Data Integration System for Linked Open Data Space

© Kuznetcov Konstantin

Lomonosov Moscow State University
K.Kuznetcov@gmail.com

Abstract

This paper describes research-in-progress work on data integration system in Linked Open Data space. Proposed system uses concept of RDF identity links to interlink heterogeneous local data sources and integrate them into global data space.

Academic supervisor: Vladimir Serebriakov, serebr@ccas.ru.

1 Introduction

For the last few decades data integration has been one of the most actual problems of computer science. With the development of IT industry countless data sources emerged in the Internet. These data sources are heterogeneous in all possible ways. Effective usage of such data sources is impossible without automatical tools for data search, retrieval, publishing and transformation.

The original hypertextual web didn't suit well for automatical processing of data from heterogeneous sources spread across the web. This led to emergence of various microformats and web APIs, and finally the concept of Semantic Web. Sematic Web implies usage of standard stack of data formats and technologies intended to support data accumulation, structuring and exchange across the web. The most important of these technologies are RDF, RDFS, OWL and SPARQL.

From the practical point of view one of the most interesting Semantic Web initiatives is Linking Open Data project [8]. This project aims for quantitative filling of the web with data structured according to Semantic Web standards and for interlinking of semantic data sources. As a result, a global Linked Open Data space should be established, similar to hypertextual web of linked documents. Publishing data in Linked Open Data space encourages reuse of data, decreases data redundancy, maximizes its (real and potential) inter-connectedness and enables network effects to add value to data. Data providers can benefit from publishing their data in LOD space. Unfortunately, the process of publishing is not that simple and consists of several steps. Small organizations often cannot afford to transform their data into LOD- acceptable form and then support the published dataset. To the moment, there is no special software to support the full cycle of publishing and managing linked open datasets. A system to support integration of data from independent data sources into the web of linked data is required.

2 Related work

To the moment, there are quite few solutions that support variuos steps required to include one's data into Linked Open Data space, even though they are based on existing hypertext web technologies. And there is no system that includes all the functionality recommended by LOD project. The most complex solution available now is Virtuoso Universal Server [11] platform. It provides tools for representing data from different sources (relational databases, RDF-storages, Web APIs, etc.) as a single virtual database and supports RDF data publishing. Virtuoso offers SPARQL access to its data and such features as RDF-crawler and simple reasoner. Virtuoso can be extended in multiple ways, e.g. published RDF-data can be accompanied with voiD descriptors. Unfortunately all the extensions that are useful for linked data publishing are made on instrumental level and not on data model level. Virtuoso is commercial software with limited open-source community edition. Among other open-source solutions it is worth to mention D2R Server [3], which supports RDF data publishing from relational databases and SPARQL querying. The MASTRO [4] and the data integration system developed in Dorodnicyn Computing Centre of RAS [2] provide richer semantic formalisms compared to D2R and Virtuoso. However, the first system is bound to a single federative database and the second one implies that its sources share some common URIs. And both of them don't provide any means for publishing or interlinking with other RDF datasets.

However, both Virtuoso and D2R Server do not go beyond simple RDF publishing. Their RDF-resource interlinking capabilities are limited to URI generation from templates. In many cases using such templated URIs cannot expose identity relations between RDF resources from different datasets and truly interlink these datasets. There are several applications for RDF data interlinking and link supporting, such as SILK [12], LIMES, SemMF and DSNotify. But at the moment none of these applications provide means for integration of one's RDF into the whole Linked Data space. I.e. there are no toos that can automatically discover new related datasets in the web, the set up and support links to the resources in these datasets. Some proposals for such systems are made in [10].

The possibilities of non-trivial usage of generated linksets are yet to be explored. Very few applications take advantage of this feature of Linked Open Data space. It is worth to mention SPLENDID system [6] here, which uses linksets statistics to optimize federative SPARQL queries. Some semantic search engines also utilize voiD descriptions of linksets.

3 Problem statement

This article proposes a concept of automated data integration system in Linked Open Data space. Proposed system should

- Form the single dataset from multiple heterogeneous sources of structured or unstructured information in similar knowledge domain and support/update formed dataset;
- Discover and store links between resources from system dataset and resources from different Linked Open Data sets available on the Internet in RDF format, as well as implicit links between resources within system dataset;
- Publish system dataset in the Internet in RDF format and provide access to it via user interface and API;
- Provide users and external applications with unified query interface to all of system's data sources;
- Support different data source types (including relational databases and SPARQL endpoints) and support on-fly connection of new data sources;
- Include flexible ontology of knowledge domain that follows Linking Open Data project recommendations and can be extended to support new data sources.

4 System architecture overview

Proposed system will follow modular architecture and will consist of following components:

- Ontology of informational objects and links of interest to system data consumers and providers;
- Linking subsystem, that should discover and store links between resources from system's data sources and/or resources from external Linked Data sources;
- Publishing subsystem, which should provide users and applications with access to resources from system's dataset according to LOD project recommendations;
- Data integration subsystem, which will contain mechanism to uniquely identify system's resources both within system and in Linked Open Data space and provide uniform access to all system's resources. This subsystem will include a set of adapters that provide unified SPARQL access to system's data sources of different types (relational databases, Web APIs, etc.);
- Harvesting and extraction subsystem with a set of harvester components, which will gather data from system's sources of unstructured data (text files, scanned documents, etc.), transform it into structured form and store it. This system is a subject of a future work.

4.1 Ontology

The system uses OWL ontology to semantically organize objects and links that match the concepts of interest from knowledge domain of system's data sources. Ontology consists of core terms and imported modules which can be added in case when some resources in newly added data source require more precise definition. In Linked Open Data space ontology serves as system's data vocabulary, it is used to establish terminological outgoing links to external datasets and allows external applications to discover metadata to establish ingoing links. Following the principles of Linked Open Data, ontology is annotated in human language with such terms as rdfs:label or rdfs:comment. Ontology's terms should be defined in URI namespace controlled by the system. Ontology adapts common Linked Data vocabularies such as Dublin Core, FOAF, vCard, PRISM, SIOC, Creative Commons, BibTex, Schema.org. Core of ontology is based on ENIP RAS ontology.

4.2 Publishing subsystem

Publishing subsystem will serve as an entry point to the system for human users and Linked Data applications. It should dereference URIs of system's resources, i.e. return descriptions of the object or concept identified by these URIs. It can be achieved by using a mechanism called content negotiation. Depending on HTTP GET request header, publishing subsystem will return either HTML representation or RDF/XML (as required by Linked Data applications) representation of the resource.

Publishing subsystem will receive data from data integration subsystem. For dereferencing resource URI, following information should be requested from data integration subsystem:

- All the literal values of resource, all incoming and outgoing RDF links. This information can be retrieved with simple SPARQL queries with patterns {<URI> ?x ?y} and {?x ?y <URI>};
- Most likely the results of these simple requests will contain URIs of other system's resources. Linked Data applications often traverse URIs they find in RDF documents. Therefore to reduce the number of HTTP requests publishing subsystem should extend aforementioned requests to some depth, or by applying some explicitly stated rules;
- Information on ontology class to which the requested resource belongs and all its ancestors;
- Information on the dataset to which this resource belongs;

All the information retrieved from data integration subsystem will be represented as a set of RDF triples. In case of RDF document these triples will be merged into resulting RDF/XML document and returned to client. In other case the triples will be published as HTML+RDFa document generated from template. These templates can be specified in general form and then redefined for specific classes.

4.3 Data integration subsystem

A data integration subsystem will provide other subsystems or external agents with uniform access interface to all of the system's data sources. Requested information should be specified with SPARQL query. This subsystem will be responsible for presenting system's data as single dataset in Linked Open Data space. There are several approaches to data integration systems – data warehousing, data mediation, peer-to-peer systems. Proposed system is supposed to work with multiple strongly autonomous data sources; therefore it adapts data mediation architecture. The drawback of such systems (e.g. Virtuoso) is huge amount in network interactions required to produce query answer. Proposed system uses Linked Open Data principles to reduce this drawback.

Data sources will be connected to the system via adapters, which are SPARQL endpoints capable of querying data sources in terms of system ontology. These adapters should be generic, configurable components (e.g. JDBC adapter, REST API adapter). As opposed to existing data integration systems with semantic capabilities (e.g. Virtuoso with its Sponger cartridges), resources from different data sources won't be merged into single dataset by providing same URI to identical resources. Instead, in the spirit of Linked Open Data, every data source should be considered to contain unique resources and get its own sub-namespace (like http://<system_URL>/datasets/<source_id>). Adapter should confront every resource from its data source with HTTP URI from this namespace. Therefore we will be able to track resource origin by its URI. When new data source will be added to the system, its adapter will be configured by specifying generic adapter settings (e.g. JDBC connection string), general Dublin Core description of the source, topic of interest categorization, licensing information, etc. Adapter configuration also includes the set of ontology classes and properties to which data in these sources belongs. This information can be entered manually or obtained with SPARQL ASK request. Next, adapter configuration will be published as voiD [1] descriptor of dataset. All such datasets are subsets (in terms of voiD) of system's whole dataset. However, all of them will be accessed via single SPARQL endpoint. Such structure preserves autonomy and independence of data sources while integrating them all together in Linked Open Data space.

Execution of queries in data integration subsystem will be carried out as follows. The first step is a SPARQL query rewriting according to the axioms of ontology, as described in [2]. Then algebraic query optimization techniques are applied. The result of this phase in terms of descriptive logic is the union of conjunctive queries with simple constraints. In the second step the set of relevant data sources for each atom of each conjunctive query is determined according to configuration of adapters. If there are no data sources relevant to an atom, the entire conjunctive query is dropped. As a result, a union of conjunctive queries with atoms of different data sets will be obtained.

Traditionally, the next step in data mediation process is construction of the physical query plan and its execution. During execution of query with atoms related to different data sources the results of subqueries to these data sources are joined. However, in the proposed system subquery results can be joined on literal field values and not on the URIs, because data sources are presented in a form of independent Linked Open Data sets and do not share common URIs. If subqueries to different data sources are to be joined on URIs we will have to use the sets of links generated by linking subsystem between these data sources. Each conjunctive query is a graph pattern with vertices being either literal values or the URIs or variables, and edges are labeled with predicates in terms of different data sources. If two adjacent edges are labeled with predicates from different sources, it is necessary to refer to the linkset for this pair of sources and select resource pairs that satisfy a given part of graph pattern. By performing this operation on all the links in the conjunctive query, we will obtain the set of resource URIs that satisfies part of the pattern that defines relationships between different data sources. Then the subquery parts related to specific data sources will be executed by adapters with corresponding join variables being replaced by URIs from linksets. On this step traditional query optimization techniques can be applied again.

4.3 Linking subsystem

RDF documents published in the Linked Open Data space are required to contain outgoing links. These outgoing links are RDF triples with the subject being the URI of the resource from the local namespace and the URI of the object and / or predicate belonging to the namespace of another dataset. The most important type of outgoing links is identity links that point at URI aliases used by other data sources to identify the same real-world object or abstract concept. Identity links can use such predicates as owl: sameAs, rdfs: seeAlso or special SKOS terms. Although the uses of predicate owl: sameAs in the LOD space are often contrary to the semantics of OWL [7], its use is recommended by W3C Technical Architecture Group. Linking subsystem will be responsible for the discovery, storage and support of identity links. Properties of the link include pair of URIs, link generation time and method, date of last link check and similarity factor. When the link is published either owl:sameAs or rdfs:seeAlso predicate is used in the triple depending on similarity factor value.

Linking subsystem will work as follows. In the first step, the two data sources to be interlinked are found. For this pair an initially empty voiD linkset is created and published. When new data sources is added to the system the linksets between this new data source and all existing data sets from other sources are automatically created. A linkset between internal dataset and external

Linked Open Data set will be created in one of the following cases:

- The user can manually select a pair of datasets for linking;
- Relevant datasets can be discovered using HTTP referrer technique described in [9];
- Relevant datasets can set be discovered by linking subsystem itself by traversing links in external dataset that is already linked to one of internal datasets.

When two target datasets for interlinking will have been selected, the subsystem will clusterize datasets by classes and determines pairs of clusters to be interlinked. This should be done to reduce the number of pairwise comparisons of datasets elements. In the case of two internal datasets both of them are described by the same ontology, so that pairs of clusters contain instances of same ontology classes. In the case of linking to an external dataset the subsystem might select pairs of classes with help of different ontology mapping techniques [5], as well as using discovered or manually specified ontology mapping rules.

The third and final step of interlinking involves pairwise comparison of clusters elements to detect pairs of identity relations. These relations will be detected using SILK LSL language rules. In the case of internal data sources, rules will be declared together with the ontology and determine which instances of the same class are identical. In the case of an external dataset rules will be either specified manually, or derived from the existing rules and ontology mapping rules.

Complete binding is achieved by pairwise comparison of all elements of all datasets (both internal and external), but in practice such comparison is impossible. Link generation optimization requires additional study.

5 Conclusion

This paper proposes a concept of data integration system orientated towards Linked Open Data space. The novelty of this concept lies in its hybrid approach; the system proposed combines data mediation and data warehousing approaches by using locally stored linksets as indexes for a search engine hasn’t been implemented yet. To the author’s knowledge, such method hasn’t been implemented yet. Besides, while there are works dedicated to bringing single data sources into the LOD space or dealing with multiple already present sources in LOD space, the idea of bringing multiple data sources into LOD space via single data integration system has received very little attention.

Currently, the proof-of-concept system is being developed in CC RAS as a part of a practical project dedicated to integration of data on protected sites and animal species. While participating in a group on this project, the author is working on query answering algorithms in presence of linksets. As a result of this project, a large set of data on national parks should emerge in the LOD space, and if incoming links from external datasets appear, the project would be considered to be successful.

Future works on this project might include the study of link network generation and support algorithms. The system can also be extended with modules to access external Semantic Web resource aggregators (sig.ma) and semantic search engines (sindice.com). Also, additional studies in the management of licensing and data access in the context of the Linked Open Data are required.

References

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets. In Proceedings of the WWW2009 Workshop on Linked Data on the Web, 2009.

[2] A. A. Bezdushny. Formal Model of Ontology-Based Data Integration Systems. Novosibirsk, 2008

[3] C. Bizer, R. Cyganiak. D2RQ — Lessons Learned. Position paper for the W3C Workshop on RDF Access to Relational Databases, 2007. http://www.w3.org/2007/03/RdfRDB/papers/d2rq-positionpaper/

[4] D. Calvanese, G. De Giacomo, D. Lembo et al. The MASTRO system for ontology-based data access. Semantic Web Journal, volume 2, number 1, pages 43-53, 2011

[5] J. Euzenat, A. Ferrara, et al. First results of the ontology alignment evaluation initiative 2011. In Proc. of 6th Ontology Matching Workshop (OM‘11), at International Semantic Web Conference (ISWC‘11), Bonn, Germany, 2011.

[6] O. Gorlitz, S. Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany, 2011.

[7] H. Halpin, P. Hayes, J. McCusker, D. Mcguinness, and H. Thompson. When owl:sameas isn't the same: An analysis of identity in linked data. In Proceedings of the 9th International Semantic Web Conference, 2010

[8] T. Heath and C. Bizer. Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool, 2011. http://linkeddatabook.com/editions/1.0/

[9] H. Muhleisen and A. Jentzsch: Augmenting the Web of Data using Referers Linked Data on the Web (LDOW2011), Mar. 2011

[10] A. Nikolov and M. d'Aquin. Identifying Relevant Sources for Data Linking using a Semantic Web Index, Workshop: 4th Workshop on Linked Data on the Web (LDOW 2011) at 20th International World Wide Web Conference (WWW 2011), Hyderabad, India, 2011.

[11] Virtuoso Universal Server, 2011. http://virtuoso.openlinksw.com/

[12] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In Proceedings of the International Semantic Web Conference, pages 650–665, 2009