Distributed ontology based information retrieval using semantic web

Chun Zhang*

Shantou Radio and TV University, Shantou, Guangdong, 515041, China

Received 1 March 2014, www.cmnt.lv

Abstract

In recent years, user demand for integrated searches over several independently operating semantic web systems have been increasing rapidly. Integrated semantic searches enable more meaningful results to be generated because information with similar meanings in diverse areas and domains is likely to be used for inference. However, it is not easy to integrate physically independent, distributed, and heterogeneous database systems to provide a single, integrated semantic web system to end-users. In this paper, we propose a novel system that integrates heterogeneous semantic web systems based on schema mapping. The user can generate only one SPARQL query using the integrated schema without the necessity of checking the schemas of individual systems each time thereby reducing additional costs to generate queries for individual systems. Furthermore, the user is not required to collect individual query results manually after performing a query and additional costs for establishing systems can be reduced because no change in existing system structures is required. If currently established systems are expanded by adding the schema structures of other ontology systems, the cost to establish another integrated retrieval system can be saved. To evaluate the effectiveness of our approach, we have implemented a prototype that integrates two national information retrieval systems.

Keywords: integrated information retrieval, ontology, schema mapping, semantic web

1 Introduction

As the semantic search has been positioned as a killer service, many conventional information retrieval systems have been transformed into semantic web systems. In the early days, individual semantic web systems were built based on their own requirements and operated independently. As a result, users of a particular semantic web system were presented with the information managed only by the system. More recently, user demands for integrated searches over several independently operating semantic web systems have been increasing rapidly. This has occurred because integrated semantic searches enable more meaningful results to be generated, as information having similar meanings in diverse areas and domains is likely to be used for inference. However, it is not an easy task to integrate physically independent, distributed, and heterogeneous database systems to provide a single, integrated semantic web system to end-users. For physical integration, existing legacy data from participating systems must be transformed according to the integrated schema and whenever new data are accumulated in the participating systems, the transformation process must be repeated.

The Semantic Web [10-13] is a collaborative movement led by the international standards body, the World Wide Web Consortium (W3C). The standard promotes common data formats on the World Wide Web. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web dominated by unstructured and semi-structured documents into a "web of data". The semantic web technology can be used in the pattern recognition layer. For example, with the help of the specific domain ontologies, the objects and relations of videos can be detected accurately.

A semantic link network (SLN) [14-17] is a relational network consisting of the following main parts: a set of semantic nodes, a set of semantic links between the nodes, and a semantic space. Semantic nodes can be anything. The semantic link between nodes is regulated by the attributes of nodes or generated by interactions between nodes. The semantic space includes a classification hierarchy of concepts and a set of rules for reasoning and inferring semantic links, for influence nodes and links, for networking, and for evolving the network. The semantic link network can be used in the video resources layer [18-20]. For example, with the help of the semantic link network model, the videos can be organized with their semantic relations.

In this paper, we propose a novel system that integrates heterogeneous semantic web systems based on schema mapping. The proposed system works by first creating integrated schema that includes all the attributes of the ontology schemas of participating semantic web systems (e.g., local schema). In the process, it maintains schema-mapping information that indicates which attribute of the local schema corresponds to that of the integrated schema. Second, user queries are generated against the integrated schema. Third, for query execution, the system re-

***Corresponding author** e-mail: cronychun@vip.qq.com

generates actual queries from the original user query in such a way that the attributes of the integrated schema are replaced with the corresponding attributes of the local ontology schema of the individual semantic web systems using the schema-mapping information. The core concept of the above-mentioned process is depicted in Figure 1.

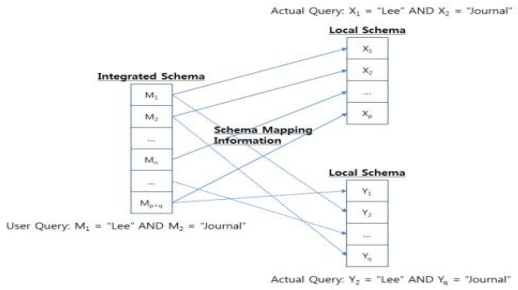

FIGURE 1 The schema mapping and query regeneration concept

To evaluate the effectiveness of our approach, we have implemented a prototype that integrates two national information retrieval systems, the National Discovery of Science Leader (NDSL) and the National Science and Technology Information Service (NTIS). The NDSL is an online database retrieval system that began service in 1962; it retains more than one hundred million pieces of information regarding research papers and reports, patents, standards, factual information, etc. The NTIS was developed in 2008 to enhance R&D investment efficiency by sharing and jointly utilizing information related to national R&D projects and science technology that are managed separately by individual government departments. Approximately 3.5 million pieces of data have been accumulated in the NTIS including national R&D and evaluation committee members, project information, result information, and research equipment information. Since the NDSL maintains only those pieces of information contained in journals, it does not retain information on what research projects‟ outcome the papers contained in the journals are. On the other hand, the NTIS does contain information about papers and reports for the outcomes of R&D projects carried out since 2008. Therefore, if the NDSL and the NTIS were linked to perform integrated retrieval, the value and the reliability of information from the two systems could be enhanced to be complementary.

The organization of this paper is as follows. Section 2 presents related work and Section 3 describes the proposed system architecture in detail. Section 4 presents classification of integrated queries and methods for sub-query generation. Section 5 presents experimental results. Finally, Section 6 provides a summary and proposals for future work.

2 Related works

Similar to ours, the ISENS [1] system provides a function to integrate and retrieve different real-world data sources having different ontologies. This system is less useful because queries in this system cannot be answered independently using a single system; instead, integrated queries can be made only to systems composed of mutually complementary data. In [1], mapping information for ontology schemas was not gathered in one place, but instead was made only for two ontologies with fields that can be mapped. Therefore, the existence of mapping information cannot be known without accessing the source system, which means that the mapping information can be accessed only through navigation. The largest difference between the present paper and [1] is that, instead of creating queries appropriate for individual local sites using mapping information, performing the queries, and integrating the results, in the case of [1]. The same query is performed using mapping information, the next system is accessed using the KAON2 reasoner to collect information, and results are presented but each database system cannot be accessed without using the system's source description and the source selection algorithm.

The DARQ system in [3] is also intended to perform integrated retrieval for distributed systems. However, this study is quite different from the study set forth in the present paper. In the DARQ system, heterogeneous data should be accessed using wrappers and the service description describes the kinds of data and access patterns that can be used for individual sites (endpoint) using sets of predicates. The DARQ system is different from the system in the present paper in that it focuses on query optimization using statistical information for integrated retrieval.

The SECO system in [2] enables efficient collection of any RDF files existing on the Web and provides interfaces in the form of HTML so that users can easily identify integrated data. This system is composed of a collector, a wrapper, a transformer, a user interface, a remote query interface, and data storage. The data storage is composed of multiple different sets of RDF data. Among them, MetaModel has Metadata information collected from files. SourceModel stores original RDF triples collected from files existing on the web and triples created here are purified through the transformer and stored in the TargetModel thereafter. The TargetModel enables access to user interface for creating HTML and remote query interfaces for query processing. The MetaModel, the UsageModel, and the TargetModel are described as ontologies and the SourceModel is composed of diverse schemas without any particular form.

3 The proposed architecture

Figure 2 depicts the overall architecture of the proposed system. The primary components of the system are the Schema Manager, the Query Manager, and the Result Manager. In what follows, we present detailed information about each component.

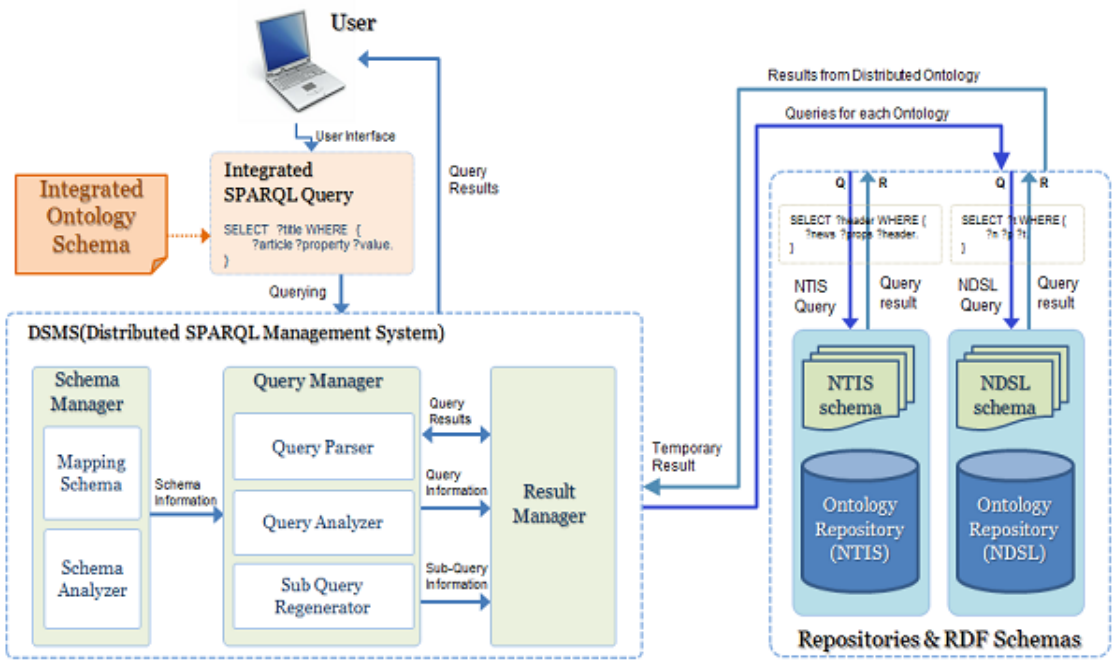

FIGURE 2 The system architecture

The NDSL and the NTIS define their own ontology schemas. Using the individual schemas, the administrator connects classes and attributes having the same meaning but different names with each other to create integrated schema that can link and retrieve two ontologies together. Therefore, the Schema Manager must maintain the Resource Description Framework (RDF) schema created by the administrator, as well as RDF schemas for individual ontologies of the NDSL and the NTIS. Both the NDSL and the NTIS have information on papers and the authors of the papers in common but they represent the information in different ways. For example, the People class in the NDSL schema and the Person class in the NTIS schema both indicate a human, but they are expressed differently. In contrast, both the NDSL and the NTIS schemas have a Journal class to represent a journal. Among the attributes defined in the classes, korName, hasName, write, and hasFirstAuthor are the attributes with the same meanings in both schemas. Therefore, the primary role of the Schema Manager is to maintain information about how individual classes and attributes defined in one schema are mapped to those defined in another schema. Note that classes and attributes in the integrated schema contain the level information, which is the core information needed by the sub-query generator. For example, classes such as Paper and Project are assigned to level 1, while hasFirstAuthor and korName attributes are assigned to level 2 and level 3, respectively. The level is related to the linkage of the class or the attribute. For instance, the hasFirstAuthor attribute has the identifier of the author but it does not have the name of the author. To acquire the author name, another triple condition such as ID :korName authorName must be executed. Therefore, the levels are constructed using the concept of linkage. How the level information is utilized will be described in the next section. The Schema Analyser analyses the schema information in the single SPARQL query submitted by the users. Then, it compares it with the RDF schemas of individual ontologies. Next, it delivers the information of the corresponding ontologies to the Query Manager.

The user creates queries based on integrated schema regardless of whether the ontologies managed by different information retrieval systems exist. The user query is first analysed using individual ontology schemas and the mapping schema. Subsequently, sub-queries are generated that are suitable for the corresponding ontologies. The Query Manager consists of the query parser, the query analyser, and the sub-query generator. When the query is verified by the query parser, the type of the query is determined through the query analyser. Using the integrated schema created by the administrator, the query parser checks the validity of the class and the attributes in the query. Then, using the schemas of individual ontologies, the query analyser determines whether the user query can be transformed into sub-queries for individual ontologies. Finally, the sub-query generator re-generates sub-queries. The sub-query generator and the query type will be dealt with in detail in the next section.

The Result Manager manipulates the intermediate results obtained by performing sub-queries in individual ontology for further processing or preparation for the final results. For example, depending on the query type, the Results Manager uses the results from one sub-query as filter information for another sub-query or it combines the results of sub-queries for the final query result.

4 Query generation

The query types determined by the query analyser are automatically generated into four types depending on the content of the information that the integrated query will retrieve. Type-1 queries represent those queries using the schema that exist only in one ontology. The integrate query includes the „Project" class, which exists only in the NTIS schema. Therefore, there is no corresponding sub-query for the NDSL. Figure 3 shows the control and data flows for processing the Type queries.

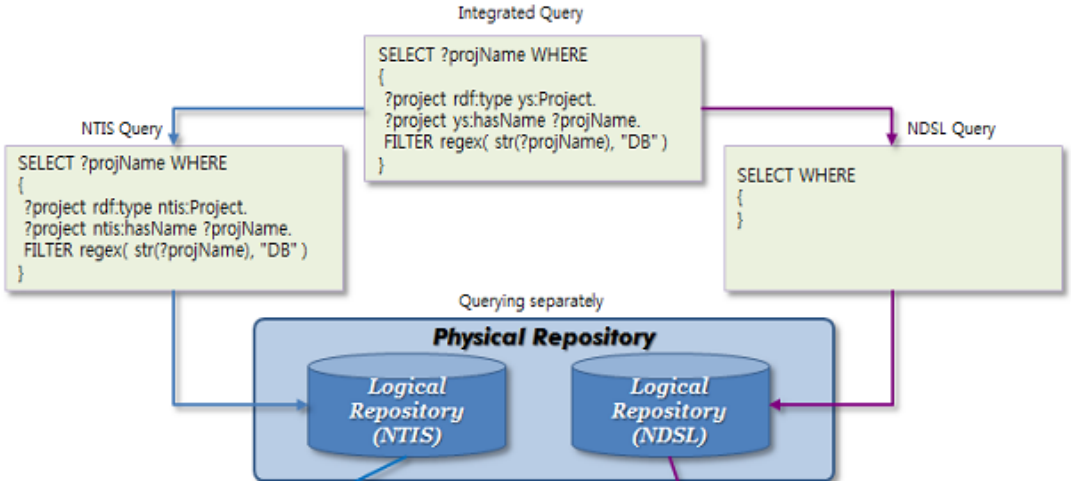

FIGURE 3 Control and data flows for processing Type-1 queries

Type-2 queries include classes and attributes that exist in every ontology. When retrieved using classes and attributes such as Paper and Author that the NDSL and the NTIS have in common, the query result is in the form of a combination of individual sub-query results. Figure 4 shows control and data flows for processing Type-2 queries. In Type-3 and Type-4 queries, some of the classes and attributes included in the queries exist in every ontologies, but the other classes and attributes are defined only in one ontology. For instance, the „Author" attribute exists both in the NDSL and in the NTIS schemas, whereas the Project attributes exists only in the NTIS schema. When queries are received of these types, the sub-query generator first separates the commonly existing classes and attributes from those that exist individually. For those classes and attributes that exist in only one ontology, they will be eliminated from the sub-queries for the ontologies that do not support them.

The example for a Type-3 query is "papers written by those who participated in the project for establishment of a driving safety DB and development of operating technology".

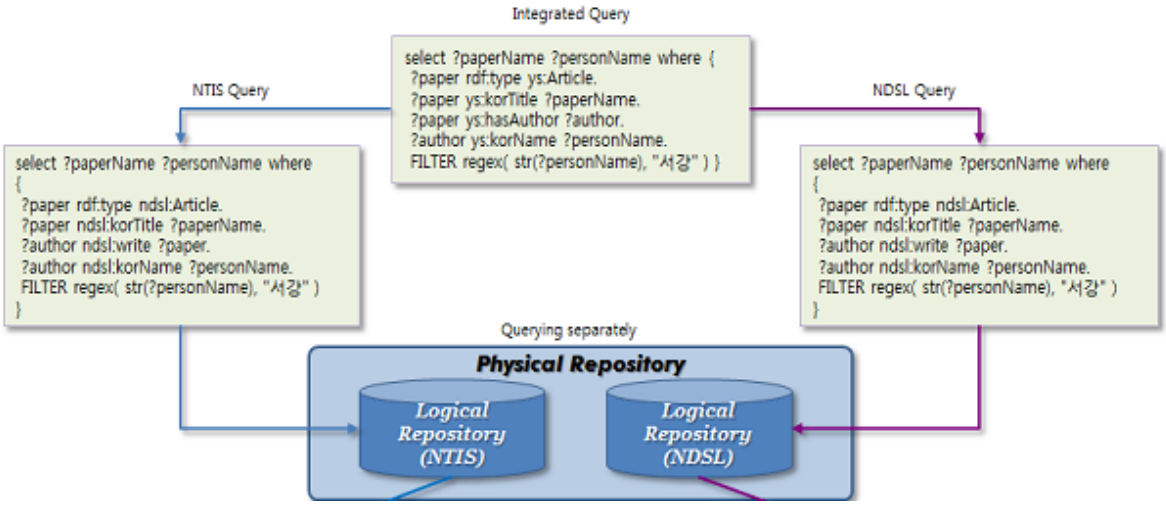

FIGURE 4 Control and data flows for processing Type-2 queries

The Figure 4 Control and data flows for processing Type-2 queries example for a Type-4 query is "projects in which the author of the paper „Diesel Engine participated". Project and participateIn in the integrated query do not

appear in the sub-query for the NDSL because the relevant classes and attributes do not exist in the NDSL schema. In the sub-query for the NTIS, the class Article is transformed into Paper and the attribute korTitle is transformed into hasName.

Type-3 and Type-4 queries have similar basic preconditions but differ from internal processing. For Type-3 queries, intermediate query results are first obtained by using classes and attributes that exist exclusively in one ontology. Then the commonly existing classes and attributes are applied to the intermediate query results. On the other hand, Type-4 queries apply query results obtained using common classes and attributes to queries for exclusive classes and attributes in a certain ontology.

To process Type-3 queries, given the two ontologies A and B, a query that includes the schema for A must be executed first. A query for B is also generated up to the WHERE clause excluding the FILTER clause. The FILTER clause to be included in the query for B will be generated by analyzing the intermediate query results from A and the queries for A and B (The query for A is complete, whereas the query for B is incomplete). First, all attributes that exist exclusively in the schema for A are extracted. Then, they are compared with the schema for B and the intersection of the two in the lowest level. Finally, the intersection and query results obtained from A are compared to generate the FILTER clause and complete the query for B. For instance, a Project exists in the NTIS schema and its level is one. The „Project" has participant for level two. The participant has a participantName, which is level three. The participantName is matched with authorName of Author of Paper in the NDSL schema. The FILTER clause is completed through these traces. For Type-4 queries, the FILTER clause is generated similarly to Type-3 queries. However, it is different from Type-3 queries in that the query results for individual ontologies extracted using classes and attributes that commonly exist in A and B. The results are then used to generate a FILTER clause for classes and attributes that exist exclusively in A or B.

5 Experiment and results

For each query type, we generated ten queries automatically and measured the performance of the proposed system in terms of the number of retrieved tuples and processing time. NDSL query results are not indicated because the integrate queries use only classes and attributes that exist exclusively in the NTIS schema. Sub-queries are generated for individual ontologies and the results are brought and integrated. In this case, the sum of the two query results differ slightly because overlapping values exist in the two query results, and they are treated as a single value.

NTIS query results are the same as the integrated query results. This indicates that the results obtained from the NDSL did not greatly affect obtaining the results from the NTIS. In other words, in the case of projects in which the author of a certain paper has participated", the author of the particular paper in the NDSL had hardly participated in NTIS projects or he participated in some cases but the cases were removed as overlapping results. This can be considered an issue of the scope and diversity of the data existent in each ontology. Queries for individual ontologies were not processed in parallel with each other because there are cases where the result values of queries for one ontology are included in the conditional clauses of queries for the other ontology. Although the results of individual queries were brought from ontologies, the integration of the results and the removal of overlapping results were made by the Result Manager using the memory of the relevant system. On reviewing individual experimental results, it can be seen that the time spent for integration is not something about which to be greatly considered.

6 Conclusions

In this paper, we proposed a novel system that will enable storing data from two different ontology systems in one physical system without converting the data to conduct integrated retrieval. Based on the individual schemas of the separated systems, the administrator can connect the schemas that have the same meanings but different forms of expression with each other between the two systems to generate integrated schemas. The user can then generate integrated SPARQL queries utilizing the schemas generated by the administrator to perform queries without recognizing the existence of individual ontologies. The user can generate only one SPARQL query using the integrated schema without the necessity of checking the schemas of the individual systems every time thereby reducing additional costs to generate queries for individual systems. Furthermore, the user is not required to collect individual query results manually after performing a query and additional costs for establishing systems can be reduced because no change in existing system structures is required. If currently established systems are expanded by adding the schema structures of other ontology systems, the cost to establish another integrated retrieval system can be saved. Although the complexity of applications will increase in this case, it should be a trivia l problem compared to the cost to integrate several millions or several dozen millions of triples manually.

In the future, the performance of result integration algorithms should be improved by adding more triple data and more query types should be added. If the system is complemented by establishing multiple ontologies in completely distributed network environments instead of a single physical storage, a more reliable system can be implemented.

References

[1] Abir Q, Dimitre A, Jeff H 2009 ISENS: A system for information integration, exp loration, and querying of multi-ontology data sources *Proceedings of the 2009 IEEE International Conference on Semantic Computing* 330-5

[2] Andreas H 2004 An integration site for semantic web metadata," proceedings of world wide web conference 1 229-34

[3] Bastian Q, Ulf L 2008 Querying Distributed RDF Data Sources with SPARQL

[4] Dimitre A, Dimitrov H J, Abir Q, Nanbor W 2006 Information integration via an end-to-end distributed semantic web system *Lecture Notes in Computer Science* 4273 764-77

[5] Mayfield J, Finin T 2003 Information retrieval on the Semantic Web: Integrating inference and retrieval *SIGIR 2003 Semantic Web Work shop*

[6] http://jena.apache.org/documentation/query/.

[7] http://linkeddata.org/.

[8] http://www.w3.org/TR/rd f-mt.

[9] http://www.w3.org/TR/rd f-sparql-query/.

[10] Liu Q, Zhang, Ni L M 2010 *IEEE Transactions on Parallel and Distributed Systems* **21**(3) 405-16

[11] Menzel M, Ranjan R, Wang L, Khan S, Chen J 2014 CloudGenius: a hybrid decision support method for automating the migration of web application clusters to public clouds *IEEE Transactions on Computers, in press*

[12] Hao F, Min G, Chen J, Wang F, Lin M, Luo C, Yang L T 2014 An optimized computational model for task-oriented multi-community-cloud social collaboration *IEEE Transactions on Services Computing in press*

[13] Qi L, Dou W, Chen J 2014 Weighted principal component analysis-based service selection method for multimedia services in cloud computing *Computing, Springer in press*

[14] Wang L, Tao J, et al. 2013 G-Hadoop: MapReduce across distributed data centers for data-intensive computing *Future Generation Computer Systems* **29**(3) 739-50

[15] Xu Z, et al. 2014 Knowle: a semantic link network based system for organizing large scale online news events *Future Generation Computer Systems*, 10.1016/j.future.2014.04.002

[16] Xu Z, Luo X, Zhang S, Wei X, Mei L, Hu C 2013 Mining temporal explicit and implicit semantic relations between entities using web search engines *Future Generation Computer Systems* DOI:10.1016/J.future.2013.9.027

[17] Liu Y, Ni L M, Hu C 2012 *IEEE Journal on Selected Areas in Communications* **30**(9) 1780-8

[18] Luo X, Xu Z, Yu J, Chen X 2011 *IEEE transactions on automation science and engineering* **8**(3) 482-94

[19] Hu C, Xu Z, et al. 2014 Semantic link network based model for organizing multimedia big data. *IEEE Transactions on Emerging Topics in Computing* 10.1109/TETC.2014.2316525.

[20] Liu Y, Zhu Y, Ni L M, Xue G 2011 *IEEE Transactions on Parallel and Distributed Systems* **22**(12) 2100-7

Author

Chun Zhang, born in May, 1970, Shantou, Guangdong Province, China

Current position, grades: a lecturer of Shantou Radio and TV University, China.
Scientific interest: data structure and management information systems
Publications: 11 papers.