Expert Systems with Applications 40 (2013) 2592–2606

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

# A formal framework for query decomposition and knowledge integration in data warehouse federations

Rafał Kern, Tomasz Stolarczyk, Ngoc Thanh Nguyen *

*Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland*

ARTICLE INFO

*Keywords:*
Warehouse federation
Schema integration
SQL query decomposition
Knowledge integration

ABSTRACT

A federation of data warehouses is understood as a set of data warehouses, which can be processed as a whole in the logic level. Physically, the federation does not gather data into one place. This paper presents a formal framework for data and knowledge processing in data warehouse federations. The management system for a data warehouse federation consists of an user interface enabling presentation of user queries, a program for query decomposition and a program for integrating knowledge coming from different data warehouses as the answers to a user query. We propose a model for query decomposition process and knowledge integration. It contains also the algorithm for knowledge inconstancy processing. This kind of inconsistency often occurs since very often the knowledge extracted from different data warehouses refers to the same subject, but is not consistent.

## 1. Introduction

Data warehouses are highly specialized databases which often treat the world in a narrow scope. They are also very important sources of knowledge needed in many fields of life. However, in many cases, one data warehouse is not sufficient to give a general point of view and it is necessary to gain some additional data sources and make them cooperate with existing ones. As Jindal and Acharya (2003) and Waddington (2004) have proved, the main reasons of forming federations are their flexibility and ability to decrease the costs of such an operation. Instead of making one centralized system it is easier and faster to create a loosely coupled network of existing data warehouses.

According to Kern, Ryk, and Nguyen (2011), a federation of warehouses we describe as a system consisting of several warehouses and a federation management system, which contains the following elements:

- An integration procedure of the schemas of the component warehouses giving the logical schema of the federation.
- A query language for user who does not need to know the schemas of the component warehouses.
- A procedure which enables decomposition of user queries to the federation into sub-queries which are sent to the component warehouses.
- A procedure integrating the answers sent from the component warehouses to federation layer.

Data warehouse federations have some similarities with distributed databases. However, in data warehouse federations the rules for data organization are not so strict. In federations more than one data warehouse may refer to the same subjects so during answers integration some conflicts may appear. A data warehouse federation must be invulnerable for inconsistency therefore it needs some tool to solve them (Akinde, 2003).

A federation structure seems to be a good solution for the mentioned problem. However, usage of heterogeneous components causes some additional problems. It is necessary to decompose the initial query into sub-queries which are processed by components. Then the federation layer must know how to integrate the partial data received from component data warehouses to present the final answer to the end user. The key part of this layer is a global schema which provides a common view on all component schemas. It works with a mapping table which contains mappings between attributes or measures in the federation schema and their equivalents in local schemas of each component data warehouse. These two elements are used during query decomposition when system must decide where the sub-query should be sent. On the other hand, it is very helpful during answer integration because of metadata it contains. The answers may contain inconsistencies, e.g. different time measures, currencies, weight units etc. All of them must be solved during answers integration because otherwise the federation will provide false data. All such dependencies must be included in the global schema. For the inconsistency processing one can use collective intelligence methods, since it is assumed that

* Corresponding author. Tel.: +48 713204139.
*E-mail addresses:* Rafal.Kern@pwr.wroc.pl (R. Kern), stolarczykt@gmail.com (T. Stolarczyk), Ngoc-Thanh.Nguyen@pwr.edu.pl (N.T. Nguyen).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.10.060

different autonomous sources may have inconsistent knowledge referring to the same object. Processing inconsistencies in this case may bring new knowledge not included in these sources.

In this paper we present a formal framework for query decomposition and a procedure for component's answers integration. To the best knowledge of the authors, up to know, there have not been worked out formal methods for this aim. The original contribution should be helpful in building federations of data warehouses as multidimensional and collective sources of knowledge. The proposed method should also be useful in processing big data.

The remaining part of this paper is organized as follows: Section 2 contains an analysis of related works. Section 3 includes the definitions of data warehouse federation and its notions. A procedure for query decomposition is presented in Section 4. Section 5 presents a answers integration algorithm. Some conclusion and future work purposes are included in Section 6.

## 2. Related works

Data warehouse federations are a very promising solution for many problems in area of business decision supports. In our previous work (Kern et al., 2011) some formal basics in the area of data warehouse federations have been introduced. Authors have moved the idea of federation proposed by Sheth and Larson (1990) to the area of data warehouses. The federation has to fulfill the conditions of autonomy, heterogeneity and dispersion. Each component may be maintained and developed independently. Its regular performance cannot be strongly affected by participation in a federation. Also no assumption about components technology or design can be made.

The data heterogeneity may appear on schema level, when the same subjects are described by different schemas or the instance level where different sources describe the same objects from real world (Dong & Naumann, 2009). The federation should be able to work with totally different designed and implemented data warehouse systems.

A federation has a layered structure with two main layers: federation and component. The federation layer contains the global federation schema and constitutes a transparent interface for external applications hiding components heterogeneity (Akinde, 2003; Berger & Schrefl, 2008; Kimball & Wiley, 1996; Schneider, 2006; Sheth & Larson, 1990). The global federation schema is expressed in common canonical dimension model introduced by Sheth and Larson (1990). Another important part of this layer is the mapping dictionary which contains mappings between dimensions and attributes presented to a user and some real attributes stored in physical component data warehouses. The federation layer may also contain some additional dictionaries used during optimization (Berger & Schrefl, 2008). Some additional data may be obtained during ETL (Extract, Transform and Load) processes (Jindal & Acharya, 2003). It also processes the export schemas obtained from each component. The component layer is a structure built on the basis of native, local data warehouse schemas. A component data warehouse is extended by the component schema and export schema. The component schema is in fact the local schema described by expressions using the common dimensional model. This model makes the federation more stable than this one proposed by Heimbigner and McLeod (1985), where each component does not have to store unique communication schemas with all other components.

A data warehouse federation works only on logical level and query processing processes are performed in its components. That is why the initial user query must be decomposed (Akinde, 2003; Evrendilek & Dogac, 1995; Monjuru, Henskens, & Hannaford, 2009; Sharma, Goswami, & Gupta, 2009; Stolba, Tjoa, Mueck, & Banek, 2007). The decomposed query must be processed and translated into understandable form for component data warehouse systems and executed by them. Query decomposition is more dependent on programming language processing, than on specific types of DB systems. This is because of a high abstraction of the logical level of federated data warehouses. A query decomposition process is based on determining sub-queries which are executable in specific components of federated data warehouses. After that, the sub-query results must be sent to the federation layer which performs the integration of them. As it was mentioned in Section 1 the federation needs some tools to provide internal data consistency and solve conflicts.

As Dong and Naumann (2009) have shown, the conflicts may appear because of incomplete data, erroneous data, and out-of-date data. Luckily, in data warehouses the last reason should not appear very often because no data is out of date. It is only a matter of properly used time dimension. Moreover, in most cases in data warehouses we work with are numerical. So the set of possible conflict types is smaller than in distributed databases. The conflicted data usually may be converted into one common system (Fan, Hongjun, Mdnick, & Chueng, 2001). The integration procedures need to know in what context the data is used in each component. The federation stores thus knowledge in its dictionary generated during schema integration and ETL process (Jindal & Acharya, 2003). In conflicts resolution phase very useful may be weights directly or indirectly provided by the user (expert) (Motro & Anokhin, 2006).

In integration process one needs to set the conditions which must be satisfied. In Konieczny and Pino-Pérez (1998) several approaches have been introduced. The first one (arbitral) tries to minimize individual dissatisfaction which is based on using distances between components and the federations answers. The second one (majority) minimizes the global dissatisfaction calculated as a sum of distances between answers from each component and federation. Because the main goal of building a federation is to gain different autonomous sources of knowledge, the majority approach seems to be a better solution. It is very difficult to create one universal integration procedure because each type of data must be handled in different way. It is much easier to provide a set of merging functions. Thus, we gain a more flexible and reliable tool. Many different data sets may also have very different quality understood mostly as its representativeness. The representativeness or generally – integrity constraints needs to be taken into consideration to provide high performance quality of the database system. Usually they should be based on some distance functions (Condotta, Kaci, & Schwind, 2008; Wallgrün & Dylla, 2010). The integration should be performed in qualitative level (Wallgrün & Dylla, 2010) and take into consideration possible user-defined weights (Motro & Anokhin, 2006).

In our approach we propose a formal framework for the federation layer. For this aim we define formally the notion of data warehouse federation. Next we build a general schema for the interaction process between users and the federation. On its basis we work out a method for integrating component data warehouse schemas for creating the logical schema of the federation. We assume that a user sees only this schema and defines a query on its basis. We work out procedures for decomposing SQL user queries to sub-queries understandable for component data warehouses and next define the algorithms for integrating the answers generated by them to determine the final answer for the user.

## 3. Basic definitions

### 3.1. Architecture overview

The general structure of federation is presented in Fig. 1, which has been done on the basis of five-level architecture proposed by Sheth and Larson (1990).