Suppose data is seen as a forest and reference data represent beaten paths through this forest. Forests are dynamic and can be considered unknown, at least to the extent that foresters cannot keep knowledge for each tree and even if they try to do so, the changes that occur constantly make this knowledge inaccurate and unreliable in time. Forest paths are known reference points and communication facilities, which facilitate the navigation within the forest, its exploration and overall, the access to the wood resources. In the same way reference data are well determined, relatively static and predictable data structures that can facilitate access to a diverse and dynamic datasets such as the web of linked data.

To access real volumes of wood foresters should, at some point, get off the beaten track and use methods and techniques to explore wild forest areas. Still, it is the case that beaten paths allow for the exploitation of large forests, by means of lowering the efforts of their exploration. In a similar way linked data management is unthinkable (and in a sense pointless) without techniques which allow for dealing with unseen data – those are all sorts of automated statistical or symbolic methods which allow for analysis, interpretation, selection and retrieval of data. Still, using reference ontologies and more general reference knowledge structures has the potential to considerably lower the cost of using linked data as well as any collection of dynamic and diverse data collection.

A concrete example of how reference ontologies can facilitate access to linked data is query formulation using a reference upper level ontology. Consider the Modigliani artwork query, [12], with PROTON predicates only:

```
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 pupp: <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 ;
          ptop:locatedIn [ rdf:type pupp:City ; ot:preferredLabel ?city ].
}
```

Querying through PROTON brings several advantages:

- One does not need to search through the vast majority of predicates defined in both DBPedia and Freebase – even the small set of about 183 properties defined in PROTON appears to be sufficient for the formulation of this query - when executed it returns the same results as a query fromulated with Dbpedia and Freebase predicates. PROTON's classes and properties are selected in a way to cover the most general common sense concepts, necessary for semantic search and indexing, and extended to a full coverage of DBPedia ontology[3] ;
- There is no need of multiple optional patterns because the most popular variations of "located in" relationships from Freebase and DBPedia are all mapped to the `ptop:locatedIn` property in PROTON;

[3] Being in experimental stage, the impacts of the use of PROTON as a unified access point to FactForge and LOD is currently the subject of a thorough analysis. This included the definition of measures for its coverage.

– The query is much shorter, easier to define and understand.

Although an upper-level ontology such as PROTON cannot cover all sorts of factual distinctions made in DBPedia and Freebase or other large datasets, querying those datasets through PROTON will always have limited "resolution" as PROTON defines classes and properties, which are much more general than most of those in the specific datasets. On the other hand, defining queries using the fine-grained original vocabularies is simply unfeasible in many scenarios because of the variety and inconsistencies presented in LOD schemata. This way, PROTON which is designed as a minimal schema covering the most popular concepts necessary for semantic search and annotation, allows for an easy entry point for exploration of LOD. One can get acquainted with its few hundred of concepts with much lower efforts as compared to those required for the larger (and in the case of DBPedia, less uniform) vocabularies of the specific datasets. Using strictly organized schemata, such as PROTON, also allows for putting together under the same upper level predicate a list of predicates from different vocabularies defining one and the same concept, like in the case of `ptop:locatedIn`. This gives a way to account for some evident inconsistencies in LOD, which can be considered as a source of diversity. On the other hand coping with diversity in the Reference Knowledge Stack can be illustrated with the formulation of mapping predicates with "looser" connotations like the ones adopted by UMBEL when linking its schema to Wikipedia: `correlatesTo`, `isAbout`, `isRelatedTo`, `isLike`, `isCharacteristicOf`, `hasMapping`, etc. instead of using the strict denotators like `owl:sameAs` and `owl:equivalentClass`. These predicates are diversity tolerant in the sense that they leverage the differences in the proper definitions of concepts and facts from different datasets.

We will evaluate if this mapping approach is feasible within RENDER. It could appear that although reference vocabularies facilitate the access to the data, they somehow limit diversity. We are prepared to extend the underlying data model according to the results of Task 3.1 in order to resolve such shortcomings of the proposed data organisation. One should note that the existence of the reference vocabulary does not mean that one cannot use the vocabulary of the original datasets. If we extend the analogy to forestry from the beginning of this section, the existence of paths in the forest, does not prevent anyone to explore the forest ignoring them.

Another possible way of taking advantage of RKS is to think of it as the overlap among various diverse sources of information. This could give the ground for the definition of a measure of diversity, by defining similarity metrics between the reference data and the collected data. Such measures could be based on frequence of occurrence, distance from reference concepts base on word space models, etc. Thus instead of limiting diversity RKS could give a basis for quantitative evaluation of diversity.

The usefulness of the above mentioned approaches will be explored further within RENDER and could advantageously augment or complement the data model.

In the next sections, the constitutive ontologies, datasets and mappings of RKS as implemented and made available in RENDER will be presented.

2 PROTON to LOD Mapping

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.

The main purpose of the reference ontology is to ensure an easier access to a given set of heterogeneous data like the ones of LOD. The reference ontology ensures uniformity of the interpretation of the data of LOD cloud at schema level as its mapping to them requires a thorough analysis of the meaning of the available concepts inspecting the instances they describe.

The easier access is mainly noticeable in the structured query formulation, as mentioned in Section 2.1 above), where single predicates from the reference ontology can refer to multiple predicates from different LOD schemata. Figure 3 illustrates the idea.

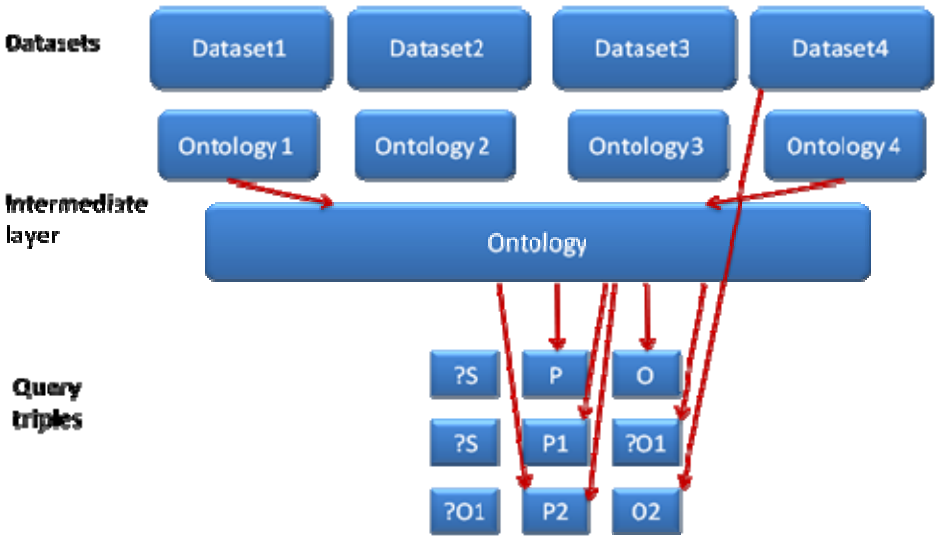

Figure 3. Relation of query triples to datasets via an intermediate layer (see the text for details).

Our work revolves around a fraction of the LOD cloud represented in a reason-able view (see Section 1.3), which allows for a better data management when inference is involved, e.g. FactForge [11]. It consists of mapping PROTON, a base upper-level ontology [23], used as reference ontology and the vocabularies of some of the datasets in FactForge, namely DBPedia, Geonames, and Freebase [24]. 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.

The 542 concepts and predicates of PROTON can cover the main upper-level concepts in the LOD datasets in question, but there remain many concepts that still have to be referred to directly with the LOD vocabularies. That is why, a second layer of a larger upper-level ontology is being introduced, which will cover the common sense part of the LOD cloud. This larger upper-level ontology, UMBEL, [28], a strict subset of OpenCyc [19], has close to 28,000 concepts in its last version (1.0). UMBEL (Upper Mapping and Binding Exchange Layer), is designed with the main objective to help content to interoperate on the Web, e.g. different datasets and domain vocabularies, allowing to reason over a coherent reference structure and its linked resources. It can also serve as a base vocabulary for the construction of other concept-based domain ontologies. The 28,000 reference concepts of UMBEL are organized in 33 Super types which are mostly disjoint. This ensures easier manipulations of the data classified under the reference concepts, including inference, more precise distinctions, and consistency. UMBEL is packed with mapping of all the DBPedia entities to UMBEL reference concepts. This allows one to combine the broad coverage of DBPedia with respect to popular entities with the sound class hierarchy of Cyc.

Thus, the mapping of PROTON to LOD is designed to take place by the "mediation" of UMBEL, as shown in Figure 4.

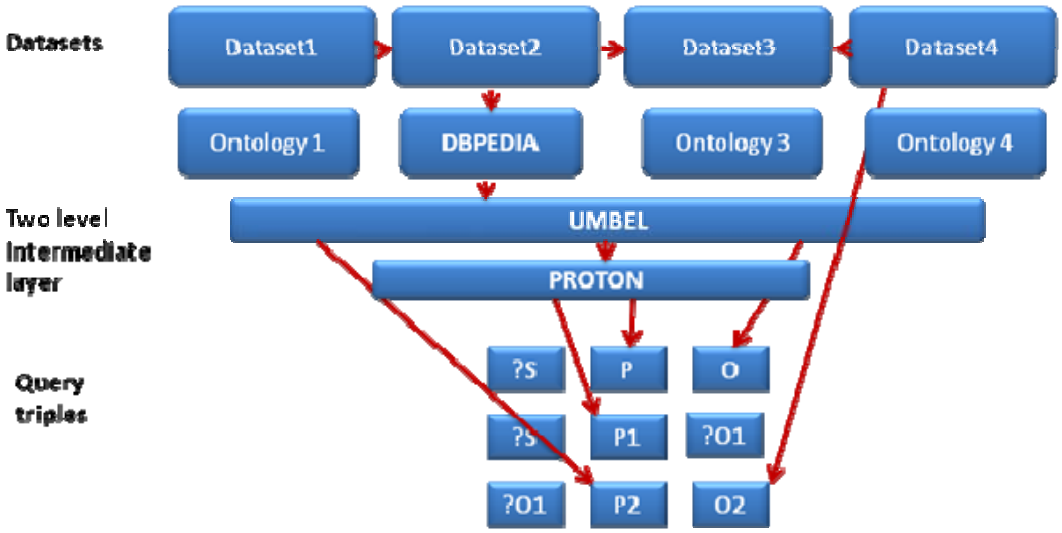

Figure 4. Mapping of PROTON to LOD by the "mediation" of UMBEL.

The development of this approach is an iterative process of extension and modification. At the present stage, PROTON is directly mapped to DBPedia, Geonames and Freebase, and UMBEL is mapped to DBPedia, allowing for access to all datasets of FactForge and to PROTON, and thus allowing one to use a small number of concepts to query the vast variety of knowledge available in FactForge. The following sub-sections describe the mapping of PROTON to DBPedia, Geonames and Freebase, the mapping of PROTON to UMBEL, and UMBEL to DBPedia.

2.1 Mapping PROTON to DBPedia, Geonames, and Freebase

Mapping PROTON to DBPedia, Geonames and Freebase faces the challenge of matching ontologies built according to different methods and design principles, e.g. data-driven ontologies and an upper-level ontology, which requires the mapping to follow the design principles of one of the ontologies, which is defined as the basic, and provide when mismatches are available a translation of the conceptualization of the other ontology into the terms of the basic one, [24].

2.1.1 The Data

PROTON is a basic subsumption hierarchy which provides coverage of most of the upper-level concepts necessary for semantic annotation, indexing, and retrieval. It is built according to the OntoClean method [20] where, for example, type and role are distinguished. OntoClean 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. It 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". Additionally, many properties are described separately (in a separate namespace) as standalone properties which pertain to ontological dimensions, but are not modelled in the ontology.
- Geonames ontology has a concise conceptualization organized in very few well structured concepts and millions of instances. Geonames is a geographic database that covers 6 million of the most significant geographical features on Earth and 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, 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).
- Freebase (http://freebase.com) is a large collaborative knowledge base of structured data from many sources like Wikipedia, Chemoz, NNDB, MusicBrainz and individually contributed data from its users. It has over 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, e.g. the left most word of predicate name 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. For example, `government.legislative_session.date_ended` or `celebrities.romantic_relationship.end_date`. Freebase has an overall of 19,632 predicates with this structure, which is constantly increasing.

2.1.2 The Methodology

Our approach summarizes a method of matching ontologies with different methodological background – data-driven ontologies and an upper level ontology. It is unidirectional semantic manual alignment of PROTON and the ontologies of the selected datasets of FactForge. The unidirectional matching scheme provides access to FactForge via PROTON, but not vice versa. Manual or semi-automatic mapping is suitable when maximum quality of mapping is sought as in the case of the building of the Reference Knowledge Stack [21]. The best results of automated schema matching approaches in the recent ontology matching competitions, where the ontologies are selected to cover very restricted knowledge domains are about 80% precision [22], [23]. The design principles of PROTON were 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, and assigning subsumption relations between entities and properties from FactForge to PROTON.

The alignment was performed manually as the most suitable approach to find the correspondences of the small amount of upper-level concepts at a maximal level of precision using the methods described above. 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, their use and instances. The matching of concepts and properties between Freebase and PROTON took place with OWL class and property construction.

The matching rules between FactForge and PROTON were designed with subsumption relations only.

2.1.3 Statistics

PROTON (3.0) aligned with DBPedia 3.6, Geonames 2.2.1 and Freebase from year 2010 has 542 classes, 128 Object properties and 55 Datatype properties. The mappings are:

- 203 mapped classes to DBPedia;

- 23 mapped properties to DBPedia;
- 3 mapped classes to Geonames;
- 4 mapped properties to Geonames;
- 382 mapped Geonames Codes;
- 9 mapped classes to Freebase;
- 60 mapped properties to Freebase.

2.2 Mapping PROTON to UMBEL

2.2.1 The Data

UMBEL has about 28,000 reference concepts drawn from the OpenCyc knowledge base, which are organized into 33 SuperTypes. The SuperTypes are designed to be disjoint and to provide a higher-level of clustering and organization of Reference Concepts for a more convenient use in user interfaces and for reasoning purposes. The UMBEL Vocabulary is designed to recognize that different sources of information have different contexts and different structures. By nature, these connections are not always exact, thus means for expressing the "approximateness" of relationships are essential, [1]. These approximate alignments can be oriented by means of the 28,000 'Reference Concepts'. By design, this set of fixed reference points is neither exact nor comprehensive. These reference concepts are not meant to model the world in all of its complexity and nuance. The actual goal is to provide a set of fixed references by which constituent content can be oriented and navigated. The coherent set of UMBEL reference concepts began with the OpenCyc knowledge base. However, since its scope and sophistication far exceeded what was tractable for a lightweight reference structure, OpenCyc was pruned and cleaned to a significant degree. So, UMBEL is a clean, 100% subset of OpenCyc. The result is a reference structure of about 28,000 concepts, broadly applicable as orienting nodes to any knowledge domain, all coherently structured and linked to one another. This lightweight UMBEL Reference Concept ontology is, in essence, a content graph of subject nodes related to one another via broader-than and narrower-than relations. In turn, these internal UMBEL Reference Concepts may be related to external classes and individuals (instances and named entities) via a set of relational, equivalent, or alignment predicates. This UMBEL Vocabulary is itself a solid basis for constructing domain ontologies that can also act as reference ontologies within their own domains.

2.2.2 Methodology

The mapping between PROTON and UMBEL takes place at the level of PROTON classes and UMBEL reference concepts ('RefConcepts') where a semi-automatic approach is being used[4]. Semantic vectors' [26] techniques are used to generate the first linkage between the two vocabularies based on their labels ('prefLabels,' 'altLabels,' and definitions). Consequently, a manual evaluation of the automatic mapping and correction or completion of the mappings is being performed to obtain a full 100% coverage and a 100% precision. At a third stage, one instance for each PROTON class was created and merged with the UMBEL-PROTON ontology. Pellet 2.2.2 [26] was used to check if the ontology was consistent with the UMBEL to PROTON linkages. The consistency checking was enforced by the UMBEL SuperTypes disjointness constraints. Several issues have been found and corrected by this procedure, most of which related to the PROTON to UMBEL linkage. So, the two ontologies are consistent within the UMBEL ontological framework.

444 PROTON classes are directly mapped to corresponding UMBEL reference concepts.

[4] The reference knowledge stack (RKS) pursues a methodology of building a layered reference structures, which provide an incremental use of more general to more specific reference data. OpenCyc is too large and complex, and not tractable for a lightweight reference structure.

For each PROTON class, an equivalent or parent concept was identified in the UMBEL reference concept structure. Matches were expressed as a `rdf:subClassOf` relationship between the PROTON class and the UMBEL reference concept. Each linkage was tested for consistency and satisfiability using Pellet. At the end of the mapping, all UMBEL reference concepts were made `rdfs:subClassOf` of one and only one PROTON class (according to FactForge's world view).

Consequently, the linkage from UMBEL to PROTON was reversed to produce the PROTON to UMBEL linkage.

2.3 Mapping UMBEL to DBPedia and Wikipedia

The mappings between UMBEL and DBpedia were geared to allow any of the constituent ontologies or their predicates to be used in conjunction with the other ontologies so that UMBEL becomes a central reference ontology for LOD. UMBEL at the level of reference concepts was mapped to DBPedia vocabulary by hand. Consequently, verification for consistency was done with the Pellet reasoner to check consistency and satisfiability.

These class mappings were then the basis for manual or semi-automatic mappings to Wikipedia instances (pages) using either the:

- DBpedia Ontology;
- Semantic Vectors [26] correspondences; or
- Analysis of DBpedia category structure.

All instance mappings were also related to one of 33 UMBEL SuperTypes (SuperClasses of reference concepts) [4].

As a result, the number of UMBEL reference concepts was expanded from 20,512 to 27,917. These are all fully integrated into the UMBEL ontology with one of 33 SuperTypes (ST) assigned.

272 DBpedia ontology classes are directly mapped to corresponding UMBEL reference concepts.

The mapping between the DBpedia ontology and the UMBEL reference concept followed the same procedure. However, since there was already a mapping between PROTON and the DBpedia ontology, only the unmapped DBpedia ontology classes needed to be added.

Across all mappings, 3,527 UMBEL reference concepts are linked directly to Wikipedia (DBpedia). The result is that 2,130,021 unique DBpedia pages are in total linked to this structure via nearly 4 million predicate relations (3,935,148). All of these pages are also characterized by one or more STs.

Of these 2 million pages, 876,125 are assigned a specific SuperType via `rdf:type`; the remaining have a less certain relationship (`relatesToXXX` predicate). Across all mappings, 60% of all UMBEL reference concepts (or 16,884) are now linked directly to Wikipedia via the new umbel:correspondsTo property. Across all of these mappings, nearly 4 million predicate relations (3,935,148) link UMBEL to Wikipedia.

Three methods were employed to link Wikipedia pages (instances) via the DBpedia (v. 3.51) extraction to the UMBEL reference concept structure:

- In method one, the instances associated with the DBpedia ontology were inherited directly based on their class mappings to the UMBEL reference concepts. These mappings also received the `rdf:type` predicate. Some 659,527 unique pages were linked in this matter, resulting in a total of 876,125 `rdf:type` assignments;
- In method two, using Semantic Vectors [26] applied to "clean" DBpedia categories, an association file to candidate UMBEL reference concepts with SV scores was created for every clean DBpedia category. These candidates were then inspected by hand with an assignment made manually. DBpedia instances associated with these categories were then mapped to the UMBEL structure and given a `relatesToXXX` predicate for the reference concept's associated, single ST (SuperType). Because multiple DBpedia instances could be related to different reference concepts, then

individual DBpedia pages may have been assigned multiple `relatesToXXX` predicates. (If the DBpedia page already had a `rdf:type` assignment, this would supercede the `relatesToXXX` predicates);

- With method two, 2,484 unique reference concepts participated in the linkage to 102,956 unique DBpedia pages. A total of 111,470 `relatesToXXX` predicates were created based on this method;
- In method three, the Wikipedia categories were deconstructed to discern their structural compositions, largely based on suffix extensions. A script was used to relate these DBpedia categories by list to candidate UMBEL reference concepts. These lists were then presented via script for assigning by hand to the associated UMBEL reference concept. Instances related to the assigned DBpedia category were then given the same `relatesToXXX` predicate that was associated with the related UMBEL reference concept;
- With method three, 1,668 unique reference concepts particated in the linkage to 1,808,782 unique DBpedia pages. A total of 2,947,553 `relatesToXXX` predicates were created based on this method;
- Lastly, a fourth source, which was not really a method, added 7,405 Wikipedia instances by virtue of hand-inspected OpenCyc to DBPedia page mappings within the current OpenCyc knowledge base.

Two ancillary objectives were included in the effort to secure UMBEL's role as a central reference ontology. These two objectives were: 1) to add further reference concepts (RCs) to UMBEL's core ontology, and 2) to refine UMBEL's existing vocabulary with additional linking predicates (`relatesToXXX` predicates).

Three major changes to the UMBEL vocabulary and reference concept structure (ontology) were made as the result of this effort.

The first major change was to add 7,405 reference concepts to the core UMBEL structure. These additions came about as a way to complete the coverage of the general UMBEL structure in order to provide appropriate linkage points into the ontology. The analysis leading to these additions came about from analyzing existing OpenCyc to DBPedia linkages and missing linking concepts due to the DBPedia and GeoNames class mapping activities. This larger "core" UMBEL structure is now felt to be closer to adequate for ongoing reference mappings to other external ontologies into the future.

The second major change was to add 31 new predicates to the UMBEL vocabulary to represent a linkage relationship to a SuperType. These predicates all have the form `relatesToXXX`, for instance `relatesToAbstraction`, `relatesToActivity`, `relatesToAnimal`, etc. The predicate indicates that the object instance has a relation to the SuperType, perhaps as a true class member or perhaps only as an attribute, but that the degree of this relationship cannot be resolved. These predicates and their association with SuperTypes are described in [27].

The third major change was to apply the UMBEL `hasMapping` predicate to all of the possible assignments, using a controlled vocabulary for characterizing the mapping assignment [28].

3 Umbel 1.0

In the beginning of the RENDER project, in a partnership between Structured Dynamics and Ontotext, the UMBEL framework has been applied and refined releasing UMBEL 0.8 followed by UMBEL 1.0. Significant use cases have been tested, notably with FactForge and Proton. The UMBEL reference ontology has been better organized and made easier to browse via the addition of 33 new SuperType classes clustered into nine dimensions. Many early vocabulary decisions have been revised, and substantial improvements across the board have been made in terms of structure and documentation. Version 0.80 is also now fully OWL 2 compliant.

UMBEL 1.0 is an upgrade of UMBEL 0.8. as a result of the effort of mapping UMBEL to DBPedia and Wikipedia as described in Section 2.3. The mapping required extensions and modifications of UMBEL schemata and reference concepts.

Thus, UMBEL 1.0 is characterized with the following features [28]:

- The number of UMBEL reference concepts was expanded from 20,512 to 27,917;
- All reference concepts are fully integrated into the UMBEL ontology with one of 33 SuperTypes (ST) assigned;
- 444 PROTON classes were directly mapped to corresponding UMBEL reference concepts;
- 272 DBPedia ontology classes were directly mapped to corresponding UMBEL reference concepts;
- 60% of all UMBEL reference concepts (or 16,884) are linked directly to Wikipedia:
  - 2,130,021 unique Wikipedia pages are accessible and linked to the UMBEL structure
  - All of these Wikipedia pages are related to one or more UMBEL STs
  - 876,125 of these Wikipedia pages are assigned a specific `rdf:type`; the remaining have a less certain relationship (`umbel:relatesToXXX` predicate)
  - nearly 4 million predicate relations (3,935,148) link UMBEL to Wikipedia
- UMBEL has been mapped to 444 PROTON classes;
- A new `correspondsTo` predicate has been added for nearly or approximate sameAs mappings (symmetric, transitive, reflexive);
- A controlled vocabulary of qualifiers was developed for the `hasMapping` predicate;
- 31 new `relatesToXXX` predicates have been added to relate external entities or concepts to UMBEL SuperTypes ;
- Some disjointedness assertions between SuperTypes were added or changed;
- Switched former UMBEL predicates that duplicated ones in SKOS because SKOS has now been changed to accommodate OWL DL.

4 FactForge Update

The major components of the Reference Knowledge Stack have been tested on top of FactForge. As a start, a basic updated version of FactForge was loaded with the latest versions of DBPedia (3.6), Geonames (2.2.1) and Freebase. Some components of DBPedia were not loaded, namely: DBPedia ontology and DBPedia categorisation hierarchy – the reason is that these parts provide semantics, which (i) can cause inference problems and (ii) is not necessary because similar semantics can be inferred based on the mappings to PROTON and UMBEL. For instance, the sub-class relationships between DBPedia ontology classes are no longer crucial, as all of those classes are already mapped to PROTON and its subsumption hierarchy provides a cleaner and more inference-friendly semantics.

The above presented basic updated version of FactForge has been extended in two directions – one instance was augmented by loading UMBEL 1.0 on top of it, while the other was extended by loading PROTON 3.0 and its mappings to the LOD datasets. The rationale behind this setup was that we needed to observe the implications of the new semantics and mappings of PROTON and UMBEL separately, before we load them together. First instances of both extended test versions of FactForge were successfully loaded and are currently being evaluated. Details about the loading times and data statistics will be provided in deliverable D.1.3.1. SPARQL queries have been run to test the connectivity of the datasets and the effectiveness of the mappings. Queries with only UMBEL concepts have been run and returned results from a variety of datasets such as DBPedia, Freebase, New York Times, OpenCyc for query asking for people with particular occupation. Queries with only PROTON concepts about places that are part of countries also returned results from multiple datasets. Further analysis and evaluation of the positive impacts of the use of the Reference Knowledge Stack for querying, reasoning and data access will be performed in the next period of the project.

The actual setting of the experiments with FactForge, UMBEL and PROTON will be described in deliverable D.1.3.1 describing the UI APIs to these services.

5 Conclusion and Future Work

This deliverable reports the work on data organization and collection after deliverable D1.1.1 in RENDER. It reiterates the so-called Reference Knowledge Stack (RKS) based data collection concept, introduced in D1.1.1, paying special attention to the central issue of preserving the diversity of the collected data and the possibility to easily access and reason over this data.

RKS includes several components of various size and nature: the PROTON upper-level ontology, OpenCyc, UMBEL and few of the central LOD datasets, including DBPedia, Freebase, Geonames, Wordnet, MusicBrainz and others. This reference structure is meant to allow for interlinking all sorts of data relevant to the project and to facilitate different types of access from users with different background, technical skills and level of familiarity with the data.

A specific contribution is the latest version (1.0) of the UMBEL reference ontology, which includes 27,911 concepts, derived from OpenCyc and interlinked with the entities in DBPedia and Wikipedia. Moreover, this deliverable describes the ontology alignment work required for the implementation of the Reference Knowledge Stack which has been completed. A new version of PROTON 3.0 extended to 542 classes, and 183 properties with upgraded alignments to DBPedia 3.6 and Geonames 2.2.1 are also part of results of this work. It also includes the mappings from PROTON to UMBEL on the one hand and to the specific schemata and ontologies of DBPedia, Geonames and Freebase on the other. All these reference resources are now publicly available in the major update of FactForge, also part of the work reported here.

All appropriate resources which are not already published as linked data will be added to the LOD cloud – this is particularly the case of PROTON together with its various mappings. The reference stack along with sample content and other information have been published through the Forest framework, to enable its efficient access and querying on the web (this work is reported in D1.3.1).

The implementation of Reference Knowledge Stack and the update of FactForge and their availability for the RENDER partners is a major step in the realization of the data organization approach of the project.

The development of the semantic annotation prototype for analysis of unstructured content in English and its annotation based on the RKS, is related to work in progress in WP2 and will be reported on in deliverable D1.2.2.

References

[1] Bergman, M. K. *Bridging the Gaps: Adaptive Approaches to Data Interoperability*. Keynote presentation at DC-2010 Conference, Pittsburgh, PA, October 22, 2010. http://www.slideshare.net/mkbergman/dcmi-20101022.

[2] Bizer, C., Heath, T., and Berners-Lee, T. *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).

[3] Caracciolo, C., Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaisé, V., Meilicke, C., Pane, J., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O., and Sváte, V. *Results of the Ontology Alignment Evaluation Initiative 2008*, (2008).

[4] Damova, M., Kiryakov, A., Simov, K., and Petrov, S. *Mapping the central LOD ontologies to PROTON upper-level ontology*. Ontology Mapping Workshop at ISWC 2010, http://om2010.ontologymatching.org.

[5] Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Trojahn dos Santos, C., Vouros, G., Wang, S. *Results of the Ontology Alignment Evaluation Initiative 2009*. In: Proceedings of the 4th Ontology Matching Workshop at ISWC, (2009).

[6] Guarino, N. and Welty, C. *Evaluating Ontological Decisions with OntoClean*. Communications of the ACM, 45(2): 61-65 (2002).

[7] Jain, P., Yeh, P. Z., Verma, K., Vasquez, R. G., Damova, M., Hitzler, P., and Sheth, A. P. *Contextual Ontology Alignment of LOD with an Upper Ontology: A Case Study with Proton*. ESWC'2011, Crete, Greece. http://knoesis.wright.edu/library/download/prateek_tech_report_2010.pdf.

[8] Kiryakov, A., Grinberg, M., Damova, M., Russo, D. *D1.1.1: Initial Collection of Data*. Deliverable of EU-FP7-ICT-2009-257790 project RENDER (2011).

[9] Kiryakov, A., and Momtchev, V. *Two Reason-able Views to the Web of Linked Data.* Presentation at the Semantic Technology Conference 2009, San Jose. http://www.slideshare.net/ontotext/two-reasonable-views-to-the-web-of-linked-data.

[10] Kiryakov, A., Ognyanoff, D., Velkov, R., Tashev, Z., and Peikov, I. *LDSR: Materialized Reason-able View to the Web of Linked Data*. In: Proceedings of OWLED 2009. Chantilly, USA, 23-24 October 2009 (2009).

[11] *Linking Open Cloud Diagramme*. http://lod-cloud.net/ as of September 2010.

[12] MacManus, R. *The Modigliani Test: The Semantic Web's Tipping Point*. http://www.readwriteweb.com/archives/the_modigliani_test_semantic_web_tipping_point.php, (2010).

[13] http://www.umbel.org/.

[14] http://dbpedia.org.

[15] http://linkeddata.org/.

[16] http://factforge.net.

[17] http://www.ontotext.com/factforge.

[18] http://linkedlifedata.com.

[19] http://www.cyc.com/cyc/opencyc.

[20] http://www.geonames.org.

[21] http://musicbrainz.org.

[22] Pellet: OWL 2 Reasoner for Java. http://clarkparsia.com/pellet.

[23] Terziev, I., Kiryakov, A., and Manov, D. *D.1.8.1 Base upper-level ontology (BULO) Guidance*. Deliverable of EU-IST Project IST – 2003 – 506826 SEKT (2005).

[24] Wikipedia. *Master data*. http://en.wikipedia.org/wiki/Master_data as of January 2011.

[25] Wikipedia. *Reference data*. http://en.wikipedia.org/wiki/Reference_data as of January 2011.

[26] http://code.google.com/p/semanticvectors/.

[27] http://www.mkbergman.com/930/announcing-a-major-new-umbel-release/.

[28] http://umbel.org/specifications/full-specification.

