SEMANTIC QUERY TRANSFORMATION FOR INTEGRATING WEB INFORMATION SOURCES

Mao Chen, Rakesh Mohan, and Richard T. Goodwin
IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532, USA

Keywords: Information integration, Query transformation, Semantic information, Ontology, Web services

Abstract: The heterogeneousness and dynamics of web information sources are the major challenges to Internet-scale information integration. The information sources are different in contents and query interfaces. In addition, the sources can be highly dynamic in the sense that they can be added, removed, or updated with time. This paper introduces a novel information integration framework that leverages the industry standards on web services (WSDL/SOAP) and ontology description language (RDF/OWL), and a commercial database (IBM DB2 Information Integrator—DB2 II (DB2 II)). Taking advantage of the data integration and query optimization capability of DB2 II, this paper focuses on the methodologies to transform a user query to the queries on different sources and to combine the transformation results into a query to DB2 II. By wrapping information sources using web services and annotating them with regard to their contents, query capabilities and the logical relations between concepts, our query transformation engine is rooted in ontology-based reasoning. To the best of our knowledge, this is the first framework that uses web services as the interface of information sources and combines ontology-based reasoning, web services, semantic annotation on web services, as well as DB2 II to support Internet-scale information integration.

1 INTRODUCTION

Efficient information integration from various sources is critical to Internet-scale business systems. In contrast to traditional full-fledged and stable information sources such as databases, web information sources are distinct in their heterogeneity and dynamics. First, web sources are heterogeneous in content hence a single information source usually provides only part of the answer for a user query. In addition, web sources have different query capabilities that are reflected in the various query schemas. Furthermore, web sources are highly dynamic in the sense that new sources are added continuously, old ones may become unavailable, and existing ones are updated frequently in terms of both the query interface and the contents.

The web service technology (W3C '02) provides a machine-usable interface to wrap the information sources that are conventionally accessible only via human-understandable query forms. Via a web service wrapper, any structured databases, file systems, unstructured web pages and other information sources can be treated equally in Internet-scale information integration.

This paper proposes a novel framework for information integration from heterogeneous and dynamic sources. Our framework leverages industry standards on web service and ontology, and an IBM database system. Namely, IBM DB2 Information Integrator (DB2 II) acts as the back end for hosting information from various sources and generating optimized query plan to the sources.

The key challenge in the proposed framework is transforming a user query to a valid DB2II query. Our query transformation mechanism consists of two phases. Phase 1 customizes a user query into the queries to different sources. The transformation results are used in the second phase to generate a query as an input to DB2 II. The corner stone of our query transformation algorithm is ontology-based reasoning. Ontology is used to describe user's view, the query schemas of the web services, and the relations between different concepts.

The major contributions of this paper are three folds:

1.) Proposing a novel framework for Internet-scale information integration using web services, ontology technology and commercial databases;

Chen M., Mohan R. and T. Goodwin R. (2005).
SEMANTIC QUERY TRANSFORMATION FOR INTEGRATING WEB INFORMATION SOURCES.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 176-181
DOI: 10.5220/0002528801760181

2.) Proposing a set of reasoning rules for transformation between different schemas;
3.) Presenting an ontology-based annotation scheme for describing query interfaces of web services which can be an extension of OWL-S/DAML-S (DAML, Burstein '02).

2 RELATED WORK

Integrating information from heterogeneous sources has been an important problem in very large databases management (Arens '96, Genesereth '97, Gio '00, Madhavan '03). The integration systems can be classified as query-centric and source-centric. The query-centric systems choose a set of users' queries and provide the procedure to customize those queries for the available sources (TSIMMIS '94, HERMES '95). As a representative of source-centric systems, InfoManifold describes sources' contents and query capabilities, and transforms each new query based on the descriptions (Levy '96).

Both types of systems focus on query planning optimization using certain criteria, but use light-weight transformation between different concept spaces. Our work is distinct from the previous efforts in three ways.

First, the query plans generated by these integration systems are usually not optimized at the execution level. In contrast, many commercial databases such as IBM DB2 II have powerful query planning engines that use sophisticated algorithms based on execution cost, statistics on usage, and other parameters as regard to the running environment (Haas '97). Our methodology takes advantage of the query optimization capabilities of DB2 II therefore guarantees efficient query execution in run time.

The second distinction between our work and the previous work is the transformation mechanism. The transformation in the previous work is light-weight. Bussler et. al. indicate that combining ontology technology and web service technology is important for making web information machine-processable (Bussler '02). Based on this idea, our information integration framework uses ontology-based reasoning to handle discrepancy between different concept spaces.

Finally, the traditional systems usually rely on ad-hoc wrapper languages and models, which makes adding or changing services in such an integration system a heavy burden on the service provider side (TSIMMIS '96). Since web services can be added or removed without recoding the integration engine and the wrappers, our framework is best suited for the dynamic environment such as web.

3 ARCHITECTURE OF OUR INFORMATION INTEGRATION SYSTEM

Figure 1 outlines the conceptual architecture of our information integration system. A user can query the integration system through SQL statement as to a conventional database. Each web source is wrapped and presented using a web service that is mapped to a virtual table in DB2 II. Using DB2 II built-in capability for federating web services, the integration system transforms a user query to queries to web services, integrates results from the web services, and returns the integrated result to user.

Our integration system consists of three functional modules. The front end of our integration system has a query transformation engine (QTE) and a query generator. QTE is in charge of customizing a user query into the valid queries of the web services. Based on the transformation result, the query generator creates a valid DB2 II query on all the related web services and triggers DB2 II with the query. At the back end of our integration framework sits IBM DB2 II. DB2 II generates optimized executable query plan that calls all the related web services and returns the aggregated results to users.

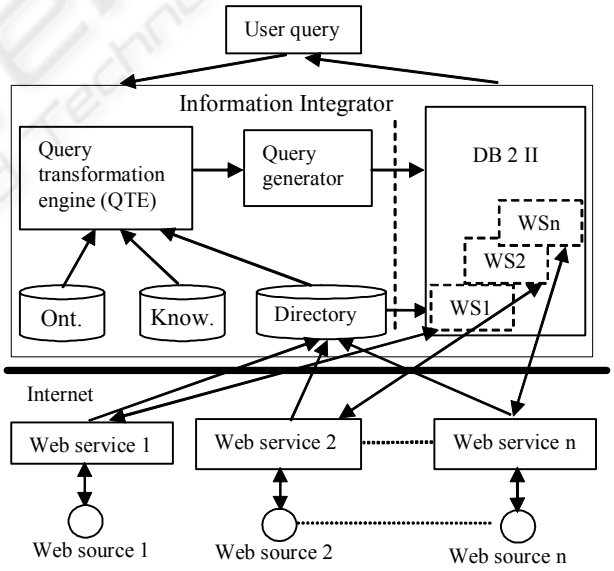

Figure 1: Architecture of our information integration system for web sources

Our query transformation (by QTE) and query generation (by query generator) are accomplished based on two types of knowledge. The first type of knowledge is semantic information about the services. The knowledge source "Ontology" stores the query capability of each service and the relations

between different concepts. The "Knowledge base" holds the information that cannot be described using ontology, for example, the mathematical relations between the concepts. The second type of knowledge is about web services. The "Directory" provides registry service to web services and updates the virtual tables of web services in DB2 II. We envision that the directory service can be implemented by enhancing semantic UDDI service (UDDI) as proposed in many works (Akkiraju '03).

Given the query optimization capability of DB2 II, the major challenges of the above infrastructure include annotating web services about their query capabilities, automatically transforming user query to the valid query for each web service, and generating an executable query plan for DB2 II. The next section presents our mechanisms to deal with the three issues.

4 SEMANTICS-BASED QUERY TRANSFORMATION

This study uses a used-car searching service as an application scenario to introduce our information integration framework. Given a user query on used car information, this service intelligently inquires and integrates the results from three sites, Yahoo Autos (Yahoo), Autos MSN (MSN) and Kelly's Blue Book (KBB). Yahoo and MSN provide on-line retailing and auction information about the used cars, and KBB is an authority site that provides a suggested retail price for a car when given car information such as make, model, and year.

A user's concept space about used car information includes two parts: the query and the result. A user can search for used cars based on *user's location*, *searching area*, *make and model*, *year*, *mileage* and *price*. The most interesting results to a user are *year*, *mileage*, *asked price*, *KBB suggested price*.

Our information integration system aims at transforming an SQL-like user query as follows:

```
SELECT * FROM car
WHERE make = 'Acura' AND price <= 15000
```

Into a valid query of DB2 II that stores the aforementioned web services:

```
SELECT make, model, mileage, price
FROM   YahooAuto
WHERE  make='Acura' AND maxprice=15000
UNION ALL
SELECT make, model, mileage, price
FROM   MSNCars
WHERE category = 'Passenger Cars' AND
       make = 'Acura' AND price = 15000
```

"*Union*" links queries each of which is valid to a web service. The final combined DB2 II query is formed based on the relations among the user's query, the query capability and the contents of each web service.

4.1 Describing Web Services as Ontology

We annotate the semantic information about web services using Protégé ontology editor and knowledge acquisition system (Protégé-2000). The resulting ontology is represented as RDFS and RDF.

A web service is an instance of the class "web source" which has three properties: the service name, the query class (input schema), and the output class (output schema). Tables 1 and 2 show the query class and the output class for Yahoo.

Table 1: Query class of Yahoo

Properties	Range	Required
User Position	{User Location}	Yes
Search Within	{Search Area}	No (50 miles)
Car Make	{Manufacture}	No
Car Model	{Model}	No
Mileage LessThan	{Car Mileage}	No
Mileage MoreThan	{Mileage}	No (0 mile)
Year LessThan	{Car Year}	No (2004)
Year MoreThan	{Car Year}	No (1940)
Price Range	{Price Range}	No

Table 2: Output class of Yahoo

Properties	Range	Required
Asked Price	{Car Price}	Yes
Mileage Is	{Mileage}	Yes
Car Type	{Make Model}	Yes
Car YearIs	{Car Year}	Yes

The symbols in the braces refer to class, and those in the brackets are the default values. Table 1 also shows that only the user position is required by Yahoo Autos. Autos MSN and Kelly's Blue Book have different input and output schemas from Yahoo Autos which are not shown due to the space limit.

4.2 Transforming a User Query to the Queries to the Web Services

This section presents the solutions for seven types of schema mismatch. The first four rules handle two pairs of dual transformations for abstract model and instance model. The fifth and the sixth rules are for transformation between different abstract models. The last rule handles the mismatches in searchable attributes at both abstract and instance levels.

4.2.1 Concept Mapping

One of the most common difficulties in dealing with heterogeneous schemas is that a same concept has different names in different sources. This mismatch can be handled using concept mapping or renaming. In this study, renaming is done by mapping different names to a common concept using "RDFS:range". For example, two equivalent concepts "Yahoo User Location" and "MSN User at" can be mapped to the same class "User Location".

If using ontology description language OWL (OWL 2004), one can use "OWL:EqualProperty" to indicate the equivalence of the above two properties.

4.2.2 Instance Mapping

In practice, same instance may have different names in different sources. For example, "New York" and "NY" refers to the same state instance. Instance mapping is an analogue to *Concept Mapping*.

Instance mapping can be achieved by using "OWL:sameAs" description. The following example shows the equivalence of "New York" and "NY":

```
<UsedCar rdf:ID="New York">
  <owl:sameAs rdf:resource="#NY" />
</UsedCar>
```

4.2.3 Concept Folding

Different sources may allow queries at different levels of granularity for a given attribute. For example, Kelly's Blue Book requires queries on "Car Type" which combines "Manufacture" and "Model" as a single attribute, while Yahoo allows queries to specify "Make" and "Model" separately. We call the transformation from fine-grained level to a coarser-grained level as *concept folding*.

Using RDFS, concept folding can be achieved by annotating fine-grained concepts as properties of the coarse-grained concept. In OWL, the two concepts "Make" and "Model" can be defined as "sub property" of the property "Make Model".

4.2.4 Instance Folding

Different from *Concept Folding* that merges fine-grained concepts into an equivalent single concept, *Instance Folding* or *Concept Expanding* extends an instance into a more general instance.

Assume a user's query includes two parameters "Make" and "Model", but a web service like MSN supports car searching only on "Car Category". A car category includes many car types hence query transformation needs to extend a specific car type searching into a more general category searching.

We define the class "Car category" with two properties that are "Make" and "Model". The relation between each category and each pair of make and model is described by the instances in a RDF file, as shown in figure 2. With this knowledge, one can transform a user's query such as

```
Where Make = Acura" and Model = "CL"
```

Into the following query to MSN:

```
Where Car Category = "Passenger Cars"
```

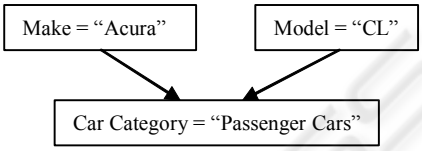

Figure 2: Instance folding of "Acura" and "CL"

Instance folding loosens the searching criteria for maximizing the usage of all the related sources, therefore the results should be filtered based on the original user request.

4.2.5 Inequality Inference for Concepts

Generally speaking, a web service may not offer a full set of comparison operators for an attribute, but a user's query may consist of any comparison operator. Limited query capability is a fundamental difference of web service from databases.

For the same attribute, some web services accept equality queries, while others use range searching. For a range searching, a service may allow the range to have one open-end or with both ends open. Therefore the semantic analysis on each service's query capability with inequality queries is necessary.

For transforming a user requested comparison operator to an available operator to a web service, we identify a complete set of transformations between any pair of comparison operators that include <, <=, =, >=, and >. For example, when a user's query includes "< N" for an attribute *A* and a service allows only equality searching on *A*, the user's query can be transformed into "{< Max + 1} - {< N +1}" where {} – {} denotes set difference.

In this study, the semantic meaning of inequality query capability is annotated using property name. For example, the class "Car Price Range" has two properties "Price Less Than" and "Price Greater Than" that describe a range searching on car price with two open ends. The semantic meaning of the comparison operators ">" and "<" are encoded as "Greater Than" and "Less Than". A user's query including "Where price < 20000" is transformed as "Price **Less Than** = 20000" in the query to the corresponding web services.

4.2.6 Mathematical Reasoning for Concepts

Not all relations between concepts can be described using ontology language. One example is that neither RDFS nor OWL can represent the mathematical relations between the concepts.

For example, MSN accepts queries on car's age, while Yahoo allows searching a car based on the upper bound and the lower bound of a car's production year. A mathematical transformation is required between the two concepts "Car age" and "Year MoreThan" using constant "current year":

```
Year MoreThan = Current Year – Car age
```

4.2.7 Mismatch Handling for Attributes

There are two reasons for the attributes specified in a user query to be unsearchable in a web service. The first reason is that the attribute set in user's query does not match that is used by a web service, which is called "domain mismatch" in this paper. Another reason is that the range of an attribute in a user query is different from that in a web service, which is referred as "range mismatch" in this paper.

In domain mismatch, the web service requires some attributes that are not specified in the user's query, or on the opposite, an attribute in the user's query is not part of the query schema for a web service. In the former case, the value of the required attribute by the web service can be defaulted, or alternatively, the query is run with each possible value of the required attribute. In the latter case, the attribute in the user's query must be ignored when generating the query to the web service. This will return a super set of the requested results. If the ignored attribute is part of the result schema in the web service, post processing can filter out the results that do not match the user's constraint. Default value can be annotated using "a:defaultValues" in RDFS.

One scenario for range mismatch is that web service requires enumerated values for an attribute, which can be annotated using "OWL:one of". To deal with the "range mismatch", the value of an attribute in a user's query should be mapped to the closest valid value for the web service so that the result from the web service is a superset of the result of the original user query. The results should be filtered based on the original user's query.

4.3 Generating Query to DB2 II

After a user's query is transformed to queries to the web services, the query generator in Figure 1 generates a DB2 II query on multiple web services. The query generation consists of three steps.

The first step is identifying all the related web services to a given user query. A web service is related if its output schema overlap the result schema of the user query, and its required attributes can be satisfied with the user's query.

The second step is to group the services which output schemas are consistent. We call two schemas are consistent if they are equivalent or one schema contains the other. The resulting schema of a service group is the intersection of the output schemas of all the services in the group. The results from the web services in a same service group are merged using the statement "UNION ALL".

The last step is to deal with the case that the output of one service group is complementary to that of another group. The query generator joins the results of those service groups.

4.4 Example of Transforming a User Query to a DB2 II Query

Assume DB2 II integrates three web services, Yahoo Autos (Yahoo), Autos MSN (MSN) and Kelly's Blue Books (KBB) and a user's query is as follows:

```
SELECT * from car
WHERE Make = Acura
and Model = CL
and Year < 8
and Price < 20000
and Price > 10000
and Mileage < 70000
and Location = 10598
```

We first create two virtual tables each of which is defined using a WITH statement. The first group includes KBB only and provides KBB Suggested Price that is not available from other service groups. The second group merges the results of Yahoo and MSN using the UNION ALL statement. The grey fields in the statement refer to the default values.

```
WITH cars_0 (year, kbb_price, car_type) AS
(SELECT KBB_CarYearIs, KBB_SuggestedPrice,
KBB_CarTypeIs
FROM KBB
WHERE KBB_CarType.Car_Make =
Acura,KBB_CarType.Car_Model = CL)

WITH cars_1 (year,price,mileage,car_type) AS
((SELECT Yahoo_CarYearIs, Yahoo_AskedPriceIs,
Yahoo_CarMileageIs, Yahoo_CarType
FROM Yahoo
WHERE Yahoo_Car_Make = Acura AND
Yahoo_Car_Model = CL AND
Yahoo_MileageLessThan = 70000 AND
Yahoo_MileageMoreThan = (0) AND
Yahoo_PriceRange.PriceLessThan =
```

20000,Yahoo_PriceRange.PriceMoreThan = 10000 **AND** Yahoo_SearchWithin = (50) **AND** Yahoo_UserPosition = 10598 **AND** Yahoo_YearLessThan = (2004) **AND** Yahoo_YearMoreThan = 1996)

UNION ALL

(**SELECT** MSN_YearIs, MSN_AskedPriceIs, MSN_MileageIs, MSN_CarTypeIs
FROM MSN
WHERE MSN_CarAgeLessThan = 8 **AND** MSN_CarCategory = PassengerCars **AND** MSN_CarType.Car_Make = Acura,MSN_CarType.Car_Model = CL **AND** MSN_MileageLessThan = 70000 **AND** MSN_PriceRange.PriceLessThan = 20000,MSN_PriceRange.PriceMoreThan = 10000 **AND** MSN_SearchWithin = (100) **AND** MSN_UserAt = 10598))

Finally, a SELECT statement joins the results from two virtual tables (service groups).

SELECT c0.year, c0.kbb_price, c0.car_type, c1.year, c1.price, c1.mileage, c1.car_type
FROM cars_0 c0, cars_1 c1
WHERE c0.year = c1.year **AND** c0.car_type = c1.car_type

5 CONCLUSION

We have proposed a novel information integration framework that uses web service as the wrapper to represent heterogeneous web information sources. Our framework is built upon industry standards such as WSDL/SOAP and Ontology languages RDFS and OWL, and leverages the service federation and the query optimization capabilities of IBM DB2 II. Using a used car searching service as the application scenario, we present a set of ontology-based transformation rules to deal with schema and content heterogeneity of web sources. Our future work is addressing scalability issues in our framework and methodologies.

REFERENCES

Akkiraju, R., Goodwin, R., Doshi, P., and Roeder, S., 2003. “A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI”. In the workshop Proc. of 18th IJCAI 2003. Information Integration on the Web, 87-92

Arens, Y., Knoblock, C. A., and Shen, W., 1996. “Query reformulation for dynamic information integration”. Journal of Intelligent Information Systems, 1996.

Burstein, M. H., Hobbs, J. R., Lassila, O., Martin, D., McDermott, D. V., McIlraith, S. A., Narayanan, S., Paolucci, M., Payne, T. R., Sycara, K. P., 2002. “DAML-S: Web Service Description for the Semantic Web”. In Proceedings of International Semantic Web Conference 2002: 348-363

Bussler, C., Fensel, D., and Maedche, A., 2002. A Conceptual Architecture for Semantic Wb Enaled Web Services. In ACM SIGMOD Record, Vol. 31, No. 4, December 2002.

http://www.daml.org/services/owl-s/

DB2 Information Integration. http://www-306.ibm.com/software/data/integration/.

Genesereth, M. R., Keller, A. M., and Duschka, O. M., 1997. “Infomaster: An information integration system”. In Proc. of SIGMOD, 1997.

Gio, W., 2000. "Future Needs in Integration of Information". In International Journal of Cooperative Systems, Vol. 9, No.4, World Scientific Publishing, November 2000, pages 449-772.

Haas, L. M., Kossmann, D., Wimmers, E. L., and Yang, J., 1997. “Optimizing Queries Across Diverse Data Sources”. VLDB (1997): pp 276-285

Subrahmanian, V. S., Adali, S., Brink, A., Lu, J., Rajput, A., Rogers, T. J., Ross, R., and Ward, C., 1995. “HERMES: A heterogeneous reasoning and mediator system”. Technical report, University of Maryland, 1995.

Levy, A. Y., Rajaraman, A., and Ordille, J., 1996. “Querying Heterogeneous Information Sources Using Source Descriptions”. In Proc. of VLDB, 1996.

Madhavan, J. and Halevy, A. Y., 2003. “Composing Mappings Among Data Sources”. In Proc. Of VLDB 2003, pages 572 - 583.

OWL Web Ontology Language Reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu/

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J., 1994. “The TSIMMIS Project: Integration of Heterogeneous Information Sources”. In Proceedings of 16th Meeting of the Information Processing Society of Japan, 1994.

Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajararnan, A., Sagiv, Y., Ullman, J., Vassalos, V., and Widom, J., 1996. “The TSIMMIS Approach to Mediation: Data Models and Languages”. Journal of Intelligent Information Systems, 8 (2), 1997, 117-132, March - April.

UDDI Technical Committee. “Universal Description, Discovery and Integration (UDDI)”. http://www.oasis-open.org/committees/uddi-spec/

Web Services Activity. http://www.w3c.org/2002/ws/.