International Journal of Engineering Research & Management Technology ISSN: 2348-4039

March- 2015 Volume 2, Issue-2

Email: editor@ijermt.org www.ijermt.org

A Survey on the Publishing Tools for Linked Data

Shilpi Saxena, **Vaishali Tyagi**
M. Tech Student
Department of Computer Science & Engineering
SRM University, NCR campus
Modinagar

Mrityunjay Singh
Assistant Professor
Department of Computer Science & Engineering
SRM University, NCR campus
Modinagar

ABSTRACT:
In this paper, we survey different publication tools for publishing a linked data on the semantic web. Working and architecture of different publishing tools are examined in this paper. We recommend hybrid architecture which contains all advantages of every publishing tool. These may be as diverse as databases maintained by two organizations in different geographical locations, or simply heterogeneous system within one organization that, historically, have 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 this work, we will survey on the different publishing tools for linked data, and try to make enhancement over them.

Keywords: Linked data, Semantic web, RDF, Query formulation, Data transformation

1. INTRODUCTION

The Semantic Web proposes to help computers "read" and use the Web. The big idea is pretty simple metadata added to Web pages can make the existing World Wide Web machine readable. This won't bestow artificial intelligence or make computers self-aware, but it will give machines tools to find, exchange and, to a limited extent, interpret information. It's an extension of, not a replacement
for, the World Wide Web. The goal of [1] Linked Data is to enable people to 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. This enables data from different sources to be connected and queried. [2]Linked data is simply about using the web to create typed links between data from different sources.

1. RDF data model to publish structured on the web.
2. Use RDF links [5] to interlink data from different data sources.

Applying both principles [7] leads to the creation of a data commons on the web, a space where people and organizations can post and consume data about anything. This data commons is often called the web of data or semantic web.

Linked data has the following characteristic:

1. Open: Linked data is accessible through an unlimited variety of applications because it is expressed in open, non proprietary formats.
2. Modular: Linked data can be combined (mash-up) with any other pieces of linked data. No advance planning is required to integrate these data. No advance planning is required to integrate these data sources as long as they both used linked data standards.
3. Scalable: It is easy to add and connect more linked data to existing linked data. Even when the terms and condition definitions that are used change over time.
Advantages of linked data are sharable, extensible and easily re-usable.

2. PUBLISHING TOOLS

2.1 D2R SERVER

D2R server [4] 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.D2R server follows the concept of data transformation. It able to represent heterogeneous data into a single form, enable uniform accessing over them. Data transfer transform will push down the operations to database server when the transfer type is database table.D2R server is open source tool for publishing a data. This server works only for structured data. Vocabulary for D2R server is RDF. In D2R RDF represents in form of triple (Subject, Predicate and Object).

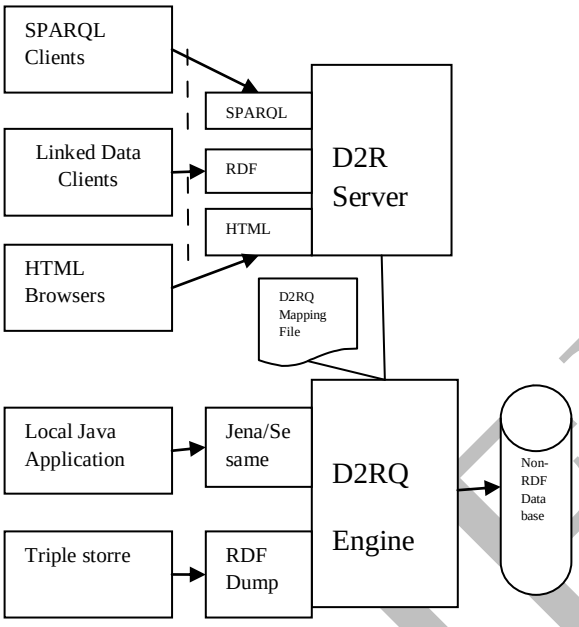

Fig 1 D2R Architecture

D2R Server: D2R server uses a customizable D2RQ mapping to map database content into this format, and allows the RDF data to be browsed and searched-the two main access paradigms to the semantic web. D2R server provides remote access to a D2RQ-mapped database via the SPARQL protocol.

SPARQL Clients: The SPARQL interface enables applications to search and query the database using the SPARQL query language over the SPARQL protocol.

HTML Browser and HTTP: A traditional HTML interface access to the familiar web browser and HTTP provides a linked data view, a HTML view for debugging and a SPARQL protocol endpoint over the database.

D2RQ Engine: D2RQ engine, a plug-in for the jena semantic web toolkit, which uses the mapping to rewrite jena API calls to SQL queries against the database and passes query results up to higher layers of the frameworks.

RDF Dumps: RDF dumps in RDF /XML or N-Triples.

JENA: It is open source semantic web framework for java.

2.2 VIRTUOSO UNIVERSAL SERVER

Virtuoso, known as virtuoso universal server [2], 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). Instead of separate servers for RDBMS, ORDBMS, RDF, XML, Web Application Server, and File Server functionality, Virtuoso combines the aforementioned into a single "universal server".

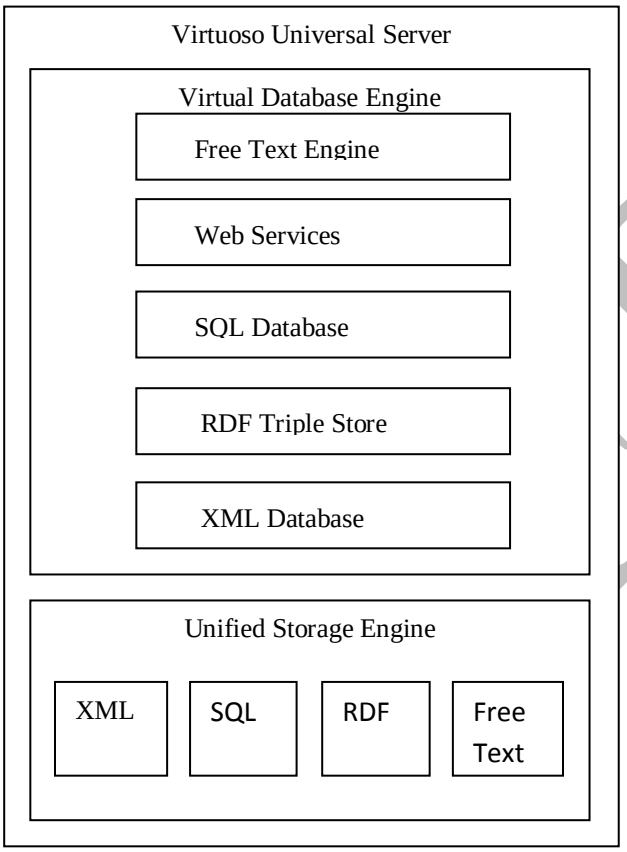

Fig 2 Virtuoso Universal Server

Virtuoso is designed with multi-threading and multi-CPU support. Data perform in quad format (Graph, Subject, Predicate and Object).In Virtuoso, an RDF mapping schema consists of declarations of one or more quad storages. The default quad storage declares that the system table RDF QUAD consists of four columns (G, S, P and O) that contain fields of stored triples, using special formats that are suitable for arbitrary RDF nodes and literals.

Virtuoso includes support for SPARUL, SPARQL and is compatibility with Jena Updates can be run transactional or with an automatic commit after each modified triple. Virtuoso can store over 1 billion triples Loads 1 billion triples LUBM benchmark at a sustained rate of 12692 triples/s and 47 million triples Wikipedia data set at a sustained rate of 20800 triples/s (Orri Erling, OpenLink).

Applications of Virtuoso are Dbpedia, Musicbrainz , Geonames and Ping the Semantic Web. In 2006 Virtuoso was available as open source. Today, Virtuoso is available in both open source and commercial licenses. The open-source version of Virtuoso is known as OpenLink Virtuoso.

Data sources of virtuoso universal server are SQL, SPARQL, XQuery, XPath, XSLT, TURTLE and JSON and schema definition languages are SQL's DDL and XML schema. It depends upon the concept of query reformulation.

2.3 PUBBY

Pubby [6] 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. Linked data is a style of publishing data on the semantic web that makes it easy to interlink, discover and consume data on the semantic web. 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. It is a implemented as a java web application. It is a open source tool for publishing. It provides a linked data interface to local or remote SPARQL protocol servers. It allows dereferenceable URIs by rewriting URIs found in the SPARQL- exposed a dataset into the pubby server's namespace and provides a simple HTML interface showing the data available about each resource. It takes care of handling of 303 redirects and content negotiation and compatible with Tomcat and jetty Servlet containers. It includes a metadata extension to add a metadata to provide data.

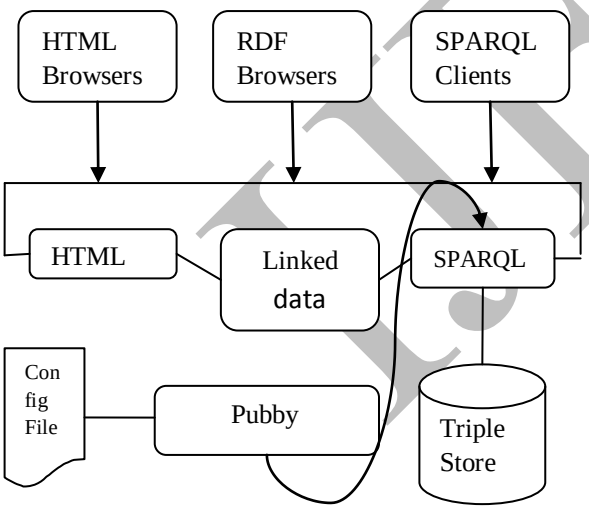

Fig 3 Pubby Architecture

In RDF, resources are identified by URIs. The URIs used in most SPARQL dataset are not dereference able, meaning they cannot be accessed in a Semantic Web browser, but return 404 Not Found errors instead, or use non-dereference able URI schemes, as in the fictional URI.
When setting up a Pub by server for a SPARQL endpoint, you will configure a mapping that translates those URIs to dereference able URIs handled by Pub by.
Pub by will handle requests to the mapped URIs by connecting to the SPARQL endpoint, asking it for information about the original URI, and passing back the results to the client. It also handles various details of the HTTP interaction, such as the 303 redirect required by Web Architecture, and content negotiation between HTML, RDF/XML and Turtle descriptions of the same resource. In Pub by URI requests rewrites into SPARQL DESCRIBE queries against the underlying RDF store.

2.4 PAGET

Paget [2] is a framework for building linked data applications. At the moment it is focused on publishing data but I intend for it to be capable of managing updates too. It requires the moriarty library and supports but does not require the use of the Tails Platform.

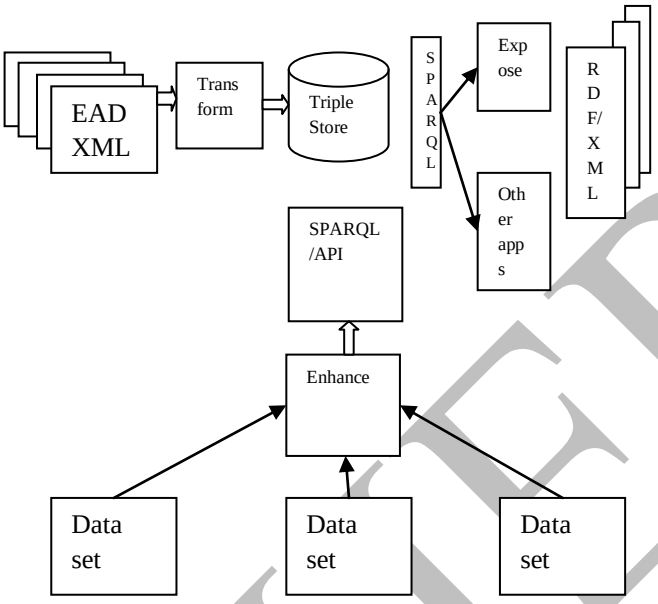

Fig 4 Paget Architecture

Paget is Opinionated Software. It has its own view of how resource-centric, data-driven web applications should be developed. If you don't share that opinion then you're going to find using Paget tough going. If you happen to agree then Paget should smooth your way to building great applications.

Paget is resource-centric. You define your application in terms of the resources it exposes to the Web. Resources are identified solely by their URI. This takes some getting used to for many web developers.
Paget is data-driven, specifically RDF data. Paget is designed around RDF data and relies on it to represent the state of the application's resources.
Because all data is represented as RDF internally, Paget can automatically represent that data in various formats such as XML, JSON, Turtle and HTML. Content negotiation for those data types is built in and Paget provides hooks for you to customize the HTML output which is where you ought to be applying your creative energy.

Paget also has opinions about how your URIs should be constructed in your application. URIs without a file extension is assumed to be "abstract resources". Those with a file extension are assumed to be descriptions of abstract resources. Paget will issue a "303 see also" redirect from the abstract resource to its description, performing content negotiation to send the client to the most appropriate format.

2.5 TRIPLIFY

Triplify [3] 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 transforms the resulting relations into RDF statements and publishes the data on the Web in various RDF serializations, in particular as Linked Data. Triplify as a light-weight software component, which can be easily integrated and deployed with the numerous widely installed Web applications. The approach does not support SPARQL, includes a method for publishing update logs to enable incremental crawling of linked data sources. Triplify is complemented by a library of configurations for common relational schemata and a REST-enabled data source registry. Despite its lightweight architecture Triplify is usable to publish very large datasets, such as 160 GB of geo data from the Open Street Map project.

Triplify is a small plugin for Web applications which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data. It is coded in PHP.

Triplify is very lightweight**.** It consists of only a few files with less than 500 lines of code. For a typical Web application configuration can be created in less than one hour and if this Web application is deployed multiple times (as most open-source Web applications are), the configuration can be reused without modifications. Triplify makes Web applications easier mash able and lays the foundation for next-generation, semantics-based Web searches**.**

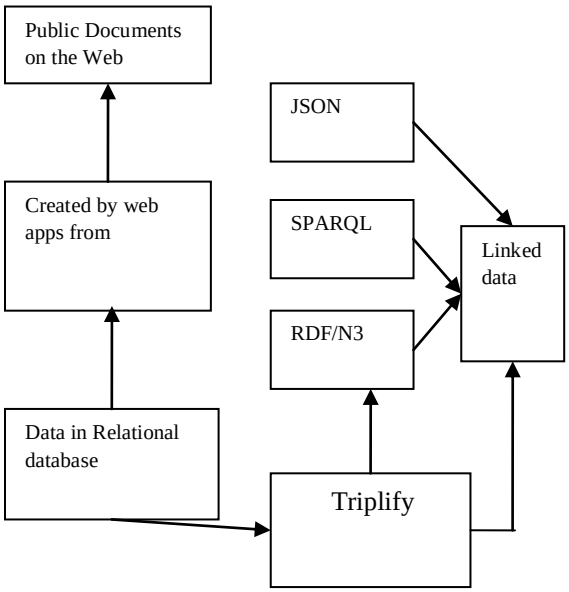

Fig 5 Triplify Architecture

3. CONCLUSION

In this paper, we survey architecture of publishing tools. Some tools are either based on query reformulation or data transformation. 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. 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.

4. REFERENCES

1. E.curry, James O'Donnell, E.Corry, Souleiman Hasan, Marcus Keane, and Sean O'Riain: "Linking building data in tha cloud: Integrating cross-domain building data using linked data", Advanced Engineering informatics 27(2013) 206-219.
2. Bizer, Christian, Tom Heath, and Tim Berners-Lee. "Linked data-the story so far." International journal on semantic web and information systems5,no.3(2009):1-22
3. Auer, Sören, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumueller. "Triplify: light-weight linked data publication from relational databases." In Proceedings of the 18th international conference on World Wide Web, pp. 621-630. ACM, 2009.
4. Bizer, Christian, and Richard Cyganiak. "D2r server-publishing relational databases on the semantic web." In Poster at the 5th International Semantic Web Conference. 2006.
5. Erling, Orri, and Ivan Mikhailov. "RDF Support in the Virtuoso DBMS." In Networked Knowledge-Networked Media, pp. 7-24. Springer Berlin Heidelberg,2009.
6. Cyganiak, Richard, and Chris Bizer. "Pubby-a linked data frontend for sparql endpoints. Retrieved from http://www4. wiwiss. fu-berlin. de/pubby/at May 28 (2008): 2011.
7. Prud'hommeaux, E., Seaborne, "A.: SPARQL Query Language for RDF". W3C (January 2008) http://www.w3.org/TR/rdf-sparql-query/