# Composing and Optimizing Data Providing Web Services

| Mahmoud Barhamgi | Djamal bensimane | Aris M. Ouksel |
|---|---|---|
| LIRIS Laboratory | LIRIS Laboratory | The University of Illinois |
| Claude Bernard University | Claude Bernard University | Dept. Of Computer Science |
| 69622 Villeurbanne, France | 69622 Villeurbanne, France | Chicago, USA |
| barhamgi@liris.cnrs.fr | Djamal.benslimane@liris.cnrs.fr | aris@uic.edu |

## ABSTRACT

In this paper, we propose a new approach to automatically compose data providing Web services. Our approach exploits existing mature works done in data integration systems. Specifically, data providing services are modeled as RDF Parameterized views over mediated ontologies. Then, an RDF oriented algorithm to compose services based on query rewriting techniques was devised. We apply also an optimization algorithm on the generated composition to speed up its execution. The results of our experiments show that the algorithms scale up very well up to a large number of services, covering thus most realistic applications.

## Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services – Web-based services.

## General Terms: Algorithms, Experimentation, Management, Performance.

**Keywords:** Web service composition, optimization, RDF.

## 1. INTRODUCTION AND MOTIVATIONS

In many of today's collaborative environments the access to an enormous number of data sources has been made in the form of Data Providing Web Services, where services correspond to a limited set of calls over the sources' schema. The typical reasons for providing limited access methods are: (i) privacy and security constraints [4]; (ii) the underlying information systems are proprietary and only offer such limited access methods [6]; (iii) or because of performance concerns of the database servers-administrators control the query plans executed against their servers by providing a set of data services instead of providing a full query access to their systems. Data providing services (hereafter referred to as DP services) are different from the traditional web services - which are basically Effect Providing services (EP services) - in that their invocation only returns data but it does not have effects that may change the state of the world. On the other hand, an EP service implements processes, often with accompanying effects as in, for example, a flight reservation service or a billing service. Many application domains like Bioinformatics and healthcare have widely embraced DP services for data sharing. Composing DP services in these domains opens up the door for building interesting applications like medical record sharing and conducting *in silico* experiments on top of the World Wide Web.
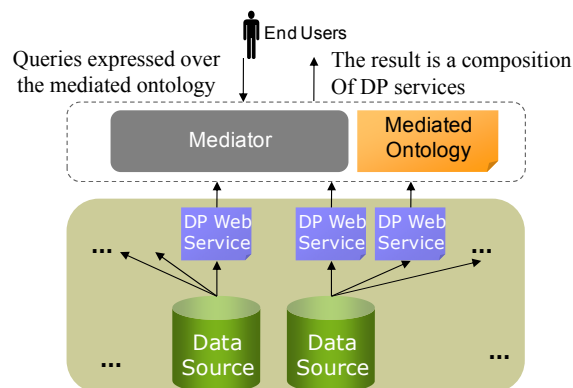
**Figure 1: A data integration approach to compose Data Providing Web services.**

The problem we are interested in this paper is the following. Suppose we have a set of DP services $S : S_1, S_2, ...., S_n$ exported by autonomous information systems in the collaboration environment (figure 1), and a user query *Q* expressed over a mediated ontology - a query expressed in SPARQL notation over RDFS ontologies. The problem is then how to answer a query *Q* by means of composition of DP services?. Web service composition has been extensively studied, many approaches were proposed [1,5]. However, these approaches cannot be applied to DP services since they cannot take into consideration the semantic relationship that holds between input and output sets of a DP service, which is considered as the determinative facto in DP service selection and composition. Rather, view-based data integration approaches in the database literature may fit best DP services composition since DP services can be regarded as conventional data sources but with access restrictions.

## 2. OUR APPROACH TO COMPOSE DP SERVICES

We propose to use a data integration approach for the purpose of DP service composition. Specifically, our approach revolves around the following points:

I. We have modeled DP services as *RDF parameterized views* over domain ontologies. RDF parameterized views allow capturing the semantic relationships between input and output sets of a DP service based on concepts and properties of the underlying ontologies. An RDF view has the form:

$$S_i(\overline{X_i}, \overline{Y_i}): \quad <\Phi(\overline{X_i}, \overline{Y_i}, \overline{Z_i}), Ct_i>, \text{ where}$$

$\overline{X_i}$ is the set of variables denoting the input parameters necessary to invoke $S_i$. $\overline{Y_i}$ is the set of variables denoting the returned literals from invoking $S_i$. $\Phi(\overline{X_i}, \overline{Y_i}, \overline{Z_i})$ is the semantic relationship holding between input and output variables. $\overline{Z_i}$ is the existential variables set relating $\overline{X_i}$ and $\overline{Y_i}$. $Ct_i$ is the constraints set that is imposed on the distinguished variables $\overline{X_i}$ or $\overline{Y_i}$. $\Phi(\overline{X_i}, \overline{Y_i}, \overline{Z_i})$ is represented in the form of RDF triples. Views have the form of SPARQL queries.

Note that this kind of semantic description can be accommodated easily in the service description files (WSDL[1] or OWL-S) as a simple annotation, and thus it fits well with Web service standard stack.

II. We have devised an efficient RDF query rewriting algorithm for the purpose of composing DP services. The algorithm takes into consideration the semantic constraints of RDFS ontologies (i.e. *subclass*, *subproperty*, *domain* and *range* constraints) while composing services. The algorithm can handle both *parameterized queries* and non-parameterized queries. A parameterized query is a class of queries that can be instantiated by specifying values for the input parameters set. A query can be parameterized over one or several input variables, and when parameterized over a variable, no specific values are specified in the query for the parameter, or may be specified as ranges of values. One implication to handling parameterized queries is that constituent DP services cannot be chosen precisely at the composition time due to the presence of equality and order constraints. Rather, their selection is dependent on the actual values of input parameters while executing the obtained composition - actual values are specified by users while invoking the composite service and are not known at the composition time.

III. We have devised an algorithm to optimize the composite services to speed up their execution. This is done by inserting a service selection mechanism during the execution based on the actual values of input parameters. This mechanism exploits order and equality constraints that are imposed on the service's input parameters to decide whether it should be invoked.

IV. Our approach features the use of an efficient probabilistic algorithm [3] to decide the minimum number of constituent services that satisfy equality and order constraints specified in a parameterized query.

## 3. IMPLEMENTATION AND EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our approach, we have implemented a Web Service Management System (WSMS) that has DBMS-like capabilities that enables clients to query and compose services in a transparent and integrated fashion. Our system has facilities that (i) allow service providers to describe their DP services as RDF views over domain ontologies; (ii) allow users to interactively initiate SPARQL queries over mediated ontologies. The system implements our algorithms to compose

services and to optimize composite services. Composite services can be hosted on the system for multiple invocations with different specific values of input parameters.

We have conducted extensive experiments to evaluate the performance of our RDF query rewriting algorithm. As in the conventional data integration system, we have considered two types of queries: Chain and Star queries. The algorithm was able to handle /300/ services in less than /4/ seconds in the chain scenario, and the same number of services was handled in less than /1/ seconds in the star scenario. These results are very encouraging, indeed our rewriting algorithm scales up very well up to a large number of services, covering thus most realistic applications.

## 4. RELATED AND FUTURE WORK

Our work is closely related to research in web service composition and data integration areas. Our approach to compose DP services is different from previous conventional service composition approaches [1,5] in that it takes into account the semantic relationship holding between input and output sets of a service- such relation requires a the definition of a declarative view over an ontology to be captured.

Our work differs from data integration systems [2] in that while these systems focus on resolving specific queries given a set of incomplete data sources, we focus on constructing a composition of services that is independent of specific input values - which necessitates the use of an optimization mechanism to filter out irrelevant services at the execution time. Furthermore, resolving a query in terms of Web services requires a special attention since a rewriting may not be executable if some input parameters of constituent services are unbound.

As a future work, we intend to study the problem of discovering and composing services in P2P settings where every peer uses its ontology to describe its own DP services. Ontology mappings among peers should be considered in such case. We are also interested in handling data privacy issues when composing services from multiples service providers.

## 5. REFERENCES

[1] S. Dustdar and W. Schreiner. A survey on web services composition. International Journal of Web and Grid Services, 1(1):1-30, 2005.

[2] A. Y. Halevy and A. Rajaraman. Data integration: The teenage years. In VLDB, pages 9-16, 2006.

[3] A. M. Ouksel, O. Jurca, I. Podnar, and K. Aberer. Effcient probabilistic subsumption checking for content-based publish/subscribe systems. In Middleware, pages 121-140, 2006.

[4] S. Rizvi, A. O. Mendelzon, and S. Sudarshan. Extending query rewriting techniques for _ne-grained access control. In SIGMOD Conference, pages 551-562, 2004.

[5] D. Wu, B. Parsia, and E. Sirin. Automating daml-s web services composition using shop2. In International Semantic Web Conference, pages 195-210, 2003.

[6] R. Yerneni, C. Li, H. Garcia-Molina, and J. D. Ullman. Computing capabilities of mediators. In SIGMOD Conference, pages 443-454, 1999.

---

[1] http://www.w3.org/TR/wsdl20/