

Study of MashQL Querying Language for the Data Web

Ms .Taware Priyanka¹, Ms. M.M.Waghmare²

¹ Student of ME (Information Technology), DGOIFE, University of Pune, Pune

² Asst.Professors, Department of Computer Engineering, DGOIFE, Bhigwan University of Pune,Pune

Abstract : The paper formulate the language for the data on the web. This language formulation helps to easily query data on the web. The main use of MashQL is that it allows people with limited knowledge of IT. It helps to explore and query one or more data sources. It does not require the knowledge about the schema, structure or any technical details of these sources. Most important, it is robust to the untrained users. And not necessary that data source should have an offline or inline. To illustrate the query formulation we choose the Data web scenarios. It also use RDF for query. RDF is the most primitive data model, so the MashQL used for the querying XML and relational databases. The two implementation of MashQL can be presented here that is a firefox add on and online mashup editor. The paper illustrate how MashQL can be used to query the data on web and mash up data web simply

Keywords : Query formulation, RDF, SPARQL, indexing methods, data web

INTRODUCTION:

The main challenge is to allow users to search and consume structured data. That receives attention from the web 2.0 and the data web communities. The increasing growth of structured data on the web has generate a more demand for make this content most reusable and consumable. Many companies are competing for gathering structured data and make it publicly.Companies are yahoo Local,Freebase, Flicker, Google Base, eBay, Amazon, LinkedIn etc.Above companies made their content or data publicly accessible using APIs.And also companies started to adopt standards of web metadata. For example, Yahoo started to support web sites embedding RDF and micro formats. And for profile and data portability MySpace also started to adopting RDF.Google, Upcoming, SlideShare, Digg, the whitehouse are started to publish their content in aRDFa. Inside the webpages W3C standard are used for embedding RDF. Hence content of data can be better understand, searched and filtered. Now the trend of structured data on the web i.e. (Data Web) is shifting towards the new paradigms of structured data retrieval. The search engines which are traditional cannot gives the results of data as a keyword based query which will not be accurate or clean. Therefore, the query itself is ambiguous. To serve the structured data on the web with full of potential, it must be able to query by the people. Main need is that Formulating queries should be fast and not requires programming skills.

1 Challenges :

Initially formulating a query, with the knowledge of structured data and the attribute labels nothing but the schema is the main challenge. At the time of search or filter the information or data it is not expected by the end users to investigate the schema or structure of the data. A data schema were dynamic in many cases because the many items of data with different schema are being changed that is it is added or dropped. Especially when query have multiple sources, allow end users to query structured data on the web flexibly and effectively is a challenge.

2 Contribution :

In this paper, considering the web as a database, where each and every data source is represented as table. This paper mainly focuses on the scenario of data web. It focus on querying RDF[1], and RDF is referred as primitive data model, hence other models can be easily mapped into it that is XML and relational databases.

Query language :

It supports to make an expressive MashQL and intuitive query language.

Query formulation algorithm :

MashQL editor uses the query formulation algorithm that help to query a data graph without the knowledge of structure or schema of database.

Graph-signature (GS) index :

The query formulation algorithm has to query the data in real time, which degrades the performance because queries involve self-joins. Hence, new way of indexing RDF called the Graph signature and it is smaller than original graph and generate the fast response time queries.

RELATED WORK :

Formulating query is nothing but the art of allowing people to easily query a data source. Query are translated into formal languages.

Query-by-form :

It is the simple but not flexible querying method. In this method, query by form is need to be developed form for each and every query. And when changes made to query will change its form. Some methods have been proposed to semi-automate generation of form [16] and modification [17] but they may fail.

Query-by-example :

In this method, query formulate as filling tables[2]. Hence, it needs to be schematized data.

Conceptual Queries :

It requires the good knowledge of conceptual schema. Databases are modeled using EER, ORM or UML diagrams.

Natural language queries :

They allow users to write queries as natural language into a formal language (example, SQL [3], XQuery [4]). The problem is that it require language ambiguity.

Visualize queries :

The many web approaches propose to formulate an SPARQL query by visualizing its triple patterns, so it requires less technical skills. Some tools that helps to formulate XQueriesgraphically are AltovaXMLSpy [5], Bea XQuery Builder [6], QURSED [7], Stylus Studio [8] and XML-GL [9].

Visual Scripting and MashupEditor :

People write query scripts inside a module, then visualize these modules and their inputs and outputs. Example of mashup editors are Popfly [10], Yahoo Pipes [11], and Smash [12]. User uses the formal language of that editors when it needs to write query. In the semantic web community two approaches (DeriPipes [13], sparqlMotion [14]) are inspired by this visual scripting.

Interactive Queries :

LoREL [15], was developed for querying schema free XML and does not required knowledge about a schema. LoREL is closest to MashQL. There are difference between LoREL and MashQL approach. First, Schema free queries are partially handles by the LoREL. LoREL uses a summary of the data, which is called as DataGuide, where asMashQL uses the Graph Signature. DataGuide groups unrelated data items, which resulted to incorrect query formulation. To solve this, strong DataGuide were proposed, but size can be grow in case of data graph; so the DataGuide can be larger than original graph. Second, it does not support to write multiple queries on database. Third, it is expressed in basic way. Path conjunction, disjunction and negation, union, reverse properties, variables this properties supported by the MashQL approach.

Interactive searching box[18], here a new user directly write a keyword. This approach is simple and not require any knowledge about the structure of data.

DESCRIPTION OF MASHQL :

1 Data Model :

Editor assumes query of data set is constructed as directed labeled graph, and it is similar to RDF syntax. A data set D is a set of subject, predicate, Object that is <subject S, predicate P, object O>. Object can be a identifier I OR Literal L and subject and a predicate can be only be unique identifier.XML mapped into a primitive data model. Fig shows a simple example of viewing database into a graph.

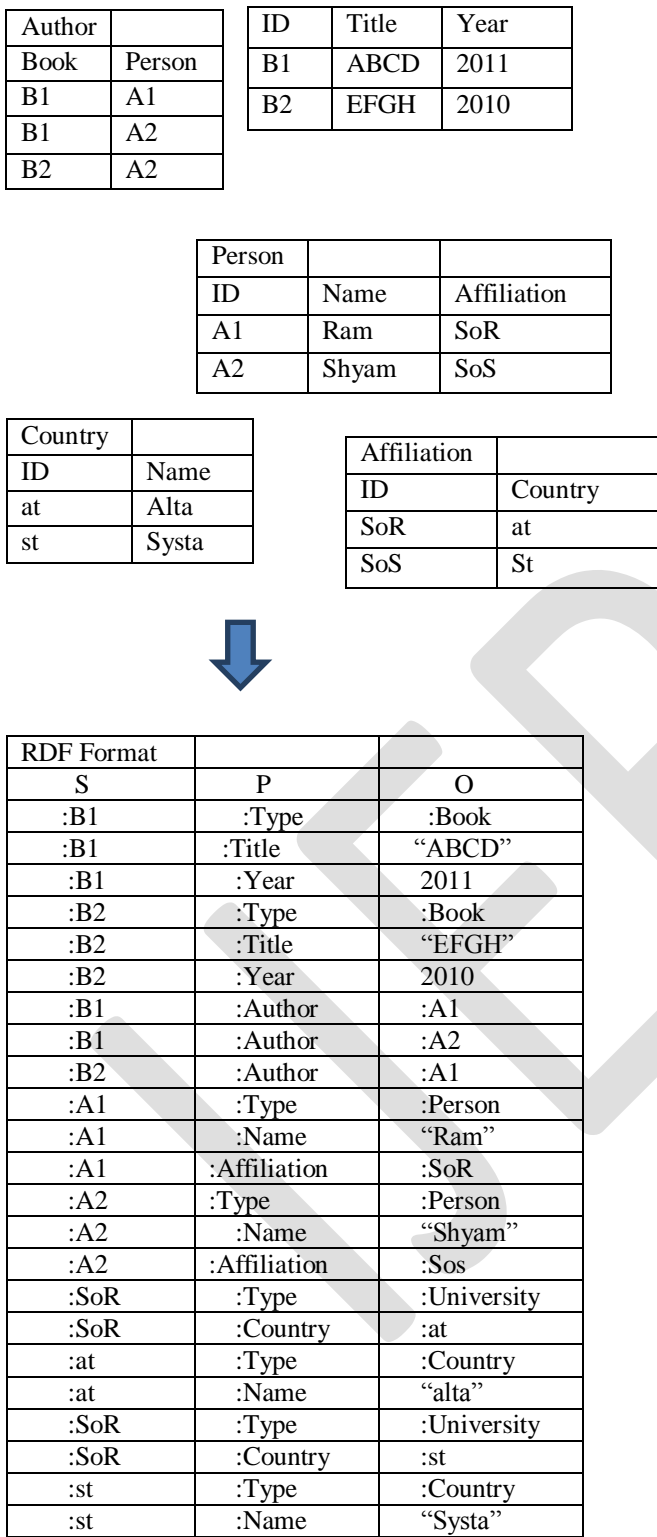


Fig1. Mapping relational database to RDF

2 Intuition of MashQL :

A MashQL question letter of the alphabet is seen as a tree. The foundation tree is termed the question subject Q(S), that is that the material being inquired. A subject matter may be a specific instance I or a user variable V. Every branch could be a restriction R, on a property of the topic. Branches may be distended to permit sub trees, referred to as question methods. During this case, the worth of a property is that the subject of sub question. This permits one to navigate through the underlying information set and build complicated queries. As are explained, every level a question is distended, it prices a be a part of once this question is executed; so, the deeper the question path is, the execution complexness will increase.

PROPOSED SYSTEM :

When gives input as a RDF data source, then it is buffered into an oracle 11g, and then the signature of graph is generated. AJAX are used to dispatch background query and translate the MahQL queries using SPARQL for execution. Oracle 11g support RDF queries and storage.

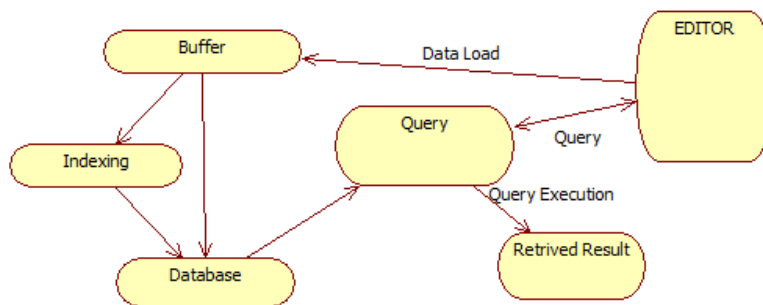


Fig2. Block Diagram of Proposed system

Alternatively develop the MashQL editor as an add on firefoxbrowser. It has same functionality of online editor. And no RDF indexing are used for indexing and questioning the data sources, but inside the editor SPARQL queries are implemented. So, the input source is limited.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to our Principal, HOD and Professors and staff members of Computer Engineering Department at Dattakala faculty of Engineering, Swami Chincholi, Bhigawan. Last but not the least, I would like to thank all my Friends and Family members who have always been there to support and helped me to complete this research work.

CONCLUSION

The system designed and formally such as the syntax and therefore the linguistics of MashQL, as a language, not just a single-purpose interface. The system conjointly such as the question formulation algorithmic rule, by that the quality of understanding an information supply (even its schema free) is touched to the question editor. And tend to self-addressed the challenge of achieving interactive performance throughout question formulation by introducing a brand new approach for categorisation RDF information. Furthermore, tend to collaborating with colleagues to use MashQL as a business rules language, therefore embrace many reaction and production operators. Arrange to conjointly support aggregation functions, as shortly as their linguistics are outlined and standardized in SPARQL.

REFERENCES :

- [1]Mustafa Jarrar and Marios D. Dikaiakos, "A Query Formulation Languagefor the Data Web"Member, IEEE Computer Society, VOL. 24, NO. 5, MAY 2012.
- [2]M. Zloof, "Query-by-Example: A Data Base Language," IBM Systems, vol. 16, no. 4, pp. 324-343, 1977.
- [3] A.Popescu, O. Etzioni, and H. Kautz, "Towards a Theory of Natural Language Interfaces to Databases," Proc.Eighth Int'l Conf. Intelligent User Interfaces, 2003.
- [4]Y. Li, H. Yang, and H. Jagadish, "NaLIX: An Interactive Natural Language Interface for Querying XML," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2005.
- [5]AltovaXMLSpy,<http://www.altova.com/solutions/xquery- tools.html>, Feb. 2010.
- [6]BEA Systems, Inc., "BEA AquaLogic Data Services Platform - XQuery Developer's Guide. Version 2.5," 2005.
- [7]M. Petropoulos, Y. Papakonstantinou, and V. Vassalos, "Graphical Query Interfaces for Semistructured Data,ACM Trans. Internet Technology, vol. 5, no. 2, pp. 390-438, 2005.
- [8]Stylus Studio, http://www.stylusstudio.com/xquery_editor. html, Feb. 2010.
- [9] S. Comai and E. Damiani, "Computing Graphical Queries over XML Data," ACM Trans. Information Systems, vol. 19, no. 4, pp. 371-430, 2001.
- [10] E. Griffin, Foundations of Popfly. Springer, 2008.
- [11] Yahoo Pipes, <http://pipes.yahoo.com/pipes>, Feb. 2010.
- [12]F. De Keukelaere, S. Bhola, M. Steiner, S. Chari, and S. Yoshihama, "SMash: Secure Component Model for Cross-Domain Mashups on Unmodified Browsers," Proc. Int'l Conf. World Wide Web (WWW), 2008.
- [13]G. Tummarello, A. Polleres, and C. Morbidoni, "Who the FOAF Knows Alice?" Proc. Int'l Semantic Web Conf. (ISWC), 2007.
- [14]SparqlMotion, <http://www.topquadrant.com/sparqlmotion>,Feb.2010.
- [15]R. Goldman and J. Widom, "DataGuides: Enabling Query Formulation and Optimization in SemistructuredDatabases," Proc. Int'l Conf. Very Large Data Bases (VLDB), 1997.
- [16]M.Jayapandian and H. Jagadish, "Automated Creation of aForms-Based Database Query Interface," Proc. VLDB Endowment,vol. 1, no. 1, pp. 695-709, 2008.
- [17] M.Jayapandian and H. Jagadish, "Expressive Query Specificationthrough Form Customization," Proc. Int'l Conf. Extending DatabaseTechnology: Advances in Database Technology (EDBT), 2008.
- [18]A. Nandi and H. Jagadish, "Assisted Querying Using Instant-Response Interfaces," Proc. ACM SIGMOD Int'l Conf. Managementof Data, 2007