KANTARA: a Framework to Reduce ETL Cost and Complexity

Ahmed Kabiri [#1], Dalila Chiadmi[#2]
[#] SIR Laboratory, Mohammadia Engineering School,
MOHAMMED V UNIVERSITY IN RABAT
[1] ahmed.kabiri@gmail.com
[2] chiadmi@emi.ac.ma

***Abstract*—Data warehouse (DW) has been widely recognized as an effective solution for integrating diverse sources. In this environment, Extraction Transformation Loading (ETL) processes constitute the integration layer. They perform data extraction, their cleaning, their conforming and loading into the target. It is widely recognized that building ETL processes, in a data warehouse project, are expensive regarding time and money. In spite of the abundance of works and proposals, around ETL, there is no proposal towards a global approach taking in account the cost and the complexity of ETL processes. In this context, we propose in this paper, a framework called KANTARA for managing ETL processes based on our experience in real world.**

Keyword - ETL, Data warehouse Refreshment, ETL Modeling, and KANTARA.

I. INTRODUCTION

In current economic context characterized by globalization (global competition), any organization wishing to survive should be responsive, decisive and effective. Indeed, in this context that is increasingly uncertain and complex, enterprises as well as States need support for making decisions and for developing strategies which will help themes to meet theirs objectives. To meet this need, organizations (enterprises or States) build data warehouse systems. Indeed, data warehouse defined by Inmon [1] as "collection of integrated, subject-oriented databases designated to support the decision making process" aims to improve decision process by supplying unique access to several sources.

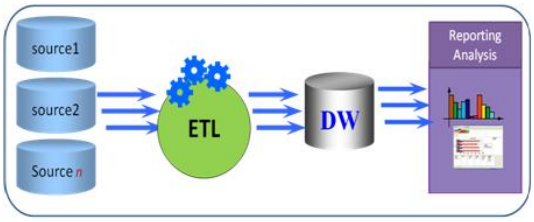

Fig. 1. A Data Warehouse Environment

Fig. 1 illustrates classical architecture of data warehouse system. It consists of a set of cooperating components that we present in the following:

- Datawarehouse (***DW***) component is the central element that saves the information produced by the Data warehouse system. In general, DW is implemented via a database management system (DBMS) such as Oracle [2] or SQL Server [3]
- "***Reporting and analysis***" component allows end users accessing DW and displaying data in multiple formats combining text, tables, graphs, etc. Tools such as Business Object [4] or Cognos [5] are used at this level
- ***Sources*** represent the supplier of Data warehouse system. They correspond to data stored by business applications (operational applications). The sources supply data in several storage formats such as flat files, databases, XML files, etc. Finally, sources will be external where data coming from external entities are used in data warehousing process.
- ***ETL*** component (or ETL processes) performs the critical mission of feeding and refreshing DW (the target) with data from sources. It is the integration layer of Data warehouse system performing three tasks. Firstly, it pulls data from several sources; secondly it applies transformations to data extracted in previous step then, thirdly, it loads data to DW. Finally, ETL component can be seen as a mediator (in linguistic sense) between sources and DW.

It is widely recognized that establishing a Data warehouse system is both costly and complex [1], [6], [7], [8], [9]. More exactly, it is the integration layer that is the most expensive and most complex. Indeed, ETL processes consume up to 70% of the budget and the time needed to set up a Data Warehouse [6]. Stated differently,

accomplishing ETL projects is very complex and time consuming. Therefore, as Simitsis [10] states "there are strong motivations for making ETL engagements less expensive and faster". Finally, let recall that ETL is a key success factor of data warehouse projects [6]. For these reasons, we consider and focus on ETL processes in the following.

In spite of the abundance of works and proposals, around ETL, there is no proposal towards a global approach. Indeed (as we can see in related works) the scientific community has enriched ETL field with several works dealing with several issues like conceptual modelling or logical modelling or performance issue. Also, one can notes that some issues are less dealt with like maintenance and tests [11]. Finally, all these issues are addressed in a separate rather than in a global approach.

The interest of a global approach is providing more integrity between steps making the life cycle of an ETL project. In others words, a global approach leads to more automation and less manual tasks. Consequently, the cost and complexity of ETL projects will be reduced. Towards this goal, we propose in this paper, a framework called KANTARA. Therefore our contributions can be outlined as follows:

- Presenting KANTARA framework our solution for managing ETL processes.

The remaining of this paper is organized as follows. Section 2 presents related works while section 3 is dedicated to our solution where we present KANTARA architecture. We conclude and present our future works in section 4.

II. RELATED WORKS

A plethora of commercial ETL tools such [12], [13] as well as a set of open sources tools such [14], [15] exist. Both of them offer graphical interface to build ETL system. However, they do not supply deep support to conceive ETL processes. They focus on technical and running aspects more than designing ones. On the other side, research community enriches the field of conceptual modelling of ETL with several proposals. During DOLAP 2002 conference, authors of [16] present an approach for conceptual modelling of ETL processes. Authors consider the following two points. At the beginning of an ETL project, the designer needs:

1. To analyse the structure and the content of sources.
2. Define the mapping rules between sources and targets.

To meet these needs, the authors developed a model based on a meta-model for designing ETL processes. In addition, this model is accompanied by a specific graphical notation where a range of activities (components or objects) frequently used by the designer is defined.

Semantic web technologies are used in designing ETL processes. Indeed, authors present in [17] and [18] an ontology based approach aiming to get mapping rules between sources and targets of an ETL process. Thus authors suggest constructing ontology via OWL (web ontology language) describing application domain. Then sources and target are annotated via the constructed ontology. Finally, a reasoning technique is used in order to identify correspondences and conflicts between sources and targets. Another interesting work related to the design of ETL processes and based on RDF (Resource Description Framework) and OWL (Web Ontology Language), two web technologies, is detailed in [19]. This work suggests an end to end method; from restitution layer until the creation of the ETL layer. The basic idea of this proposal is to convert the data sources to a RDF file conforms to a generic ontology called OLAP ontology (this ontology is constructed). Then the target tables are populated with data extracted by queries generated using OLAP ontology.

In 2003, Trujillo [20] proposes an UML based approach for the design of ETL processes. Authors conceive an ETL process as class diagram. Thus, the atomic element of an ETL process is represented as classes (according to UML concept: class and attributes) while the interconnection between these classes is defined by the UML dependencies. Let note that authors have decided to restrict their model to ten types of commonly used ETL activities such as aggregation, Conversion, Filter and Join where each element is associated with a specific graphic icon. Finally, for data description the model uses the attributes of the classes. The strength of this model is using a standard modelling language that is UML. But its main drawback is mapping management: mapping between classes representing an ETL processes is not managed as class attributes. This limitation comes from the UML itself which not manage the mappings between attributes of classes included in a class diagram.

The scientific community has enriched the conceptual modelling of ETL area with several approaches presented above. These proposals differ with respect to the formalism used. But they have the same drawback that is the lack of support for managing the risk of change. In other words, these proposals were not accompanied by approaches for the maintenance of the models resulting from the use of these approaches. Therefore other independent approaches have emerged like the one presented in [21]. In this work, the authors abstract all parts of ETL process, as a sequence of enriched queries modelled by graphs. The graphs are annotated (by the designer) with actions to perform in response to change event. An algorithm to readapt the graph, given an evolution event in sources, is supplied. However, this approach is difficult to implement, because of enormous amount of additional information required in nontrivial cases [22]

The works presented above, are interesting but they don't offer a global approach dealing with ETL. In other words, there is no proposal toward global approach taking in account the cost and the complexity of ETL processes. In this context, we propose in next section, our framework called KANTARA for managing ETL processes.

III. Our solution for Managing ETL Processes

KANTARA (framework for managing extraction transformation load processes) is our framework for managing ETL processes.

Actually, KANTARA adopts components architecture as shown in Fig 2. We distinguish three levels:

1. The level "Model Edition" manages ETL processes by supplying a set of features around ETL processes. This level is based on six components (see Fig 2). Finally the main outputs of this level are conceptual models of ETL processes. These models are independents of any platform (PIM models).
2. The level "Transformations" performs a set of transformations on the PIM models derived from the upper level (Model Edition). The output of "Transformations" level is a code corresponding to PIM models. This code, which we call job within the ETL, is transferred to the next level "Execution & Integration" for execution.
3. The level "Execution & Integration" executes the code produced by the previous level. It does not have an output. Otherwise, its mission is to combine and physically transfer data from sources to targets by running the generated code (jobs).

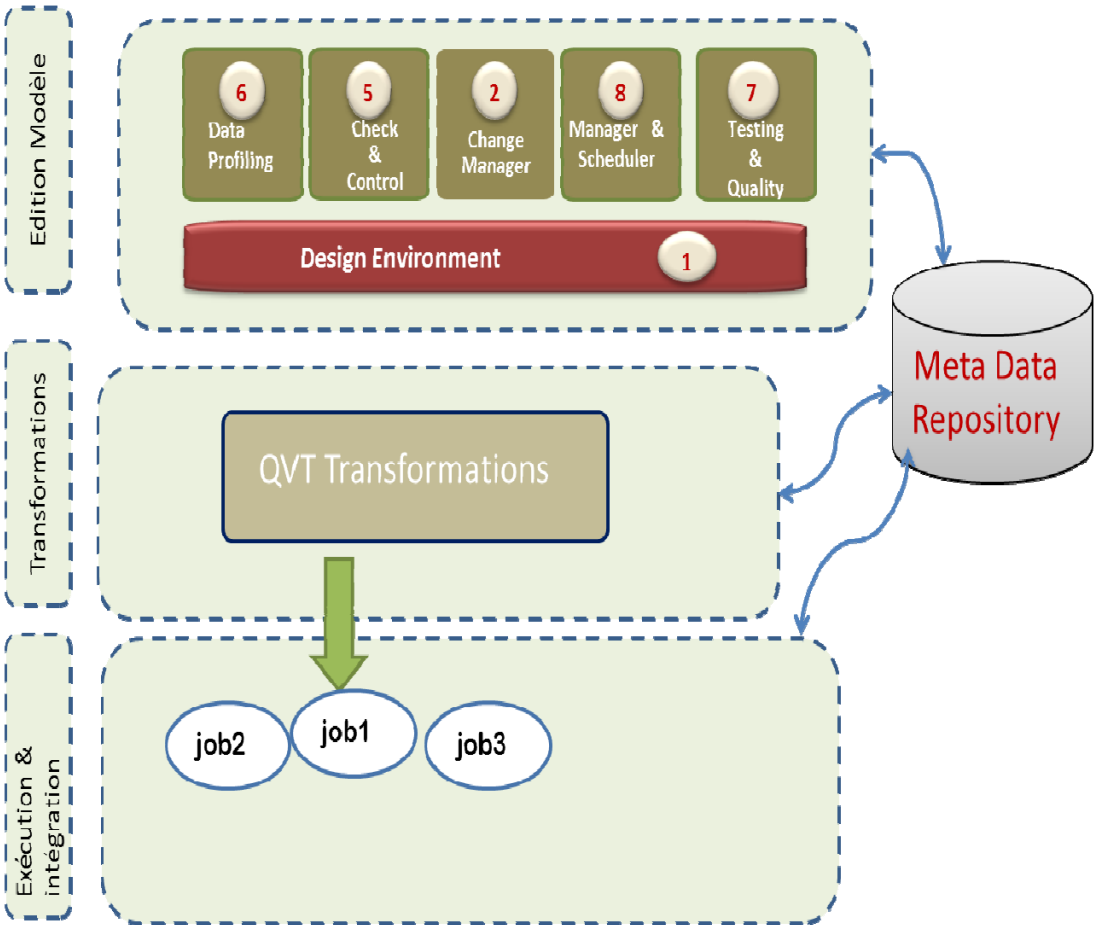

Fig. 2. Overview of KANTARA

The component "Meta Data Repository" is the repository of KANTARA metadata. This component is transverse to all levels since it communicates with all their sub-components. It manages and stores metadata produced and defined by other parts. For example, it contains information about sources, targets (schema, type of sources, etc.), mapping rules, etc.

Regarding end user (designer), the visible level is "***Model edition***" which consists of six components representing features where KANTARA adds value. These features are the following components:

1. ***Data Profiling (DP):*** The mission of this component is to help the designer exploring the sources being part of an ETL process in order to understand data and find out their relationships. Thus its output is metadata about data, which states about the accuracy of data. The delivery of this module is useful for designer, since it helps him to collect and specify necessary information to identify and trace the road to the targets. Stated differently, ***DP*** allows the designer both to assess the quality of data sources and to define (specify) necessary transformations to apply on data with. Fixing errors of data is an example of theses necessary transformations. Indeed, data belonging to sources may be erroneous and invalid (syntaxes errors semantic errors…).
2. ***Design Environment (DENV):*** is a working area where the designer makes models of ETL processes. Having requirements and ***DP*** information (previous component), designer builds models of ETL processes. These models are independent of platform, in one hand, and should be interpretable by other components in other hand. We give further explanations about the mission and the requirements of this component in next sections.
3. ***Checks and Control (CC):*** This component is in charge to check and to control the models made by the designer (in design environment). Thus this component improves the quality of the code generated by KANTARA and allows to increase productivity since that the detection and the correction of errors as soon as possible saves time of rework. Time of rework means time spent to investigate and correct errors encountered in the context of the development and deployment of an ETL project. To perform its mission, ***CC*** component parses models generated by ***DENV*** component to detect errors of PIM models. Finally, it is based on «repository of rules" that contains a list of rules to apply to ETL models. Thus the mission of ***CC*** module is to read the models to check whether the rules defined at component "repository of rules" are met or not.
4. ***Change Manager (CM):*** The main mission of this module is to make easy the task of maintaining ETL processes. It intends to sustain and support designer to achieve new requirements dealing with adding or deleting even fields or business rules. Therefore this component manages the risk of changes by identifying the impact of evolution either in sources or in targets. In other words, ***CM*** allows within KANTARA to identify and to record the impact of a given change event. Finally, let note that with this feature the cost and the time of maintenance task will be decreased.
5. ***Manager & Scheduler (MS):*** This component manages, in a broad sense, ETL processes within KANTARA. It complements other components and offers features such as scheduling. Particularly and in connection with the overall objective of KANTARA, which is reducing the cost of ETL processes, ***MS*** component enhance reuse. Indeed, reusing existing models or scripts when developing new ones saves the effort required for their design or their development. Thus ***MS*** supplies a set of operators like import, export and compare that the designer will use to make new ETL models based on existing ones instead of making theme from scratch.
6. ***Testing & Quality (TQ):*** This component manages within KANTARA, testing of ETL processes. In other words, this component helps the designer testing and validating the models and the code created within KANTARA. Consequently, it improves the deliverables of KANTARA by improving the testing process of ETL processes. To this end, ***TQ*** parses ETL models created by ***DENV*** component in order to generate data to be loaded both in sources and targets. We stress that ***TQ*** generates data taking in account the content of ETL models (particularly business rules). It does not generate data randomly as commercial tools or open source tools did.

Having all these features, the cost and time needed to set up an ETL project will be decreased besides the reduction of complexity associated with ETL processes. Indeed, with such features, manual intervention will be decreased and automatic handling of tasks will be increased.

IV. CONCLUSION

It is widely recognized that building ETL processes, in a data warehouse project, are expensive regarding time and money. In order to overcome this situation, a new approach becomes necessary. In this perspective, several works have dealt with an aspect of ETL processes. However there is no proposal towards global approach for managing ETL processes.

Towards the goal expressed above, we have presented, in this paper, KANTARA our solution for managing ETL processes. KANTARA architecture is layered on 3 levels namely *Model Edition, Transformations, Execution & Integration*. Especially, we have described briefly the roles of components constituting *Model Edition*.

KANTARA is an ongoing project. We have completed the first phase where KANTARA architecture is defined. In the future, we would refine and instantiate this architecture.

REFERENCES

[1] W. Inmon, D Strauss and G. Neushloss, *DW 2.0: The Architecture for the next generation of data warehousing*, 1rst ed., Morgan Kaufman, 2007.

[2] (2015) Oracle website. [Online]. Available: http://www.oracle.com.

[3] (2015) Microsoft website. [Online]. Available: http://www.microsoft.com

[4] (2015) SAP website. [Online]. Available: http://www.sap.com/index.html

[5] (2015) IBM Cognos website. [Online]. Available: http://www-01.ibm.com/software/analytics/cognos/

[6] R. Kimball and J. Caserta, *the Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley Publishing, 2004

[7] M. Golfarelli, Ed., *Encyclopedia of Database Systems: Data Warehouse Life-Cycle and Design*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, pp. 658-664, 2009.

[8] (2015) Gartner Magic Quadrant for Data Integration Tools website. [Online]. Available: http://www.gartner.com

[9] Worldwide Business Analytics Software 2013– 2017 Forecast and 2012 VendorShares. http://idcdocserv.com/241689e_sas

[10] A. Simitsis, K. Wilkinson, M. Castellanos and U. Dayal, "QoX-Driven ETL Design: Reducing the Cost of ETL Consulting Engagements," in *Proc. SIGMOD*, pp. 953-960, 2009.

[11] A. Kabiri and D. Chiadmi, "Survey on ETL Processes" *Journal of Theoretical and Applied Information Technology*, vol. 54, pp. 219–229, August. 2013.

[12] (2015) IBM InfoSphere DataStage website. [Online]. Available: http://www-01.ibm.com/software/data/infosphere/datastage/

[13] (2015) Informatica website. [Online]. Available: http://www.informatica.com/FR/Pages/index.aspx

[14] (2015) Talend website. [Online]. Available: http://fr.talend.com/about-us.

[15] (2015) Pentaho website. [Online]. Available: http://www.pentaho.com/

[16] P. Vassiliadis, A. Simitsis and S. Skiadopoulos, "Conceptual modelling for ETL processes" in *Proc. 5th ACM International Workshop on Data Warehousing and OLAP (DOLAP2002)*, pp. 14-21, 2002.

[17] D. Skoutas and A. Simitsis, "Designing ETL processes using semantic web technologies" in Proc. 9th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2006), pp 67-74, 2006.

[18] D. Skoutas and A. Simitsis, "Ontology-based conceptual design of ETL processes for both structured and semi-structured data" *International Journal on Semantic Web and Information Systems.*, vol. 3, pp. 1–24, Oct. 2007.

[19] M. Niinimaki and T. Niemi, Ed., *Journal on Data Semantics XIII: An ETL Process for OLAP Using RDF/OWL Ontologies*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2009, vol. 5530.

[20] J. Trujillo and S. Lujan-Mora, "A UML Based Approach for Modelling ETL Processes in Data Warehouses" in *Proc. of ER*, pp 307-320, 2003.

[21] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, "What-If Analysis for Data Warehouse Evolution", *in Proc. DaWaK*, pp. 23–33, 2007.

[22] A. Dolnik, "ETL evolution from data sources to data warehouse using mediator data storage", *in Proc. MANAGING EVOLUTION OF DATA WAREHOUSES Workshop (MEDWa), pp.249-257, 2009.*