Information Systems 44 (2014) 94–119

Contents lists available at ScienceDirect

Information Systems

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

A requirement-driven approach to the design and evolution of data warehouses

Petar Jovanovic [a,*], Oscar Romero [a], Alkis Simitsis [b], Alberto Abelló [a], Daria Mayorova [a]

[a] Universitat Politècnica de Catalunya, BarcelonaTech, Barcelona, Spain
[b] HP Labs, Palo Alto, CA, USA

ARTICLE INFO

Available online 24 January 2014

Keywords:
DW
ETL
Multidimensional design
Conceptual design

ABSTRACT

Designing data warehouse (DW) systems in highly dynamic enterprise environments is not an easy task. At each moment, the multidimensional (MD) schema needs to satisfy the set of information requirements posed by the business users. At the same time, the diversity and heterogeneity of the data sources need to be considered in order to properly retrieve needed data. Frequent arrival of new business needs requires that the system is adaptable to changes. To cope with such an inevitable complexity (both at the beginning of the design process and when potential evolution events occur), in this paper we present a semi-automatic method called *ORE*, for creating DW designs in an iterative fashion based on a given set of information requirements. Requirements are first considered separately. For each requirement, *ORE* expects the set of possible MD interpretations of the source data needed for that requirement (in a form similar to an MD schema). Incrementally, *ORE* builds the unified MD schema that satisfies the entire set of requirements and meet some predefined quality objectives. We have implemented *ORE* and performed a number of experiments to study our approach. We have also conducted a limited-scale case study to investigate its usefulness to designers.

1. Introduction

Data warehousing ecosystems have been widely recognized to successfully support strategic decision making in complex business environments. One of their most important goals is to capture the relevant organization's data provided through different sources and in various formats with the purpose of enabling analytical processing of such data. The most common design approach suggests building a centralized decision support repository (like a DW) that gathers the organization's data and which, due to its analytical nature, follows a multidimensional (MD) design. The MD design is distinguished by the fact/dimension dichotomy, where facts represent that the subjects of analysis and dimensions show different perspectives from which the subjects can be analyzed (e.g., we can analyze *shopping orders* for *customers* and/or *suppliers*). Furthermore, the design of the extract-transform-load (ETL) processes responsible for managing the data flow from the sources towards the DW constructs, must also be considered.

Complex business plans and dynamic, evolving enterprise environments often result in a continuous flow of new information requirements that may further require new analytical perspectives or new data to be analyzed. Due to the dynamic nature of the DW ecosystem, building the complete DW design at once is not practical. Also, assuming that all information and business requirements are available from the beginning and remain intact is not realistic either. At the same time, for constructing a DW design (i.e., its MD schema) the heterogeneity and relations among existing data sources need to be considered as well.

The complexity of the monolithic approach for building a DW satisfying all information requirements has also been

[*] Corresponding author.
E-mail addresses: petar@essi.upc.edu (P. Jovanovic), oromero@essi.upc.edu (O. Romero), alkis@hp.com (A. Simitsis), aabello@essi.upc.edu (A. Abelló), mayorova@essi.upc.edu (D. Mayorova).

0306-4379/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2014.01.004

Table 1
The DW Bus Architecture for IR1–R5.

	Customer	Supplier	Nation	Region	Orders	Part	Partsupp	Lineitem_dim
ship. qty.(IR1)	√	√	√		√		√	
profit(IR2)		√	√					√
revenue(IR3)		√	√	√		√	√	
avil. stock val.(IR4)		√	√			√		
ship. prior.(IR5)	√				√			√

largely characterized in the literature as a stumbling stone in DW projects (e.g., see [1]). As a solution to this problem, a step-by-step approach for building a DW has been proposed in [1] (a.k.a. *Data Warehouse Bus Architecture*). This approach starts from data marts (DM) defined for individual business processes and continues exploring the common dimensional structures, which these DMs may possibly share. To facilitate this process, a matrix as the one shown in Table 1 is used, which relates DMs (i.e., their subsumed business requirements) to facts and dimensions implied by each DM. Such matrix is used for detecting how dimensions (in columns) are shared among facts of different DMs (in rows), i.e., if a fact of a DM in the row *x* is analyzed from a dimension of the column *y*, there is a tick in the intersection of *x* and *y*. The content of the matrix in Table 1 follows our running example based on the TPC-H benchmark [2], which is introduced in more detail in Section 2.1. We consider five information requirements (i.e., IR1–IR5) and for each of them a single DM. Each requirement analyzes some factual data (e.g., `IR3` analyzes the `revenue`), from different perspectives (e.g., `revenue` is analyzed in terms of parts – i.e., `partsupplier-part` hierarchy–, and supplier's region –i.e., `supplier-nation-region` hierarchy–). Finally, based on this matrix, different DMs are combined into an MD schema of a DW. However, such design guidelines still assume a tremendous manual effort from the DW architect and hence, DW experts still encounter the burdensome and time-lasting problem of translating the end-user's information requirements into the appropriate MD schema design.

Automating such process has several benefits. On the one hand, it supports the complex and time-consuming task of designing the DW schema. On the other hand, automatically produced results guarantee that the MD integrity constraints [3] are met as well as some DW quality objectives used to guide the process [4]. Accordingly, several works tried to automate the process of generating MD schemas (e.g., [5–7]). However, for the sake of automation, these approaches tend to overlook the importance of information requirements and focus mostly on the underlying data sources. Such practices require an additional manual work in conforming the automatically produced MD designs with the actual user requirements, which often does not scale well for complex scenarios. For example, it has been shown that even for smaller data source sizes the number of potential stars produced by means of a blind search of MD patterns over the sources is huge [7]. Consequently, it is not feasible to assume that the DW architect will be able to prune and filter such results manually.

To the best of our knowledge only three works went further and considered integrating the information requirements in their semi-automatic approaches for generating MD models [8–10]. At different levels of detail, these approaches support transforming every single requirement into an MD model to answer such requirement. However, how to integrate such individual MD models into a single, compact MD view is left to be done manually (although [8,9] introduce strict manual guidelines in their approaches to assist the designer). Our experiments, described in Section 6, have shown that integrating MD requirements is not an easy task and this process must also be supported by semi-automatic tools.

In this work, we present a semi-automatic method for Ontology-based data warehouse REquirement-driven evolution and integration (*ORE*). *ORE* complements the existing methods (e.g., [5–7]) and assists on semi-automatically integrating partial MD schemas (each representing a requirement or a set of requirements) into a unified MD schema design. Moreover, *ORE* could also be used to integrate existing MD schemas of any kind (e.g., as in [11]). *ORE* starts from a set of MD interpretations (MDIs) of individual requirements (which resemble the rows of Table 1). Intuitively, an MDI is an MD characterization of the sources that satisfies the requirement at hand (see Section 2.2 for further details). Iteratively, *ORE* integrates MDIs into a single MD schema which satisfies all requirements so far. Importantly, *ORE* generates MD-compliant results (i.e., fulfilling the MD integrity constraints) and determines the best integration options according to a set of quality objectives, to be defined by the DW designer. To guarantee so, *ORE* systematically traces valuable metadata from each integration iteration. During this entire process, the role of the data sources is crucial and *ORE* requires a characterization of the data sources in terms of a domain ontology, from where to automatically explore relationships between concepts (e.g., synonyms, functional dependencies, taxonomies, etc.) by means of reasoning.

Our method, *ORE*, is useful for the early stages of a DW project, where we need to create an MD schema design from scratch, but it can also serve during the entire DW lifecycle to accommodate potential evolution events. As we discuss later on, in the presence of a new requirement, our method does not create an MD design from scratch, rather it can automatically absorb the new requirement and integrate it with the existing MD schema.

Contributions: The main contributions of our work are as follows.

- We present a semi-automatic approach, *ORE*, which, in an iterative fashion deals with the problem of designing a unified MD schema from a set of information requirements.
- We introduce novel algorithms for integrating MD schemata, each satisfying one or more requirements. Results produced are guaranteed to subsume all requirements so far, preserve the MD integrity constraints, and meet the user defined quality objectives.
- We introduce the traceability metadata structure to systematically record information about the current integration opportunities both for finding the best