Available Online at http://www.journalajst.com

ISSN: 0976-3376

Asian Journal of Science and Technology
Vol. 6, Issue 05, pp. 1401-1405, May, 2015

ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

REVIEW ARTICLE

ARCHITECTURE OF DATA PUBLISHING TOOL FOR THE LINKED DATA

*Shilpi Saxena, Vaishali Tyagi and Mrityunjay singh

Department of Computer Science and Engineering, SRM University, India

ARTICLE INFO

Article History:
Received 24th February, 2015
Received in revised form
29th March, 2015
Accepted 05th April, 2015
Published online 31st May, 2015

Key words:
Linked Data,
Semantic Web,
RDF,
SPARQL,
Query Formulation

ABSTRACT

In the recent past, the attention has been paid for publishing the structured data on web, know as linked data. Linked data is simply about using web to create typed links between data from different sources. These may be as diverse as databases maintained by different organizations in distributed over different geographical locations, or heterogeneous within the same organization, which are not easily interoperated at the data level. Technically, linked data refers to data published on the web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets. A variety of linked data publishing tools has been developed. The tools shield publishers from dealing with technical detail such as content negotiation and ensure that data is published according to the linked data community best practices. In literature, we have found that the existing data publishing tools are either based on query reformulation or data translation. These two approaches suffer from the various drawbacks. In this work, we have combined these two approaches, and proposed a new architecture for data publishing tool for linked data. These tools are very helpful because data can published easily with the help of tools on semantic web.

INTRODUCTION

The Semantic Web is a Web of Data. The collection of Semantic Web technologies (RDF, OWL, SKOS, SPARQL, etc.) provides an environment where application can query that data, draw inferences using vocabularies. Web search is a process of querying, learning and reformulating queries to satisfy certain information needs. When a user submits a search query, the search engine attempts to return the best results to that query. The goal of the Linked Data is to enable people to share structured data on the Web as easily as they can share documents today. Linked Data is the term used to describe a method of exposing and connecting data on the Web from different sources. Currently, the Web uses hypertext links that allow people to move from one document to another. In Semantic Web terminology, Linked Data is the term used to describe a method of exposing and connecting data on the Web from different sources. Currently, the Web uses hypertext links that allow people to move from one document to another. The idea behind Linked Data is that hyper data links will let people or machines find related data on the Web that was not previously linked. Linked data is published by different tools and with the help of publishing tool linked data can easily published on the semantic web.

These publication tools (Bizer, Christian *et al.*, 2009) are D2R server, virtuoso universal server, triplify, paget and pubby server. These tools are based on either data transformation or query reformulation. These two approaches used in publishing tools. Some publishing tools follows concept of data transformation and some publishing tools follows concept of query reformulation but data transformation has some drawbacks like inconsistency, uncertainty during transformation (at data/schema), storage cost increased, integrity constraint and consistency overhead and drawbacks of query reformulation are hard to use, adding removing or modifying source description, access time increased, query reformulation uncertain at query and reformulation methods are different. In order to overcome the disadvantages of tools we have proposed a Hybrid scheme for publishing data, which inherits the merits of the data transformation and query reformulation techniques, and overcome the demerits of these techniques. Sub query algorithm used for query reformulation. Section 2 describes related work for publishing linked data with the help of the publishing tools. Section 3 provides basic concepts of the linked data. In Section 4 discussed about proposed architecture. Conclusion is discussed in section 5.

RELATED WORKS

Basically, our research about different publishing tools for linked data. These tools are D2R server, Pubby, Paget, Triplify and virtuoso universal server.

***Corresponding author:** **Shilpi Saxena,**
Department of Computer Science and Engineering, SRM University, India.

They are either based on query reformulation or data transformation. D2R server and pub by based on the data transformation and Triplify, virtuoso universal server and paget based on the query reformulation. These tools are either open source tool or resource centric tool. Query reformulation used for the distributed system. In distributed system we have different data sources so with the help of tools we can convert into the RDF form and this RDF data stored in the database with the help of the data transformation.D2R server (Bizer *et al*., 2006) is a tool for publishing a content of relational database on the semantic web. It supports RDF and HTML browsers to navigate the content of the database, and allows the querying the database using the SPARQL query language. It is based on the data transformation approach.

Virtuoso, known as virtuoso universal server, is a multi-purpose protocol RDBM. Includes an object-relational database engine (for SQL, XML, RDF and free text) includes java and .net run hosting web application server, web services, web content management, Data Portability (controlling, sharing, and moving data freely from system to system). Virtuoso is based on query reformulation. Pubby (Erling *et al*., 2009) can be used to add Linked data interfaces to SPARQL endpoints. It is used for extension of RDF. It is hard to connect information these stores with other external data sources. It allows a wide variety of existing RDF browser, RDF crawlers and query agents to access the data. Pubby make it easy to turn a SPARQL endpoint into a linked data server. Pubby based on the concept of data transformation. Paget (Bizer, Christian *et al*., 2009) is a framework for building linked data applications and it is based on data transformation. It is focused on publishing data. Triplify (Auer *et al*., 2009), (Auer *et al*., 2009) is a simple approach to publish RDF and Linked Data from relational databases. Triplify is based on mapping HTTP□URI requests onto relational database queries expressed in SQL with some additions. Triplify follows the concept of the query reformulation.

Linked Data Concepts

Linked Data

Linked data (Bizer, Christian *et al*., 2009) used for share structured data on the web as easily as they can share documents today. Linked data is a set of best practices for publishing and deploying instance and class data using the RDF data model, and uses (URIs) uniform resource identifiers to name the data objects (HTTP), but rather than using them to serve web pages for humans readers, it extends then to share information in a way that can be read automatically by computers.

Use of Linked Data

Linked Data lies at the heart of what Semantic Web is all about: large scale integration of, and reasoning on, data on the Web. Almost all applications listed in, say collection of Semantic Web Case Studies and Use Cases are essentially based on the accessibility of, and integration of Linked Data at various level of complexities.

Publishing Rules for Linked Data

There are four Publishing rules (curry *et al*., 2013).

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful information using the standards
- Including links to other relevant URIs so that people can discover more things.

RDF (Resource Description Framework)

RDF (Curry *et al*., 2013) standards provide a common interoperable format and model for data linking and sharing on the web. RDF provides an appropriate technology platform to enable the sharing of cross domain information relevant to the operation of a building. In RDF, a description of a resource is represented as a number of triples. The three parts of each triple are called its subject, predicate and object.

RDF Link

Represent typed links between two resources. RDF links (Curry *et al*., 2013) consist of three URI references. The URIs in the subject and the object position of the link identify the interlinked resources. The URI in the predicate position defines the type of the link. RDF links are the foundation for the web of data.

RDF Schema

RDF Schema provides a data modeling vocabulary for RDF data. It is complemented by several companion documents which describe the basic concepts and abstract syntax of RDF. RDF schema is a semantic extension of RDF. It provides a mechanism for describing groups of related resources and the relationships between these resources. RDF schema is written in RDF using the terms described in this document.

Data Transformation

Data transfer transform can be used to move data from a source or from another transform into the data store target. This can be used after query transformation with group by, distinct or order by functions which do not allow push. In data transformation accessing time is less and fast retrieval.

Query Reformulation

When we have multiple data sources it is necessary to decide which data source can contribute to an answer query? The process of finding relevant sources and feasible sub-queries is referred to as query planning. Query reformulation used for heterogeneous distributed data sources. After query planning we choose source selection process. In source selection we can use SPARQL.

Sparql

SPARQL (Prud'hommeaux and Seaborne, 2008) (pronounced "sparkle", a recursive acronym for SPARQL Protocol and DF Query Language) is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format.

Proposed Architecture

Architecture of publishing data on web is divided into three layers.
A query processed in 3 layers.

Application Layer

In first step user interface developed with the help of user interface any user can send a query to the query processor. At a time many users can send a query.

Query Processing Layer

- Query processor processed the query and search in the database where data populated in the RDF form. This process is known as data transformation. If query is present in the database then result of query is evaluated in the form of the RDF form if query is not present in the database then used further step.
- If result of query is not present in the database then again send to the query processor and query processed by the query processor. Query reformulation used with the help of the sub query reformulation algorithm and query split into the sub Queries and sent to the suitable appropriate data sources for integrated result.

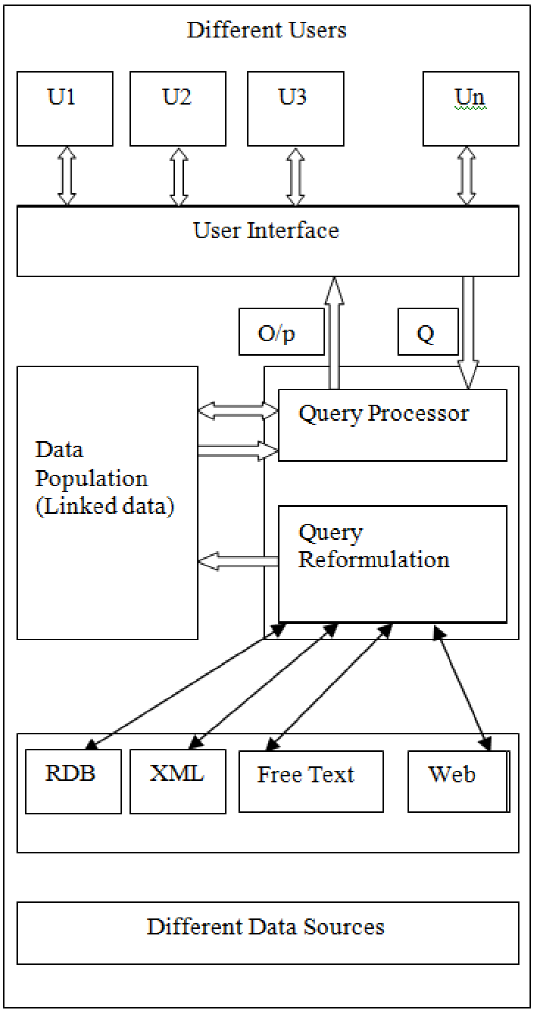

Fig1. Query processing procedure

Data Source Selection Layer

In this layer find appropriate and suitable data source related with query and choose correct data source for given query. After it query is send to the database and convert into the triple form (RDF) and execute the result to the user with the help of the query processor

Sub Query Algorithm

Sub query algorithm (Bastian Quilitz and Ulf Leser, ?) used for query reformulation for heterogeneous data sources. In Sub-queries algorithm query can be answered by the data sources. Sub-queries consist of one filtered basic graph pattern per data source. Sub query represent as triple (T; C; d), where T is a set of triple patterns, C is a set of value constraints and d is the data source that can answer the sub-query. Algorithm shows how the sub-queries are generated. If a triple pattern matches exactly one data source (D_i= {d}) the triple will be added to the set of a sub-query for this data source. All triples in this set can later be sent to the data source in one sub-query. If a triple matches multiple and different data sources the triple must be sent individually to all matching data sources in separate sub-queries.

Example

Let data source A and B be two data sources. One data source contains current and historic Oscar winning films; the other data source contains biographies of Hollywood actors and actresses. Data source A and B has capabilities (Oscar winning films, true) and (biographies, true). A stores the triple (a, Oscar winning film,"transformer"), B stores the Triple (a ,biographies,"Robert petrson"). if sent to A and B with both triple patterns or the correct result if triple patterns are sent in separate sub-queries and the results are joined afterwards.

In our example, two data sources are started independently from each other. One data source hosts information on current and historic Oscar winning films; the other a large database of biographies of Hollywood actors and actresses. Both contain complementary information in their website databases. They will cover firstly how information sharing between these sites could happen without the use of semantics. Then, they will describe how the same information can be shared between the two data sources - and potentially beyond - with the use of semantics.

Our two data sources, one fronting an XML database of all Oscar winning films, and another one fronting a My SQL database of Hollywood actors. The two sites were started independently, and do not collaborate. The Oscar Winners site lists, as its name suggests, the entire Oscar winning films ever produced and also a list of actors and actresses who starred in them. However, it doesn't hold any other actor information other than their name and date of birth. The Actor Biographies site contains a complete listing of many current and former Hollywood actors, including a complete biography, plus a list of movies that they starred in. But, it does not contain any film plots, or screenshots of the films. Let's look at how these two sites might collaborate under their current, more traditional data model.

Obviously, the users of data source B would benefit from being able to click on the name of a starring actor and find out more about them - this information is stored in the MY SQL database. Likewise, the users of data source A would benefit from being able to click on the names of films that the actors starred in and find more information. This is stored in the XML database. Any sharing of data between the two data sources cannot be done by joining tables in their databases. Firstly, they have been independently designed in the first place and so their primary keys referring to individual actors or films in both databases will not be synchronized. They would have to be mapped. But secondly, they are using different database server systems which are not cross-compatible.

To collaborate using their current databases, the owners of either site would have to decide on a common data format by which to share information that they could both understand by using a common film and actor unique ID scheme of their own invention. They could do this, for example, by creating a secure XML endpoint on each of their websites from which they can request information from each other on demand. This way, their shared information is always up to date.

With the introduction of RDF and semantics, it is far easier. Let's investigate how this could be achieved using RDF and the semantic web - it all happens automatically, not manually.

Sharing With The Semantic Web Model

Vocabulary - A collection of terms given a well -defined meaning that is consistent across contexts.

RDF – Information are stored in triple form using the RDF schema.

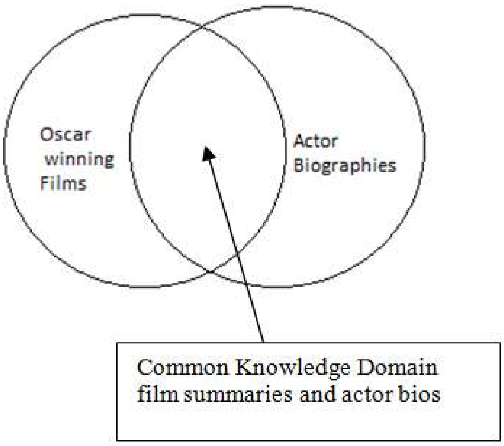

Fig. 2. Model of the two site scenario using semantic modeling

Firstly, the two sites need to apply a common, standard vocabulary to describe their data that is contextually consistent. For example, the term 'film title' should mean the same thing for both sites, as should the term 'actor name' and 'actor birth date'. This may be done by the two data sources adopting the same base RDF Schema, or a common vocabulary, for expressing the meaning behind the data they expose, and publishing that data on a query able endpoint so that the two data sources can communicate with each other across the web.

With this standard vocabulary in place

The two data sources can now query each other using the same terms. The Oscar Winning Movies site can now query the actor names on the Actor Biographies data source on-demand and gain more detail about a specific actor or actress that has starred in a movie. The Actor Biographies site can now query the film plots on the Oscar Winning Movies data source on - demand and gain more detail about films an actor has starred in. With the contextual relationships defined in formal web ontology, further related information about the actors or films, e.g. film locations, other news events happening on the same day of filming or birth date or the actor, or films made by the same director, may be found via the linked standard terminology without the user even imagining that information initially existed. This happens without the need for transformation, mapping, or contracts being set up between the two sites. It all happens through semantics.

Conclusion

The publication of variety of data on web is a crucial problem. For publishing such data on web requires the data extraction from the source, and perform the require transformation task. In this paper, we proposed architecture for the data publication tool. We have combined the query reformulation and data translation approach into one, and found our approach is more suitable for the data publication tool. This architecture has provided one of the possibilities for developing the data publishing tool for linked data, which is not be taken as whole. Our architecture is useful for didactic purpose to the researchers of this area. However, the proposed architecture will helpful as a guide for properly implementing the publishing tool which will be as automatic as possible, and work on the principle of linked data.

REFERENCES

Auer, Sören, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumueller, 2009. "Triplify: light-weight linked data publication from relational databases." In Proceedings of the 18th *international conference on World Wide Web*, pp. 621-630. ACM.

Auer, Sören, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumueller, 2009. "Triplify: light-weight linked data publication from relational databases." In Proceedings of the 18th *international conference on World wide web*, pp. 621-630. ACM.

Bastian Quilitz and Ulf Leser,” Querying Distributed RDF Data Sources with SPARQL”

Bizer, Christian, and Richard Cyganiak. 2006. "D2r server-publishing relational databases on the semantic web." In Poster at the 5th *International Semantic Web Conference*.

Bizer, Christian, Tom Heath, and Tim Berners-Lee, 2009. "Linked data-the story so far." International journal on semantic web and information systems5, no.3:1-22

Curry, E., James O’Donnell, Corry, E. Souleiman Hasan, Marcus Keane, and Sean O’Riain, 2013. “Linking building data in tha cloud: Integrating cross-domain building data using linked data”, Advanced Engineering informatics 27 206-219.

Cyganiak, Richard, and Chris Bizer, 2008. "Pubby-a linked data frontend for sparql endpoints. Retrieved from http://www4. wiwiss. fu-berlin. de/pubby/at May 28: 2011.

Erling, Orri, and Ivan Mikhailov, 2009."RDF Support in the Virtuoso DBMS." In Networked Knowledge-Networked Media, pp. 7-24. Springer Berlin Heidelberg.

Prud'hommeaux, E. and Seaborne, January 2008. “A. SPARQL Query Language for RDF”. W3C http://www. w3.org/TR/rdf-sparql-query/

****