Data Warehouse Back-End Tools

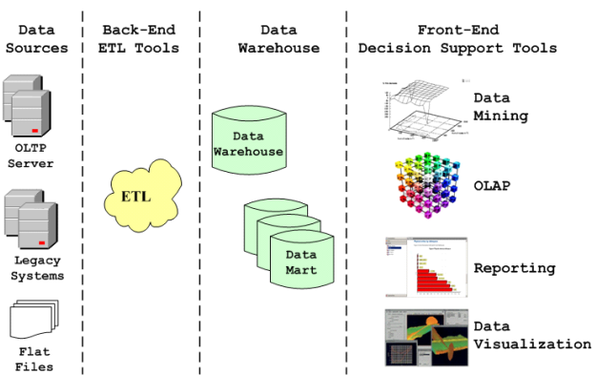
Alkis Simitsis
National Technical University of Athens, Greece

Dimitri Theodoratos
New Jersey Institute of Technology, USA

INTRODUCTION

The back-end tools of a data warehouse are pieces of software responsible for the extraction of data from several sources, their cleansing, customization, and insertion into a data warehouse. They are known under the general term *extraction, transformation and loading* (ETL) tools. In all the phases of an ETL process (extraction and exportation, transformation and cleaning, and loading), individual issues arise and, along with the problems and constraints that concern the overall ETL process, make its lifecycle a very complex task.

Figure 1. Abstract architecture of a data warehouse



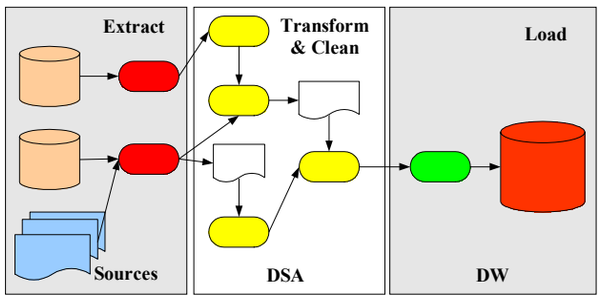
BACKGROUND

A Data Warehouse (DW) is a collection of technologies aimed at enabling the knowledge worker (executive, manager, analyst, etc.) to make better and faster decisions. Data warehouses typically are divided into the front-end part concerning end users who access the data warehouse with decision-support tools, and the back-stage part, where the collection, integration, cleaning and transformation of data takes place in order to populate the warehouse. The architecture of a data warehouse exhibits various layers of data in which data from one layer are derived from data of the previous layer (Figure 1). The processes that take part in the back stage of the data warehouse are data intensive, complex, and costly (Vassiliadis, 2000). Several reports mention that most of these processes are constructed through an in-house development procedure that can consume up to 70% of the resources for a data warehouse project (Gartner, 2003).

In order to facilitate and manage the data warehouse operational processes, commercial tools exist in the market under the general title Extraction-Transformation-Loading (ETL) tools. To give a general idea of the functionality of these tools, we mention their most prominent tasks, which include (a) the identification of relevant information at the source side; (b) the extraction of this information; (c) the customization and integration of the information coming from multiple sources into a common format; (d) the cleaning of the resulting data set on the basis of database and business rules; and (e) the propagation of the data to the data warehouse and/or data marts. In the sequel, we will adopt the general acronym ETL for all kinds of in-house or commercial tools and all the aforementioned categories of tasks.

In Figure 2, we abstractly describe the general framework for ETL processes. In the left side, we can observe the original data providers (sources). Typically, data providers are relational databases and files. The data from these sources are extracted by extraction routines, which provide either complete snapshots or differentials of the data sources. Then, these data are propagated to the Data Staging Area (DSA), where they are transformed and cleaned before being loaded to the data warehouse. Intermediate results, again in the form of (mostly) files or relational tables, are part of the data-staging area. The data warehouse is depicted in the right part of Figure 2 and comprises the target data stores (i.e., fact tables for the storage of information and dimension tables with the description and the multidimensional, roll-up hierarchies of the stored facts). The loading of the central warehouse is performed from the loading activities depicted right before the data warehouse data store.

Figure 2. The environment of extract-transformation-load processes



State of the Art

In the past, there have been research efforts toward the design and optimization of ETL tasks. We mention three research prototypes: (a) the AJAX system (Galhardas et al., 2000); (b) the Potter's Wheel system (Raman & Hellerstein, 2001); and (c) ARKTOS II (Arktos II, 2004). The first two prototypes are based on algebras, which we find mostly tailored for the case of homogenizing Web data; the latter concerns the modeling and the optimization of ETL processes in a customizable and extensible manner.

An extensive review of data quality problems and related literature, along with quality management methodologies, can be found in Jarke, et al. (2000). Rundensteiner (1999) offers a discussion of various aspects of data transformations. Sarawagi (2000) offers a similar collection of papers in the field of data, including a survey (Rahm & Do, 2000) that provides an extensive overview of the field, along with research issues and a review of some commercial tools and solutions on specific problems (Monge, 2000; Borkar et al., 2000). In a related but different context, we would like to mention the IBIS tool (Calì et al., 2003). IBIS is an integration tool following the global-as-view approach to answer queries in a mediated system.

Moreover, there is a variety of ETL tools in the market. Simitsis (2003) lists the ETL tools available at the time that this paper was written.

MAIN THRUST

In this section, we briefly review the problems and constraints that concern the overall ETL process, as well as the individual issues that arise separately in each phase of an ETL process (extraction and exportation, transformation and cleaning, and loading). Simitsis (2004) offers a detailed study on the problems described in this paper and presents a framework toward the modeling and the optimization of ETL processes.

Scalzo (2003) mentions that 90% of the problems in data warehouses arise from the nightly batch cycles that load the data. At this stage, the administrators have to deal with problems like (a) efficient data loading and (b) concurrent job mixture and dependencies. Moreover, ETL processes have global time constraints, including the time they must be initiated and their completion deadlines. In fact, in most cases, there is a tight time window in the night that can be exploited for the refreshment of the data warehouse, since the source system is off-line or not heavily used during this period. Other general problems include the scheduling of the overall process, the finding of the right execution order for dependent jobs and job sets on the existing hardware for the permitted time schedule, and the maintenance of the information in the data warehouse.

Phase I: Extraction and Transportation

During the ETL process, a first task that must be performed is the extraction of the relevant information that has to be propagated further to the warehouse (Theodoratos et al., 2001). In order to minimize the overall processing time, this involves only a fraction of the source data that has changed since the previous execution of the ETL process, mainly concerning the newly inserted and possibly updated records. Usually, change detection is performed physically by the comparison of two snapshots (one corresponding to the previous extraction and the other to the current one). Efficient algorithms exist for this task, like the snapshot differential algorithms presented in Labio and Garcia-Molina (1996). Another technique is log sniffing (i.e., the scanning of the log file in order to reconstruct the changes performed since the last scan). In rare cases, change detection can be facilitated by the use of triggers. However, this solution is technically impossible for many of the sources that are legacy systems or plain flat files. In numerous other cases, where relational systems are used at the source side, the usage of triggers also is prohibitive due to the performance degradation that their usage incurs and to the need to intervene in the structure of the database. Moreover, another crucial issue concerns the transportation of data after the extraction, where tasks like ftp, encryption-decryption, compression-decompression, and so forth can possibly take place.

Phase II: Transformation and Cleaning

It is possible to determine typical tasks that take place during the transformation and cleaning phase of an ETL process. Rahm and Do (2000) further detail this phase in the following tasks: (a) data analysis; (b) definition of