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

RENDER

Deliverable 1.3.1

Web-based API

Editor:	Atanas Kiryakov, Ontotext
Author(s):	Atanas Kiryakov, Ontotext; Maurice Grinberg, Ontotext
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)[1]	Public (PU)
Contractual Delivery Date:	March 2011
Actual Delivery Date:	March 2011
Version:	1.1
Keywords:	Data publishing, data access, SPARQL, linked data, SPARQL end-point

[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:	D1.3.1 - Web-based API
Editor (Name, Affiliation)	Atanas Kiryakov, Ontotext AD
Work package Leader (Name, affiliation)	Atanas Kiryakov, Ontotext AD
Estimation of PM spent on the deliverable:	3

Executive Summary

This deliverable describes the web based user interfaces and APIs for access and publication of data related to RENDER, which will be provided in the beginning of the project. The main service will be an RDF repository that can be explored and queried through the Forest framework. In essence, it will be a modified version of the FactForge service, updated with the developments of the Reference Knowledge Stack (RKS). The later was developed in WP1 (see D1.2.1, [15]) as a reference data structure meant to facilitate the integration and access to the data to be used within and published in the course of the project.

The general public will be able to search in and explore RDF graphs, evaluate SPARQL queries, and search for relations through the facilities of the Forest library. The application will be able to access the data through a SPARQL end-point, which provides full access to the data. The first set of data to be available through this interface will represent the RKS loaded into an instance of the BigOWLIM semantic repository. This way users will be able to benefit from the inference capabilities of the engine, its optimized model for handling **`owl:sameAs`** equivalence, various options for full-text search and geo-spatial constraints.

The management and publication of data will be allowed trough the SPARQL Update specification, [27], which allows inserting and deleting data in specific RDF graphs as well as creating and managing named graphs. At this stage this service will not be publicly available as it is still necessary to find the best model for controlling the access to such functionality, so that maximum freedom is combined with a good level of reliability of the service.

Table of Contents

Abbreviations

API application programming interface, a specification of technical mechanisms, which allow for integration between software systems and components;

DBPedia an 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; more information is available in [7].

LOD the Linking Open Data project is a W3C SWEO Community project and is an initiative for publishing “linked data”; more details are provided in section 1.1 and at [32];

RDF Resource description framework, a basic specification determining the data model of the Semantic Web, [21];

SPARQL a query language for RDF specified in [26];

UI user interface, referring to the front-end components of software systems, which take care of the communication between human users and the system. Such could be web forms as well as other web pages, which allow humans to explore data provided by a computer system, to provide data and, more generally, to interact with it.

Definitions

This material assumes prior knowledge of the basic semantic web standards, namely RDF, [21], RDFS, [9], and OWL, [14].

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. See section 1.1;

OpenCyc OpenCyc is the open source version of the Cyc technology, [11], the world's largest and most complete general knowledge base and commonsense reasoning engine;

PROTON an upper-level schema ontology, which defines about 400 classes and 100 properties relevant for entity classification, description and relation across multiple domains, [29];

Reason-able view Reason-able views, [19], 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 and 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 data data describing a physical or virtual object and its properties. Reference data is used in data management to define characteristics of an identifier that are used within other data centric processes, [31].

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 1.5 for further details.

RDF Molecule the description of an URI node in an RDF graph, including only the minimal information that describes the URI, as defined in section 3.2 of [20]. Technically, the molecule of node S is the part of the graph that you can reach following all paths in the graph, starting from S, until you reach non-blank nodes. This notion is close to the one defined in [23], but quite different from the triple-centric definition provided in [13].

SPARQL End-point a web service, which allows a remote computer to execute a SPARQL query and to retrieve the results in accordance with the SPARQL RDF Protocol, [10].

1 Introduction

This is a relatively small technical deliverable aiming to document the interfaces and, more generally, the organisation of both data publication and data usage as far as the general public is concerned. In covers several points:

- Data querying and exploration via a user interface (UI, see section 2.1);
- Data access via API (see section 2.2);
- Publication of data (see section 3) and, more specifically, publication of linked data.

The data access facilities documented here reflect the data collection, organisation and integration principles, discussed in deliverables D1.1.1, [18], and D1.2.1, [15]. The remainder of the first section introduces several notions and projects on which the RENDER's mechanisms for data publishing and access are based.

1.1 Linked Data

The notion of "linked data" is defined by Tim Berners-Lee, [4][6], as RDF graphs, published on the WWW so that one can explore them across servers by following the links in the graph in a manner similar to the way the HTML web is navigated. It is considered a method for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web, using URIs and RDF. "Linked data" are constituted by publishing and interlinking open data sources, following the principles of:

1. using URIs as names of things;
2. using HTTP URIs, so that people can look up these names;
3. providing useful information when someone looks up a URI;
4. including links to other URI, so people can discover more things.

In fact, most of the RDF datasets fulfil principles 1 and 2 by design. The new element in these principles concerns the requirement for enabling Semantic Web browsers to load HTTP descriptions of RDF resources, based on their URIs. To this end, data publishers should make sure that:

- the "physical" addresses of the published pieces of data are the same as the "logical" addresses, used as RDF identifiers (URIs);
- upon receiving an HTTP request, the server should return a set of triples that describe the resource;
- re-use identifiers and vocabularies from other datasets, which are also linked data.

Linking Open Data (LOD, [32]) is a W3C SWEO community project aiming to extend the Web by publishing open datasets as RDF and by creating RDF links between data items from different data sources. Linked Open Data provides sets of referenceable, semantically interlinked resources with defined meaning. The central dataset of the LOD is DBPedia, [7]. Due to the many mappings between other LOD datasets and DBPedia, the latter serves as a sort of a hub in the LOD graph, assuring a certain level of connectivity. 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).

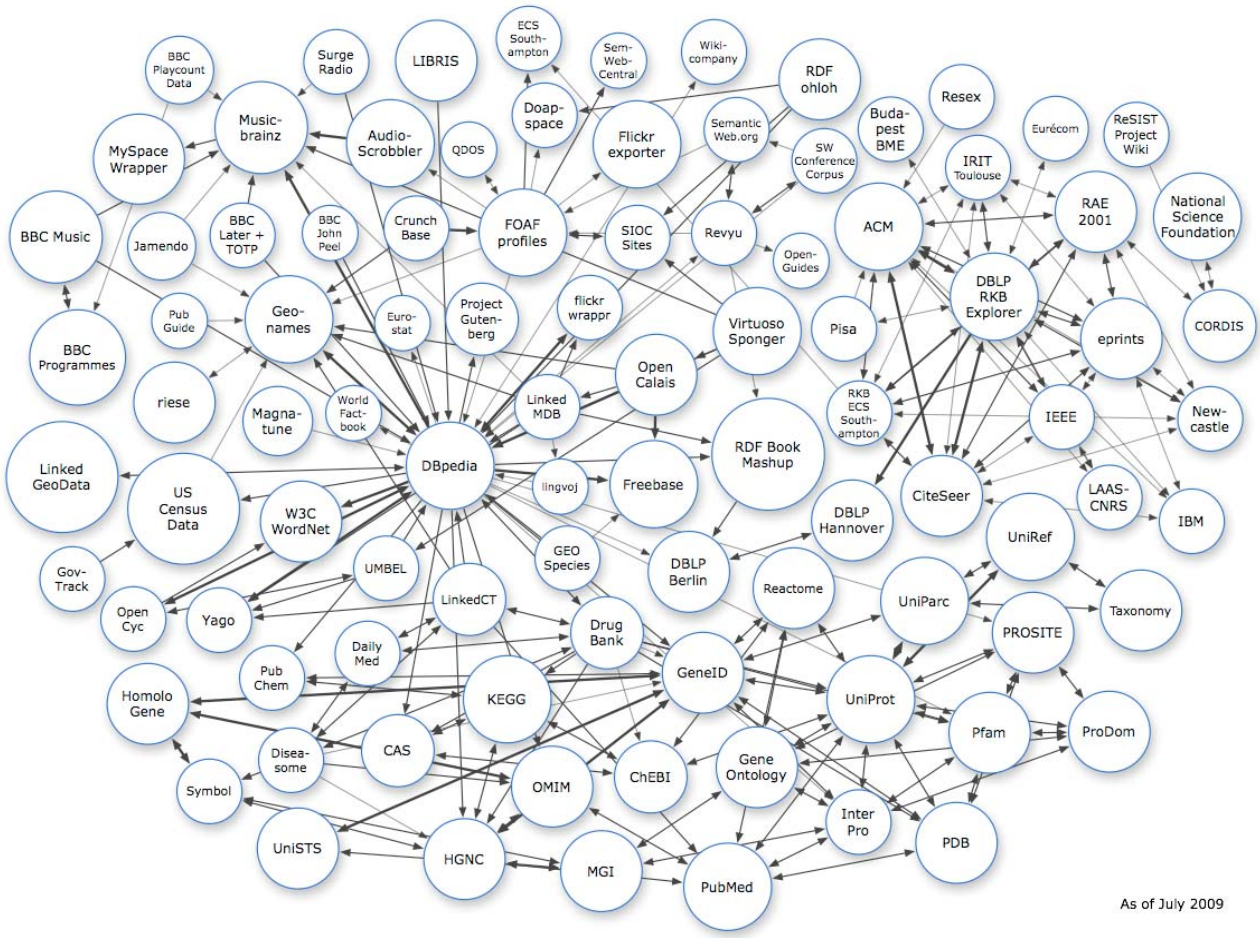

Figure 1. *Map of the Datasets in Linking Open Data (LOD) Project, [32]*

1.2 SPARQL Query and Update Languages

The first version of *SPARQL,* [26], is an SQL-like query language for RDF data, specified by the RDF Data Access Working Group of W3C. It differs from SQL in that it does not contain specific Data Definition Language (DDL) provisions, because the schemata are represented in both RDFS and OWL as standard RDF graphs, thus requiring no specific language to deal with them.

Version 1.1 of the SPARQL specification extends the query language with new features, [16], but also adds data modification capabilities with the SPARQL Update specification, [27].

The SPARQL Query Language supports four types of queries:

- SELECT queries – return *n*-tuples of results just like the SELECT queries in SQL;
- DESCRIBE queries – return an RDF graph. The resulting graph describes the resources, which match the query constraints. Usually, a description of a resource is considered an RDF-molecule, [20], forming the immediate neighbourhood of a URI;
- ASK queries – provide a positive or a negative answer indicating whether or not the query pattern can be satisfied;
- CONSTRUCT queries – return an RDF graph constructed by substituting the variables in the graph template and combining the triples into a single RDF graph by set union.

The SPARQL Update language allows a range of actions, related to remote management and modification of an RDF repository:

- Updates of data contained in the single RDF graph, including insertion of new data, deletion of data, and loading of new files of data into the repository;
- Creation, deletion and managements of named graphs.

1.3 RDF Search and RDFRank

A feature of BigOWLIM[2] that deserves special attention is the so-called RDF Search that provides a novel method for schema-agnostic retrieval of data from RDF datasets. The main rationale behind RDF Search is to allow searching in an RDF graph by keywords and getting usable results (in many cases standalone literals are not useful). Technically, it involves full-text indexing of the URIs in the RDF graph with respect to their "text molecules" – a text snippet, achieved by the concatenation of text from all nodes in the RDF molecule of the corresponding URI. The result is a list of URIs, the text molecules of which match the keywords from the query, ranked with a metric that combines standard full-text search Vector Space Model and RDFRank.

RDFRank is a rank that BigOWLIM can calculate and make available for each URI in the repository via the system predicate **`http://www.ontotext.com/owlim/hasRDFRank`**. BigOWLIM calculates RDFRank for the nodes in an RDF graph similarly to the way in which Google's PageRank, [25], calculates it for web pages. RDFRank is a static, contextually neutral, measure about the degree of "importance" of an RDF node in a graph, based on the importance of the nodes that are linked to it through statements containing other nodes as a subjects and this one as an object.

1.4 Reason-able Views

Using linked data (see section 1.1) for data management is considered to have a great potential. On the other hand, several challenges need to be handled in order to make this possible. *Reason-able views, [19],* represent an approach for reasoning with and management of linked data defined at Ontotext and implemented in two systems, namely, FactForge (presented in section 1.5**Fehler! Verweisquelle konnte nicht gefunden werden.**) and LinkedLifeData (http://www.linkedlifedata.com). *Reason-able views* is an assembly of independent datasets, which can be used as a single body of knowledge with respect to reasoning and query evaluation. The key principles can be summarized as follows:

- Group selected datasets and ontologies in a compound dataset;
- Clean up, post-process and enrich the datasets if necessary. Do this conservatively, in a clearly documented and automated manner, so that (i) the operation can easily be performed each time a new version of one of the datasets is published and (ii) the users can easily understand the intervention made to the original dataset;
- Load the compound dataset in a single semantic repository and perform inference with respect to tractable OWL dialects;
- Define a set of sample queries against the compound dataset. These determine the "level of service" or the "scope of consistency" contract offered by the reason-able view.

Each reason-able view is aiming at lowering the cost and the risks of using specific linked data datasets for specific purposes. The design objectives behind each reason-able view are as follows:

- Make reasoning and query evaluation feasible;

[2] A semantic repository developed by Ontotext, http://www.ontotext.com/owlim/

- Lower the cost of entry through interactive user interfaces and retrieval methods such as URI auto-completion and *RDF search* (a search modality where RDF molecules are retrieved and ranked by relevance to a full-text style query, represented as a set of keywords)*;*
- Guarantee a basic level of consistency – the sample queries guarantee the consistency of the data in the same way regression tests guarantee the quality of the software;
- Guarantee availability – in the same way web search engines are usually more reliable than most of the web sites; they also do caching;
- Easier exploration and querying of unseen data – sample queries provide re-usable extraction patterns, which reduce the time for getting to know the datasets and their interconnections.

1.5 FactForge and the Reference Knowledge Stack

FactForge (http://www.factforge.net/, [5]) represents a reason-able view to the web of linked data (see section 1.1) that includes several of the most central datasets of the LOD cloud: DBPedia, Freebase, Geonames, UMBEL, Wordnet, CIA World Factbook, Lingvoj, MusicBrainz (RDF from Zitgist). Along with the dataset specific schemata and ontologies, the following ones have been loaded in FactForge: Dublin Core, SKOS, RSS, FOAF. They were referred to or imported in the ontologies of the above-listed datasets, so, they were necessary to allow the proper interpretation of the semantics of the data.

The BigOWLIM semantic repository is used to load the data and "materialize" the facts that could be inferred from it. FactForge is probably the largest and most heterogeneous body of general factual knowledge that was ever used for logical inference. The inference was performed with respect to a ruleset, derived from the so-called OWL Horst dialect, [28]. The only dataset that required modification before loading it in FactForge was DBPedia. We had to:

- remove the YAGO module as some incorrect classifications of entities and other faults in it were causing the inference of too many faulty statements in FactForge;
- clean up the category hierarchy.

FactForge has been in development for more than a couple of years. The latest developments were motivated by its expected inclusion in the RENDER project. Besides the ongoing updates of the content reflecting the latest versions of the included datasets, FactForge has been extended with OpenCyc and the New York Time's dataset. Apart from other effects, these datasets were added in order to further minimize the dominance of the DBPedia and Wikipedia in the instance identification vocabulary, i.e. the coverage of topics and entities and their naming.

The Reference Knowledge Stack (RKS) can be seen as an extension of FactForge, serving the purposes of data collection, integration and management within the RENDER project as described in deliverable D1.1.1, [18]. Its general idea is to provide a combination of interlinked vocabularies of different size and nature (e.g. instance- vs. schema- level) and to allow them to be used as an access mechanism for LOD and other data. The datasets included in the RKS and the interconnections between them are depicted on Figure 2.

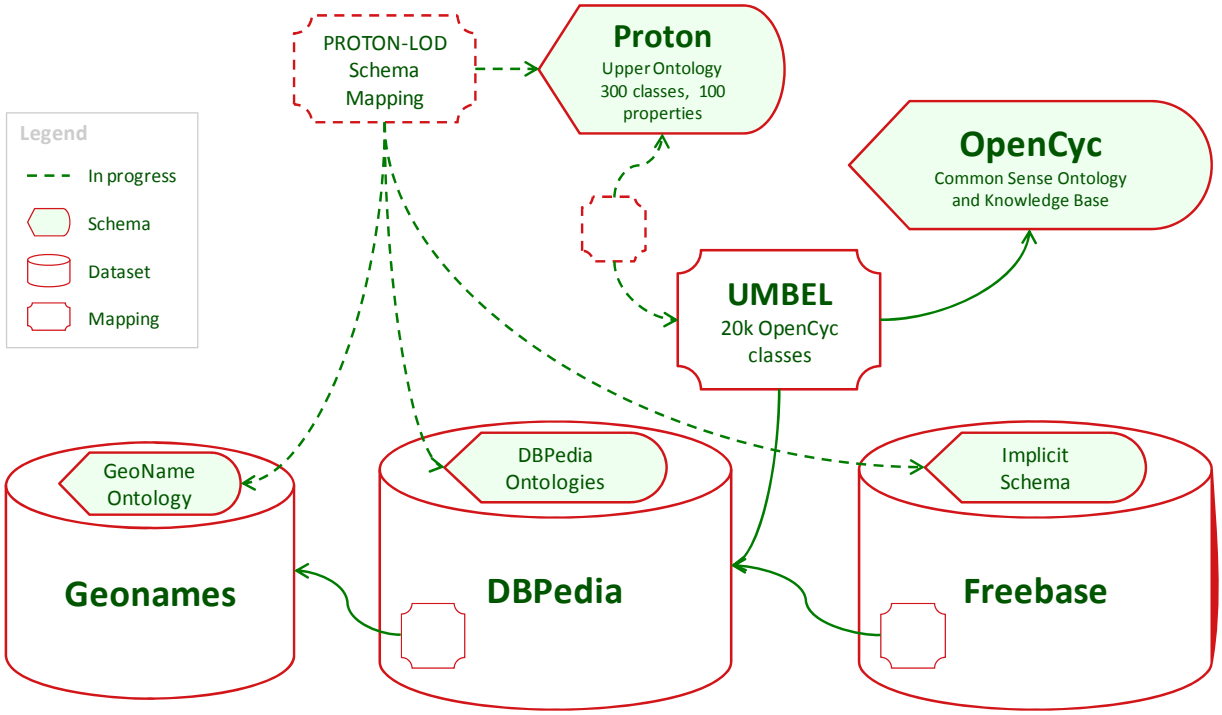

Figure 2. *Reference Knowledge Stack Datasets and Mappings*

The major changes that the RKS introduces to FactForge are described in deliverable D1.2.1, [15], and can be summarized as follows:

- PROTON upper-level ontology is added to DBPedia, Geonames and Freebase together with its schema-level mappings;
- UMBEL is updated and extended with mappings of DBPedia's categories to it;
- There is also a mapping between PROTON and UMBEL, allowing the classifications of the DBPedia entities (which are in fact Wikipedia articles) with respect to UMBEL to be "visible" through PROTON's class hierarchy as well.
- DBPedia is loaded without its own ontologies and its categorisation hierarchy.

The reason for the latter is that these parts of DBPedia 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 these classes are already mapped to PROTON and its subsumption hierarchy provides a cleaner and more inference-friendly semantics.

2 Data Exploration and Querying

The standard data model for representing data in RENDER is RDF, [21]. The semantics of the data is expressed in RDF Schema, [9], and OWL, [1]. The integration of data and their access is further facilitated by the fact that the datasets are interlinked to one or more of the elements of the Reference Knowledge Stack (see section 1.5). While for specific tasks within the project some non-RDF-based representations are also considered, the facilities for data publication and access are entirely based on the above standards.

Complying to the Semantic Web-related specifications of W3C means that there should be facilities for querying datasets using SPARQL, [26], as well as for searching in and exploring RDF graphs. Both the user-interfaces and the APIs will be based on the Forest linked-data front-end framework, which is developed in Ontotext as an open-source library. It allows the exploration of data, hosted in a semantic repository compliant with Sesame[3]. The Forest framework allows customizing the services provided on its basis. For instance, the user interfaces of the FactForge (see section 1.5) and LinkedLifeData[4] services are both entirely based on Forest, but they still look quite different and provide different look and feel.

While the Forest framework allows the exploration of data in any Sesame-compliant repository, in FactForge it uses a repository, hosted by BigOWLIM[5], which includes advanced features such as full-text search (FTS), geo-spatial[6] indexing and querying, optimized handling of **`owl:sameAs`** equivalence[7], etc.

The specific URLs where the RENDER data will be available for exploration and programmatic access will be published on the RENDER's project web site.

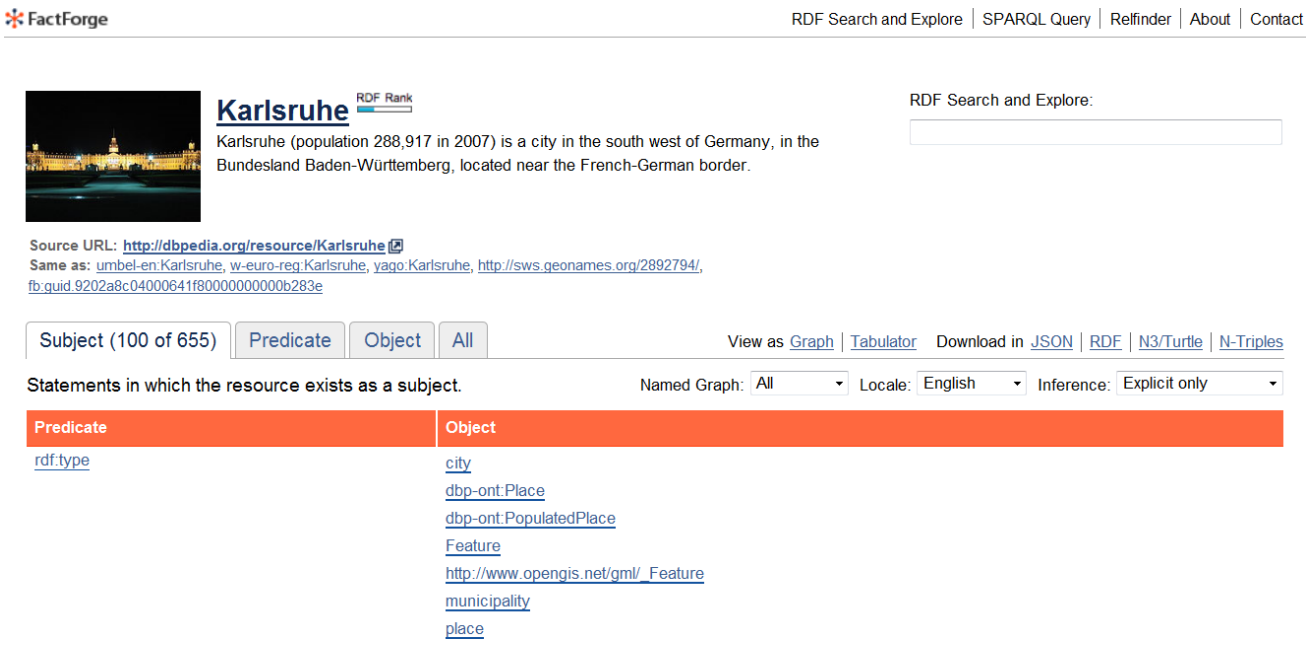

Figure 3. *The Resource Exploration View of FactForge*

[3] http://www.openrdf.org
[4] http://www.linkedlifedata.com
[5] http://www.ontotext.com/owlim
[6] http://www.ontotext.com/owlim/geo.html
[7] http://www.ontotext.com/owlim/owl-sameAs-optimisation.html

2.1 User Interface for Exploration and Querying

In the first phase of the project, the RENDER data will be accessible through public web user interfaces, identical with those currently offered by FactForge (see section 1.5). The main access methods to be exposed are as follows:

- **Incremental URI auto-suggest mechanism.** It allows users to easily find the entity identifiers that they are looking for. On one hand, it allows finding URIs that they do not know or cannot fully type in. On the other, it improves the speed of exploring known URIs, because users do not need to type their full length, but can type just few characteristic symbols and select them;
- **One-node-at-a-time exploration**. This is a standard "linked data browser" metaphor where each URI is represented by a table of two columns –predicates and objects of statements, where this URI appears as a subject. The user can navigate to other URIs by following the hyperlinks of related resources presented in this table. Forest has several features, which allow more ergonomic exploration of repositories containing large volumes of data from multiple datasets: presenting resources in the UI by their "preferred labels", instead of their URIs (when available[8]); filtering data by language locale and named graph; filtering data by their explicit/implicit status; presenting all URIs that are **`owl:sameAs`**-equivalents of the one currently being explored; presenting a text snippet and an image on top of the page (when available). A screenshot of DBPedia's URI for the city of Karlsruhe is shown in Figure 3;
- **RDF Search, retrieving a ranked list of URIs by keywords**. More information about the RDF Search is provided in section 1.3. From a user interface point of view, Forest renders the results of the RDF Search as a list of URIs, represented by their preferred labels and text snippets, when such are available;
- **SPARQL querying interface**. To facilitate query writing, the form provides: shortcuts for the namespace prefixes used in the repository; references to sample queries and check-boxes, which allow one to determine whether inferred facts should be considered during query evaluation and whether query results should be "expanded" with respect to the standard **`owl:sameAs`** semantics;
- **RelFinder.** The graphical search facility called 'RelFinder', [17] allows finding paths between selected nodes. This is a computationally intensive activity and the results are displayed and updated dynamically. The resulting graph can be reshaped by the user with simple click and drag operations. Entities within the emerging graph can be selected and a properties box provides links to the sources of information about the entity.

2.2 APIs for Data Access

Programs will be able to access the data published within RENDER via SPARQL end-point, which at a technical level represents a web service conformant with the SPARQL RDF Protocol, [10]. In essence, it allows programs to submit SPARQL queries for evaluation to a remote query processor (a semantic repository) and to retrieve results in a standard format.

Several comments can be made about the SPARQL end-point, providing the RENDER data. As a start, it should be mentioned that SPARQL, as a query language, is agnostic with respect to any form of inference. In other words, the SPARQL specification does not allow one to specify whether statements that are inferred, or can be inferred in the process of query evaluation, should be considered by the query engine during evaluation. More generally the question is whether reasoning with respect to the semantics of the data and the ontologies loaded in the repository should affect the results of the query. BigOWLIM allows the clients to determine their preference how to use the inference in the query evaluation with the help of a special-purpose system predicate in the FROM NAMED clause of a SPARQL query. These system predicates are documented in section 8.6 of the BigOWLIM's User Guide, [24].

[8] Preferred labels should be associated with the node via predicate **`http://factforge.net/preferredLabel`**

As the data being published by RENDER, or through its publishing facilities, are meant to be entirely open and public, there is no need of any access control. Still, to ensure the smooth operation of the SPARQL end-point, limitations will be enforced on the maximum size of query results that can be loaded through the public interface and the maximum time for query evaluation. The most important rationale behind such a constraint is to ensure that the SPARQL queries that happen to require massive computational resources are not going to hamper the performance of the entire service.

3 Data Management and Publishing

The API to be used for Data Management and Publication will be SPARQL 1.1 Update, [27]. It was selected because:

- it is the most comprehensive standardized mechanism for management of RDF data;
- it covers the requirements for all low-level tasks that need to be handled in relation to publication and management of data in the context of RENDER.

Essentially, there are two levels of granularity at which data can be updated using the SPARQL update: modifications to the content of a single (named) RDF graph; creation and removal of entire named graphs in a repository. As long as the metadata about the named graphs is RDF, it can be managed with the standard mechanisms for modifying RDF graph content, applied to the default graph of the dataset or to any other graph that can be used for storing such metadata. The following is a short summary of the basic operations for modifying the content of an RDF graph, provided by the SPARQL Update:

- INSERT DATA / DELETE DATA allow adding or removing specific statements that are included inline in the query;
- INSERT / DELETE allow inserting and/or deleting statements ,based on query patterns; it is much more powerful and generic than the DATA variants of the operators;
- LOAD inserts all the data from a remote graph into a named graph or into the default graph of the repository;
- CLEAR clears the content of a named graph or of the default graph; the graph itself is not removed.

Since the beginning of the project, Ontotext has dedicated considerable efforts in enabling BigOWLIM to support the SPARQL Update specification and, more generally, SPARQL version 1.1, [16]. Initially the efforts were to achieve this by integrating BigOWLIM with Jena[9], which already supported SPARQL 1.1. Version 3.4 of BigOWLIM was the first one to include Jena integration and SPARQL 1.1 support. With version 3.5, released at the end of March 2011, all functionality, even the advanced features of BigOWLIM is available through the Jena APIs and through SPARQL. An intermediate version was recently provided for independent evaluation and demonstrated outstanding results in the Berlin SPARQL Benchmark, [8]. There is ongoing work to extend the open-source Sesame framework with SPARQL 1.1 support – this will enable BigOWLIM to deliver even better performance with respect to SPARQL 1.1.

At the current phase of the project however, the SPARQL Update functionality will not be available through the public end-points and interfaces, because it is still necessary to properly define the requirements for access control and implement the corresponding policies. Without access control, free public access to interfaces that allow data modification is likely to result in an unpredictable and unusable service. This reveals the need to carefully investigate the requirements in order to deliver the best balance between freedom of modification and stability.

3.1 Linked Data Publishing

As outlined in section 1.1, publishing RDF data as “linked data” requires the following:

1. The “physical” addresses of the published pieces of data should be the same as the “logical” addresses, used as RDF identifiers (URIs);
2. Upon receiving an HTTP request, the server should return a set of triples that describe the resource;
3. Re-use identifiers and vocabulary from other datasets that are also linked data.

[9] http://jena.sourceforge.net/

Meeting the first requirement is only a matter of properly selecting the namespaces, used for constructing the URIs. Within RENDER, data will be published in the sub-folders of http://data.render-project.eu/ URL[10]. The HTTP requests to this sub-domain will be directed to a Forest-based service, which is used to provide access to the RENDER data (see section 2); Forest provides support for linked data publishing and can answer HTTP GET requests in accordance with the "linked data etiquette". The separate bodies of data will be loaded in the repository in one or several named graphs, dedicated to them, so that they can be managed separately, but also queried together with the rest of the data.

To meet the third requirement, if the datasets subject to publishing are not already properly linked to other linked data datasets, the standard procedure will be to get them linked to one or more of the dataset in the Reference Knowledge Stack (RKS, see section 1.5). This approach is appropriate for the following reasons:

- RKS already includes several of the most popular datasets of linked data, e.g. DBPedia, Freebase, Geonames and MusicBrainz, and many of the most popular ontologies and schemata (e.g. FOAF and SKOS);
- RKS does not limit the diversity because it includes multiple alternative references for the most popular concepts. For instance, most of the locations can be found in more than three of the datasets in RKS (DBPedia, Freebase, OpenCyc, Geonames). Most of the popular classes and relationships are also available through several ontologies and schemata (PROTON, UMBEL, OpenCyc, DBPedia).

[10] In special occasions partners will be allowed to publish linked data in different sub-domains, where linked data requests can be properly handled.

4 Conclusion and Future Work

This deliverable documents the data publishing and management web interfaces to be provided in the beginning of the RENDER project. The data will be accessible for read-only queries and exploration by human users through the Forest framework, which offers a range of different user interfaces for this purpose. A SPARQL end-point will allow for programmatic (API-level) access.

As a start, the public RENDER data service will expose the Reference Knowledge Stack – a combination of several datasets and ontologies, selected and interlinked so that they provide: various types of vocabularies (a choice between different design principles, granularities, etc.), multiple alternative references for most of the popular concepts, and good linking to the LOD cloud.

The publishing, management, and modifications of the data will take place though the SPARQL Update specification. To make this possible Ontotext extended the functionality of BigOWLIM so that it could handle SPARQL ver. 1.1, including updates. The performance of this extended version of BigOWLIM was already proven – it demonstrated the best overall performance across all repositories subject to a recent independent evaluation with respect to the Berlin SPARQL Benchmark, [8].

In the course of the project, the requirements for data management will be analysed in order to implement the optimal access control policy that ensures a balance between freedom and ease of use, on one hand, and good reliability of the public data services on the other.

References

[1] Bechofer, S, van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. *OWL Web Ontology Language Reference*. In: Dean, M., Schreiber, G. (Eds.), W3C Recommendation, February 10, 2004. http://www.w3.org/TR/owl-ref/.

[2] 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. (2010)

[3] Bergman, M. *Announcing a Major, New UMBEL Release.* http://www.mkbergman.com/930/announcing-a-major-new-umbel-release/ November (2010)

[4] Berners-Lee, T. *Design Issues: Linked Data*. http://www.w3.org/DesignIssues/LinkedData.html (2006)

[5] Bishop, B.; Kiryakov, A.; Ognyanoff, D.; Peikov, I.; Tashev, Z.; Velkov, R. *FactForge: A fast track to the web of data*. Submission for Semantic Web Journal, Special Issue: Real-World Applications of OWL. http://www.semantic-web-journal.net/content/new-submission-factforge-fast-track-web-data. To appear. (2011)

[6] 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)

[7] Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; Hellmann, S. *DBpedia – A Crystallization Point for the Web of Data.* Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Issue 7, Pages 154–165, 2009.

[8] Bizer, Ch., Schultz, A. *BSBM V3 Results.* (February 2011). http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/results/V6/index.html#comparison

[9] Brickley, D., Guha, R.V, (eds.) *Resource Description Framework (RDF) Schemas*. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-schema/ (2004)

[10] Clark, K.G.; Feigenbaum, L.; Torres, E. (eds.) *SPARQL Protocol for RDF*. W3C Recommendation 15 January 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-protocol-20080115/

[11] Cycorp. *OpenCyc*. http://www.cyc.com/cyc/opencyc

[12] 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. (2010)

[13] Ding, Li., Finin, T., Peng, Y., da Silva, P. , McGuinness, D. *Tracking RDF Graph Provenance using RDF Molecules.* http://ebiquity.umbc.edu/_file_directory_/papers/178.pdf, (2005)

[14] Giasson, F.; Bergman, M.; eds.: *Upper Mapping and Binding Exchange Layer (UMBEL) Specification.* http://umbel.org/specifications/full-specification, version 0.8, Nov 2010. (2010)

[15] Grinberg, M., Damova, M., Kiryakov, A. *D1.2.1: Initial Data Integration*. Deliverable of EU-FP7-ICT-2009-257790 project RENDER (2011).

[16] Harris, S.; Seaborne, A. (eds.) *SPARQL 1.1 Query Language.* W3C Working Draft 14 October 2010. http://www.w3.org/TR/2010/WD-sparql11-query-20101014/

[17] Heim, P; Hellmann, S; Lehmann, J; Lohmann, S; Stegemann, T. *RelFinder: Revealing Relationships in RDF Knowledge Bases*. In Proc. of the 4th International Conference on Semantic and Digital Media Technologies SAMT 2009.

[18] 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).

[19] 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. (2009)

[20] Li, Y., Cunningham, H., Roberts, A., Kiryakov, A., Momtchev, V., Greenwood, M., Aswani, N., Damljanovic, D. *Selection Components (report accompanying two software deliverables).* LarKC project deliverable D2.2.1, 2.5.1, (2009)

[21] Manola F., and Miller, E. (eds.): *RDF Primer*. W3C Recommendation, 10 Feb 2004, http://www.w3.org/TR/rdf-primer/, (2004)

[22] Momchev, V., Assel, M., Cheptsov, A., Bishop, B., Bradesko, L., Fuchs, C., Gallizo, G., Kotoulas, S., and Tagni, G. *D5.5.3 Report on platform validation and recommendation for next version*. LarKC EU-IST-2008-215535, (2010)

[23] Newman, A.; Li, Y.-F.; Hunter, J. *A Scale-Out RDF Molecule Store for Improved Co-Identification, Querying and Inferencing.* In The 4th International Workshop on Scalable Semantic Web knowledge Base Systems (SSWS) 2008, Karlsruhe, Germany, (2008)

[24] Ontotext. BigOWLIM User Guide, ver. 3.4. http://www.ontotext.com/owlim/BigOWLIM_user_guide_v3.4.pdf

[25] Page, L.; Brin, S.; Motwani, R., Winograd, T. The PageRank citation ranking: Bringing order to the Web. http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=1999-66&format=pdf&compression. (1999)

[26] Prud'hommeaux, E., Seaborne, A: *SPARQL Query Language for RDF*, W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/ (2008)

[27] Schenk, S.; Gearon, P. *SPARQL 1.1 Update*. W3C Working Draft 26 January 2010. http://www.w3.org/TR/2010/WD-sparql11-update-20100126/

[28] ter Horst, H. J.: *Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity*. In Proc. of ISWC 2005, Galway, Ireland, LNCS 3729, pp. 668-684, November 6-10, (2005)

[29] 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)

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

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

[32] World Wide Web Consortium (W3C): *Linking Open Data*. W3C SWEO community project home page, as of January 2010. http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData (2010)

