Chapter 6
On Handling the Evolution of External Data Sources in a Data Warehouse Architecture

Robert Wrembel
Poznan University of Technology, Poland

ABSTRACT

A data warehouse architecture (DWA) has been developed for the purpose of integrating data from multiple heterogeneous, distributed, and autonomous external data sources (EDSs) as well as for providing means for advanced analysis of integrated data. The major components of this architecture include: an external data source (EDS) layer, and extraction-transformation-loading (ETL) layer, a data warehouse (DW) layer, and an on-line analytical processing (OLAP) layer. Methods of designing a DWA, research developments, and most of the commercially available DW technologies tacitly assumed that a DWA is static. In practice, however, a DWA requires changes among others as the result of the evolution of EDSs, changes of the real world represented in a DW, and new user requirements. Changes in the structures of EDSs impact the ETL, DW, and OLAP layers. Since such changes are frequent, developing a technology for handling them automatically or semi-automatically in a DWA is of high practical importance. This chapter discusses challenges in designing, building, and managing a DWA that supports the evolution of structures of EDSs, evolution of an ETL layer, and evolution of a DW. The challenges and their solutions presented here are based on an experience of building a prototype Evolving-ETL and a prototype Multiversion Data Warehouse (MVDW). In details, this chapter presents the following issues: the concept of the MVDW, an approach to querying the MVDW, an approach to handling the evolution of an ETL layer, a technique for sharing data between multiple DW versions, and two index structures for the MVDW.

DOI: 10.4018/978-1-60960-537-7.ch006
On Handling the Evolution of External Data Sources in a Data Warehouse Architecture

INTRODUCTION

Contemporary model of managing enterprises, institutions, and organizations is based on decision support systems. In these systems, knowledge is gained from data analysis. Nowadays, core components of the majority of decision support systems include data warehouses (Kimball & Ross, 2002).

A data warehouse architecture has been developed for the purpose of: (1) providing a framework for the integration of multiple heterogeneous, distributed, and autonomous external data sources spread across a company and (2) providing means for advanced analysis of integrated data. A data warehouse architecture is typically composed of four layers, as shown in Figure 1. The first one - an external data sources (EDSs) layer represents heterogeneous and distributed production systems that are integrated. EDSs can include various fully functional databases as well as non-database storage systems. The second layer - an extraction-translation-loading (ETL) layer is responsible for executing multiple tasks, including: extracting data from EDSs, transforming data into a common data model, cleaning data, removing missing and inconsistent values, integrating data, removing duplicates, computing aggregates as well as loading data into a central repository (Kimball & Caserta, 2004). In practice, an ETL layer is modeled as a directed graph whose nodes represent the aforementioned tasks and arcs represent the execution order of these tasks, cf. (Vassiliadis et al., 2005).

The third layer includes a large central repository (database), called a data warehouse (DW), that stores integrated and summarized data. The fourth layer, called an on-line analytical processing (OLAP) layer, is responsible for various types of data analysis and visualizations. In this layer, OLAP applications, which execute complex queries, are used for the purpose of discovering trends, patterns of behavior, and anomalies as well as for finding hidden dependencies between data.

An inherent feature of EDSs is their evolution in time with respect to not only their contents (data) but also their structures (schemas). Content changes result from a day-to-day usage of EDSs. Structure changes result mainly from changes of the real world being modeled in EDSs and new user requirements. Both types of changes must be propagated into a DW architecture. In practice, content changes are handled and propagated by materialized views and standard DW refreshing techniques, cf. (Gupta, Mumick, 1999). Structure changes are more difficult to handle and manage since they have an impact on multiple layers of a DW architecture. First, structural changes have an impact on an ETL layer that must be redesigned and redeployed. Second, they have an impact on a data warehouse schema that must be modified in order to follow changes in EDSs (Rundensteiner, Koessler, & Zhang, 2000). DW schema changes result, in turn, in changes in OLAP applications. Additionally, new requirements of a DW users and creating various simulation scenarios (often called a ‘what-if’ analysis) may require changes in a DW schema.

As reported in (Sjøberg, 1993; Moon et al., 2008), structures of data sources change frequently. For example, the Wikipedia schema changed on average every 9-10 days during the last 4 years. Also our experience reveals that schemas of EDSs may change very frequently. For example, telecommunication data sources changed their schemas every 7-13 days, on average. Banking data sources are more stable but they changed their schemas every 2-4 weeks, on average. The most frequent changes concerned increasing the length of a column, changing a data type of a column, and adding a new column.

For a long period of time the existing DW technologies and research contributions have tacitly assumed that the structure of a DW is time invariant. As a consequence, many of the research developments and most of the commercially available DW technologies offer functionalities for managing data warehouses of static (time in-