ISSN 2250-3153

Application of ETL Tools in Business Intelligence

Nitin Anand

Ambedkar Institute of Advanced Communications Technologies and Research, New Delhi

***Abstract*- Business** intelligence (BI) is considered to have a high impact on businesses. Research activity has risen in the last years. An important part of BI systems is a well performing implementation of the Extract, Transform, and Load (ETL) process. In typical BI projects, implementing the ETL process can be the task with the greatest effort. Here, set of generic metamodel constructs with a palette of frequently used ETL activities, is specialized , which are called templates.

***Index Terms*-** Business Intelligence, BI, BPMN Extract, Transform, and Load, ETL Design, Model-Driven Engineering, OMB

I. INTRODUCTION

Business intelligence (BI) has gained wide recognition in the last years. It also got high business impact and is seen as a \key enabler for increasing value and performance" [1].

Unsurprisingly, the progress of BI is monitored by management and IT consultants [2]. It is recognized as having a high relevance for the profit of businesses [3]. Companies that do not operate business intelligence systems could get permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. It is agreed that a strategic business intelligence approach will be needed [4].

At the same time, Business intelligence is a rather new discipline with a lot of research activity. Even though the term has been coined in 1958 [5]1, the number of published papers has risen considerably in the last few years. The rapid progress has also brought a high level of heterogeneity [6]; this causes both problems for businesses and offers research opportunities. It is possible to grasp the current state of BI [1] and practitioner's literature tries to lay out a roadmap on how to implement BI in a company [7]. There is no reliable roadmap for BI progress, though.

One important component of BI is the Extract, Transform, and Load (ETL) process. It describes the gathering of data from various sources (extract), its modification to match a desired state (transformation) and its import into a database or data warehouse (load). ETL processes take up to 80% of the effort in BI projects [8]. A high performance is thereby vital to be able to process large amounts of data and to have a up-to-date database. The term ETL is known for a while [9] and the relevant market is already divided by a number of major players [10]. While open source tools are competitive with commercial software in many areas, there is little published work on open source ETL tools (see Sect. 3) even though the practical relevance is given. Open Source BI tools recently gained more attention [2] and start to compete with commercial solutions [11]. Therefore, we present an analysis of open source ETL tools. We especially focus on the performance, which is a main criterion for ETL processes. Well performing open source tools could be alternatives to proprietary solutions. The contributions of our paper are the summary of the background, the considerations on how to design a study for performance comparison and the actual results including our decision advise. Moreover, with a general discussion of the study we contribute to the body of BI knowledge.

II. BACKGROUND

Speaking of Business intelligence, many approached and technologies are addressed. We will give a brief introduction.

The core BI consists of Online Analytical Processing (OLAP) and Enterprise Information Systems (EIS).These components directly support decision making. In theory,a BI application can be built with these two components. Analysis-oriented BI comprises various concepts and applications that allow model-based or method-based analyses of data. Besides the two core components, it can include ad hoc reporting and data or text mining components as well as advanced functionality such as an analytic Customer Relationship Management (CRM).The view of full BI not only extends the functionality by standard reporting but includes two components required for any powerful BI solution: a Data Warehouse to store the data for analysis, and ETL tools and processes to both make data accessible and to feed it into the data warehouse beforehand. An overview of the BI concepts and technologies with regard to their process-oriented aim and their orientation can be seen in Fig. 1.

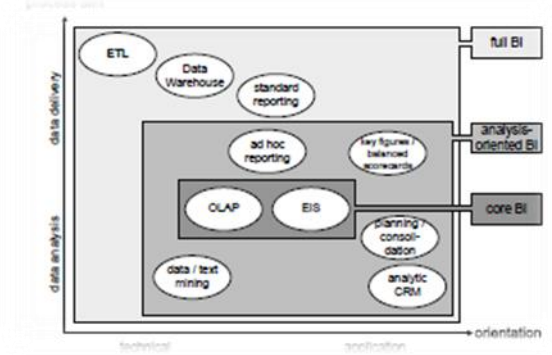

Figure 1: Overview of business intelligence components (inspired by [12])

The ETL process is made up of its three core components:

Extraction: In the first phase, data is extracted of heterogeneous operative systems. The amount of data is reduced by omitting any non-relevant data sets. Extraction must not negatively affect the performance of productive systems. It runs as a background task or is executed at times of low activity (e.g. during the night).

Transformation: Any transformation needed to provide data that can be interpreted in business terms is done in the second step. Data sets are cleaned with regard to their data quality. Eventually, they are converted to the scheme of the target database and consolidated.

Loading: Finally, the actual loading of data into the data warehouse has to be done. The Initial Load which generally is not time-critical is distinguished from the Incremental Load. Whereas the first phase affected productive systems, loading can have an immense effect on the data warehouse. This especially has to be taken into consideration with regard to the complex task of updating currently stored data sets. In general, incremental loading is a critical task. ETL processes can either be run in batch mode or real time. Batch jobs typically are run periodically. If intervals become as short as hours or even minutes only, these processes are called near real time. Of course, the above phases and especially their near real time execution is extremely complex.benchmark.

III. RELATED WORK

Business intelligence is no new emergence. Due to the high business impact of BI and the importance of open source software, there are some studies on ETL. However, they are typically done by business consulting forms such as Gartner [13] and capture the market situation.

Existing approaches for developing ETL process highlight two main axes: (i) designing ETL processes independently of a specifc vendor, and then (ii) implementing into an executable code tailored to a certain technology.

ETLprocesses can be adequately designed by a workow language. Furthermore, other approaches [14, 15] recommend the use of ontologies for describing data sources and targets, thus providing the necessary information for solving an adequate

ETL process design by inference. Other approaches as [16] apply graph operation rules in order to guide the ETL design steps, also by using ontologies. Unfortunately, building such ontologies is a tedious task since it requires a high correctness and a blow-by-blow description of the data stores.

Therefore, we advocate the use of a rich workow language that provides various enhancements for the designer work without requiring the definition of any ontology.

On the other hand, the implementation of the ETL design has been discussed from many points of view. For example, an attempt has been concerned about the optimization of the physical ETL design through a set of algorithms [17]. Also, a UML-based physical modeling of the ETL processes was introduced by [18]. This approach formalizes the data storage logical structure, and the ETL hardware and software configurations. Although both works deal with interesting adjacent issues about the implementation topic, they mainly miss to produce a code for executing ETL processes. Paradoxically, an exclusive ETL programming approach using the Python language has been claimed by [19]. Yet,this approach omits to supply an independent-vendor design which decreases the reusability and easy-of-use of provided framework.

IV. MODEL-DRIVEN FRAMEWORK

The present work addresses the ETL process development using aModel-Driven Development (MDD) approach. In this section, we concretely show how this approach allows to organize the various components of this framework in order to efficiently perform the design and implementation phases of the ETL process development.[20]

4.1 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)20): the Model Instance layer (M0), the Model layer (M1), the Meta-Modellayer (M2), and the Meta-Meta-Model layer (M3).

The Model Instance layer (M0) is a representation of thereal-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-specifc 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 ions, 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.

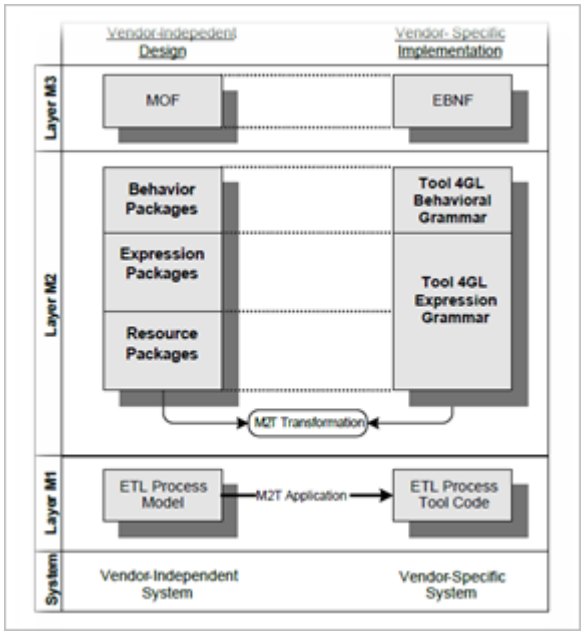

Fig 2: MDD layers for the ETL development framework

V. TRANSFORMATIONS

Transformations are used for automatically generating the ETL code from an input ETL model. In this section, we define the transformations as matching statements between the metamodel and the grammar at the M2 layer, in order to be executed at the M1 layer. At the same time, we show how the abstract patterns for such transformations provide useful guidance when developing code generators for new ETL tools. Our approach uses OMG's model-to-text (M2T) standard for expressing the transformations.

5.1 M2T Transformations

M2T organizes transformations in a set of templates, responsible for generating code for the input model elements. Each template contains the code portions corresponding to a single metamodel concept. This code may be static or dynamic. Static code is replicated literally during the execution. Dynamic code corresponds to OCL expressions specified using the metamodel concepts. These expressions are computed on the input model during execution. In practice, the input model elements are sequentially iterated, and for each element, the corresponding template code is generated and appended to an output file. Thus, the transformations between the BPMN4ETL metamodel and the OMB grammar are performed by means of templates. However, the template code depends on target ETL tool. For this reason, we show next how an abstract level for the transformation templates can be useful, since it specifies the required templates, and describes their common processing order across ETL tools.

VI. APPLYING TRANSFORMATIONS

This section describes the Eclipse-based implementation of our framework, specifically, the code generation. It emphasizes an excerpt of the transformation code that should be applied to the ETL model in Fig 3 in order to get the corresponding OMB code. [21]

3.2.1 BPMN4ETL Model (M1)

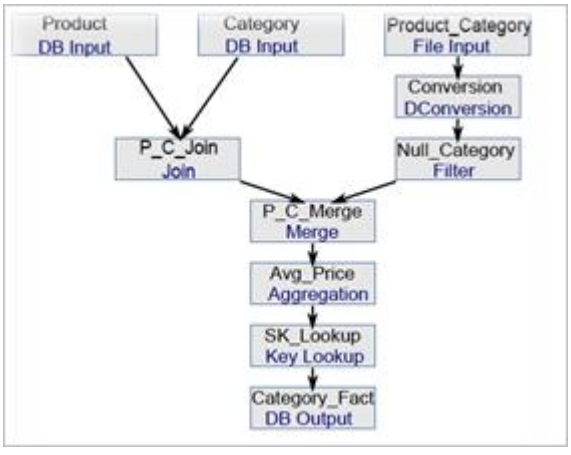

Fig3: ETL process model for the Category _Fact _load

Eclipse is a software development environment comprising an extensible plug-in system. It provides the Eclipse Modeling Framework (EMF), which comprises a number of model- driven development capabilities like modeling, inter-model transformations, and code generation. More precisely, EMF allows the definition of models and metamodels by means of the Ecore tools. In practice, metamodels are created using the Ecore meta-metamodel, which is the implementation of MOF in Eclipse. From these metamodels it is possible to generate the genmodel, which is a set of Java classes that represent each metamodel concept. The genmodel is useful for building dedicated tools, such as editors, to create and manage the models corresponding to our metamodel.

VII. CONCLUSIONS AND FUTUREWORK

Even though data warehouses are used since the early 1990s, the design and modeling of ETL processes is still accomplished in a vendor-dependent way by using tools that allow to specify them according to the schemas represented in concrete platforms. Therefore, their design, implementation and even more their maintenance must be done depending on the target platform. In this paper, we have provided, to the best of our knowledge, the first modeling approach for the specifcation of ETL processes in a vendor-independent way and the automatic generation of its corresponding code in commercial platforms. This is done thanks to the expressiveness of our BPMN-based metamodel and to the model-driven development capabilities, provided by Eclipse and Acceleo, in code generation.

Our approach considers Oracle and Microsoft as representative target implementation and execution tools, although only Oracle's solution has been described in this paper for space reasons. Moreover, it o_ers some new techniques in order to guide developers in building other code generators for new platforms. Currently, our framework covers the design and implementation phases of the ETL process development. One future work expects to extend this framework in order to handle the whole ETL development life-cycle, i.e., by involving the analysis phase as well. This work should enhance the quality of the generated code by our approach by means of: i) enhancing the transformation implementation using some existing knowledge-based techniques, and ii) by building the necessary metrics for proposing the 'best' implementation related to a certain ETL process.

REFERENCES

[1] H. J. Watson and B. H. Wixom. The current state of business intelligence. Computer, 40(9):96{99

[2] Gartner, Inc. Press release: Gartner reveals five business intelligence predictions for 2009 and beyond, January 2009

[3] S. Williams and N. Williams. The Profit Impact of Business Intelligence. Morgan Kaufmann, San Francisco, CA, 2006. Online, 2007.

[4] Z. Panian. Business intelligence in support of businessstrategy. In Proc. MCBE'06, pages 19-23, StevensPoint, 2006. WSEAS.

[5] H. P. Luhn. A business intelligence system. IBM J.Res. Dev., 2(4):314{319, 1958.

[6] Z. Panian. Expected progress in the field of business intelligence. In Proc. AIKED'09, pages 170-175,Stevens Point, 2009. WSEAS.

[7] L. T. Moss and S. Atre. Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications. Addison-Wesley Longman, Boston, MA, USA, 2003

[8] W. H. Inmon. Building the Data Warehouse. Wiley, New York, NY, USA, 3rd edition, 2002.

[9] R. Kimball, L. Reeves, W. Thornthwaite, M. Ross, and W. Thornwaite. The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses. Wiley, New York, NY, USA, 1998.

[10] Gartner, Inc. Press release: ETL magic quadrant update: A market in evolution, May 2002. Online: http://www.gartner.com/reprints/informatica/106602.html.

[11] Gartner Inc. Magic quadrant for data integration tools, September 2008. Online: New York, NY, USA, 3rd edition, 2002. http://www.sap.com/solutions/pdf/

[12] C. Dittmar and P. Gluchowski. Synergiepotenziale und Herausforderungen von Knowledge Management und Business Intelligence, pages 27{42. Springer, 2002.

[13] Gartner, Inc. Who's who in open-source business intelligence, April 2008. Online: http://www.stratebi.es/todobi/may08/whos who in opensource busin 156326.pdf.

[14] D. Skoutas and A. Simitsis. Designing ETL processes using semantic web technologies. In I. Song and P. Vassiliadis, editors, Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP, DOLAP'06, pages 67{74, Arlington, Virginia, USA, Nov. 2005. ACM Press.

[15] 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, 3(4):1{24, 2007.

[16] D. Skoutas, A. Simitsis, and T. Sellis. Ontology-driven conceptual design of ETL processes using graph transformations. In Journal on Data Semantics XIII, number 5530 in LNCS, pages 122{149. Springer, 2009.

[17] V. Tziovara, P. Vassiliadis, and A. Simitsis. Deciding the physical implementation of ETL workflows. In I. Song and T. Pedersen, editors, Proceedings of the 10th ACM International Workshop on Data Warehousing and OLAP, DOLAP'07, pages 49-56, Lisbon, Portugal, Nov. 2007. ACM Press

[18] 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.

[19] C. Thomsen and T. Pedersen. pygrametl: A powerful programming framework for extract-transform-load programmers. In Song and Zim_anyi [11], pages 49-56.

[20] 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.

[21] http://www.omg.org/spec/CWM/1.1/

AUTHORS

First Author – Nitin Anand, Ambedkar Institute of Advanced Communications Technologies and Research, New Delhi, Email:proudtobeanindiannitin@gmail.com