Journal of King Saud University – Computer and Information Sciences (2016) **28**, 125–132

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Reformulating XQuery queries using GLAV mapping and complex unification

Saber Benharzallah *, **Hammadi Bennoui, Okba Kazar**

Smart Computer Science Laboratory, Biskra University 07000, Algeria

Received 16 March 2014; revised 18 December 2014; accepted 3 June 2015
Available online 21 October 2015

KEYWORDS

Data integration;
Mediator;
XML;
XQuery;
GLAV mapping

Abstract This paper describes an algorithm for reformulation of XQuery queries. The mediation is based on an essential component called mediator. Its main role is to reformulate a user query, written in terms of global schema, into queries written in terms of source schemas. Our algorithm is based on the principle of logical equivalence, simple and complex unification, to obtain a better reformulation. It takes XQuery query, global schema (written in XMLSchema), and mappings GLAV as input parameters and provides resultant query written in terms of source schemas. The results of implementation show the proper functioning of the algorithm.

1. Introduction

Now the Web is presented as the most favored means to disseminate information. Many companies and organizations, whatever their field of activity (e-commerce, education, geographical or historical applications, etc.), make this choice for disseminating information.

The diversity of distributed information sources and their heterogeneity are one of the main difficulties encountered by users of the Web. It requires the user to respect the access methodology for each data source, implying to know the location of the base, the description of their content, the possibilities of interrogation and the format of results, in order to receive the expected response (Hacid and Reynaud, 1998).

The mediator-based systems offer interesting solutions for the integration of heterogeneous data. Accordingly, most recent works have taken this approach including the Internet-Oriented Systems (Moussa, 2002; Elazami et al., 2007; Mustafa and Rahman, 2013). The mediator acts as an interface between users and data sources. It is composed of a global schema, which provides a unified view of data sources and a set of views describing the content of sources. Queries are then expressed on the global schema, giving users the illusion of querying a single database. Based on information provided by the views, the mediator analyzes and reformulates the queries into sub-queries that would be executed by data sources. Before being sent to the target data source, each sub-query is translated into the native language of the source by the corresponding wrapper. The mediator uses the schema mappings to reformulate queries. Schema mappings establish a correspondence between data stored in two databases, called source and target respectively. Query processing under schema mappings has been investigated extensively in the two cases

* Corresponding author.
E-mail addresses: sbharz@yahoo.fr (S. Benharzallah), bennoui@gmail.com (H. Bennoui), kazarokba@yahoo.fr (O. Kazar).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2015.06.001

where each target atom is mapped to a query over the source (called GAV, Global-As-View), and where each source atom is mapped to a query over the target (called LAV, Local-As-View). The general case, called GLAV, in which queries over the source are mapped to queries over the target, has recently attracted a lot of attention (Calvanese et al., 2012). The mediator approach has the advantage of being able to build a query data sources system without touching the data remaining in their original sources.

XML is an extremely versatile markup language, capable of labeling the information content of diverse data sources including structured and semi-structured documents, relational databases, and object repositories (XQuery 1.0, 2007). A query language which uses the structure of the XML can intelligently express queries in all types of data that are physically stored in XML or viewed as XML via middleware. Because the query languages were traditionally designed for specific data types, the majority of existing proposals for the XML query languages are robust for some types of data sources, but weak for others. The specification of XQuery (XQuery 1.0, 2007) describes a new query language, which is conceived to be largely applicable to all types of XML data sources.

Most query reformulation algorithms (Koch, 2002; Arenas et al., 2004; Libkin and Sirangelo, 2008) using GLAV mapping approach exploit conjunctive queries; and consequently are not expressive or are applied in the fields of data exchange.

In this paper we describe a reformulation algorithm of XQuery queries for mediator based systems. The main role of the mediator is to reformulate a user query, written in terms of global schema, into queries written in terms of source schemas. Our algorithm is based on the principle of logical equivalence and simple/complex unification to obtain a better reformulation. The algorithm avoids the shorts reformulations. It takes XQuery query, global schema (written in XMLSchema), and expressive mappings GLAV (written in XQuery language) and provides resultant query written in terms of source schemas. The results of implementation show the proper functioning of the algorithm.

The rest of this paper is organized as follows: Section 2 presents the related work, some solutions presented in the literature and the characteristics of our solution. Section 3 outlines some concepts used in this paper. Section 4 presents the proposed architecture of our system of mediation and describes the reformulation algorithm. The programming environment and implementation are presented in Section 5. Finally, Section 6 concludes and prospects the paper.

2. Related work

The two main problems posed by the construction of a mediator are (Rousset et al., 2002): (i) the choice of both the language used to model the global schema, and the languages used to model, according to this schema, the views on the sources to be integrated as well as queries of users. And, (ii) the choice of query reformulation algorithm in terms of views in order to get all the answers to a query.

Studies have focused on the languages for modeling the global schema to represent the views of the sources to integrate and those used to express queries from human users or computing entities (Reynaud and Safar, 2008; Goasdoue et al., 2000). Others have focused on the design and implementation of algorithms for query rewriting in terms of views on relevant data sources and, more recently, some research focus on designing intelligent interfaces assisting the user in Query formulation (Maiz et al., 2006; Charlet et al., 2003).

Most of research (Calvanese et al., 2012; Halevy et al., 2006) on query processing under schema mappings in data integration distinguish three approaches to establish mappings between the global schema and source schemas. In GAV mappings, for each relationship used in the global schema, we define a view written using source schemas. The main advantage of this approach is its simplicity of reformulation. Nevertheless, it lacks the flexibility with respect to the addition, deletion and modification of the sources to the data integration system. This is due to the fact that each modification of a local source schema results in a modification of the global one. The projects TSIMMIS (Garcia-Molina et al., 1997), INFORMIX (Leone et al., 2005) follow the GAV approach.

In LAV mappings, every relationship of a source schema is defined as a view on the global schema. In this approach, each source is independently specified, which permits to provide more flexibility with respect to the addition/deletion of data sources to integrate. It has no effect on the global schema, only views should be added (or deleted). On the other hand, the price to pay for this flexibility is the complexity of the construction of answers to a query in the designed mediator. The projects STYX (Amann et al., 2002), Agora (Manolescu et al., 2001), follow the LAV approach.

GLAV mappings (Reynaud and Safar, 2008; Djema et al., 2007) overcome the limitations of both GAV and LAV (Friedman et al., 1999). In the query reformulation of the GLAV approach, each mapping rule is represented by a conjunctive query written in the global schema associated with a conjunctive one written in source schemas. These queries are virtual views that do not represent the results stored on sources, rather than LAV approach where each source may be regarded as it contains a response to a query written in the global schema; and consequently, the sources represent materialized answers to written queries on the global schema. Thus, in GLAV approach, the rules allow to reformulate the query more efficiently. Additionally, it reaches the limits of the expressive power of a data source description language. And also the query reformulation is a co-NP-hard in the size of the data in the sources. Query reformulation in this approach is shown to be no harder than that of the LAV approach. In fact, most of the research on query processing under schema mappings in data integration concentrate on GAV and LAV mappings (see, for instance the surveys in Calvanese et al. (2012) and Halevy et al. (2006)).

In data integration, GLAV mappings were specifically taken into account in Cal (2004), but only in the case of relational databases. It was mainly studied in the exchange of information. In particular, the focus of Friedman et al. (1999) and Levy et al. (2000) is put on providing foundation for exchange of information based on schema mappings; whereas in Florescu (1996), Arenas et al. (2010) and Fagin et al. (2009), the goal is to study operators on schema mappings relevant to model management, notably, composition, merge, and inverse (Calvanese et al., 2012). A more general form of GLAV that accounts for XML like structures, and which we will use here, has been used to give semantics for mappings between XML schemas and to generate the data transformation scripts (in SQL, XQuery or XSLT) that implement the desired data exchange (Libkin and Sirangelo, 2008; Yu and Popa, 2004).