Extraction–Transformation–Loading Processes

E

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

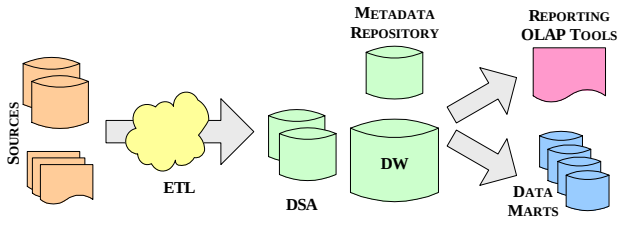
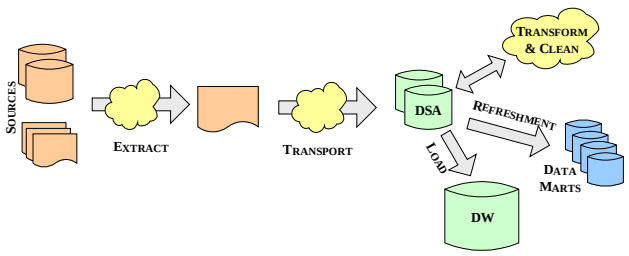


Figure 2. ETL processes in detail



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 & Hellerstein, 2001) presents 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 CHAPTER

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

Filters	**Unary transformations**	**Binary transformations**
Selection (σ)	Push	Union (U)
Not null (NN)	Aggregation (γ)	Join (⋈)
Primary key violation (PK)	Projection (π)	Diff (Δ)
Foreign key violation (FK)	Function application (f)	Update Detection (Δ_{UPD})
Unique value (UN)	Surrogate key assignment (SK)	
Domain mismatch (DM)	Tuple normalization (N)	**Composite transformations**
	Tuple denormalization (DN)	Slowly changing dimension (Type 1,2,3)(SDC-1/2/3)
Transfer operations		Format mismatch (FM)
Ftp (FTP)	**File operations**	Data type conversion (DTC)
Compress/Decompress (Z/dZ)	EBCDIC to ASCII conversion (EB2AS)	Switch (σ*)
Encrypt/Decrypt (Cr/dCr)	Sort file (Sort)	Extended union (U)

Global Problems and Constraints

Scalzo (2003) mentions that 90% of the problems in DW arise from the nightly batch cycles that load the data. At this period, administrators have to deal with problems such as (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 during the night that can be exploited for the refreshment of the DW, because the source system is off-line or not heavily used during this period.

Consequently, a major problem arises with the scheduling of the overall process. The administrator must find the right execution order for dependent jobs and job sets on the existing hardware for the permitted time schedule. On the other hand, if the OLTP applications cannot produce the necessary source data in time for processing before the DW comes online, the information in the DW will be out of date. Still, because DWs are used for strategic purposes, this problem can be afforded because long-term reporting and planning is not severely affected by this type of failure.

Extraction and Transportation

During the ETL process, one of the first tasks that must be performed is the extraction of relevant information that must be further propagated to the warehouse. 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 and mainly concerns the newly inserted, and possibly updated, records. Usually, change detection is physically performed by the comparison of two snapshots (one corresponding to the previous extraction and the other corresponding to the current one). Efficient algorithms exist for this task, such as the snapshot differential algorithms presented by Labio and Garcia-Molina (1996). Another technique is log sniffing, (i.e., the scanning of the log file 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 consist of legacy systems or plain flat files. In numerous other cases, where relational systems are used at the source side, the usage of triggers is also prohibitive due to the performance degradation that their usage incurs and the need to intervene in the structure of the database. Moreover, another crucial issue concerns the transportation of data after the extraction, where tasks such as FTP, encryption–decryption, compression–decompression, and so forth, can take place.

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 transformation workflow and mapping rules, (c) verification, (d) transformation, and (e) backflow of cleaned data.

In terms of the transformation tasks, we can categorize the problem in two main classes of problems (Lenzerini, 2002):

- conflicts and problems at the schema level, and,
- data-level transformations (i.e., at the instance level).

The main problems with the schema level are (a) naming conflicts, where the same name is used for different objects (homonyms) or different names are used for the same object (synonyms), and (b) structural conflicts, where one must deal with different representations of the same object in different sources.

In addition, there are many variations of data-level conflicts across sources: duplicated or contradicting records, different value representations (e.g., marital status), different interpretation of the values (e.g., measurement units dollar vs. euro), different aggregation levels (e.g., sales per product vs. sales per product group), or reference to different points in time (e.g., current sales as of yesterday for source 1 vs. current sales as of last week for source 2). The list is enriched by low-level technical problems such as data type conversions, applying format masks, assigning fields to a sequence number, substituting constants, setting values to NULL or DEFAULT based on a condition, or using simple SQL operators (e.g., UPPER, TRUNC, SUBSTR).

In sequel, we present three common ETL transformations as examples: (a) semantic reconciliation and denormalization; (b) surrogate key assignment; and (c) string problems.

- **Semantic reconciliation and denormalization:** Frequently, DWs are highly denormalized, to answer more quickly certain queries. For example, in Figure 4, one can observe that a query on the total income of an employee in table DW.EMP can easily be computed as an addition of the attributes Salary, Tax, Bonus, whereas in the schema of the OLTP table EMP_INCOME, we should apply an aggregation operation. The transformation of the information organized in rows to the information organized in columns is called rotation or denormalization, because, frequently,

Figure 4. Semantic reconciliation and denormalization

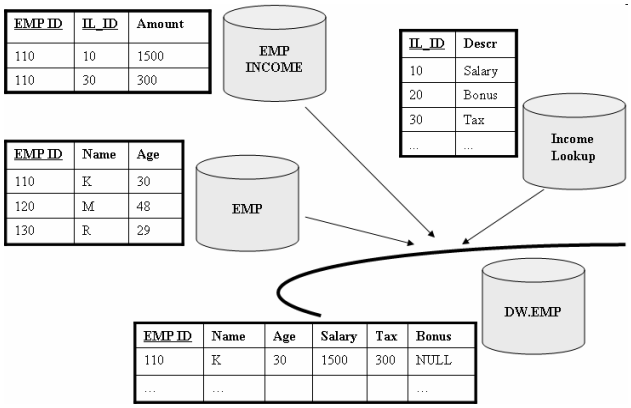
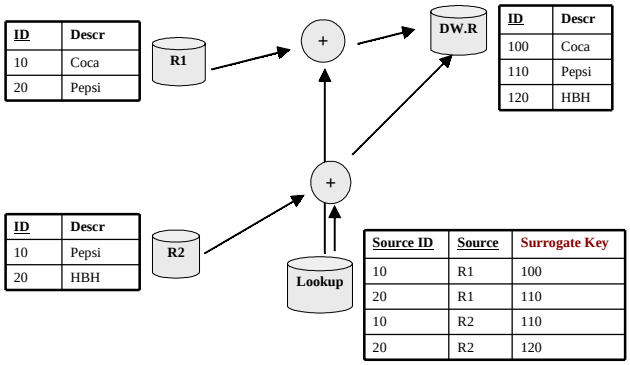


Figure 5. Surrogate key assignment



the derived values (e.g., the total income) are also stored as columns, functionally dependent on other attributes. Occasionally, it is possible to apply the reverse transformation to normalize denormalized data before being loaded to the DW. Observe also, how the different tables at the source side (i.e., EMP, EMP_INCOME, Income_Lookup) are integrated into a single DW table (i.e., DW.EMP).

- **Surrogate keys:** In a DW project, we usually replace the keys of the production systems with a uniform key, which we call a *surrogate key* (Kimball, Reeves, Ross, & Thornthwaite, 1998). Reasons for this replacement are performance and semantic homogeneity. Performance is affected because textual attributes are not the best candidates for indexed keys and thus need to be replaced by integer keys. More important, semantic homogeneity causes reconciliation problems because different production systems might use different keys for the same object (synonyms), or the same key for different objects (homonyms), resulting in the need for a global replacement of those values in the DW. Observe row 10, Pepsi, in table R2 of Figure 5. This row has a synonym conflict with row 20, Pepsi, in table R1 because they both represent the same real-world entity with different IDs, and it has a homonym conflict with row 10, Coca, in table R1 (over attribute ID). The production key ID is replaced by a surrogate key through a lookup table of the form Lookup(SourceID,Source,SurrogateKey). The Source column of this table allows that there can be synonyms in the different sources, which are mapped to different objects in the DW (e.g., value 10 in tables R1 and R2). At the end of this process, the DW table DW.R has globally unique, reconciled keys.
- **String problems:** A major challenge in ETL processes is the cleaning and homogenization of string data, that is, data that stands for addresses, acronyms, names, and so forth. Usually, the approaches for the solution of this problem include the application of regular expressions (e.g., using Perl programs) for the normalization of string data to a set of reference values. Figure 6 illustrates an example of this problem.

Loading

The final loading of the DW has its own technical challenges. Simple SQL commands are not sufficient because the open-loop-fetch technique, in which records are inserted one by one, is extremely slow for the vast volume of data to be loaded in the warehouse. An extra problem is the simultaneous usage of the rollback segments and log files during the loading process. Turning them off might include some risk in the case of a loading failure. So far, the best technique is the usage of the batch loading tools offered by most RDBMSs that avoids these problems.

Another problem is discriminating between new and existing data at loading time. This problem arises when a

Figure 6. String problems

Source Value	DW value
HP	HP
H.P.	HP
H-P	HP
Hewlett-Packard	HP
Hioulet-Pakard	HP
DEC	DEC
Digital Co.	DEC
…	…

set of records has to be classified to (a) the new rows that need to be appended to the warehouse and (b) rows that already exist in the DW but whose value has changed and must be updated (e.g., with an UPDATE command). Modern ETL tools already provide mechanisms towards this problem, mostly through language predicates.

Techniques that facilitate the loading task involve the simultaneous creation of tables and their respective indexes, the minimization of interprocess wait states, and the maximization of concurrent CPU usage.

FUTURE TRENDS

In a recent study, Giga Information Group (2002) reported that the ETL market reached a size of $667 million for year 2001; still, the growth rate reached a rather low 11% (compared with a 60% growth rate for year 2000). The study also proposed future technological challenges and forecasts that involve the integration of ETL with (a) XML adapters; (b) EAI (Enterprise Application Integration) tools (e.g., MQ-Series); (c) customized data quality tools; and (d) the move towards parallel processing of the ETL workflows.

There are several issues that are technologically open and that present interesting topics of research for the future. Active ETL, the need to refresh the warehouse with the freshest data possible (ideally, online) is a hot topic that has recently appeared (Adzic & Fiore, 2003). The need for optimal algorithms, for the individual transformations and for the whole process, is also an interesting topic of research. Finally, we anticipate that the extension of the ETL mechanisms for nontraditional data such as XML/HTML, spatial and biomedical data will also be a hot topic of research.

CONCLUSION

ETL tools are pieces of software responsible for the extraction of data from several sources, their cleansing, customization, and insertion into a DW. 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 very troublesome. Although, state of the art in the field of both research and commercial ETL tools includes some pieces of remarkable progress, a lot of work remains to be done before we claim that this problem is resolved. To this end, recent studies account this subject a research challenge and pinpoint the main topics of future work.

REFERENCES

Adzic, J., & Fiore, V. (2003). *Data warehouse population platform.* Proceedings of 5th International Workshop on the Design and Management of Data Warehouses (DMDW '03), Berlin, Germany.

Borkar, V., Deshmuk, K., & Sarawagi, S. (2000). Automatically extracting structure from free text addresses. *Bulletin of the Technical Committee on Data Engineering, 23*(4).

Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M., Naggar, P., & Vernacotola, F. (2003). *IBIS: Semantic data integration at work.* Proceedings of the 15th CAiSE: Vol. 2681. Lecture notes in computer science (pp. 79-94). Springer.

Galhardas, H., Florescu, D., Shasha, D., & Simon, E. (2000). Ajax: An extensible data cleaning tool. In *Proceedingsof the ACM SIGMOD international conference on the management of data* (p. 590). Dallas: TX.

Giga Information Group. (2002). *Market overview update: ETL* (Tech. Rep. No. RPA-032002-00021).

Inmon, W.-H. (1996). *Building the data warehouse* (2nd ed.). New York: John Wiley & Sons.

Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (Eds.). (2000). *Fundamentals of data warehouses*. Springer-Verlag.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The data warehouse lifecycle toolkit: Expert methods for designing, developing, and deploying data warehouses.* John Wiley & Sons.

Labio, W., & Garcia-Molina, H. (1996). *Efficient snapshot differential algorithms for data warehousing.* Proceedings of the 22nd international conference on very large data bases (VLDB)(pp. 63-74). Bombay, India.

Lenzerini, M. (2002). *Data integration: A theoretical perspective.* Proceedings of the 21st symposium on principles of database systems (PODS) (pp. 233-246). Wisconsin.

Monge, A. (2000). Matching algorithms within a duplicate detection system. *Bulletin of the Technical Committee on Data Engineering, 23*(4).

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *Bulletin of the Technical Committee on Data Engineering, 23*(4).

Raman, V., & Hellerstein, J. (2001). *Potter's wheel: An interactive data cleaning system.* Proceedings of the 27th

international conference on very large data bases (VLDB) (pp. 381-390), Rome, Italy.

Rundensteiner, E. (Ed.). (1999). Data transformations [Special issue]. *Bulletin of the Technical Committee on Data Engineering*, *22*(1).

Sarawagi, S. (2000). Data cleaning [Special issue]. *Bulletin of the Technical Committee on Data Engineering*, *23*(4).

Scalzo, B. (2003). *Oracle DBA guide to data warehousing and star schemas.* Prentice Hall.

Simitsis, A., (May 10, 2004). List of ETL tools. Retrieved May 10, 2004, from *http://www.dbnet.ece.ntua.gr/~asimi/ETLTools.htm*

Simitsis, A., & Vassiliadis, P. (2003). *A methodology for the conceptual modeling of ETL processes.* Proceedings of the decision systems engineering (DSE '03), Velden, Austria.

Simitsis, A., Vassiliadis, P., Skiadopoulos, S., & Sellis, T. (2003). *Modeling of ETL processes using graphs.* Proceedings of the 2nd Hellenic Data Management Symposium (HDMS03), Athens, Greece.

Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). *Conceptual modeling for ETL processes.* Proceedings of the 5th data warehousing and OLAP (DOLAP '02), McLean, VA.

Vassiliadis, P., Simitsis, A., Georgantas, P., Terrovitis, M., & Skiadopoulos, S. (2004). A generic and customizable framework for the design of ETL scenarios. *Information Systems Journal*.

KEY TERMS

Data Mart: A logical subset of the complete data warehouse. We often view the data mart as the restriction of the data warehouse to a single business process or to a group of related business processes targeted towards a particular business group.

Data Staging Area (DSA): An auxiliary area of volatile data employed for the purpose of data transformation, reconciliation, and cleaning before the final loading of the data warehouse.

Data Warehouse (DW): A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data used to support the strategic decision-making processes for the enterprise. It is the central point of data integration for business intelligence and is the source of data for the data marts, delivering a common view of enterprise data (Inmon, 1996).

ETL: Extract, transform, and load (ETL) are data warehousing functions that involves extracting data from outside sources, transforming it to fit business needs, and ultimately loading it into the data warehouse. ETL is an important part of data warehousing; it is the way data actually gets loaded into the warehouse.

On-Line Analytical Processing (OLAP): The general activity of querying and presenting text and number data from data warehouses as well as a specifically dimensional style of querying and presenting that is exemplified by a number of OLAP vendors.

Source System: An operational system of record whose function is to capture the transactions of the business. A source system is often called a legacy system in a mainframe environment.

Target System: The physical machine on which the data warehouse is organized and stored for direct querying by end users, report writers, and other applications.