A UNIFIED MODEL DRIVEN METHODOLOGY FOR DATA WAREHOUSES AND ETL DESIGN

Faten Atigui, Franck Ravat, Ronan Tournier and Gilles Zurfluh
IRIT (UMR 5505), Institut de Recherche en Informatique de Toulouse
118 route de Narbonne, F-31062, Toulouse, France
University Toulouse 1 Capitole, 2 rue G. Marty, F-31042, Toulouse Cedex 9, France

Keywords: Data Warehouse, Multidimensional modelling, ETL, MDA, QVT.

Abstract: During the last few years, several frameworks have dealt with Data Warehousing (DW) design issues. Most of these frameworks provide partial answers that focus either on multidimensional (MD) modelling or on Extraction-Transformation-Loading (ETL) modelling. However, less attention has been given neither to unifying both modelling issues into a single structured framework nor to automating the warehousing process. To overcome these limits, this paper provides a generic unified and semi-automated method that integrates DW and ETL processes design. The framework is handled within the Model Driven Architecture (MDA). It (i) first helps the designer in modelling the decision-makers requirements and then (ii) generates the MD model as well as (ii) the logical and the physical models and finally (iv) generates the source code. In this approach, the transformation rules are formalized using the Query/View/Transformation (QVT) language.

1 INTRODUCTION

A Data Warehouse (DW) is a huge amount of data, often historical, used for supporting business processes within an organization (Ravat et al., 1999). The relevant data for the decision-making process are collected from data sources by means of software processes commonly known as Extraction-Transformation-Loading (ETL) processes (Vassiliadis, 2009). Extracted data are often structured according to the multidimensional (MD) paradigm that organizes information according to facts and dimensions (Kimball, 1996); (Ravat et al., 2007).

It is well recognized that the warehousing task is complex, tedious, time-consuming and often error-prone (Kimball, 1996). When building a DW, the designer deals with two major issues. The first issue addresses DW design, whereas the second addresses ETL processes design. Current frameworks provide only partial solutions that focus either on MD structures or on ETL processes. Indeed, the whole warehousing process (in charge of creating MD structures and loading data in the DW) requires combining the two methods. The data integration problems must be considered in order to select the appropriate methods. Besides, most of existing approaches -apart from industrial tools (Barateiro and Galhardas, 2005) do not provide means to automatically generate or document all the aspects of the warehousing task.

Model Driven Architecture (MDA) is well known as a framework that manages complexity, it significantly reduces development costs, improves software quality and accomplishes high levels of reuse (Bettin, 2003), (Kleppe et al., 2003), (Object Management Group, 2003). Therefore, we assume that with the support of MDA, the warehousing task will require less efforts and time. Besides, MDA provides a support for integration, interoperability, adaptability, portability and reusability of information systems (Kleppe et al., 2003). Thus, providing a complete and integrated model driven methodology for DW and ETL design will be useful.

In this paper, we propose a unified, mixed and semi-automated method for DW and ETL design. This method has several advantages, specifically:

(i) it addresses both the DW's and the ETL processes modelling, avoiding inconsistencies, integration and interoperability problems that may arise when using separate methods;

Atigui F., Ravat F., Tournier R. and Zurfluh G..
A UNIFIED MODEL DRIVEN METHODOLOGY FOR DATA WAREHOUSES AND ETL DESIGN.
DOI: 10.5220/0003502002470252
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 247-252
ISBN: 978-989-8425-53-9

(ii) it proposes a unified conceptual model that defines MD concepts (structures and operations) that can be reused to define new DW schemata;
(iii) it is a model driven approach that tackles the warehousing process within an integrated and well-structured way;
(iv) it automatically generates code by using a set of formal transformations rules, thus saving time and effort.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces our approach. Section 4 details the conceptual model. Section 5 presents the logical and physical models as well as the transformation rules. Section 6 concludes and lists some future work.

2 RELATED WORK

Several research deal with DW design problems. In the literature, this issue has been tackled from two complementary but different points of view (Rizzi et al., 2006). The first deals with the MD modelling and attempts to describe the DW MD structure (Romero and Abelló, 2009). The second aims at representing processes responsible for loading and updating data in the warehouse (Vassiliadis, 2009).

Regarding the MD design, existing approaches can be classified into three main categories (Rizzi et al., 2006). Requirement-driven approaches (Tsois et al., 2001), (Prat et al., 2006), provide MD schemas based on a detailed analysis of decision-makers' needs. Data-driven approaches (Golfarelli and Rizzi, 1998), (Hüsemann et al., 2000), start from an analysis of data sources to identify the structure of the MD schema and select relevant data for decision making. Finally, mixed approaches (Zepeda et al., 2008), (Mazón and Trujillo, 2009), (Romero et Abelló, 2010), (Essaidi and Osmani, 2010), consider both decision-makers requirements and data availability within operational sources. These approaches provide MD schemata that both meet decision-makers' needs and fit with existing data sources.

Regarding ETL processes, several ways for their design and development exist. Academic researches offer either specific models or to reuse existing standards such as UML or the Business Process Model Notation (BPMN).

In (Vassiliadis et al., 2002) a conceptual model is defined that provides a specific graphical notation allowing designers to formally define technical issues often encountered in ETL processes. Then, the authors complement their model by providing a method for conceptual modelling (Simitsis et Vassiliadis, 2003). (Simitsis, 2005) provide a set of rules to map a conceptual model to logical one whereas (Simitsis et al., 2010) suggest using semantic web technologies to ease the selection of relevant data sources to be transformed and loaded.

Some Authors extend UML notations to describe ETL workflows. In (Trujillo et Luján-Mora, 2003) the authors define a set of ETL activities through stereotyped classes. (Luján-Mora et al., 2004) extend UML using a mapping diagram to represent the transformation rules between sources attributes and MD attributes. (Muñoz et al., 2008) use the UML activity diagrams to design the ETL process. (Muñoz et al., 2009) provide a model driven approach to generate the ETL process. (El Akkaoui et Zimanyi, 2009) propose a conceptual model for ETL processes based on the BPMN standard.

Besides, there are tools for designing or running ETL workflows (Barateiro and Galhardas, 2005), *e.g.* Oracle Warehouse Builder and Microsoft Integration Services. However, these tools use specific notations and languages, thus decreasing the integration and the interoperability levels of the system.

Compared to existing approaches, the advantages of our approach lie on merging the description of MD data structures and ETL operations within a unified model. This solution inherits the advantages of mixed approaches. Moreover, using a unified model avoids costly and redundant steps, as linking data sources. This model also avoids problems of inconsistencies, integration and interoperability encountered when using separate models and/or methods. Besides, the fact and the dimensions defined within a unified MD model are complete (structures and operations) and ready to be used in other DW's schemata. Moreover, our approach has the advantage of reusing and adapting existing models and language such as UML.

3 OVERVIEW OF OUR APPROACH

Our framework is tackled with MDA -an Object Management Group (OMG) standard- that aims at covering the software development life cycle. With MDA, the software development process is based on the use of models and automatic transformations between these models (Kleppe et al., 2003), (Object Management Group, 2003). The designer builds a unified conceptual model (PIM: Platform Independent Model) that describes the MD schema and the

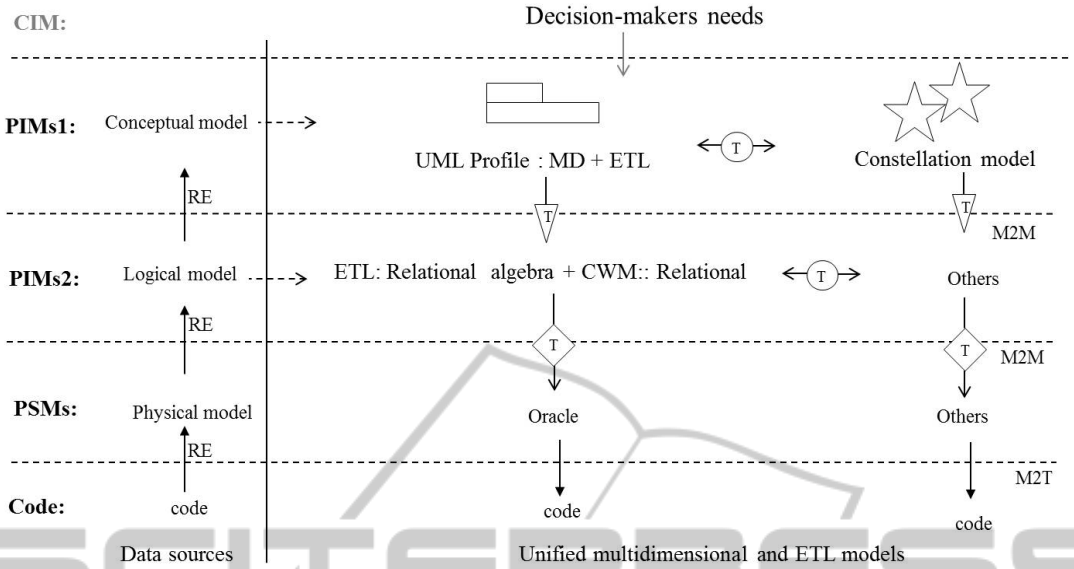

Figure 1: Model driven framework for DW and ETL design.

related ETL processes. The automatic transformations translate it into a successive set of models in order to get the code tailor-made to the chosen platform. The conceptual model represents the MD schema that considers the decision maker's needs and the data sources. Afterwards, the PIM is mapped into several logical models (PIMs2) (CWM::Relational, CWM::XML, etc.) depending on a chosen deployment platform (Oracle, Mondrian, etc.). Finally, the framework provides the appropriate code that will create the MD structure and the ETL workflow. The data sources modelling levels are generated by reverse engineering (RE) from physical models. An overview of our framework is shown in figure 1.

In next sections we focus on the conceptual and logical PIMs as well as the PSM. Moreover, we present the inter-levels and merging transformations rules. However, we do not present the requirements formalization (or decision-makers needs). Indeed, the CIM can be modelled by applying a requirement engineering framework, *e.g.* CADWA (Computer Aide Data Warehouse Analysis) method mentioned in (Salinesi and Gam, 2006), and then a set of transformation rules can be applied as in (Mazón and Trujillo, 2009) in order to automatically derive the conceptual schema.

4 UNIFIED MD AND ETL UML PROFILE

Conceptual modelling provides a high level of abstraction and aims at achieving the independence of deployment problems (Rizzi, 2008). We propose a conceptual model based on the Unified Modelling Language (UML). UML is a well-known standard modelling language and supported by many tools. It has the major advantage of adapting and reusing existing technologies.

A UML profile is a set of mechanisms and techniques that attempts to adapt UML to a specific application domain. From a technical view, a UML profile is a set of stereotypes that can be defined as domain-specific concepts (Kleppe et al., 2003). Since the proposed framework is based on unifying the DW and ETL design, the profile illustrated in figure 2 considers concepts related to both design issues. It describes basic MD concepts such as "Constellation" which extends the metaclass "Package". A constellation is composed of three types of classes namely "Fact", "Dimension" and "LevelAttributes". A dimension can be related to one or more "LevelAttribute" by means of a specific type of composition association named "Hierarchy". Also, "LevelAttributes" that belong to the same hierarchy are related with a specific composition association called "Rolls up". A UML class is composed of static properties and a set of dynamic operations. In the DW context, static properties define fact measures and dimension attributes. Dimension attributes may be parameters (that specify data aggregation levels) or weak attributes (that complement parameter semantics). While, operations represent the ETL processes modelled by the stereotype "ETLOperation" that extends the metaclass "Operation". All the

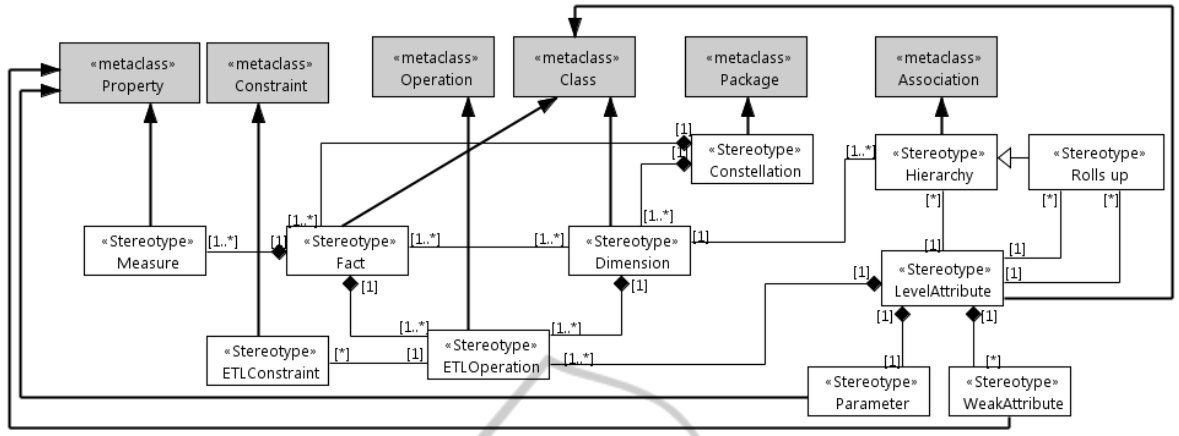

Figure 2: UML profile for multidimensional and ETL modelling.

stereotypes have a public visibility except for ETL operations that must be hidden to the decision-makers.

5 AUTOMATIC GENERATION FOR LOGICAL PIM AND PHYSICAL PSM

In this section we first present how to obtain a logical PIM from the conceptual PIM. Second, we present how conceptual and logical PIMs are then mapped into physical PSMs. We particularly focus on the static properties transformations. The transformation rules are formalized by QVT (Object Management Group, 2009). QVT is the OMG standard language for model-to-model transformations. It is a declarative language and offers both graphical and textual syntaxes. Besides, QVT allows multi-directional transformations as well as merging models (two or more models are mapped into one or more model). A QVT transformation between two candidate models is specified by a set of relations. Relations are defined by two or more domains which specify a candidate model and a set of corresponding elements to be matched as well as a pair of "when" (pre-conditions) and "where" (post-conditions) predicates.

In the next subsections, we detail the logical PIM and the PSM, as well as the model-to-model transformation rules that we formalized using QVT graphical syntax. Due to limited space, we present only the dimension transformation relations.

5.1 Logical Design

Logical models are automatically generated from conceptual models by applying a set of rules. In order to cover as much as possible of warehousing application, our approach provides a set of logical models. The designer can choose the most suitable one with the application he is developing such as the normalized ROLAP (Relational On-Line Analytical Processing), the denormalized or the optimized ones (that define views for calculating pre-aggregates), OOLAP (Object On-Line Analytical Processing), XML schemata, etc. To illustrate our framework, we chose to define transformation rules for denormalized ROLAP (Kimball, 1996) often used for MD schemata. In order to automatically derive this model, we define a set of QVT rules to generate a denormalized logical PIM. Formal definitions of these rules contain several QVT relations, we present only the dimension-to-table rule.

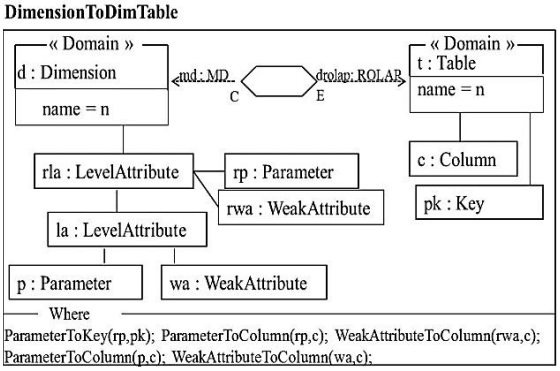

Figure 3: Dimension to dimension table relation.

DimensionToTable Relation. Each dimension of the constellation model (source model shown on the left side of the following figure) is mapped into a ROLAP table (target model shown on the right side) that have the same name. The "Where" clause speci-

fies that all the parameters and weak attributes are transformed into columns of the table by applying, respectively, ParameterToColumn and WeakAttributeToColumn relations. Moreover, the root parameter is mapped into the table's primary key through the relation ParameterToKey.

5.2 Physical Design

Physical models are dependent on a specific platform; we choose to detail transformations rules generating Oracle materialized view. The use of materialized views is significantly advantageous since calculating, storing, loading, and refreshing are performed automatically by the database.

Oracle physical schema is structured within materialized views and dimensions. The materialized view definition is based on ROLAP tables (PIM2); however dimensions depend on the hierarchies of the dimensions (PIM1). Therefore, the physical model is generated by merging both conceptual and logical models.

We have formalized these rules using QVT graphical notations (we expose only the dimension transformation relations).

DimTableToMaterializedView relation. Each ROLAP dimension table is mapped to a materialized view with the same name. The "Where" clause identifies relations mapping source and target columns as well as source and target primary key.

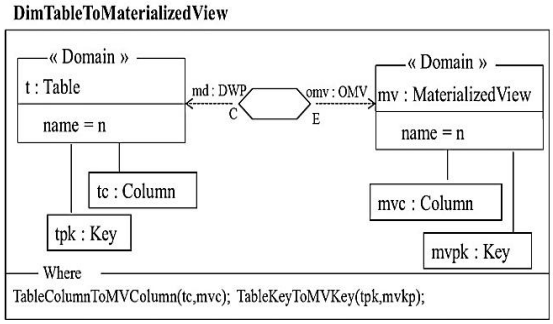

Figure 4: Dimension table to materialized view relation.

MDDimensionToOracleDimension Relation. This relation maps each conceptual dimension to an Oracle dimension with the same name prefixed by '_dim'. The "When" clause specifies that the appropriate ROLAP table must be already mapped to a materialized view. The "Where" clause specifies that hierarchies, parameters and weak attributes are mapped to Oracle hierarchies, levels and attributes respectively.

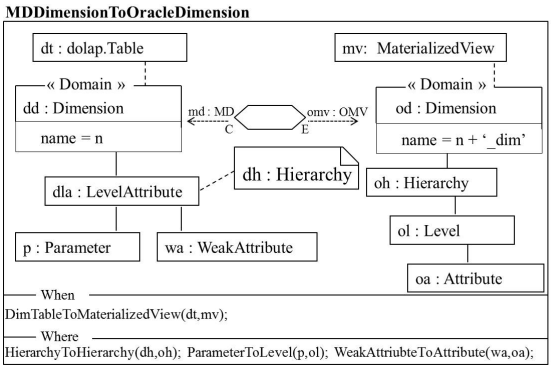

Figure 5: MD dimension to Oracle dimension relation.

6 CONCLUSIONS AND FUTURE WORK

In this paper we presented a model driven approach for DW and ETL design. With the support of the MDA, our framework semi-automates the warehousing task. Indeed, each modelling phase (requirement's analysis, conceptual, logical and physical design) is specified by one or more models. The method applies a series of QVT transformations in order to automatically generate the code.

For conceptual design, the method provides an UML profile that provides a unified description of MD structure and ETL processes. Logical and physical models are also generated by applying QVT transformations. We have detailed the transformation rules from MD PIM to the denormalized ROLAP PIM and then from these PIMs to Oracle materialized views and dimensions.

As future work, we plan to focus on the requirements formalization phase (CIM) and on automating the transition between the CIM and the conceptual PIMs by defining a set of QVT rules. Besides, we intend to focus on the behavioural properties transformations by providing a specific model and transformation rules for the ETL processes. We also intend to develop our method by considering others platforms. Moreover, we plan to apply our approach on real-world case studies and to evaluate it by different users.

REFERENCES

Barateiro, J., Galhardas, H., 2005. *A survey of data quality tools. Datenbank-Spektrum 14, 48.*

Bettin, J., 2003. Model-Driven Architecture Implementation & Metrics. *SoftMetaWare, Ltd., Version 1.*

El Akkaoui, Z., Zimanyi, E., 2009. Defining ETL worfklows using BPMN and BPEL,*12th international workshop on Data warehousing and OLAP. p. 41–48.*

Essaidi, M., Osmani, A., 2010. Model driven data warehouse using MDA and 2TUP. Journal of Computational *Methods in Science and Engineering 10, 119–134.*

Golfarelli, M., Rizzi, S., 1998. A methodological framework for data warehouse design, *1st international workshop on Data warehousing and OLAP. p. 3–9.*

Hüsemann, B., Lechtenbörger, J., Vossen, G., 2000. *Conceptual data warehouse design. Citeseer.*

Kimball, R., 1996. *The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc. New York, NY, USA.*

Kleppe, A.G., Warmer, J., Bast, W., 2003. *MDA explained: the model driven architecture: practice and promise. Addison-Wesley Longman Publishing Co. Boston, MA, USA.*

Luján-Mora, S., Vassiliadis, P., Trujillo, J., 2004. Data mapping diagrams for data warehouse design with UML. *Conceptual Modeling–ER 2004 191–204.*

Mazón, J.-N., Trujillo, J., 2009. A hybrid model driven development framework for the multidimensional modeling of data warehouses. *SIGMOD Rec. vol. 38, 12.*

Muñoz, L., Mazón, J.N., Pardillo, J., Trujillo, J., 2008. Modelling ETL processes of data warehouses with uml activity diagrams, *On the Move to Meaningful Internet Systems: OTM 2008 Workshops. p. 44–53.*

Muñoz, L., Mazón, J.-N., Trujillo, J., 2009. Automatic generation of ETL processes from conceptual models, 12th international workshop on Data warehousing and OLAP - *21 th international workshop, Hong Kong, China, p. 33.*

Object Management Group, 2003. OMG Document -- omg/03-06-01 (MDA Guide V1.0.1)

Object Management Group, 2009. Query/View/Transformation.

Prat, N., Akoka, J., Comyn-Wattiau, I., 2006. *A UML-based data warehouse design method. Decision Support Systems 42, 1449–1473.*

Ravat, F., Teste, O., Tournier, R., Zurfluh, G., 2007. *Graphical querying of multidimensional databases, Advances in Databases and Information Systems. p. 298–313.*

Ravat, F., Teste, O., Zurfluh, G., 1999. Towards data warehouse design, *8th international conference on Information and knowledge management. p. 359–366.*

Rizzi, S., 2008. Conceptual modeling solutions for the data warehouse. *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications 208–227.*

Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J., 2006. *Research in data warehouse modeling and design: dead or alive?, 9th intern. workshop on Data warehousing and OLAP. p. 3-10.*

Romero, O., Abelló, A., 2009. A survey of multidimensional modeling methodologies. *International Journal of Data Warehousing and Mining 5, 1–23.*

Romero, O., Abelló, A., 2010. *Automatic validation of requirements to support multidimensional design. Data & Knowledge Engineering.*

Salinesi, C., Gam, I., 2006. A Requirement-driven Approach for Designing Data Warehouses. *In Requirements Engineering :Foundation for Software Quality (REFSQ).*

Simitsis, A., 2005. Mapping conceptual to logical models for ETL processes, *8th International workshop on Data warehousing and OLAP. p. 67–76.*

Simitsis, A., Skoutas, D., Castellanos, M., 2010. *Representation of conceptual ETL designs in natural language using Semantic Web technology. Data & Knowledge Engineering 69, 96–115.*

Simitsis, A., Vassiliadis, P., 2003. *A methodology for the conceptual modeling of ETL processes, CAiSE workshops.*

Trujillo, J., Luján-Mora, S., 2003. A UML based approach for modeling ETL processes in data warehouses. *Conceptual Modeling-ER 2003 307–320.*

Tsois, A., Karayannidis, N., Sellis, T., 2001. MAC: *Conceptual data modeling for OLAP, the International Workshop on DMDW. p. 28–55.*

Vassiliadis, P., 2009. A Survey of Extract–Transform–Load Technology. *International Journal of Data Warehousing and Mining 5, 1-27.*

Vassiliadis, P., Simitsis, A., Skiadopoulos, S., 2002. Conceptual modeling for ETL processes, dans: *Proceedings of the 5th international workshop on Data Warehousing and OLAP. p. 14–21.*

Zepeda, L., Celma, M., Zatarain, R., 2008. A mixed approach for data warehouse conceptual design with MDA. *Computational Science and Its Applications–ICCSA 2008 1204–1217.*