Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 1, January 2015, pg.134 – 139

RESEARCH ARTICLE

The Study of Basics for a Query Formulation Language – MashQL

Roshani Sudhirpant Khule
Student of Master of Engineering in (CS & IT)
HVPM's college of Engineering and Technology, Amravati, India
Khule.roshani@gmail.com

Prof. Ranjit R. Keole
Lecturer in the Department of Information Technology
HVPM's College of Engineering and Technology, Amravati, India
ranjitkeole@gmail.com

Abstract -

In this paper, we present MashQL, a novel query formulation language also called as Query-by-diagram language for querying and mashing up structured data on the Web. It doesn't require users to know the queried data's structure. Being a language not merely an interface and assuming data to be schema-free is one of the key challenges addressed in this paper. Assuming web data as input in RDF format used to represent the metadata of the Web applications in structured manner. We can query by using SPARQL language which is the recent recommendation of W3C. By using this approach we mentioned how user having less technical knowledge is able to retrieve the data in structured format. We mentioned that how retrieval results will be faster by providing keyword search and then use MashQL approach. We mentioned the basic concepts for study of MashQL that is mashup, SPARQL, RDF in MashQL by which user can perform a calculation on a set of values and return a single value. A novel technique for optimizing queries over large data sets to allow instant user interaction is proposed and evaluated.

Keywords: Query-by-diagram, Query Formulation, Web 2.0, Semantic Web, Data Web, Mashup, SPARQL, RDF

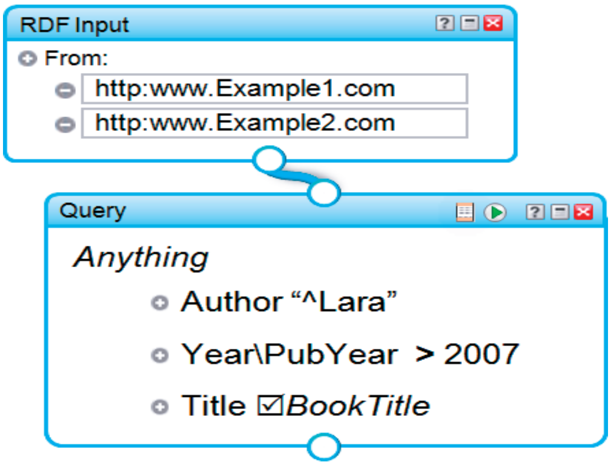
I. Introduction

The rapid growth of Web 2.0 content has created a high demand for making this content more reusable. For populating the semantic web many companies such as Facebook, Yahoo! and Google, Amazon, eBay, LinkedIn, another emerging has started to appear [1]. The vision of semantic web as proposed by the World Wide Consortium (W3C) is to "create a universal medium for the exchange of data". For this vision to realize, large amounts of structured datasets are being published forming a web of interlinked structured data. Examples of structured datasets being published and interlinked by the project include Wikipedia, Wikibooks, Yago, DBLP bibliography, Wordnet, Geonames, MusicBainz, Freebase and many more. Governments are also following the trend of publishing structured data in the web and also encouraging people to reuse and benefit from it. Most of these datasets are published in Resource Description Framework (RDF) which is a W3C recommendation. This emerging direction is the use of the Resource Description Framework in attributes, which adds a set of attributes to XHTML for embedding RDF triples in web pages. RDF plays an important role in bridging the gap between the web of files (web 2.0) and the web of data (web 3.0) [3] in that, with the support of the largest companies, HTML authors are starting to embed RDF triples in their XHTML pages and thus contributing to the growth of the semantic web.

In open worlds such as the Web, structured data is being created and consumed by different users, and the need for a mechanism to mash up and consume this distributed and heterogeneous data easily is a real demand. As the trends of publishing structured data in the Web are increasing rapidly, It of Web technologies towards new paradigms of structured-data retrieval [2]. Traditional search engines cannot serve such data as the results of a keyword-based query will not be precise or clean, because the query itself is still ambiguous although the underlying data is structured. So, SPARQL was proposed as a standardized and RDF Query Language that enables querying collections of RDF data which is analogous to SQL for querying databases. However, SPARQL is oriented for technical people and is of no use to people with limited IT skills. In fact, to exploit the massive amount of structured data in the Web to its full potential, people should be able to query this data easily and effectively. Formulating queries should be fast and should not require programming skills. Thus, to allow non technical people to query RDF data, MashQL was introduced as an intuitive language for querying the Data Web [4].

MashQL project was started at the University of Cyprus and is continued at Birzeit University. Our work and contribution to the MashQL project is twofold (i) bringing RDFa support to MashQL and (ii) developing the Graph-Signature Indexing query optimization solution and using it in MashQL to extend it to support queries over large RDF datasets. Graph-Signature indexing can also be viewed as a separate contribution from MashQL in that the indexing solution provides a significant enhancement to Oracle's solution for querying large RDF datasets which is known as Oracle's Semantic Technology.

This article proposes a mashup language called MashQL, which uses SPARQL as a backend query language. It encapsulates the complexity of SPARQL and allows people to query RDF sources intuitively see Figure below [5]. In the background, MashQL queries are translated into and executed as SPARQL queries. The novelty of MashQL is that it allows one to formulate a query over a data sources without any prior knowledge about its schema. MashQL does not also assume its users to have any general knowledge about RDF or SPARQL to get started. Hence, the average internet user can use MashQL to develop data mashups easily. Furthermore, MashQL supports pipelines and materialized queries as built-in concepts.



From:
http:www.Example1.com
http:www.Example2.com

Anything
Author "^Lara"
Year\PubYear > 2007
Title ☑*BookTitle*

Figure. An example of MashQL query.

MashQL is a query-by-diagram language. The idea of MashQL is to allow people to query, mash up, and pipeline structured data intuitively. In the background MashQL queries are automatically translated into and executed as SPARQL queries. In this paper we are further elaborating the concept of main basic techniques used in MashQL i.e. mashup, SPARQL, RDF(Resource Description Framework) and at last MashQL itself.

II. Main techniques present in MashQL

A. mashup

A mashup, in web development, is a web page, or web application, that uses content from more than one source to create a single new service displayed in a single graphical interface. For example, any organization can combine the addresses and photographs of their different offices branches with a Google map to create a map mashup [6]. The term implies easy, fast integration, frequently using open application programming interfaces and data sources to produce enriched results. To expose the massive amount of public content and to allow people to build mashups easily, several mashup editors have been launched, including Google Mashup, Microsoft's Popfly, IBM's Smash, Yahoo Pipes, and few others. Yahoo Pipes are simpler so it have received most attention by users. To build on the remarkable success of Web 2.0 mashups and expose the data web to its full potential, we propose to regard the Internet as a database. Each data source is seen as a table, and each mashup as a query. The average user could build mashups intuitively, and mashups should be reusable. Hence, people can reuse and build on each other's results. Mashups should not be limited to sophisticated queries, but can also be simple inquiries.

The main characteristics of a mashup are combination, visualization, and aggregation. It is important to make existing data more useful, for personal and professional use. To be able to permanently access the data of other services, mashups are generally client applications or hosted online. Mashups can be considered to have an active role in the evolution of social software and Web 2.0. Mashup composition tools are usually simple enough to be used by end-users. They generally do not require programming skills and rather support visual wiring of GUI widgets, services and components together. Therefore, these tools contribute to a new vision of the Web, where users are able to contribute. Mashups can also regard as a mechanism for general-purpose data retrieval.

In what follows we present our motivation of using RDF and SPARQL in data mashups.

Types of mashup

There are many types of mashup, like business mashups, consumer mashups, and data mashups [8]. Out of which, the most common type of mashup is the consumer mashup, aimed at the general public.

- Business (or enterprise) mashups:

  This mashup contains applications that combine their own resources, application and data, with other external Web services [7]. They focus data into a single presentation and allow for collaborative action among businesses and developers. This works well for an agile development project, which requires collaboration between the developers and customer for defining and implementing the business requirements. Enterprise mashups are secure, visually rich towards Web applications that expose actionable information from diverse internal and external information sources.

- Consumer mashups

  This mashup combines data from multiple public sources in the browser and organize it through a simple browser user interface [9]. For example, Wikipediavision combines Google Map and a Wikipedia API.

- Data mashups

  It is opposite to the consumer mashups, combine similar types of media and information from multiple sources into a single representation. The combination of all these resources create a new and distinct Web service that was not originally provided by either source

B. SPARQL

SPARQL is pronounced as "sparkle". It is a recursive acronym for SPARQL Protocol and RDF Query Language. SPARQL is a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format [10] [11]. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium, and is recognized as one of the key technologies of the semantic web.

SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns [12]. Implementations for multiple programming languages exist [13]. There exist tools that allow one to connect and semi-automatically construct a SPARQL query for a SPARQL endpoint [14]. In addition, there exist tools that translate SPARQL queries to other query languages, for example to SQL [15] and to XQuery [16]. SPARQL allows users to write queries against data that can loosely be called "key-value" data or, more specifically. The entire database is thus a set of "subject-predicate-object" triples. This is analogous to some NoSQL databases' using the term "document-key-value", such as MongoDB.

SPARQL provides a full set of analytic query operations such as JOIN, SORT, AGGREGATE for data whose schema is intrinsically part of the data rather than requiring a separate schema definition. Schema information is often provided externally, it allow different datasets to be joined in an unambiguous manner. In addition, SPARQL provides specific graph traversal syntax for data that can be thought of as a graph. Some implementations, such as SPARQLverse also allow additional triple attributes such as timestamp and allow additional analytic functionality like windowed aggregates.

For example, the following query returns names and emails of every person in the dataset:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}
```

Where foaf means friends-of-a-friend. This query joins together all of the triples with a matching subject, where the type predicate, "a", is a person (foaf:Person) and the person has one or more names (foaf:name) and mailboxes (foaf:mbox).

This query can be distributed to multiple SPARQL endpoints (services that accept SPARQL queries and return results), computed, and results gathered, a procedure known as federated query. Whether in a federated manner or locally, additional triple definitions in the query could allow joins to different subject types, to allow simple queries, for example, to return a list of names and emails for people who drive automobiles with a high MPG rating.

C. RDF (Resource Description Framework)

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications [17] originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications.

RDF data can also be considered in SQL relational database terms as a table with three columns - the subject column, the predicate column and the object column. Like relational databases, the object column is heterogeneous, the per-cell data type is usually implied by predicate value. Alternately, again comparing to SQL relational, all of the triples for a given subject could be represented as a row, with the subject being the primary key and each possible predicate being a column and the object is the value in the cell. However, SPARQL/RDF becomes easier and more powerful for columns that could contain multiple values, and where the column itself could be a joinable variable in the query, rather than directly specified. A collection of RDF statements intrinsically represents a labeled, directed multi-graph. RDF data is often persisted in relational database or native representations also called Triplestores, or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple [18]. It includes the cardinality constraints from OSLC Resource Shapes and Dublin Core Description Set Profiles as well as logical connectives for disjunction and polymorphism. As RDFS and OWL demonstrate, one can build additional ontology languages upon RDF.

RDF is being used to have a better understanding of road traffic patterns. This is because the information regarding traffic patterns is present on different websites, and RDF is used to integrate information from different sources on the web. Some uses of RDF include research into social networking. It will also help people in business fields understand better their relationships with members of industries that could be of use for product placement [19]. It will also help scientists understand how people are connected to one another.

III. Extended Implementation of MashQL

As we seen, that MashQL is proposed a query-by-diagram language in order to allow building data mashups intuitively. Not only MashQL is user-friendly for non-IT people, but also it allows querying and navigating RDF data sources without having to know the schema or the technical details of these sources [20]. Because of this we can demonstrated the use of MashQL using different use cases, and one can learned the lessons regarding the elegancy, coverage, and performance of MashQL. MashQL is not merely an interface of SPARQL. Although it can be used as such, but it can be used also as a general query language by its own. In addition, MashQL can be used also for filtering metadata streams. In this article, we focus on using the W3C's SPARQL standard as the backend query language.

E-learning is an interactive learning system mainly with a computer through Internet connection [21]. The retrieval of book details by e-learning users is generally somewhat difficult, since those data are structured. The structured data face the challenging issues in retrieval process. The schema of the data is not known to the user in order to query to the end users. Here E-learning materials may be schema-free or poorly-schematized. So by using, graphical query formulation language, called MashQL, in order to easily query structured data in E-learning application. Even, when the end users have limited technical background they can query and explore multiple data sources. This is the main significance of MashQL. This work aims in introducing semantic keyword search to retrieve the structured data, and extend the implementation techniques for MashQL.

MashQL can generally be implemented by online mashup editors. It is mostly similar to or as an extension to Yahoo Pipes or as a query interface for online RDF datasets (e.g., Freebase, DBpedia, or DBLP). It can be implemented also as a query plug-in to offline RDF stores (e.g., AllegroGraph or Oracle). MashQL can also be used to filter metadata streams in, e.g., iTunes, jobs.ac.uk, eBay, or Upcoming. We are currently developing a MashQL prototype in Yahoo Pipes style. This prototype is an AJAX web-based plug-in to Oracle 11g. Choosing Oracle is not only because of its scalability, but also because Oracle's SPARQL inherits all functionalities of SQL, including aggregation and grouping functions, which are not supported in the standard SPARQL. Today's implementation prototype does not only generate the W3C's standard SPARQL, but being extended to also generate Oracle's SPARQL [CDES]. Although this article focuses on using MashQL for querying RDF data sources using SPARQL, however, MashQL can be similarly used for querying relational databases or XML documents. In this case, one needs to either develop a stylesheet that translates MashQL markups into SQL, XQuery, or any preferred backend query language; or maybe map the dataset into an RDF-like model as our system is more scalable.

Based on the formal definition of MashQL, the technical specification of MashQL queries, called the MashQL markup. The goal of this markup is to serialize graphical MashQL queries in a textual and interchangeable format. In this way, tools will be able to save, load, process, and exchange queries easily.

IV. Conclusion

We propose a mashup language, called MashQL, which allows people to mash up data intuitively. In the background, MashQL queries are translated into and executed as SPARQL queries. The novelty of MashQL is that it allows one to query an RDF graph without any prior knowledge about its structure or technical details; as well as it supports pipelines and materialized queries as built-in concepts. End-users can navigate, query, and mash up unknown graphs. Data is schema-free, and from multiple sources. Users also do not need any knowledge about RDF/SPARQL to get started. MashQL is not merely a SPARQL interface, or limited RDF. It has its own path-pattern intuition can be similarly used for XML and DB. Although we focus on RDF/SPARQL mashups, but our approach can be easily reused for other data formats and query languages. We studied the main basic terms on which MashQL is based along with the study of Query-by-diagram language.

References

[1] “A Query Formulation Language for the Data Web” Mustafa Jarrar and Marios D. Dikaiakos, Member, IEEE Computer Society, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 5, MAY 2012.
[2] “Querying the Data Web: The MashQL Approach” , Jarrar, M. ; Univ. of Birzeit, Bir Zeit, Palestinian Authority ; Dikaiakos, M.D., Internet Computing, IEEE (Volume:14 , Issue: 3).
[3] “MashQL, A novel approach for querying the Data Web (Web 3.0)”, 2010 © Copyright MashQL Project.
[4] “Jarrar M, Dikaiakos M: MashQL: A query-by-diagram”. Technical Article TAR200805. University of Cyprus, 2008. Download from: http://www.jarrar.info/mashql/TA/
[5] “Querying the Data Web The MashQL Approach”, Published by the IEEE Computer Society 1089-7801/10/$26.00 © 2010 IEEE IEEE INTERNET COMPUTING.
[6] Fichter Darlene, What Is a Mashup?http://books.infotoday.com/books/Engard/Engard-Sample-Chapter.pdf (retrieved 12 August 2013).
[7] Clarkin, Larry; Holmes, Josh. "Enterprise Mashups". MSDN Architecture Journal. MSDN Architecture Center.
[8] Sunilkumar Peenikal (2009). "Mashups and the enterprise". MphasiS - HP.
[9] "Enterprise Mashups: The New Face of Your SOA". http://soa.sys-con.com/: SOA WORLD MAGAZINE. Retrieved 2010-03-03. A consumer mashup is an application that combines data from multiple public sources in the browser and organizes it through a simple browser user interface.
[10] Jim Rapoza (2 May 2006). "SPARQL Will Make the Web Shine". eWeek. Retrieved 2007-01-17.
[11] Segaran, Toby; Evans, Colin; Taylor, Jamie (2009). Programming the Semantic Web. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. p. 84. ISBN 978-0-596-15381-6.
[12] "XML and Web Services In The News". xml.org. 6 October 2006. Retrieved 2007-01-17.
[13] "SparqlImplementations - ESW Wiki". Esw.w3.org. Retrieved 2009-10-01.
[14] "ViziQuer a tool to construct SPARQL queries automaticly". lumii.lv. Retrieved 2011-02-25.
[15] "D2R Server". Retrieved 2012-02-04.
[16] "SPARQL2XQuery Framework". Retrieved 2012-02-04.
[17] "XML and Semantic Web W3C Standards Timeline". 2012-02-04
[18] Optimized Index Structures for Querying RDF from the Web Andreas Harth, Stefan Decker, 3rd Latin American Web Congress, Buenos Aires, Argentina, October 31 to November 2, 2005, pp. 71-80.

[19] An RDF Approach for Discovering the Relevant Semantic Associations in a Social Network By Thushar A.K, and P. Santhi Thilagam.
[20] "RDF data retrieval in structured format using aggregate function and keyword search in MashQL "Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on , Medhe, S. ; Comput. Dept., D.Y. Patil Coll. of Eng., Akurdi, Pune, India ; Phalke, D.A., 18-19 Oct. 2013
[21] "Keyword Search Retrieval for Structured data using RDF for E-learning application"Sumalatha, M.R. ; Dept. of Inf. Technol., Anna Univ., Chennai, India ; Parvathy, M., Advanced Computing (ICoAC), Fifth International Conference on, 2013.