



Design of a Data Warehouse Following ETL Process Modeling

Maxime Kasonga Badibanga

Academic Personnel, Unikan

SUMMARY:

An Extract-Transform-Load (ETL) process is very complex in terms of data flow and the tasks responsible for cleaning, filtering, normalizing and loading the data into the data warehouse.

Extracting data from sources, transforming it to deliver quality data with value for analysis) loading prepared data into the warehouse. In this article, we propose a modeling of ETL processes at the conceptual and logical level, the models obtained are stored in the form of XML documents. We based ourselves on the approach of Panos Vassiliadis et al, while adapting the conceptual metamodel and proposing a metamodel at the logical level¹.

Keywords: ETL; XML, Data warehouse

1. Introduction

Faced with a very competitive market, companies must have great analytical capabilities. The information available to the company constitutes a valuable resource for analyzing, understanding and acting accordingly.

However, the big difficulty lies in the quantities of voluminous data stored in various formats and models. In addition to this heterogeneity, these source data were not created with analytical perspectives. Despite the difficulties they present, this data is of inestimable value for analysis and decision support applications.

Before being loaded into a warehouse, the source data goes through an extraction, transformation and loading process, known as ETL, to prepare it and give it the necessary properties in terms of relevance.

Given the complexity and importance of ETL in a decision-making project, we are interested in this article in the modeling of an ETL process at the conceptual and logical levels making it possible to anticipate complexity and hazards in order to have visibility over the entire process before its implementation.

In this problem, Panos Vassiliadis et al. proposed an adhoc graphic formalism allowing data to be modeled in the form of concepts characterized by a set of attributes².

The power of the formalism of Vassiliadis et al. is in the detailed description of the transformation tasks. The approach is described by a metamodel which ensures extensibility. The contribution of Juan Trujillo & Sergio Lujan Mora (ER2003) is based on a UML extension with profiles for multidimensional modeling³.

Our contribution is summarized in a certain number of proposals and additions to the Vassiliadis metamodel at the conceptual level to more precisely model the reality of an ETL process, Proposal of a metamodel for the logical level in order to visualize at a very high level the different elements as well as their relationships.

The first point of this article will present the ETL process in general as well as the problems it poses in the context of a decision-making project. We will show why model the ETL process and what should be modeled. The second part will be devoted to Vassiliadis' approach to ETL modeling by presenting its basic concepts, principles and metamodel. We summarize, in the third part, we will present our contribution in the form of proposals in relation to Vassiliadis' approach and we end this article with a conclusion and perspectives.

¹ Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). Conceptual Modeling for ETL Processes. In Proc. ACM 5th International Workshop on Data Warehousing and OLAP (DOLAP 2002), McLean, VA, USA November 8, 2002.

² Vassiliadis, P., Quix, C., Vassiliou, Y., & Jarke, M. (2001). Data Warehouse Process Management. Information Systems, 26, 3, pp. 205-236

³ Trujillo, J., & Luján-Mora, S. (2003). A UML Based Approach for Modeling ETL Processes in Data Warehouses. In Proceedings of 22nd International Conference on Conceptual Modeling (ER 2003), pp. 307-320, Chicago, IL, USA, October 13-16, 2003

2. ETL Process

An ETL (Extract-Transform-Load) process is used to extract data from various heterogeneous sources to be prepared and loaded into the data warehouse. The difficulty of the ETL process lies in the diversity of the data and their heterogeneity⁴.

Figure 1 describes the environment of an ETL process: the lower layer presents the static part, i.e. the source data, the Data Staging Area (DSA) where the transformation tasks take place and the data warehouse (DW). In the upper layer, the three ETL phases are represented, namely extraction (Extract), transformation (Transform & Clean) and loading (Load).

In order to master this complexity, designers prefer to shed light on this ETL process before tackling the physical implementation.

Modeling the ETL process at the conceptual and logical levels is one of the solutions that contribute to controlling the complexity and hazards of the ETL process. The modeling must formalize all the elements of an ETL process in order to understand the data circuit from the source systems to the warehouse. Mapping data at different levels is a very important aspect to understand the flow of data.

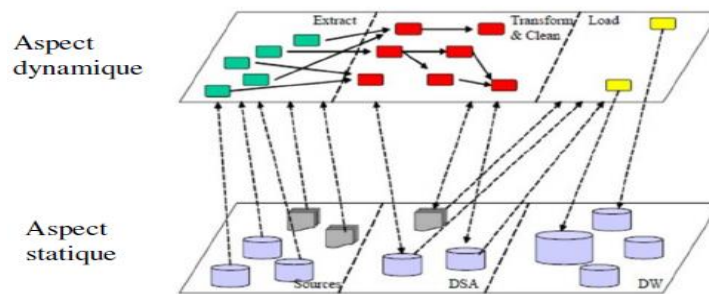


FIG.1.1. Environment of an ETL process. Panos Vassiliadis et al. (Dolap2002)

3. Vassiliadis' approach

It is interesting to visualize the static (data) and dynamic (ETL tasks) aspects at the conceptual, logical and physical levels. The conceptual level consists of giving a very general idea of the environment of the decision-making project, namely: the needs of users in terms of analyzes (measures and dimensions), the sources available, transformations to be made to the data.

The transition to the logical level consists of integrating other details such as the execution sequence of activities, different data schemas relating to an activity (inputs, outputs, parameters, rejected data). The physical level which completely describes the ETL process must specify the environment under which the process will run: OS, nature of the source systems, hardware infrastructure as well as the different user profiles. To shed light on transformation tasks, the approach represents the activity at a very fine level of granularity: the attributes. The latter are valued and modeled as entities in their own right and participate in the representation of the details of the transformation tasks.

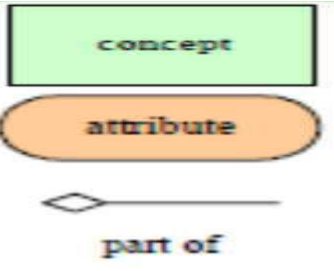
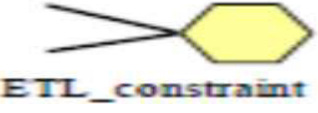


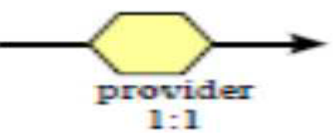

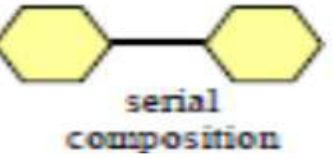

3.1. Conceptual modeling

The formalism proposed by Vassiliadis makes it possible to represent the different objects manipulated in the ETL process as well as the associations linking these objects (mapping) to those of the data warehouse. Figure 3 shows an example of an ETL process relating to the management of an academic establishment.

The merchandise (course in designation) and education (category) sources are candidates for storage. Currently, it is the first source that is selected (active candidate).

The data extracted from this source is copied into the S1.Client concept to carry out the necessary transformations before loading it into the warehouse (DW.Effective). The diagram shows how the attributes of the S1.CLIENT concept are processed at the level of the transformations SK (serrogate key), NN (not null), fl (year() function), aggregation γ (count()). PK being a constraint (Primary key) on the Year, designation and Category data of the warehouse.

⁴ Simitsis, A. (2005). Mapping conceptual to logical models for ETL processes. In Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP (DOLAP 2005), pp.: 67-76 Bremen, Germany, November 4-5, 2005

SYMBOLS	DESIGNATION
	<p>Represents an entity in the data source, DSA, or ED</p> <p>As in E/R modeling, they make it possible to define the concepts</p> <p>This association makes it possible to link the concept to its attributes</p>
	<p>Allows you to express certain constraints on the content of the ED through the attributes thereof (PK, FK, NOT NULL, etc.)</p>
	<p>Abstraction of a piece or a complete module of code executing a task</p> <p>ETL: (1) data cleaning/filtering (as a violation of the constraint PK/FK, (2) data transformations (such as aggregation).</p>
	<p>Allows you to explain design choices, specify semantics or a constraint to be checked in real time (execution time, events, errors, ...)</p>
	<p>Represent the mapping between source data (input) and data from the ED (output) via a transformation. The simple case (1:1) represents the case where a source data (input) gives output, through a transformation, to a ED data (output).</p>
	<p>The general case (N:M) represents the case where a set of transformed source data (inputs) will give rise to several data from the ED (outputs)</p>
	<p>Allows you to model the case where in a "Provider" association the data go through several transformations before giving birth to the data of the ED.</p>
	<p>Allows you to specify the candidate data sources for feeding the ED by highlighting the active source. One of the sources can be used in same time (XOR)</p>

TAB. 1 – Vassiliadis formalism for modeling at the conceptual level

For the formalism to be generic, Vassiliadis et al. proposed a metamodel (figure 4) grouping together all the elements, in the form of metaclasses, that can intervene in a process.

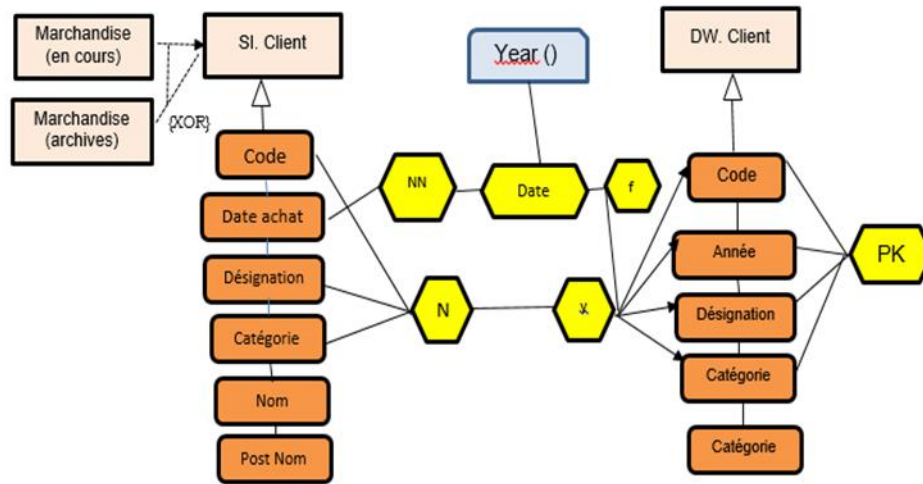


FIG. 3 – Example of an ETL process relating to a merchandise sales establishment

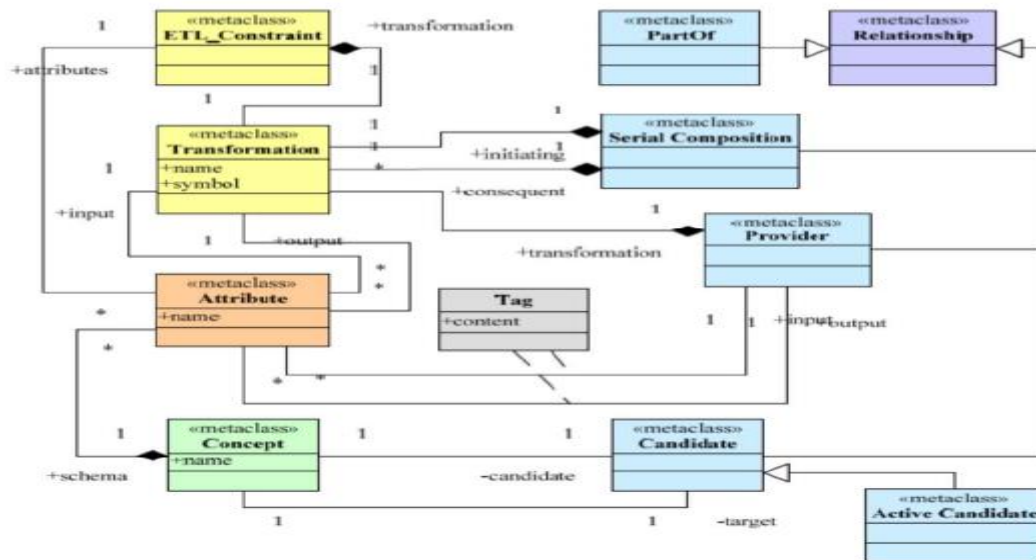


FIG. 4 – Metamodel proposed for modeling at the conceptual level

3.2 Logic modeling

The logical model, also called an architecture graph, captures data flows from sources to the data warehouse through a sequence of synchronized activities responsible for preparing the data records. This involves specifying the type of data used during processing, the inputs/outputs, parameters of each activity, the sequence of activities.

The architecture graph has vertices and edges. There are several types of vertices:

a) Elementary entities

- DataTypes: Each data type T is characterized by a name and a domain,
- Attributes: Attributes are characterized by a name and a data type.
- Schema: is a finite list of attributes. Any entity characterized by one or more schemas is called a structured entity.

b) RecordSet: A record is defined as the instantiation of a schema to a list of values belonging to the domains of the respective attributes to the schema.

c) Function: It is assumed the existence of a finite set of function types. A function type has a name, a finite list of parameter data types, and a single return data type.

d) Activity: Activities are considered as logical abstractions representing parts or modules of complete codes. They are represented by an LDL language which, on the one hand, defines the source code of an activity and on the other hand avoids the treatment of the specificities of a particular language.

An activity is formally described by a name (Name), input schema (Input Schemata), output schema (Output Schema), rejections schema (Rejections Schema), list of parameters (Parameter List), operational semantics of the outputs (Output Operational Semantics), Rejection Operational Semantics.

The different types of relationships (edges) constituting an architectural graph are:

- PartOf: Relates attributes and parameters to the activities, records or functions to which they belong.
- Instance-Of: Allows you to capture information about the typing of attributes and functions.
- Regulator: This relationship is defined between the parameters of an activity and the terms (attributes or constants) which feed this activity.
- Provider: This relationship captures the passage of data between Providers and Consumers through the Provider relationship between the attributes of the schemas concerned.
- Derived provider: Special case of the Provider relationship. This relationship is used when output attributes are generated by the composition of input attributes and activity parameters.

Figure 5 presents a process diagram at the logical level of the example of a merchandise sales establishment:

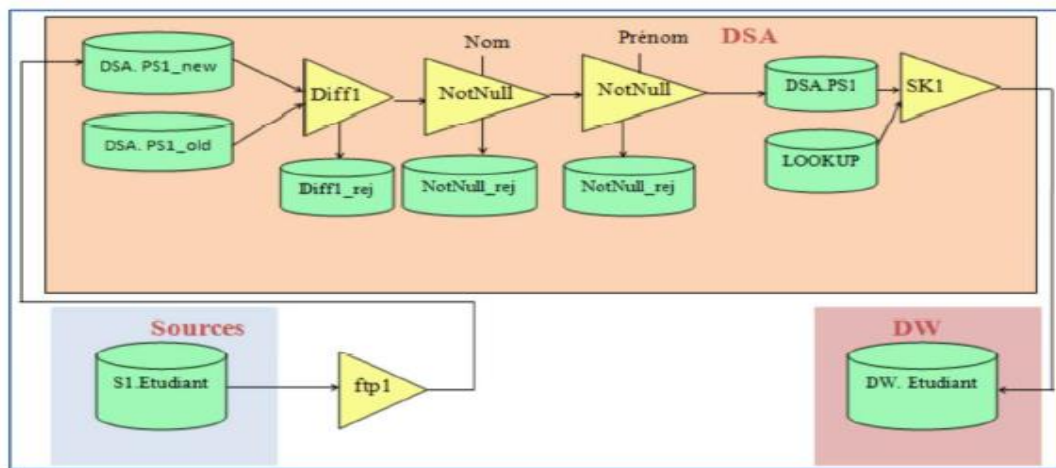


FIG. 5 – Example of the ETL process at the logical level relating to an establishment University

4. Contribution

An analysis of Vassiliadis' approach allowed us to understand the link between the different parts, to highlight certain implicit aspects and to make some improvements. At the conceptual level, we proposed some modifications to the metamodel with the aim of making it finer and more exhaustive.

For the logical level, we proposed a metamodel which describes, in a similar way to that of the conceptual level, the elements manipulated at the level of a logical model. Here is a summary of our contribution:

1) By analyzing the environment of an ETL process (figure 1) and the notations proposed for modeling an ETL process (table 1), we discovered the link not explained in Vassiliadis' papers. We will be able to deduce and delimit the different phases (data sources, extraction, DSA, transformation, data warehouse) from the diagram of a process as shown in the example in Figure 6.

2) In the Vassiliadis metamodel, only part of the source schemas, DSAs and data warehouses is represented, i.e. the data concepts necessary for analysis needs devoid of the relationships between them. By adding a reflexive association at the level of the "Concept" metaclass with "Attribute" as an associative metaclass to represent the attribute ensuring the link (Foreign Key), we will have represented all the content of the data schemas. For the warehouse in particular, we will have the star diagram

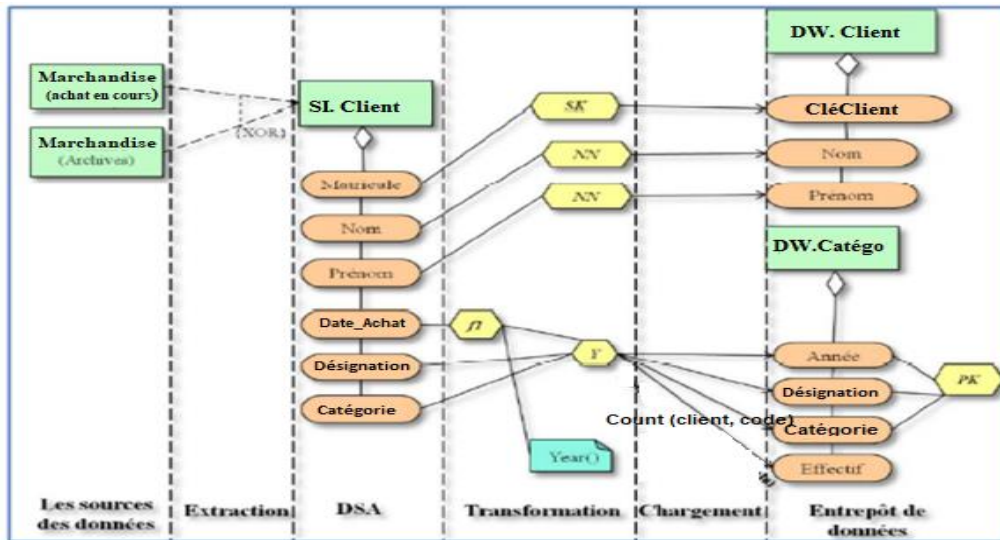


FIG. 6 – Delineation of the different stages in the process diagram

3) In the metamodel, the Cardinality (1:1) of the “candidate” metaclass with “concept” does not naturally represent the reality of the relationship. A “candidate” relationship has several source candidates and a single target candidate. To express this, we then propose to put a cardinality (1..*) in the candidate association instead of '1'. If we denote R as the candidate relation, C1, C2, C3 as the candidate concepts and C4 as the target concept, the instances of the two associations Candidate and Target are as follows:

<i>Relation</i>	<i>Concept candidat</i>	<i>Relation</i>	<i>Concept cible</i>
R	C1	R	C4
R	C2		
R	C3		

TAB. 3 – Instances of Candidate and Target relationships with cardinality '1'

4) To give it more meaning in the metamodel, the "Part-Of" relationship represented as a metaclass must be associated with the "Concept" metaclass and the "Attribute" metaclass to represent that such attribute is a constituent of such concept.

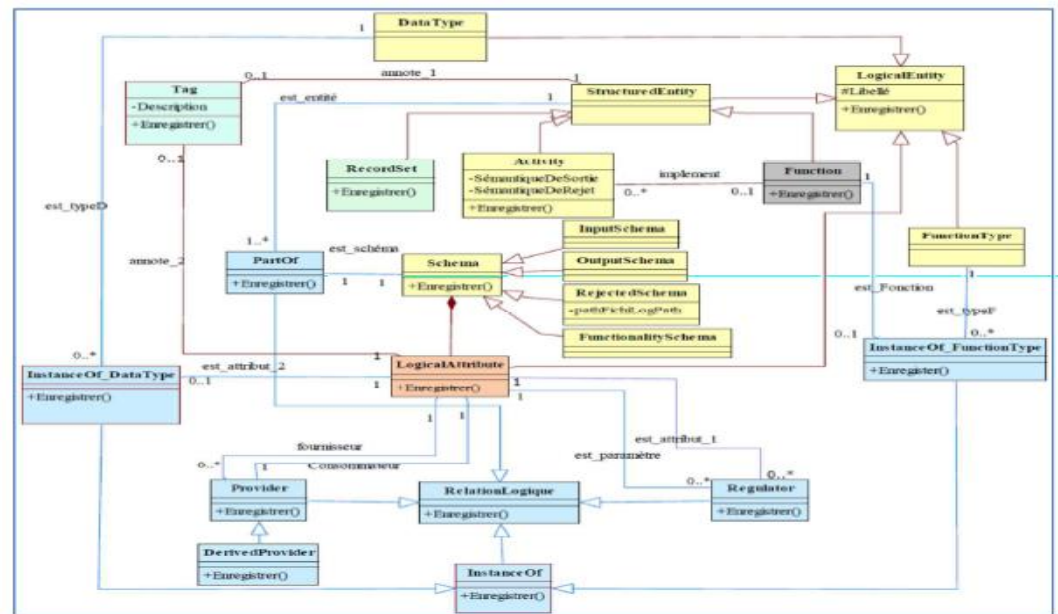


FIG. 8 – Proposed metamodel for the logical level

5) The "Tag" metaclass as designed does not allow the use of a note anywhere in the conceptual model. It would be more interesting to link it to the "Relationship" metaclass (in order to involve all relations), attributes, concepts and transformations.

6) For the logical level, we proposed a metamodel (see figure 8).

6. Conclusion and outlook

The article highlighted the importance of the ETL process as part of a decision-making project. Modeling this process is a means that facilitates understanding of the operating details of ETL and then makes it possible to control its complexity and anticipate possible problems and risks before implementing or configuring the ETL tool.

Vassiliadis' approach is one of the most interesting in this field. Studying this approach allowed us to discover very interesting aspects in ETL processes.

Our contribution to the approach is summarized in a set of remarks and proposals to further refine the metamodel and best reflect the reality of an ETL process.

The physical level, which models the ETL process in an environment characterized by the technical means (equipment and software environment) and the user profiles necessary for the proper functioning of the process, also deserves a good analysis, proposal of a formalism, of a metamodel. The Conceptual-Logic and Logic-Physical mapping is a very interesting aspect which deserves in-depth work in order to ensure the generation of logical and physical models in a semi-automatic manner but with less effort for the designer.

Bibliographic references

1. Vassiliadis, P., Simitsis, A., & Skiadopoulos, S., *Conceptual Modeling for ETL Processes*. In *Proc. ACM 5th International Workshop on Data Warehousing and OLAP*, McLean, VA, USA November 8, 2002.
2. Vassiliadis, P., Quix, C., Vassiliou, Y., & Jarke, M., *Data Warehouse Process Management*. *Information Systems*, 26, 3, pp. 205-236, 2001.
3. Trujillo, J., & Luján-Mora, S. (2003). *A UML Based Approach for Modeling ETL Processes in Data Warehouses*. In *Proceedings of 22nd International Conference on Conceptual Modeling (ER 2003)*, pp. 307-320, Chicago, IL, USA, October 13-16, 2003
4. Simitsis, A., *Mapping conceptual to logical models for ETL processes*. In *Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP*, pp.: 67-76 Bremen, Germany, November 4-5, 2005.