

A Semantic Query Method for Deep Web*

Jiang Hao**, Liu Wen-ju, Lu Li-li
School of Computer Science and Engineering
Southeast University
Nanjing, Jiangsu Province 210096, China
hjiang@seu.edu.cn

Abstract—Based on the idea of "functionality-centric", this paper proposes a complete set of oriented semantic query methods for Deep Web, builds up the relevant software architecture, provides a new method for full use of Deep Web data resources in semantic web environment through describing the establishment of the semantic environment, re-writing the SPARQL-to-SQL query, semantic packaging of semantic query result, and the architecture of semantic query services.

Keywords—Deep Web; semantic query; SPARQ and RDF view

I. INTRODUCTION

The Deep Web, an important part of the current Web information, refers to the hidden information in the Web database. With the continuous development of computer networks, the reality of how to use effectively the Deep Web data resources becomes a hot issue.

Most of the data resources in Deep Web are stored in relational databases accounting for more than 77% of Deep Web, while the rest part of Deep Web is unstructured data sources such as texts, images.[1] Deep Web boasts for the characteristics of large-scale, structural resistance, hiding, dynamic, isolation, and heterogeneity, leading in the difficult access to the Deep Web data resources[2].

Up to now, the method of accessing to the Deep Web data sources mainly touches on approaches of "data-centric", such as Web data integration and Deep Web semantic annotation, which can't effectively solve the Deep Web resources utilization. As DERI's M.Hepp pointed out[3]: the current data-centric thinking go the wrong way, because a lot of evidence to show that in the real business environment, organizations are more willing to expose the "function" rather than "data". Therefore, he proposed the use of semantic Web services technology to mark the Deep Web functions, rather than its data.

This paper proposes the idea of "functionality-centric", using a combination of the latest Semantic Web Services and RDF data access technology. The research tries to construct Deep Web Semantic Query (DWSQ) services through providing function service so as to enable organizations to meet with client application in semantic inquiry services of the Deep Web data source under the premise of non-exposure of data sources in the semantic

Web environment.

II. RELEVANT RESEARCH WORK

Lots of relevant researches have been done at home and abroad in recent years, including following implementations:

A. Web data integration

It mainly includes two methods in traditional Web data integration and ontology-based semantic integration. The traditional Web data integration is typical for WISE-Integrator[4] and MetaQuerier[5] developed by American scholar with the idea of schema extracting of the Deep Web query interface, constructing the integrated interface, implementing data query and integration through the meta-queries. While Ontology-based semantic integration is mainly concentrated on the Deep Web depth, describing the semantics of Web data sources through local ontologies, building up the semantic mapping between local ontology and global ontology, implementing semantic integration through rewriting from ontology query to database query of local data source.

B. Deep Web semantic annotation

The Deep Web semantic annotation is using the Web ontology vocabulary to mark up the Deep Web content. It means creating semantic metadata with an easy access to the resources "understood" by the machine for the Deep Web content processing. Semantic annotation can be divided further into surface annotation and deep annotation. Deep Web surface annotation' is the method proposed by American scholars[6]. Deep Web depth annotation comes from the "Deep Annotation" program proposed by the AIFB research group of the Universities of Karlsruhe, Germany[7].

C. Web database semantic mapping/query

As Semantic Web standard, RDF[8] (Resource Description Framework) provides the way of semantic description of the properties and content for the Web resource. The RDF query language SPARQL[9] provides a semantic query on RDF data at the same time, and the two support the Semantic Web applications. The typical Web database semantic mapping / query method is D2R MAP[10].

III. SOFTWARE ARCHITECTURE

Based on the above analysis of current access technology

* This paper is supported by the programs of "Jiangsu natural Science Foundation"(No.BK2008354)

** Corresponding author: Jiang Hao. E-mail: hjiang@seu.edu.cn

and its developments about the Deep Web back-end relational database, this paper proposes the idea of “functionality-centric”. using a combination of the latest Semantic Web Services and RDF data access technology, from providing function service perspective to realize the semantic query of Deep Web back-end relational database. Semantic query service software architecture is shown in Figure 1.

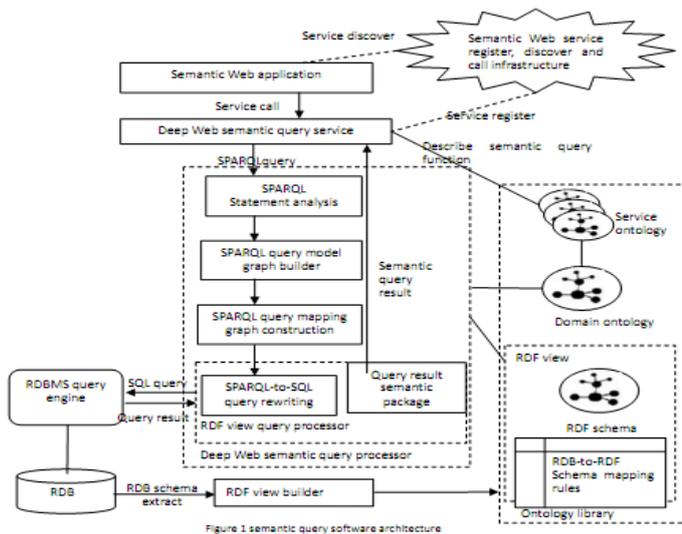


Figure 1 Semantic query service software architecture

Figure 1 shows that the main process implemented in the Deep Web semantic query service includes:

- Users input SPARQL query statement through semantic Web application.
- Preprocessing SPARQL statement, then rewriting SPARQL statement to SQL statement through SPARQL-to-SQL query rewriting mode based on RDF view.
- RDBMS accesses to Deep Web back-end relational database and return SQL query result.
- Query result semantic package schema based on RDF view pack SQL statement result into semantic query result.

Deep Web semantic query service software architecture mainly includes: RDF view builder, ontology library, Deep Web, semantic query processor and registration, foundation and invocation of Deep Web semantic query service.

A. RDF view builder

RDF view maintenance (RVM) is used for the definition and maintenance of virtual RDF view of the Deep Web back-end relational database, this view is different from the view adopted in Web database existing semantic query method. It is not a semantic model of the entire relational database. But a semantic description of the part data that the user can access through certain Deep Web query interface, the paper called it a "limited RDF view".

The definition of limited RDF view through deep Web back-end relational database implements the mapping

between RDB schema to RDF schema, and generates the corresponding RDB-to-RDF Schema mapping to provide a enough and effective data source for the Semantic Web application. This limited RDF view included RDF schema and RDB-to-RDF Schema mapping. The implementation of SPARQL-to-SQL query rewriting and query results semantic packaging modules are based on the RDF view.

The main steps of the definition of limited RDF view for Deep Web back-end RDB by RDF view builder as follows:

- 1) RDB schema abstraction[11];
- 2) RDB-to-RDF schema mapping rules[12];
- 3) Build limited RDF view.

B. Ontology library

Ontology library includes three types of ontologies and relationships among the three. Ontology includes the data source ontology, domain ontology and service ontology.

Domain ontology formally describes concepts and their relationship in the field of the Deep Web semantic query. The main role of the domain ontology is providing a shared domain semantics for the service requestor and service provider. Compared with the data source ontology, domain ontology has higher concept abstraction, and a wider range of description.

The mapping between the domain ontology and data source ontology is established in order to express the concept of relationship between the domain ontology and data source ontology.

C. Deep Web semantic query processor

After DWSQ service being called, Deep Web semantic query processor needs to execute a specific semantic query function. Deep Web Semantic query processor is an intermediary between the Deep Web Semantic Query Service and DBMS. It rewrites SPARQL statement to SQL statement and packages the SQL results. Specific process includes: the SPARQL statement analysis, SPARQL query model graph construction, SPARQL query mapping graph generation, the SPARQL-to-SQL query rewriting and query results semantic packaging.

1) SPARQL statement analysis

After Deep Web semantic query processor receiving SPARQL statement, we analyze the correctness of its grammar, then analyze SPARQL statement to extract different graph schema information. SPARQL query pattern tree is generated in accordance with the appropriate build rules. This query pattern tree is made from five kind of nodes- Element Triple Pattern, Element Optional Pattern, Element Union Pattern, Element Constraints and Element Group Pattern. In specific, Element Group Pattern can be any combination of the first four patterns.

2) SPARQL query mapping graph generation

The construction of the SPARQL query model graph still can not determine SPARQL statement should be exactly mapped to the Deep Web back-end relational database's which tables and which properties. So we need to get the mapping information between SPARQL query model graph and Deep Web back-end relational database in usage of

limited RDF view. Then we can generate corresponding SPARQL query mapping graph.

3) SPARQL-to-SQL query rewriting

SPARQL statement is composed of different graph patterns. for different graph pattern, the specific process method used to rewrite to SQL statement is different. There are mainly three methods to be described as follows:

a) Element Triple Pattern query rewriting

In the process of rewriting Element Triple Pattern to SQL statement, there are different treatments for different nodes in SPARQL query mapping graph such as non-leaf node, attribute node and table node.

b) Element Constraints Pattern query rewriting

In order to make sure of stability of the semantic value in constraint expression during the rewriting of Element Constraints to the SQL statement, the first step goes with the analysis of the value constraint expression to extract operator and operator operating expression. Then goes with second step with analysis of operating expressions until the operator to work constantly or variably. Finally, according to the correspondence between the operator in SPARQL statement and the operator in SQL statement in the SPARQL statements, there goes with the third step to rewrite the operator, constant or variable to corresponding SQL conditional expression.

c) Element Optional Pattern query rewriting

Element Optional Pattern query rewrite can be converted to a number of Element Triple Pattern and Element Constraint Pattern query rewrite. In a sense, Element Optional Pattern is composed of Element Triple Pattern, Element Constraint Pattern and Element Optional Pattern.

4) query results semantic package

After the SPARQL statement to SQL statement query rewrite, we use generated SQL statement to query Deep Web back-end RDB. At first the generated SQL statement is passed to the query engine RDBMS in Deep Web back-end RDB. Then RDBMS queries RDB and return the query results to Deep Web semantic query processor. Finally, the query result semantic package mode in Deep Web semantic query processor package the query results through the information of RDB-to-RDF pattern mapping rules and finally return the query results to the user in the form of RDF data sets.

5) Semantic query service's registration, discovery and calling

DWSQS is a semantic Web service. It check in general semantic Web service register, discover and call infrastructure. In this way, semantic Web application will finds and calls it to realize the semantic query for Deep Web back-end RDB. The output of DWSQS is RDF data sets whose semantics is described by RDF ontology or OWL domain ontology. As a semantic Web service, DWSQS semantics is described by a number of service anthologies. All of the above anthologies are created and maintained by typical ontology engineering environment and are accessible to the Web.

a) DWSQS registration

The DWSQS providers package for Deep Web semantic

query function to be realized by Deep Web semantic query processor, serving as a DWSQ semantic Web service and a mean to get registered to publish service semantic description. The processed registry grants the DWSQS requesters to finish DWSQS registration. after the DWSQ services registered so that the service requester can discover and call the Web service.

b) DWSQS discovery

When service requestors send DWSQS request to the registration center, the registration center will find services to meet the demand and to provide reference information for the service to finish DWSQS discovery.

c) DWSQS calling

when the service requester finds the real Web services through DWSQS reference information from registration center to complete the bind between service requestor and service provider, the service requestors will receive semantic query result through calling SPARQL statement and executing DWSQS.

IV. CONCLUSION

Through analyzing and summarizing the existing Deep Web semantic query method, the paper adopts the idea of "functionality-centric" to build a Deep Web semantic query service method describing its software architecture, semantic query service and implementation process. This Deep Web semantic query service without exposure to the data source stored in back-end relational database, through the service calling in the form of Web applications, can provide Web application Deep Web data source semantic query.

ACKNOWLEDGEMENT

We appreciate our great thanks to those postgraduates fighting for their academic papers in figuring out precise data in the programs of the programs of "Jiangsu natural Science Foundation"(No.BK2008354), "Model and Application for Internet-based English Oral Test System"(10YJA740061) supported by Human Social Sciences Foundation under Chinese Ministry of Education. And "Research on the Model for Combined Postgraduate Education in Arts, Engineering and Science from the Cognitive Perspective"(KJGKT-12-06).

REFERENCES

- [1] K. C.-C. Chang, B. He, C. Li, et al (2004). Structured databases on the Web: observations and implications. SIGMOD Record, 33(3): 61-70
- [2] Meng Xiaofeng, Yu Ge (2008). Deep WebSuJu preface special. Journal Of Software, 19(2): 177-178
- [3] Martin Hepp (2006). Semantic Web and Semantic Web Services: Father and Son or Indivisible Twins?. IEEE Internet Computing, 10(2): 85-88
- [4] H. He, W. Meng, C. T. Yu, et al (2004). Automatic integration of Web search interfaces with WISE-Integrator. The VLDB Journal, 13(3): 256-273
- [5] B. He, Z. Zhang, K. C.-C. Chang (2005). MetaQuerier: querying structured web sources on-the-fly. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp. 927-929

- [6] Y. Lu, H. He, H. Zhao, et al (2007). Annotating Structured Data of the Deep Web. Proc. of the 23rd Int'l Conf. on Data Engineering, pp. 376-385
- [7] R. Volz, S. Handschuh, S. Staab, et al (2004). Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the Semantic Web. Journal of Web Semantics, 1(2): 187-206
- [8] G. Klyne, J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. [10 Feb 2004]. <http://www.w3.org/TR/rdf-concepts/>.
- [9] E. Prud'hommeaux, A. Seaborne, Bristoleds, et al. SPARQL Query Language for RDF. W3C Recommendation. [15 January 2008].
- [10] C. Bizer (2003). D2R MAP - A Database to RDF Mapping Language. Poster presentation at the 12th Int'l World Wide Web Conf.
- [11] Xu Zhuoming, Su Wenping (2005) . A relational database model of information extraction. Journal of Hohai University(Natural Sciences) [J],2005,33(2):202-206(In Chinese).
- [12] Syed Hamid Tirmizi, Juan Sequeda, and Daniel Miranker. Translating SQL Applications to the Semantic Web. The 19th Int'l conf. on Database and Expert Systems Applications, 450-464.