FactForge: A fast track to the Web of data

Editor(s): Michel Dumontier, Carleton University, Canada; Rinke Hoekstra, Universiteit van Amsterdam, The Netherlands
Solicited review(s): Thorsten Liebig, derivo GmbH Ulm, Germany; Eric Prud'hommeaux, W3C, Cambridge, MA, U.S.A.; Michel Dumontier, Carleton University, Canada
Open review(s): Aidan Hogan, DERI, National University of Galway, Ireland

Barry Bishop[a*], Atanas Kiryakov[a], Damyan Ognyanov[a], Ivan Peikov[a], Zdravko Tashev[a], Ruslan Velkov[a]
[a]Ontotext AD, 135 Tsarigradsko Chaussee, Sofia 1784, Bulgaria

Abstract. The advent of Linked Open Data has seen a large number of structured datasets from various domains made available to the public. These datasets are seen as a key enabler for the Semantic Web, where applications can consume and combine this data in powerful and meaningful ways. However, the uptake of Linked Data during this 'introductory phase' is hampered in ways similar to the uptake of any new technology - until the technology has found widespread use, the range of opportunities for exploiting it is limited and until the opportunities are fully explored, the uptake of the technology is restricted. FactForge is a free, publicly available service that provides an easy point of entry for would-be consumers of Linked Data. This Web application is based on OWLIM, a high performance semantic repository that offers outstanding RDF data management and reasoning capabilities based on OWL. The data-exploration functionality provided by FactForge exploits the advanced features of OWLIM to allow users to combine SPARQL with various full-text search and ranking functions for powerful, user-guided data-mining over a number of the most popular LOD datasets. This paper gives an overview of FactForge, its many unique capabilities and its role within the emerging trend for the exploitation of Linked Open Data using OWL-based inference.

Keywords: Knowledge representation, Linking Open Data, OWL, OWLIM, RDF, reasoning, SPARQL, triple-store

1. Introduction

'Linked Data' is defined by Tim Berners-Lee [4], as a number of RDF graphs, published so that they can be navigated across servers by following the links in the graph in a manner similar to the way the HTML Web is navigated. Therefore, the publishers of Linked Data should comply with four simple design principles:

1. Use URIs as identifiers for things;
2. Use HTTP URIs, so that these identifiers can be looked up;
3. Provide useful information when a URI is looked up;
4. Include links to URIs from other datasets.

Although not related to semantics, the Linked Data concept turns into an enabling factor for the realization of the Semantic Web [3] as a global web of structured data around the Linking Open Data initiative introduced in section 1.1. There are various obstacles for reasoning with Linked Data and these are related to the scale and nature of such data. In order to provide context for the experiment presented in this paper, section 1.2 gives a brief overview of the state of the art in scalable reasoning. Section 2 makes a proposal for a 'reason-able' view as a practical approach for reasoning with Linked Data.

The major contribution of this paper is a 'reasonable' view called FactForge [13], presented in section 3, which allows mining and navigation of large-scale general knowledge datasets. This builds upon previous work under the name 'Linked Data Semantic Repository' [23]. Sections 4 and 5 give details about the process of materialization of the deductive

[*] Corresponding author. E-mail: barry.bishop@ontotext.com.

closure of the selected datasets within FactForge, performed by the BigOWLIM [22] semantic repository. Finally, an analysis of the results is provided in section 6 and a discussion on future work is given in section 7.

The results reported here are based on work performed within the European research projects RASCALLI [36] and LarKC [25], in which FactForge is designed and used as a test-bed for scalable reasoning [24] and for the modelling of incomplete context-aware reasoning based on spreading activation and priming [41]. The latter experiments were extended to use 'priming' for pre-selection of relevant subsets of data for Web-scale reasoning [43].

1.1. Linking Open Data

Linking Open Data (LOD) [28] is a W3C Semantic Web Education and Outreach community project aiming to extend the Web by publishing open datasets as RDF [31] and by creating RDF links between data items from different data sources. The central dataset of LOD is DBPedia – an RDF extract of the Wikipedia open encyclopaedia – which serves as a 'hub' in the LOD graph, because of the many mappings between it and the other LOD datasets. Currently LOD contains more than 200 datasets[1], totalling nearly 30 billion statements, joined together with many millions of link statements.

1.2. Scalable Reasoning

For most of the popular knowledge representation (KR) formalisms and ontology languages, the worst case complexity of the algorithms for basic reasoning tasks indicates that they are intractable. Such algorithms can not therefore be applied to very large scale knowledge bases and datasets.

RDFS [19] is a schema definition language, much of which is used to define OWL [11] an ontology language for the Semantic Web. RDFS and OWL standardize the epistemology, vocabulary and syntax of the ontologies and the data encoded with respect to them. Yet, the semantics of RDFS and the various dialects of OWL are still quite diverse.

OWL DL [11] is a description logic that is well established in the semantic Web community, however many reasoning activities associated with this representation formalism have exponential worst case complexity and are considered intractable for large datasets. The most scalable experiments with sound and complete OWL DL reasoning are in the range of 5 million statements, under the UOBM [30] benchmark as reported in [21]. Inconsistency checking with respect to OWL DL has been performed, under specific constraints, against 60 million statements, as presented in [38].

OWL Horst is a partial-axiomatisation of OWL RDF-based semantics defined in [40] as an extension of RDFS semantics [19] towards supporting some, but not all, OWL primitives. Ter Horst defines a rule language called R-entailment in which both the body and the head of the rule are RDF graph patterns, described via statements, which can contain URIs, blank nodes, and variables in any position, as well as literals in the object position; blank nodes are not allowed in the body; all variables in the head of the rule should also appear in its body. This OWL dialect is defined as a set of R-entailment rules, named pD* entailment. OWL Horst is representative for a class of OWL dialects defined by systems of rules similar to R-entailment, of which the OWL2 RL rule language [32] is an example. Such OWL dialects place stringent requirements on the range of possible rules in order to bound the reasoning complexity.

As presented in [24], there are plenty of systems (AllegroGraph [1], BigDATA [5], BigOWLIM [22], DAML DB [8], ORACLE 11g [34]) which can perform reasoning with languages/fragments of a similar complexity to OWL Horst over datasets with a maximum size of between one and ten billion explicit statements. However, there are no systems that scale an order of magnitude greater than this or add significantly more expressivity at this scale, except for highly specialised, massively parallel systems, such as WebPIE [43]. From this we make the assumption that a current practical bound for expressivity and scalability, is a language/fragment as complex as OWL Horst up to some billions of RDF statements.

2. Reasoning with Linked Data

There are various problems related to reasoning with Linked Data. Some of the major issues include:

- Most of the traditional reasoning setups implement sound and complete inference under the so-called "closed-world assumption": the knowledge is considered complete, so if a specific fact is not known or inferable, it is not true. Such setups are irrelevant in an environment where the knowledge is incomplete by design and logical consistency is not guaranteed;

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/state/

- Some of the datasets of LOD, or at least some parts of them, are not suitable for reasoning. It seems that many data publishers use OWL and RDFS vocabulary without properly understanding their formal semantics leading to modelling errors that cause problems during inference, such as providing multiple values for functional properties or using properties with the wrong type of individual (and hence infer the wrong class membership due to the domain of the property);

- Some of the datasets are derived by the means of text-mining and, due to the intrinsic limitations in the accuracy of the extraction techniques, include incorrect information. For instance, the YAGO module of DBPedia contains plenty of faulty classifications of Wikipedia articles. Such inaccuracies are of a relatively small number and probably not a serious problem for human readers exploring DBPedia. However, they can lead to significant noise and inconsistencies after reasoning;

- Although reasoning with data distributed across different World-Wide-Web servers is possible, it is nearly always much slower than reasoning with local data.

Reason-able views represent an approach for reasoning with the Web of Linked Data, introduced in [43]. We call a *reason-able view (RAV)* an assembly of independent datasets that can be used as a single body of knowledge (referred to as an *integrated dataset*) with respect to reasoning and query evaluation. The integrated dataset represents the union of the independent datasets or versions of those, where parts of the original datasets could be excluded or refined in order to meet reasonability or some other criterion.

The notion of "reasonability" above means that the integrated dataset has certain specific qualities with respect to a specific reasoning task and language. Examples for reasonability criteria could be "consistent with respect to OWL Lite" or "to allow RDFS entailment within O(n) time and space" – note that due to the materialization approach of BigOWLIM, the semantics used is fixed at load time.

We define a Linked Data *reason-able view* (*RAV*) as a reason-able view where:

- All the datasets in the view represent Linked Data – see section 1;
- A single reasoning strategy is applied to all datasets;
- There are entities in each dataset that are connected to entities in at least one of the other datasets.

Considering the size of the LOD datasets (see section 1.1), in order to make query evaluation and reasoning practically feasible, the integrated dataset of a linked RAV should be loaded in a single repository (even if it employs some sort of distribution internally). Such a linked RAV can be considered as an index, which caches parts of the LOD cloud and provides access to the datasets included in a manner similar to the one in which Web search engines index Web pages and facilitate their usage.

As a final practical consideration, an RAV may hold datasets that are updated frequently. If this is the case then the database used to store it should be capable of being updated at least as frequently and the appropriate mechanisms put in place to ensure that updates from sources are propagated in a timely fashion. For datasets of a general knowledge nature, this is unlikely to be necessary, but for data sets that hold continuously changing data, such as news streams, then this must be taken in to account.

3. FactForge

FactForge (known in previous versions as the Linked Data Semantic Repository) is a reason-able view to the Web of Linked Data, made up of eight of the central LOD datasets, which have been selected and refined in order to serve as a useful index and entry point to the LOD cloud and to present a good use-case for large-scale reasoning and data integration. The design objectives for FactForge are as follows:

1. Consistency with respect to the formal semantics;
2. Generality – no specific domain knowledge should be required to comprehend most of the semantics;
3. Heterogeneity – data from multiple data sources should be included;
4. Reasonability with respect to OWL2 RL (see section 4 for details).

3.1. Datasets

FactForge includes the following LOD datasets:

- **DBPedia** [2] is an RDF dataset derived from Wikipedia, designed to provide as full as possible coverage of the factual knowledge that can be extracted from

Wikipedia with a high level of precision. It serves as a hub for the LOD project;

- **Freebase** [16] is a dataset containing information about 11 million things, including movies, books, TV shows, celebrities, locations, companies and more;
- **Geonames** [17] is a geographic database that covers 6 million of the most significant geographical features on Earth (e.g. countries, populated places, mountains, rivers, and bridges), characterised by coordinates and relations to other features (e.g. 'parent feature' in which the feature is nested);
- **UMBEL** [42] is a lightweight ontology structure, essentially, a hierarchy of about 20,000 classes, derived from OpenCyc and mapped to DBPedia. The classes range from general philosophical notions like **`TangibleThing`** to very specific classes like **`AbaCloth;`**
- **Wordnet** [45] is a lexical knowledge base that covers about 150,000 English words. Wordnet defines the meanings of English words by grouping them into sets of synonyms, called synsets. Each synset expresses a distinct concept. The words linked to a given synset are synonyms with respect to the meaning of the lexical concept represented by this synset. A word can have multiple meanings, i.e. it can be associated with multiple synsets. More general terms are associated with less general terms through hyponym-hypernym relations. FactForge uses the W3C's Wordnet RDF/OWL representation [46];
- **CIA World Factbook** [7] is a collection of structured data, including statistical, geographic, political, and other information about all countries;
- **Lingvoj** [26] provides descriptions of the most popular human languages; currently it contains information about more than 500 languages;
- **MusicBrainz** [33] (RDF from Zitgist) contains comprehensive music information suitable for browsing or useful for tagging.

3.2. Ontologies

The connectivity in FactForge is facilitated by DBPedia (which provides linksets to GeoNames, lingvoj, and Wordnet) and by UMBEL (which is linked to DBPedia). These link sets are also loaded in to FactForge along with the following ontologies and schemata:

- **DCMI Metadata Terms** [12] (Dublin Core - DC) is a relatively small, but very popular metadata schema. It defines attributes (e.g. author/contributor, date of publication, language, etc.) that can be used to describe information resources;
- **SKOS** [39] (Simple Knowledge Organization System) represents a relatively simple RDF schema that allows describing taxonomies of concepts linked to each other by any sort of subsumption hierarchy. The most important properties defined by SKOS are **`skos:broader`** and **`skos:narrower`**, defined as inverses of each other. The subsumption semantics of these relationships is more appropriate for the encoding of "topic ontologies" and subject classifiers as compared to **`rdfs:subClassOf`**.
- **RSS** [37] is an RDF schema designed to enable syndication of machine-readable information about updates from Web sites;
- **FOAF** [15] is a project aimed at creating a network of machine-readable personal profiles published on the Web. In essence, the FOAF ontology defines the attributes of these personal profiles, which, in turn, allows for publication of contact information and links to other profiles.

3.3. Data Access Methods

FactForge provides several methods to explore the combined dataset that exploit some of the advanced features of BigOWLIM.

Firstly, 'RDF Search and Explore' allows entities to be searched by keyword (treating them as text strings) with a real-time auto-suggest feature ordered by 'RDF Rank' (a feature similar to Google's Page Rank [6] that calculates a node's relative importance based on the number of ways it is related to other nodes). The results page shows all triples where the selected node appears as the subject, predicate or object, together with the preferred label, RDF Rank indicator, image, etc.

Secondly, a SPARQL [35] page allows users to write their own queries with clickable options to add each of the known namespaces. Many interesting queries are provided as examples. The results are presented in a conveniently formatted table with the option to download results in various formats (SPARQL/XML, JSON, etc).

Lastly, a graphical search facility called ‘RelFinder’ [20] that discovers paths between selected nodes. This is a computationally intensive activity and the results are displayed and updated dynamically during each iteration. 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.

4. Reasoning Setup

The ‘reasonability criteria’ (see section 2) for FactForge were defined with respect to the OWL2 RL rule language [32]. Formally, FactForge should be correct using forward-chaining reasoning, which includes entailment and consistency checking with respect to OWL2 RL.

Furthermore, the results of the inference should be consistent with ‘common sense’ without specific assumptions about the context of interpretation. In other words, the deductive closure should not include statements which go against ‘common sense’, under the style and level of consensus similar to that of Wikipedia. We define ‘common sense’ to mean some information that is clearly wrong when interpreted by a person. For example, it is logically consistent to say that a ‘hotel’ is a subclass of ‘whale’, but most people will immediately declare this as false. It follows that the ‘common sense’ property of a dataset is very hard to determine without surveying a number of people with all explicit and entailed knowledge. The conclusion in section 7 hints that so far this has only been attempted in a very ad hoc manner.

The BigOWLIM semantic repository is used to load the datasets and perform forward-chaining and materialization. This repository uses a rule language that supports R-entailment (see section 1.2) and can be configured to perform forward-chaining using predetermined rule-sets.

For improved loading performance, FactForge uses BigOWLIM's optimized rule-set for OWL Horst [40] which is an extension of RDFS entailment. However, the optimized rule-set excludes the trivial inferences of rdf1, 4a, 4b that have no equivalent in OWL2 RL. The remaining semantics are entirely captured in OWL2 RL, except for rdfs12 (which is not relevant to the FactForge dataset) and rdfs13 (which is captured in OWL2 RL by axiomatic triples for the XSD data types). The remaining inferences that rdfs13 would produce when applied to the FactForge dataset are in fact already present. Hence we conclude that using the optimized OWL Horst rule-set with the FactForge dataset leads to a model that is entirely consistent with the application of the OWL2 RL rule-set to the same data. The use of this rule-set allows the entire dataset to be loaded and forwarding chaining reasoning to be performed in approximately four days.

In order to determine that the OWL Horst rule set is not only consistent with OWL2 RL applied to the same dataset, but also sufficient we conducted a loading experiment using the OWL2 RL rule-set. The loading time was considerably longer, approximately four-fold, but the total number of statements was only slightly larger (less than 1%). It was not possible to determine what further inferences the OWL2 RL rule-set produced, but we conclude that the OWL Horst rule-set is sufficient to capture the semantics of the FactForge dataset to a satisfactory accuracy.

4.1. owl:sameAs optimization

The loading speed and query performance of FactForge benefit from a specific feature of BigOWLIM that allows for the efficient handling of **`owl:sameAs`** statements. **`owl:sameAs`** is an OWL predicate used to declare that two different URIs denote one and the same thing. It is often used to align identifiers from different datasets that refer to the same thing.

BigOWLIM combines identifiers for the same things in to equivalence classes that dramatically reduce the indexing space required and yet still allow all correct query solutions to be enumerated. As can be seen from Table 3, this approach reduces the number of statements to be indexed by some 78%.

In addition to an ‘Include inferred’ check box on the SPARQL query page, FactForge also has an ‘Expand results over equivalent URIs’ checkbox that exploits the BigOWLIM option to enumerate over equivalent URI’s or not. When this option is deselected, only one URI is used for a particular resource, selected from the resource’s equivalence class. This can make a dramatic difference to the number of query results returned, where statements that differ only by the substitution of equivalent URIs are removed from the result set.

4.2. Logical consistency

A knowledge base may be inconsistent with respect to the semantics of OWL2 RL (rule language) in a number of ways, e.g. when an individual is a member of two disjoint classes or when two individuals are both *the same* as each other and *different from* each other. The entailment rules [32] include many consistency checks that derive 'false' when an inconsistency is detected.

BigOWLIM includes a consistency checking mechanism that uses rule-like expressions and can detect inconsistencies whenever statements are committed to the repository. However this incurs additional overhead during loading, when all inferred statements are computed. After loading, the repository will be used in a read-only manner to answer the queries passed to it from the FactForge user interface. Therefore, the approach chosen is to turn off inconsistency checking at load time and check for inconsistencies after loading is completed. This is achieved by executing SPARQL queries that have the same form as the premises of the OWL2 RL consistency checking rules. If future incarnations of FactForge are updated more frequently, for example if the underlying datasets change rapidly, then consistency checking can be switched on in order to catch inconsistencies as the conflicting statements are committed.

Table 1 Dataset loading and inference statistics

Dataset	**Explicit Indexed Triples** ('000)	**Inferred Indexed Triples** ('000)	**All Indexed Triples** ('000)	**Entities** (nodes) ('000)	**Inferred closure ratio**
Schemata and ontologies	11	7	18	6	0.6
DBPedia (SKOS categories)	2,877	42,587	45,464	1,144	14.8
DBpedia (owl:sameAs)	5,544	566	6,110	8,464	0.1
UMBEL	5,162	42,212	47,374	500	8.2
Lingvoj	20	863	883	18	43.8
CIA Factbook	76	4	80	25	0.1
Wordnet	2,281	9,296	11,577	830	4.1
Geonames	91,908	125,025	216,933	33,382	1.4
DBpedia core	560,096	198,043	758,139	127,931	0.4
Freebase	463,689	40,840	504,529	94,810	0.1
MusicBrainz	45,536	421,093	466,630	15,595	9.2

5. Loading and Materialization Statistics

Dataset loading and inference statistics are presented in Table 1. The first column lists the datasets (or parts of them) in the order in which they were loaded into the repository. The number of triples listed in the Explicit Indexed Triples column indicates the increase in the number of statements in the BigOWLIM indices after the dataset has been loaded. Note that some data providers claim that their datasets contain an amount of statements slightly different from the one presented in the table.

Table 2 provides a summary of loading and inference statistics for all the datasets. This collection includes 283 million entities (nodes in the RDF graph). Further processing then computes the RDF rank, text snippet and preferred label for each node. The final dataset statistics after this post-processing are shown in Table 3. At this point, the dataset includes 404,796,665 entities.

Table 2 Summary of dataset statistics after loading

Total number of statements after loading	**Value (millions)**
Indexed explicit statements	1,177

Indexed inferred statements	881
Indexed statements (explicit + inferred)	2,058

The larger number of retrievable statements in Table 3 is a result of the `owl:sameAs` optimization discussed in section 4, where the optimization has ‘compressed’ more than 7.7 billion statements, reducing the size of the indices by 78%.

Table 3 Summary of dataset statistics post-processing

Total number of statements after post-processing	**Value**
Added (preferred labels and ranks)	179,812,809
Indexed	2,237,550,383
‘Compressed’ through sameAs optimization	7,760,929,834
Unique retrievable statements <s p o g>	9,818,667,408

6. Analysis of the Results

The most important outcome of this experiment is that it has shown that it is possible to build a reasonable view that matches the requirements set forth in section 3:

- FactForge successfully integrates several of the central LOD datasets into a single body of general knowledge;
- FactForge contains heterogeneous datasets. The nature of the knowledge encoded in them varies from encyclopaedic (DBPedia), through geographic (Geonames), to linguistic (Wordnet and lingvoj) and taxonomical (UMBEL);
- The facts inferred from the knowledge in FactForge look reasonable; this conclusion is drawn from intensive exploration and querying of FactForge over many months. The only exceptions discovered are the SKOS categories in DBPedia;
- The integrated dataset of FactForge is logically consistent apart from one type of inconsistency that is described in section 6.4.

In most cases, the high ratio of expansion in the deductive closure is due to long chains of statements over transitive properties that are used to construct hierarchies. This is the case with the nesting of locations over the `gno:parentFeature` in Geonames, the class hierarchy in UMBEL, and the category hierarchy in DBPedia.

6.1. Fixing the category hierarchy in DBPedia

The most difficult problem with respect to ensuring “reason-ability” for FactForge is related to the category hierarchy in DBpedia. This hierarchy includes around half a million categories linked with almost one million relations. The hierarchy is defined via `skos:broader` relations and in many cases the actual relationship is either too weak and insignificant or simply inaccurate. Often concepts with overlapping meanings are incorrectly encoded as a pair of broader-narrower categories, instead of just related categories, which combined with the extensive use of auxiliary categories and multiple-inheritance, results in an extremely tangled hierarchy that contains many cycles related through transitive subsumption relationships. The result of such cycles is that after materialization all categories in a cycle become equivalent to one another. During this experiment just over two thousand simple cycles were detected, over half of which were trivial (a category being marked as broader to itself) and these were easily discarded. The remainder were analyzed manually, which resulted in nearly one thousand relations being changed from `skos:broader` to `skos:related`. The resulting graph contains `skos:broader` paths with lengths ranging from 1 to nearly two hundred.

6.2. Differences between FactForge and LUBM with respect to inference

Generally, one can observe that reasoning with real-world data appears to be much more challenging, compared to synthetic tests like LUBM [18]. The differences between FactForge’s integrated dataset and the datasets generated and used in LUBM can be summarized as follows:

- The RDF graph in LUBM has a star-like topology: the sub-graph for each university is connected only to the sub-graphs of the LUBM ontology and the first university which stands in the centre of the ‘star’. This allows for easy partitioning and caching in the process of loading. In contrast, the FactForge combined dataset is highly irregular and it is possible that there is no easy way to isolate and cache only the most used parts.

- The full deductive closure of LUBM expands the number of statements by 70%, while in FactForge the expansion is 834%. The major reason for this being the long chains of predicates related over transitive properties and the intensive use of **`owl:sameAs`**.
- In FactForge more than 100,000 different predicates are used, mostly due to the encoding style of DBPedia. On the other hand, in LUBM there are just a handful of predicates used, which allows for efficient loading and querying of LUBM in a repository configuration where indices with predicates as the primary sorting criteria are not maintained.

6.3. New data access opportunities

The integration of so many of the central LOD datasets combined with reasoning and the advanced data access features of BigOWLIM brings exciting new opportunities to discover previously hidden knowledge, specifically, the ability to formulate expressive SPARQL queries and integrate full-text search and ranking.

Consider the query in Fig. 1, which can be found as an example on the SPARQL page of FactForge – shown here without prefixes for brevity. This query finds individuals born in Germany that are designated as entertainers and orders the results by RDF rank (measure of interconnectedness):

```
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
```

Fig. 1 SPARQL query to find popular German entertainers

This query makes use of the datasets DPPedia, Geonames, UMBEL and MusicBrainz, requiring inference over types, sub-classes, and transitive relationships. Before FactForge, answering this kind of query in real-time was not possible. Curiously, the first result is F.W.Nietzsche, due to the little known fact that Nietzsche was a musician and composer as well as a philosopher. Knowing this, it is no surprise that the answer to the question “who are the most interconnected individuals from Germany who are *also* entertainers” should return Nietzsche first.

Another interesting query example is The Modigliani Test[2]. Richard McManus wrote that “the tipping point for the Semantic Web may be when one can deliver – using Linked Data – a comprehensive list of locations of original Modigliani art works”. Although there is still some way to go before a ‘comprehensive’ list can be discovered using Linked Data, it seems that FactForge can at least be used to get some answers to this question – see the query in Fig. 3, where the prefixes are omitted for brevity. At the current time, eight painting titles and locations are returned. Again, this query is given as an example on the SPARQL page of FactForge.

6.4. Logical consistency

An examination of the predicates used in the FactForge datasets shows that there are many consistency checking rules that are made redundant due to the fact that some of the predicates and classes in their premises are simply not used. The absence of the any of the following classes:

```
owl:AllDifferent
owl:AsymmetricProperty
owl:IrreflexiveProperty
owl:Nothing
```

and any of the following predicates:

```
owl:assertionProperty
owl:complementOf
owl:differentFrom
owl:maxQualifiedCardinality
owl:propertyDisjointWith
owl:sourceIndividual
owl:targetIndividual
owl:targetValue
```

eliminates nearly all of the consistency checking rules. Furthermore, even though **`owl:maxCardinality`** appears often, it is never associated with a value of zero and so none of the cardinality checking rules can be triggered.

However, there are twenty occurrences of **`owl:disjointWith`** in the merged dataset and these lead to some difficulties. The rule **`cax-dw`**, which can be emulated using the SPARQL query shown in Fig. 2, when applied to the current dataset is indicating some 10,000 inconsistencies.

[2] http://www.readwriteweb.com/archives/the_modigliani_test_for_linked_data.php

```
SELECT * WHERE {
  ?c1 owl:disjointWith ?c2 .
  ?x rdf:type ?c1 .
  ?x rdf:type ?c2 .
}
```

Fig. 2 SPARQL query to check for inconsistencies caused by individuals being members of disjoint classes

7. Conclusion and Future Work

Several of the central LOD datasets were selected (approximately 1.117 billion statements), modelling errors were fixed (apart from some disjoint classes that have some common members) and the result was loaded in to a BigOWLIM semantic repository with nearly 180 million additional annotation statements. Forward-chaining inference was performed to materialize a further 881 million statements meaning that a total of 2.237 billion statements are indexed. Enumeration of **`owl:sameAs`** equivalence classes is performed at query time leading to some 9.8 billion retrievable statements.

Many months of ad hoc use has shown that the vast majority of the inferred statements match common sense expectations.

A reduced rule-set (optimized OWL Horst) was used for inference and an analysis of the rules and a comparison of the number of inferred statements shows that the chosen rule-set is sufficient to capture the semantics of OWL2 RL to within 1%. In other words, reasoning with respect to a more expressive dialect will not entail a significantly larger number of additional implicit statements.

```
SELECT DISTINCT ?painting_l ?owner_l ?city_fb_con ?city_db_loc ?city_db_cit
WHERE {
  ?p fb:visual_art.artwork.artist dbpedia:Amedeo_Modigliani ;
    fb:visual_art.artwork.owners [ fb:visual_art.artwork_owner_relationship.owner ?ow ] ;
    ff:preferredLabel ?painting_l.
  ?ow ff:preferredLabel ?owner_l .
  OPTIONAL { ?ow fb:location.location.containedby
               [ rdf:type umbel-sc:City ; ff:preferredLabel ?city_fb_con ] } .
  OPTIONAL { ?ow dbp-prop:location ?loc.
             ?loc rdf:type umbel-sc:City ;
               ff:preferredLabel ?city_db_loc }
  OPTIONAL { ?ow dbp-ont:city [ ff:preferredLabel ?city_db_cit ] }
  FILTER ( bound(?city_fb_con) || bound(?city_db_loc) || bound(?city_db_cit) )
}
```

Fig. 3 SPARQL query to find the names and locations of Modigliani paintings

Future optimisations in the BigOWLIM semantic repository platform and a steady improvement in hardware will allow more datasets to be included. Depending on the frequency of updates of these new datasets, a more rigorous update cycle can by implemented that minimises the latency between updates in the original data source and these changes being manifest in the FactForge repository. Alternatively, if an update stream is available for rapidly changing data then these updates can be applied directly to the underlying BigOWLIM repository.

Other work involves experimenting with some applications of FactForge, e.g. semantic annotation of text with respect to the entities in FactForge or using it for query expansion for services like Flickr [14].

FactForge is one of the largest publicly available bodies of general knowledge (not specific to a particular scientific domain) against which inference has been performed. Visitors to the Website [13] can query and explore the data using the methods described in section 3.3. Other large, combined knowledge bases made available to the public include the LOD Cloud Cache [29] containing some 15.4 billion explicit statements including the entire data.gov catalogue [9] and Linked Life Data (LLD) [27] (also from Ontotext) that assembles a large fraction of the life-science-related datasets from LOD. LLD includes over 5 billion explicit triples from over 28 data sources plus additional link datasets.

Ontotext maintains a public demonstration service [13] that allows one to explore FactForge and evaluate queries against it through a Web interface. Applications can access FactForge via a SPARQL end-point. Such a setup would make a useful ‘backend’ for a lightweight client that browses Linked Data or annotates/enriches application data, e.g. a mobile application like DBPedia Mobile [10] that could use GPS position data to find nearby points of interest or provide information about the current region. Should FactForge become a useful asset for application developers, then Ontotext will make a cloud-based version available. This will allow applications to operate with their own dedicated BigOWLIM instance, datasets and bandwidth. Furthermore, it will permit applications to modify the bundled FactForge datasets or add their own data.

References

1. AllegroGraph RDFStore, page: http://www.franz.com/agraph/allegrograph/
2. Auer, S; Bizer, C; Kobilarov, G; Lehmann, J; Cyganiak, R; Ives, Z; DBpedia: A Nucleus for a Web of Open Data. In The Semantic Web, volume 4825 of LNCS, pages 722–735. Springer Berlin / Heidelberg, 2007.
3. Berners-Lee, T; Hendler, J; Lassila, O; *The Semantic Web*, Database and Network Journal 2006, Vol36; No3, pp. 7-10
4. Berners-Lee, T. (2006). *Design Issues: Linked Data*. http://www.w3.org/DesignIssues/LinkedData.html
5. BigData RDF Database, page: http://www.systap.com/bigdata.htm
6. Brin, S; Page, L; The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, Volume 30, Issues 1-7, Proceedings of the Seventh International World Wide Web Conference, April 1998, Pages 107-117, ISSN 0169-7552
7. CIA World Factbook, D2R Server for the CIA Factbook, homepage: http://www4.wiwiss.fu-berlin.de/factbook/
8. DAML DB, prototype implementation for a scalable, high-performance persistent store for DAML content, homepage: http://www.daml.org/2001/09/damldb/
9. DATA.gov, U.S. Federal Executive Branch datasets, homepage: http://www.data.gov/catalog
10. DBPedia Mobile, page: http://wiki.dbpedia.org/DBpediaMobile
11. Dean, M; Schreiber, G. – editors; Bechhofer, S; van Harmelen, F; Hendler, J; Horrocks, I.; McGuinness, D. L; Patel-Schneider, P. F.; Stein, L. A. (2004). *OWL Web Ontology Language Reference*. W3C Recom., 10 Feb. 2004. http://www.w3.org/TR/owl-ref/
12. Dublin Core Metadata Initiative, page: http://dublincore.org/
13. FactForge - The Fast Track to The Centre of the Web of Data, homepage: http://factforge.net/
14. Flickr, photo sharing website, page: http://www.flickr.com/
15. FOAF, The Friend of a Friend (FOAF) project, homepage: http://www.foaf-project.org/
16. Freebase, an entity graph of people, places and things, homepage: http://www.freebase.com/
17. GeoNames geographical database, page: http://www.geonames.org/
18. Guo, Y; Pan, Z; and Heflin, J. (2004). *An Evaluation of Knowledge Base Systems* for *Large OWL Datasets*. Journal of Web Semantics, 3(2), 2005, pp. 158-182. http://www.websemanticsjournal.org/ps/pub/2005-16
19. Hayes, P. (2004). *RDF Semantics*. W3C Recommendation 10 Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/
20. Heim, P; Hellmann, S; Lehmann, J; Lohmann, S; Stegemann, T; (2009) RelFinder: Revealing Relationships in RDF Knowledge Bases. In Semantic Multimedia, volume 5887 of LNCS, pages 182–187. Springer Berlin / Heidelberg, 2009.
21. Kiryakov, A. (2008). *Measurable Targets for Scalable Reasoning.* LarKC project deliverable D5.5.1. http://www.larkc.eu/deliverables/
22. B. Bishop, A. Kiryakova, D. Ognyanov, I. Peikov, Z. Tashev, and R. Velkov. Owlim: A family of scalable semantic repositories. Semantic Web Journal, 2(1):33–42, 2011.
23. Kiryakov, A; Ognyanov, D; Velkov, R; Tashev, Z; Peikov, I; LDSR: a Reason-able View to the Web of Linked Data, in: SemanticWeb Challenge (ISWC2009), 2009.
24. Kiryakov, A; Tashev, Z; Ognyanoff, D; Velkov, R; Momtchev, V; Balev, B; Peikov, I. (2009). *Validation goals and metrics for the LarKC platform.* LarKC project deliverable D5.5.2. http://www.larkc.eu/deliverables/
25. LarKC – European research project for Web scale reasoning, homepage: http://www.larkc.eu/
26. Lingvoj, Resources for the multilingual Semantic Web, homepage: http://www.lingvoj.org/
27. Linked Life Data, a semantic integration platform for biomedical data, homepage: http://linkedlifedata.com/
28. Linking Open Data, W3C Semantic Web Education and Outreach community project, page: http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
29. LOD Cloud Cache, OpenLink Software, page: http://lod.openlinksw.com/
30. Ma, L; Yang, Y; Qiu, Z; Xie, G; Pan, Y. (2006) *Towards A Complete OWL Ontology Benchmark*. . In York Sure and John Domingue, editors, The Semantic Web: Research and Applications, volume 4011 of LNCS, pages 125–139. Springer Berlin / Heidelberg, 2006. 10.1007/117622561 2.
31. Manola F., Miller, E. (eds.) (2004). *RDF Primer. W3C Recommendation 10 Feb 2004*, http://www.w3.org/TR/REC-rdf-syntax/
32. Motik, B; Cuenca Grau, B; Horrocks, I; Wu, Z; Fokoue, A; Lutz, C. (eds.) (2009). *OWL 2 Web Ontology Language Profiles*. W3C Recommendation 27 October 2009. http://www.w3.org/TR/owl2-profiles/
33. MusicBrainz, community music metadatabase, homepage: http://musicbrainz.org/
34. Oracle 11g database, page: http://www.oracle.com/us/products/database/index.htm
35. Prud'hommeaux, E; Seaborne, A; SPARQL Query Language for RDF. W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/
36. RASCALLI – European research project, page: http://www.ofai.at/rascalli/project/project.html
37. RSS, RDF Site Summary, page: http://web.resource.org/rss/1.0/spec
38. Schonberg, E; Srinivas, K; Kalyanpur, A; Cimino, J; Patel, C; Dolby, J; Kershenbaum, A; Ma, L; Fokoue, A. Matching Patient Records to Clinical Trials Using Ontologies. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 816–829, Berlin, Heidelberg, 2007. Springer-Verlag.
39. SKOS, Simple Knowledge Organization System, homepage: http://www.w3.org/2004/02/skos/
40. ter Horst, H. J. (2005) *Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity* . In Yolanda Gil, Enrico Motta, V. Benjamins, and Mark

Musen, editors, *The Semantic Web ISWC 2005*, volume 3729 of Lecture Notes in Computer Science, pages 668–684. Springer Berlin / Heidelberg, 2005.

41. Todorova, P., Kiryakov, A., Ognyanoff, D., Peikov, I., Velkov, R., Tashev, Z. (2009). *Spreading Activation Components.* LarKC project deliverable D2.4.1. http://www.larkc.eu/deliverables/
42. Upper Mapping and Binding Exchange Layer (UMBEL), homepage: http://www.umbel.org/
43. Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F., Bal, H.: OWL Reasoning with WebPIE: Calculating the Closure of 100 Billion Triples . In Aroyo, L., Antoniou, G., Hyvnen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., and Tudorache,T., editors, *The Semantic Web: Research and Applications*, volume 6088 of LNCS, pages 213–227. Springer Berlin /Heidelberg, 2010.
44. Velkov, R., Ognyanoff, D., Kiryakov, A. (2009). *Open-Domain Incomplete Reasoner.* RASCALLI project deliverable *D3b.* http://www.ofai.at/rascalli/
45. Wordnet, a lexical database for English, page: http://wordnet.princeton.edu/
46. Wordnet, W3C WordNet RDF/OWL Files, homepage: http://www.w3.org/2006/03/wn/wn20/