Volume 5, No. 1, January 2014

Journal of Global Research in Computer Science

ISSN-2229-371X

REVIEW ARTICLE

Available Online at www.jgrcs.info

THE CONCEPTUAL MODELING OF ETL PROCESSES

Nitin Anand
AIACT&R New Delhi
proudtobeanindiannitin@gmail.com

Abstract: An ETL process includes various ETL activities, such as filtering, aggregating, checking for null values, etc., which can be represented by the constraint functions and transforming operations defined in previous section. However, the activities cannot exist in an ETL process independently; they must be organized in certain order that is specified in an ETL task of the ETL process. We think that ETL tasks are basic units in an ETL process and an ETL task is also the basic procedure to transfer data from a data source into a data target. An ETL task includes an ETL mapping and the descriptions of the data sources and the data target, such as the lists of attributes, the types of attributes, etc. For a set of data sources and a target DW, we encapsulate all tasks between the data sources and the target DW into one ETL session, which also contains the information for connecting the sources and the target DW.

Keywords: Data Quality (DQ), Data Extraction, Extraction-Transformation-Loading (ETL), Security, Simulation, Staging Area

INTRODUCTION

ETL (Extract-Transform-Load), which is the process of extracting data from a variety of heterogeneous data sources, and transforming those extracted data into needed format, and then loading those data into the DW (Data Warehouse) [1].

ETL processes contain 3 parts: Extraction, Transformation and Loading, each of which has its own metadata.

a. ***Extraction****:* Data extraction is the process of capturing data source, that is to say, reading the data from all kinds of original operation systems and cleansing the data, which is the premise of all the work. If there are no related mapping rules and metadata [2].
b. ***Transformation****:* Data transformation is the process of transforming above data by some prearrange rules, and dealing with some redundant, ambiguous, incomplete and anti-rules data to realize a unity of data granularity and data format. If we want to finish the data transformation from the source data storage format to the target data storage format, we have to know the information about source data and target data, which are also metadata [2].
c. ***Loading****:* Data loading is the process of importing above data to DW system by all or by planned increment. In order to load transformed data to the DW system, we also need metadata about mapping rules [2].

From the 3 processes, we can see metadata plays an important role in ETL, whose mismanagement can lead to the ineffectiveness of ETL processes directly. ETL processes often fails through its triviality and fallibility.

The architecture of ETL is shown as Fig. 1[3]. The phases of extract, transform and load were executed in one single process. Under the framework of conventional ETL, the ETL process is defined: for different data source, develop and compile program or script; retrieval records from database; after extract, exchange the data according to users' requirement; load the data to target data warehouse; and process the records piece by piece until the end of source database. The framework of ETL is simple and would be easily implemented under the conventional architecture, but the weakness is obvious: The efficiency and reliability of load is lame which makes the overall scenario weak and difficult [4]

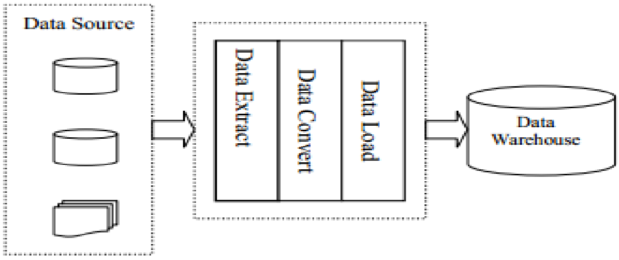

Figure 1 The Architecture of ETL.[3]

RELATED WORK

Many researches done in DQ area are connected to match and merge techniques. Theory on this topic, as well as case studies is covered in [5]. Most authors try to solve the problems with object identity, and the main difference between them is the approach they use. Implementation as an extension to Common Warehouse MetaModel is described in [6]. So far, research has mostly dealt with the problem of maintaining the warehouse in its traditional periodical update setup [7, 8]. Temporal data warehouses address the issue of supporting temporal information efficiently in data warehousing systems [9]. In [8], the authors present efficient techniques (e.g. temporal view self-maintenance) for maintaining data warehouses without disturbing source operations. A related challenge is supporting large-scale temporal aggregation operations in data warehouses [10]. In [11], the authors describe an approach which clearly separates the DW refreshment process from its traditional handling as a view maintenance or bulk loading process. They provide a conceptual model of the process, which is treated as a composite workflow, but they do not describe how to efficiently propagate the date. Theodoratus et al. discuss in [12] data currency quality factors in data warehouses.

MODEL-DRIVEN FRAMEWORK

The ETL process development uses a Model-Driven Development (MDD) approach. In this section, we concretely show how this approach allows organizing the various components of this framework in order to efficiently perform the design and implementation phases of the ETL process development.

MDD-Based Framework:

MDD is an approach to software development where extensive models are created before source code is written. As shown in Fig. 2, the MDD approach defines four main layers (see Meta-Object Facilities (MOF) [13]: the Model Instance layer (M0), the Model layer (M1), the Meta-Model layer (M2), and the Meta-Meta-Model layer (M3).

The Model Instance layer (M0) is a representation of the real-world system where the ETL process design and implementation are intended to perform. This may be represented, respectively, by a vendor-independent graphical interface and by a vendor-specific ETL engine. At the Model layer (M1), both the ETL Process Model is designed and the ETL process code is derived by applying a set of transformations thus moving from the design to the implementation.

The Meta-Model layer (M2) consists of the BPMN4ETL Metamodel that defines ETL patterns at the design phase, and a 4GL grammar at the implementation phase. Finally, the Meta-Meta-Model level (M3), corresponds to the MOF meta-metamodel at the design phase, while it corresponds to the Backus Naur Form (BNF) at the implementation phase.

http://www.omg.org/spec/MOF/2.0/

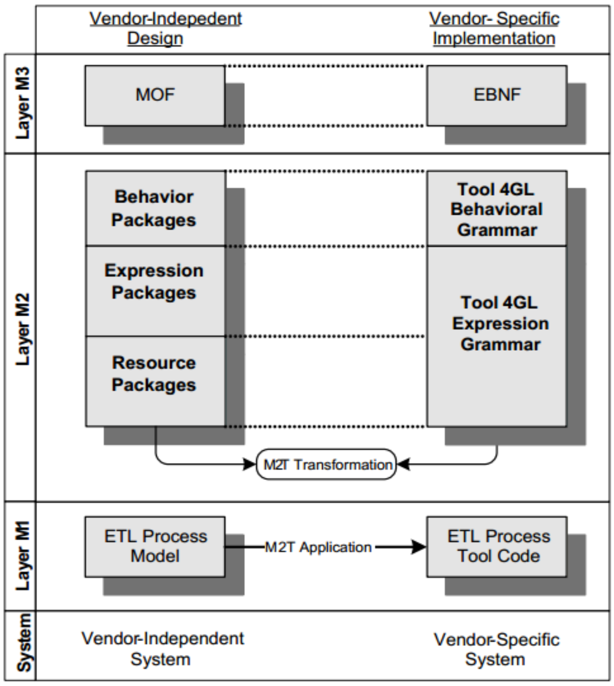

Figure 2: MDD layers for the ETL development framework

CONCLUSION AND FUTUREWORK

In a recent study [14], the authors report that due to the diversity and heterogeneity of data sources, ETL is unlikely to become an open commodity market.. Since quality plays an important role in developing software products, We have presented functional requirements along with non-functional requirement i.e., security requirements. This approach is better compared to existing systems. In [15], the authors report on their data warehouse population system.

The architecture of the system is discussed in the paper, with particular interest (a) in a "shared data area‟ , which is an in-memory area for data transformations, with a specialized area for rapid access to lookup tables and (b) the pipelining of the ETL processes.

The future work may include to dealing with other non-functional requirements like reliability, performance etc. In this paper, we have focused on the data-centric part of logical design of the ETL scenario of a data warehouse.

First, we have defined a formal logical met model as a logical abstraction of ETL processes. The data stores, activities and their constituent parts, as well as the provider relationships that map data producers to data consumers have formally been defined. Then, we have provided a reusability framework that complements the genericity of the aforementioned metamodel. Practically, this is achieved from an extensible set of specializations of the entities of the metamodel layer, specifically tailored for the most frequent elements of ETL scenarios, which we call template activities. In the context of template materialization, we have dealt with specific language issues, in terms of the mechanics of template instantiation to concrete activities.

REFERENCES

[1]. Zhang X.F, Sun W.W, Wang W., et a1 . Generating Incremental ETL Processes Automatically. Computer and Computational Sciences, 2006 : 516—521.

[2]. Zhang Zhongping and Zhao Ruizhen, Design of architecture for ETL based on metedata-driven, Computer Applications and Software, vol. 26, Jun. 2006, pp. 61-63

[3]. Sun Wei, and Zhang Zhongneng. "ETL Architecture Research ". Micro-computer Application, 2005, 21(3):13-15.

[4]. Zhao Xiaofei, and Huang Zhiqiu, "A Formal Framework for Reasoning on Metadata Based on CWM, " The 25th International Conference on Conceptual Modeling, 2006: 371-384

[5]. T. N. Herzog and F. Scheuren and W. E. Winkler. Data Quality and Record Linkage Techniques.Springer, Heidelberg, 2007

[6]. P. Gomes and J. Farinha and M. J. Trigueiros. A Data Quality Metamodel Extension to CWM. In 4th Asia Pacific Conference on Conceptual Modelling Proceedings, pages 17–26. APCCM, February 2007

[7]. W. Labio, J. Yang, Y. Cui, H.Garcia-Molina, and J. Widom, 2000. "Performance Issues in Incremental Warehouse Maintenance", International Conference on Very Large Data Bases (VLDB).

[8]. J. Yang, and J. Widom, 2001. "Temporal View Self-Maintenance", 7th Int. Conf. Extending Database Technology (EDBT).

[9]. J.Yang, 2001. “Temporal Data Warehousing”, Ph.D. Thesis, Dept. Computer Science, Stanford University.

[10]. J. Yang, and J. Widom, 2001. “Incremental Computation and Maintenance of Temporal Aggregates”, 17th Intern. Conference on Data Engineering (ICDE).

[11]. M. Bouzeghoub, F. Fabret, and M. Matulovic, 1999. “Modeling Data Warehouse Refreshment Process as a Workflow Application”, Intern. Workshop on Design and Management of Data Warehouses (DMDW)

[12]. D. Theodoratus, and M. Bouzeghoub, 1999. “Data Currency Quality Factors in Data Warehouse Design”, International Workshop on the Design and Management of Data Warehouses (DMDW)

[13]. S. Lujan-Mora and J. Trujillo. Physical modeling of data warehouses using UML. In I. Song and K. Davis, editors, Proceedings of the 7th ACM International Workshop on Data Warehousing and OLAP,DOLAP'04, pages 48{57, Washington, D.C., USA, Nov. 2005. ACM Press.

[14]. Giga Information Group. Market Overview Update:ETL. Technical Report RPA-032002-00021, March 2002.

[15]. J. Adzic, V. Fiore, Data Warehouse Population Platform,in: Proceedings of the Fifth International Workshop on the Design and Management of Data Warehouses (DMDW‟ 03), Berlin, Germany, September 2003.