Data Lineage Tracing in Data Warehousing Environments

Hao Fan

International School of Software, WuHan University, China, 430072
hfan@iss.whu.edu.cn

Abstract. Data lineage tracing (DLT) is to find derivations of integrated data in integrated database systems, where the data sources might be autonomous, distributed and heterogeneous. In previous work, we present a DLT approach using partial schema transformation pathways. In this paper, we extend our DLT approach to using full schema transformation pathways and discuss the problem of lineage data ambiguities. Our DLT approach is not limited in one specific data model and query language, and would be useful in general data warehousing environments.

1 Introduction

Data from distributed, autonomous and heterogeneous data sources is collected into a central repository in a data warehouse system, in order to enable analysis and mining of the integrated information. However, in addition to analyzing the data in the integrated database, we sometimes also need to investigate how certain integrated information was derived from the data sources, which is the problem of *data lineage tracing* (DLT).

AutoMed[1] is a heterogeneous data transformation and integration system which offers the capability to handle data integration across multiple data models. In the AutoMed approach, the integration of schemas is specified as a sequence of primitive schema transformation steps, which incrementally add, delete, extend, contract or rename schema constructs, thereby transforming each source schema into the target schema. AutoMed uses a functional programming language based on comprehensions as its intermediate query language (IQL).

In previous work [8], we discussed how AutoMed metadata can be used to express the schemas and the cleansing, transformation and integration processes in heterogeneous data warehousing environments. In [7] and [9], we give the definitions of lineage data in terms of bag algebra, and present a DLT approach using partial schema transformation pathways, $i.e.$ only considering IQL queries and add and rename transformations. In this paper, we extend our DLT approach to considering full schema transformation pathways, which include queries beyond IQL, and delete, extend and contract transformations, and discuss the problem of lineage data ambiguities, namely the fact that equivalent queries may have different lineage data for identical tracing data. The tracing data is the data which lineage should be computed.

[1] See http://www.doc.ic.ac.uk/automed/

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 25–36, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The outline of this paper is as follows. Section 2 gives a review of related work. Section 3 gives an overview of AutoMed and our DLT approach using partial schema transformation pathways. Section 4 extends our DLT approach by considering queries beyond IQL, and delete, extend and contract transformations, and Section 5 discusses the ambiguity of lineage data. Finally, Section 6 gives our concluding remarks.

2 Related Work

The problem of data lineage tracing in data warehousing environments has been formally studied by Cui *et al.* in [6,5]. In particular, the fundamental definitions regarding data lineage, including *tuple derivation for an operator* and *tuple derivation for a view*, are developed in [6], and [5] introduces a way to trace data lineage for complex views in data warehouses. However, the approach is limited to the relational data model.

Another fundamental concept of data lineage is discussed by Buneman *et al.* in [2], namely the difference between "why" provenance and "where" provenance. Why-provenance refers to the source data that had some influence on the existence of the integrated data. Where-provenance refers to the actual data in the sources from which the integrated data was extracted.

In our approach, both why- and where-provenance are considered, using bag semantics. In [7], we define the notions of *affect-pool* and *origin-pool* for data lineage tracing in AutoMed — the former derives all of the source data that had some influence on the tracing data, while the latter derives the specific data in the sources from which the tracing data is extracted. In [9], we develop DLT formulae and algorithms for deriving the affect-pool and origin-pool of a data item along a virtual or partially materialised transformation pathway, where intermediate schema constructs may or may not be materialised.

Cui and Buneman in [4] and [2] also discuss the problem of ambiguity of lineage data. This problem is known as *derivation inequivalence* and arises when equivalent queries have different data lineages for identical tracing data. Cui and Buneman discuss this problem in two scenarios: (a) when aggregation functions are used and (b) when where-provenance is traced. In this paper, we investigate when ambiguity of lineage data may happen in our context and we describe how our DLT approach for tracing why-provenance can also be used for tracing where-provenance, so as to reduce the chance of derivation inequivalence occurring.

3 Data Lineage Tracing in AutoMed

3.1 Overview of AutoMed

AutoMed supports a low-level hypergraph-based data model (HDM). Higher-level modelling languages, such as relational, ER, OO, XML, flat-file and multidimensional data models, are defined in terms of this HDM. An HDM schema consists of a set of nodes, edges and constraints, and each modelling construct of a higher-level modelling language is specified as some combination of HDM nodes, edges and constraints. For any modelling language $\mathcal{M}$ specified in this way, via the API of AutoMed's Model