Extraction–Transformation–Loading Processes

Alkis Simitsis  
*National Technical University of Athens, Greece*

Panos Vassiliadis  
*University of Ioannina, Greece*

Timos Sellis  
*National Technical University of Athens, Greece*

**INTRODUCTION**

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. The architecture of a DW exhibits various layers of data in which data from one layer are derived from data of the lower layer (see Figure 1). The operational databases, also called data sources, form the starting layer. They may consist of structured data stored in open database and legacy systems, or even in files. The central layer of the architecture is the global DW. The global DW keeps a historical record of data that result from the transformation, integration, and aggregation of detailed data found in the data sources. An auxiliary area of volatile data, data staging area (DSA) is employed for the purpose of data transformation, reconciliation, and cleaning. The next layer of data involves client warehouses, which contain highly aggregated data, directly derived from the global warehouse. There are various kinds of local warehouses, such as data mart or on-line analytical processing (OLAP) databases, which may use relational database systems or specific multidimensional data structures. The whole environment is described in terms of its components, metadata, and processes in a central metadata repository, located at the DW site.

In order to facilitate and manage the DW operational processes, specialized tools are available in the market, under the general title extraction-transformation-loading (ETL) tools. ETL tools are pieces of software responsible for the extraction of data from several sources, their cleansing, customization, and insertion into a DW (see Figure 2). The functionality of these tools includes:

- the *extraction* of relevant information at the source side;
- the *transportation* of this information to the DSA;
- the *transformation* (i.e., customization and integration) of the information coming from multiple sources into a common format;
- the *cleaning* of the resulting data set, on the basis of database and business rules; and
- the *propagation and loading* of the data to the DW and the *refreshment* of data marts.

**BACKGROUND**

In the past, there have been research efforts towards the design and optimization of ETL tasks. We mention three research prototypes: (a) AJAX, (b) Potter’s Wheel, and (c) Arktos II. 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 of ETL processes in a customizable and extensible manner.

![Figure 1. A data warehouse environment](image1)

![Figure 2. ETL processes in detail](image2)
The AJAX system (Galhardas, Florescu, Sasha, & Simon, 2000) deals with typical data quality problems, such as the object identity problem, errors due to mistyping, and data inconsistencies between matching records. This tool can be used either for a single source or for integrating multiple data sources. AJAX provides a framework wherein the logic of a data cleaning program is modeled as a directed graph of data transformations that starts from some input source data. AJAX also provides a declarative language for specifying data cleaning programs, which consists of SQL statements enriched with a set of specific primitives to express mapping, matching, clustering, and merging transformations. Finally, an interactive environment is supplied to the user to resolve errors and inconsistencies that cannot be automatically handled and to support a stepwise refinement design of data cleaning programs.

Raman and Hellerstein (2001) present the Potter’s Wheel system, which is targeted to provide interactive data cleaning to its users. The system offers the possibility of performing several algebraic operations over an underlying data set, including format (application of a function), drop, copy, add a column, merge delimited columns, split a column on the basis of a regular expression or a position in a string, divide a column on the basis of a predicate (resulting in two columns, the first involving the rows satisfying the condition of the predicate and the second involving the rest), selection of rows on the basis of a condition, folding columns (where a set of attributes of a record is split into several rows), and unfolding. Optimization algorithms are also provided for the CPU usage for certain classes of operators. The general idea behind Potter’s Wheel is that users build data transformations in an iterative and interactive way; thereby, users can gradually build transformations as discrepancies are found and clean the data without writing complex programs or enduring long delays.

ARKTOS II is a coherent framework for the conceptual, logical, and physical design of ETL processes. The goal of this line of research is to facilitate, manage, and optimize the design and implementation of the ETL processes, during both the initial design and deployment stage, as such during the continuous evolution of the DW. To this end, Vassiliadis, Simitsis, and Skiadopoulos (2002) and Simitsis and Vassiliadis (2003) proposed a novel conceptual model. Further, Simitsis, Vassiliadis, Skiadopoulos, and Sellis (2003) and Vassiliadis et al. (2004) presented a novel logical model. The proposed models, conceptual and logical, were constructed in a customizable and extensible manner so that the designer can enrich them with his own reoccurring patterns for ETL processes. Therefore, ARKTOS II offers a palette of several templates, representing frequently used ETL transformations along with their semantics and their interconnection (see Figure 3). In this way, the construction of ETL scenarios as a flow of these transformations, is facilitated. Additionally, ARKTOS II takes into account the optimization of ETL scenarios, with a main focus on the improvement of the time performance of an ETL process, and ARKTOS II tackles the problem of how the software design of an ETL scenario can be improved, without any impact on its consistency.

An extensive review of data quality problems and related literature, along with quality management methodologies can be found in Jarke, Lenzerini, Vassiliou, and Vassiliadis (2000). Rundensteiner (1999) offered a discussion on various aspects of data transformations. Sarawagi (2000) offered 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 for specific problems (e.g., Borkar, Deshmuk, & Sarawagi, 2000; Monge, 2000). In a related but different context, the IBIS tool (Calì et al., 2003) is an integration tool following the global-as-view approach to answer queries in a mediated system. Moreover, there exists a variety of ETL tools in the market. Simitsis (2004) listed the ETL tools available at the time this paper was written.

**MAIN THRUST OF THE ARTICLE**

In this section we briefly review the individual issues that arise in each of the phases of an ETL process, as well as the problems and constraints that concern it.

### Figure 3. Typical template transformations provided by ARKTOS II

<table>
<thead>
<tr>
<th>Filters</th>
<th>Unary transformations</th>
<th>Binary transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection (σ)</td>
<td>Push</td>
<td>Union (U)</td>
</tr>
<tr>
<td>Not null (NN)</td>
<td>Aggregation (σ)</td>
<td>Join (σ*τ)</td>
</tr>
<tr>
<td>Primary key violation (PK)</td>
<td>Projection (π)</td>
<td>Diff (A)</td>
</tr>
<tr>
<td>Foreign key violation (FK)</td>
<td>Function application (f)</td>
<td>Update Detection (Δuτ)</td>
</tr>
<tr>
<td>Unique value (UN)</td>
<td>Tuple normalization (T)</td>
<td></td>
</tr>
<tr>
<td>Domain mismatch (DM)</td>
<td>Tuple denormalization (DN)</td>
<td></td>
</tr>
<tr>
<td>Transfer operations</td>
<td>File operations</td>
<td></td>
</tr>
<tr>
<td>Ftp (FTP)</td>
<td>EBCDIC to ASCII conversion (EB2AS)</td>
<td></td>
</tr>
<tr>
<td>Compress/Decompress (Z/Zip)</td>
<td>Sort file (Sort)</td>
<td>Switch (σ*)</td>
</tr>
<tr>
<td>Encrypt/Decrypt (Cri/Cr)</td>
<td></td>
<td>Extended union (U)</td>
</tr>
</tbody>
</table>
Related Content

Constraint-Based Multi-Dimensional Databases
[www.irma-international.org/chapter/constraint-based-multi-dimensional-databases/7953/](www.irma-international.org/chapter/constraint-based-multi-dimensional-databases/7953/)

Versioning Approach for Database Evolution
[www.irma-international.org/chapter/versioning-approach-database-evolution/20692/](www.irma-international.org/chapter/versioning-approach-database-evolution/20692/)

Maintaining Mappings between Conceptual Models and Relational Schemas
[www.irma-international.org/chapter/maintaining-mappings-between-conceptual-models/63669/](www.irma-international.org/chapter/maintaining-mappings-between-conceptual-models/63669/)

Development of an E-Healthcare Information Security Risk Assessment Method

Mining Allocating Patterns in Investment Portfolios
[www.irma-international.org/chapter/mining-allocating-patterns-investment-portfolios/8056/](www.irma-international.org/chapter/mining-allocating-patterns-investment-portfolios/8056/)