The Journal of Systems and Software 83 (2010) 85–95

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

An intelligent query processing for distributed ontologies

Jihyun Lee [a], Jeong-Hoon Park [a], Myung-Jae Park [a], Chin-Wan Chung [a,*], Jun-Ki Min [b]

[a] *Division of Computer Science, Department of Electronic Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejon 305-701, Republic of Korea*

[b] *School of Internet-Media Engineering, Korea University of Technology and Education, Byeongcheon-Myeon, Cheonan, Chungnam 330-708, Republic of Korea*

A R T I C L E I N F O

Article history:
Received 30 October 2008
Received in revised form 30 March 2009
Accepted 7 June 2009
Available online 12 June 2009

Keywords:
Distributed query processing
Semantic mapping
Query optimization
Distributed ontologies
Semantic Web

A B S T R A C T

In this paper, we propose an intelligent distributed query processing method considering the characteristics of a distributed ontology environment. We suggest more general models of the distributed ontology query and the semantic mapping among distributed ontologies compared with the previous works. Our approach rewrites a distributed ontology query into multiple distributed ontology queries using the semantic mapping, and we can obtain the integrated answer through the execution of these queries. Furthermore, we propose a distributed ontology query processing algorithm with several query optimization techniques: pruning rules to remove unnecessary queries, a cost model considering site load balancing and caching, and a heuristic strategy for scheduling plans to be executed at a local site. Finally, experimental results show that our optimization techniques are effective to reduce the response time.

1. Introduction

In the Semantic Web, the definitions of resources and the relationship between resources are described by an ontology in order to automatically interpret the resources and retrieve useful information. The resources in the Web are independently generated in many locations. Thus, even if the ontologies describe resources in the same (similar) domain, they can use different representations (i.e., language and schema). Also, the ontologies are managed by various local ontology management systems which have different capabilities and strategies for storing and query processing. Under these environment, some Web applications want to access the ontologies without regard to the heterogeneity and the dispersion of the ontologies and the local systems. In order to support such a request, an efficient query processing over the distributed ontologies is essential. Of course, existing distributed query processing techniques can be applied to query the distributed ontologies. However, they confront the limitations of the efficiency and the functionality since some important characteristics of a distributed ontology environment are not considered.

Fig. 1 shows an example of the distributed ontology environment. There are three kinds of ontologies, **UNIV**, **COLLEGE**, and **PUB** which are managed in three different sites and two types of local systems (i.e., LS_1, LS_2). **UNIV** and **COLLEGE** describe the information of the university and the college, respectively, and **PUB** describes the publication information. For the simplicity, we describe only the schema and omit the instance part. These ontologies are independently generated but related to each other even if they have different schemas. For example, let us suppose the following conditions: first, the concept of *Professor* in **UNIV** is defined as the concept of *Lecturer* in **COLLEGE**. Second, the information of the authors in **PUB** can be found in **UNIV** and **COLLEGE**. In this distributed ontology environment, consider the following example queries:

Example 1. Q_1: Find professors who teach 'Algorithm'.

Example 2. Q_2: Find authors who wrote publications about 'Semantic Web' and also retrieve the name and the email addresses of the authors.

In order to find the answer of query Q_1, we should retrieve professors and lecturers who teach 'Algorithm' from **UNIV** and **COLLEGE**, respectively. For query Q_2, **UNIV** and **COLLEGE** should be searched along with PUB to find the personal information of the authors who wrote papers about 'Semantic Web'. For such a query, in order to efficiently find the answer dispersed in several ontologies and local sites, a distributed query processing method considering the heterogeneity of the ontologies is required.

The use of the semantic mapping is a representative approach to deal with the heterogeneity among different ontologies (Borgida and Serafini, 2003; Haase and Motik, 2005; Motik et al., 2004; Serafini and Andrei, 2005). In (Borgida and Serafini, 2003; Serafini and Andrei, 2005), the semantic mapping is the semantic relationship (i.e., subsumption or equivalence) between concepts (i.e., classes or properties) in two different ontologies and it has

* Corresponding author. Tel.: +82 42 350 3537; fax: +82 42 350 3510.
E-mail addresses: hyunlee@islab.kaist.ac.kr (J. Lee), jhpark@islab.kaist.ac.kr (J.-H. Park), jpark@islab.kaist.ac.kr (M.-J. Park), chungcw@islab.kaist.ac.kr (C.-W. Chung), jkmin@kut.ac.kr (J.-K. Min).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.06.008

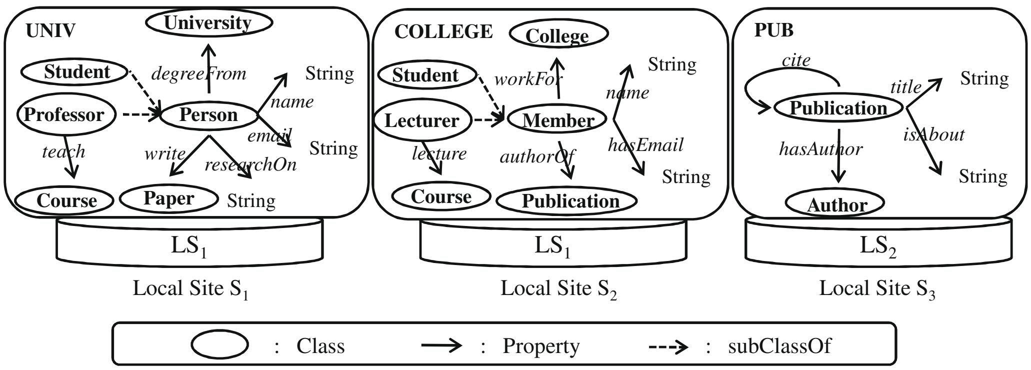

Fig. 1. An example of the distributed ontology environment.

been extended to that between views (i.e., queries) (Haase and Motik, 2005; Motik et al., 2004). However, the previous works do not support more general semantic mapping and distributed query covering more than two ontologies. Besides, most of them have focused on only the rewriting of the query using the semantic mapping, and do not make an issue of the efficient distributed query evaluation (i.e., query rewriting, scheduling, and execution).

In this paper, we resolve issues of the distributed query processing over multiple heterogeneous ontologies. We extend the models of the distributed query and the semantic mapping to support more general distributed ontology query answering compared with previous works. Furthermore, we present a distributed ontology query processing algorithm with several query optimization techniques considering the characteristics of the distributed ontology environment.

The contributions of the paper are as follows:

Extended models of the distributed ontology query and the semantic mapping: We present a general distributed ontology query model to cover multiple different ontologies. We also present a general semantic mapping model in which more than two ontologies can be associated. The extension of query and semantic mapping models makes it possible to include relevant data which could not be accessed before in the query result. Also, our approach logically integrates independently grown distributed ontologies through the query rewriting based on the semantic mapping. As a result, we can efficiently extract an integrated answer of a distributed query over different ontologies.

Optimization techniques for an efficient query processing on the distributed ontologies: Multiple distributed queries are generated from an original distributed query to obtain results from dispersed ontologies. In order to remove unnecessary operations and to increase the parallelism among executions of the multiple queries, we suggest several optimization techniques. First, we present pruning rules to remove invalid and redundant queries. Second, we suggest a heuristic strategy for scheduling plans to be executed at a local site. Third, we propose a cost model considering site load balancing and caching for processing multiple distributed queries.

The remainder of the paper is organized as follows: In Section 2, we review related work. In Section 3, we present a distributed ontology query model and a semantic mapping model. Section 4 describes a distributed query processing technique with several query optimization techniques over distributed ontologies. Section 5 contains the results of experiments. Finally, in Section 6, we conclude this paper.

2. Related work

Recently, the research on a query processing over distributed ontologies has been performed. Stuckenschmidt et al. (2005) suggests a global data summary for locating data matching query answers in different sources and the query optimization. However, Stuckenschmidt et al. (2005) assumes that all distributed ontologies can be accessed in a uniform way like a global schema. In other words, the heterogeneity of schemas of the distributed ontologies is not considered. Besides, many tasks are concentrated on the mediator. As well as query scheduling, the merge (i.e., join) of all local query results is also executed in the mediator. Thus, when the mediator receives requests for many queries at the same time, the bottleneck on the mediator is inevitable.

The most of research on the query answering over distributed ontologies are based on the P2P architecture. Edutella (Nejdl et al., 2002) uses an unstructured P2P network which has no method to route a query to the relevant ontologies. Instead, the query is broadcasted in the entire network. Thus, a huge amount of unnecessary network traffic incurs. As a successor of Edutella, to provide better scalability, Nejdl et al. (2003) presents a schema-based query routing strategy in a hierarchical topology using the super-peer concept. Nejdl et al. (2003) also suggests a rule-based mediation between two different schemas in order to collect results from many peers using heterogeneous schemas. SomeRDFs (Adjiman et al., 2007) supports the semantic mapping between two atomic concepts and between the domain (or range) of a property and a class. Piazza (Halevy et al., 2003) proposes a language (heavily relies on XQuery/XPath) to describe the semantic mapping between two different ontologies. In these works, for distributed query answering, a peer reformulates a query by using the semantic mapping and forwards the reformulated query to another peer related by the semantic mapping.

DRAGO (Serafini and Andrei, 2005) focuses on a distributed reasoning based on the P2P-like architecture. In DRAGO, every peer maintains a set of ontologies and the semantic mapping between its local ontologies and remote ontologies located in other peers. A reasoning service is performed by a local reasoner for the locally registered ontologies and the reasoning is propagated to the other peers when the local ontologies are semantically connected to the other remote ontologies. The semantic mapping supported in DRAGO is only the subsumption relationship between two atomic concepts. Besides, it does not support the ABox reasoning.

KAONP2P (Haase and Wang, 2007) also suggests the P2P-like architecture for query answering over distributed ontologies. KAONP2P supports more extended semantic mapping which describes the correspondence between views of two different ontologies, where each view is represented by a conjunctive query. For the distributed query answering, it generates a virtual ontology including a target ontology to which the query is issued and the semantic mapping between the target and the other ontologies. Then, the query evaluation is performed against the virtual ontology.

OBSERVER (Mena et al., 2000a) does not consider the P2P environment. Non the less, the goal is also to find an answer of an