

# Modeling and automatic generation of data warehouse using model-driven transformation in business intelligence process

Redouane Esbai, Soufiane Hakkou, Mohamed Achraf Habri

Laboratory of Applied Mathematics and Information Systems, Mohammed First University, Oujda, Morocco

## Article Info

### Article history:

Received Aug 25, 2022

Revised Feb 4, 2023

Accepted Feb 12, 2023

### Keywords:

Business intelligence

Data warehouse

Metamodels

Model transformation

Online analytical processing

cube

## ABSTRACT

The work presented in this paper focuses on the modeling and implementation of business intelligence processes, specifically how to apply model-driven architecture (MDA) throughout the entire development process of a data warehouse to automatically generate the multidimensional schema. As a result, this work specifies different rules for automating the process of obtaining an online analytical processing (OLAP) cube and implementing it using a collection of metamodels and automatic transformations. The data warehouse relational and OLAP cube models are then created using a model transformation from the conceptual model (class diagram). Finally, the XML file and SQL code are generated for creating the OLAP cubes and relational data warehouse, respectively. Using transformation languages like query-view-transformation (QVT) and Aceleo, all model transformations are performed.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Redouane Esbai

Laboratory of Applied Mathematics and Information Systems, Mohammed First University

Oujda 60000, Morocco

Email: es.redouane@gmail.com

## 1. INTRODUCTION

The data warehouse modeling phase is the first and most crucial step in the business intelligence (BI) process since it determines the quality of all other steps, including data transformation, data analysis, information extraction, and online analytical processing (OLAP). Thus, because the design of a data warehouse is frequently described as a multi-tier system in which data from one layer is generated from data from the previous layer, establishing a data warehouse is a difficult and complex operation [1]. As a result of this architecture, several strategies, and approaches for developing different parts of data warehouses have been offered.

Through this work, the authors use a conceptual model summarizing the analysis and design of the information system of the organization. From this model, future objects decision models such as relational and OLAP cube models will be automatically generated. The decisional model will be represented as a multidimensional schema; indeed, multidimensional modeling is the most commonly used representation of data warehouse schemas [2]. Wherefore, a lot of time and effort has gone into data warehouse design, and several methods for automated data warehouse modeling have been developed [3]. Although some of these methods adopted model driven architecture (MDA), none of them have gained widespread consensus [4].

In this paper, the authors propose a new method for designing and implementing data warehouse relational and OLAP cubes based on the MDA approach [5]. To do so, the UML class diagram is used to present the multidimensional schema as a platform independent model (PIM). Afterwards, from this latter, two platform specific models (PSMs) which are relational and OLAP cube are obtained. At last, the SQL code is generated from the relational model for building a data warehouse in relational database management system

(RDBMS) and the XML code is generated to implement an OLAP cube, which can be integrated in a server like Pentaho schema workbench. The model-to-model transformation can be done using query-view-transformation (QVT), and Aceleo language is also used to generate code from the models. The advantage of these standards is the bidirectional execution of transformation rules. In this respect, the purpose of this research is to rethink the work [6], [7] by using MDA concepts to develop the transformation rules aimed at generating OLAP cubes and data warehouse relational implementation, and then any user can create a data warehouse without the need for detailed knowledge on BI tools.

This paper is organized as follows: related works are presented in the section 2, the section 3 explains our proposed method. A case study is presented in the section 4, and the section 5 focuses on the results and possible extensions. The last section concludes this paper and presents some perspectives.

## 2. RELATED WORK

Several research projects have been conducted in recent years to develop data warehouse systems. This section presents a brief overview of some approaches that make use MDA viewpoints in the data warehouse life cycle. Firstly, by utilizing the multidimensional partitioning approach and a multidimensional and spatio-temporal design pattern, which will direct the generation of the data schema, Batoure [8] use model-driven engineering to coordinate the overall process of modelling, implementing, and powering a decision system. On the other hand, Sraï *et al.* [9] proposed a fact-dimension model. It is a data warehouse graphical conceptual model. They also demonstrated how to generate a data warehouse schema from the entity-relationship diagram's data sources.

Several data warehouse case studies are presented in [10]-[13]. The star schema and their variants are used in the data warehouse design, which is based on a relational approach employing tables, columns, and foreign keys. The authors' concern was limited to the representation of relational data warehouses, and they disregarded any other technologies. Nevertheless, this work is recognized as a standard in the field of data warehouses. In contrast, Beggar *et al.* [14] suggested a method that uses QVT relations to create a multidimensional model from user requirements. The generation of the final code and PSM model was not explained by the authors, and in a previous study [15] used the MDA approach to create a data warehouse that allows them to store temporal structures associated with time-varying dimension levels. The authors identified the transformation rules to generate the SQL code to create the associated tables and the historical data warehouse. The use of data in the OLAP model was not discussed by the authors.

What is more, Blanco *et al.* [16] proposed a method to generate the relational and CWM OLAP models using the MDA approach. The two models obtained are independent of specific platforms. Except for the relationship transformation of type many to many, the authors defined the transformations rules with the QVT language to generate the OLAP model, and the study did not discuss how the cube final code was resolved.

Last but not least [6], [7] suggested an MDA approach to obtaining an OLAP cube and its implementation by automating the transformations process from metamodels. Compared to the mentioned cited works, this paper aims to rethink and to complete the work presented in [6], [7], by applying the standard QVT and Aceleo to develop the transformation rules aiming at automatically generating the SQL code of data warehouse relational model and XML file of OLAP cube. It is the only work for reaching this goal.

## 3. PROPOSED METHOD

The MDA approach opted for this work is the modeling and template approaches to generate the relational model of data warehouse and OLAP cube. As mentioned above, these approaches require a source and a target metamodels. This section presents the various meta-classes forming the source metamodel of conceptual multidimensional and the targets metamodels of data warehouse relational and OLAP cube.

### 3.1. Multidimensional source metamodel for data warehouse

The source metamodel proposed for the transformation, is a simplified metamodel containing facts and dimensions. A fact is made up of attributes that represent measures and degenerate dimensions that represent the dimensions that surround the fact. The dimension is made up of bases that represent various levels of hierarchy. Each base has a descriptor, dimension characteristics, and an ID that acts as the base key. The fact\_attribute class now includes the attribute aggregation function. This metamodel is an extension of the work done by Moukhi *et al.* [17]. Figure 1 shows the source metamodel.

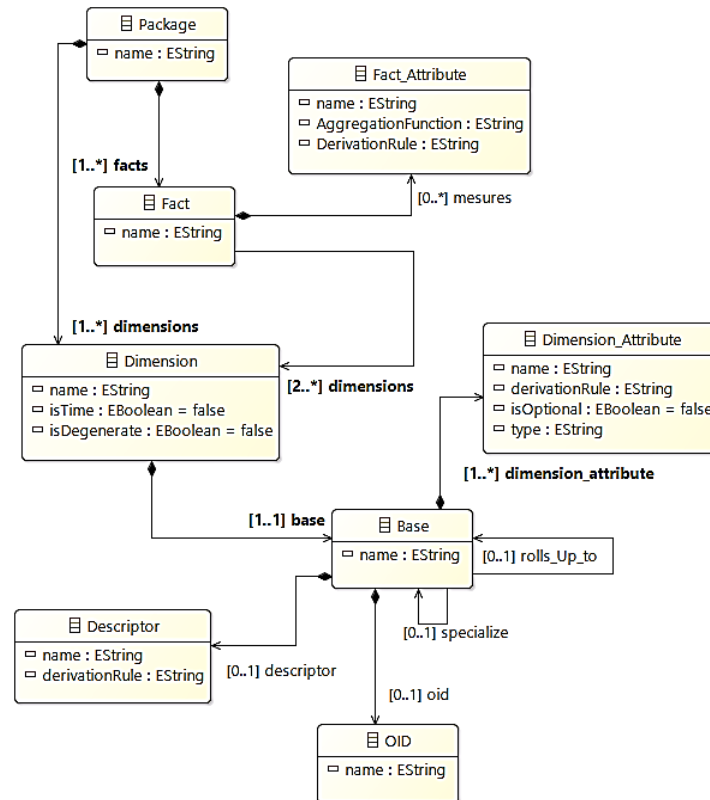


Figure 1. Simplified conceptual multidimensional source metamodel

### 3.2. The OLAP cube and data warehouse relational target metamodels

The proposal of this study aims to automatically generate two PSM (the relational model and the OLAP model) needed to create respectively the structures of data warehouse and cube. Therefore, to design the PSMs, the UML [18] class diagrams metamodels are adopted to present following the OLAP and data warehouse relational metamodels as shown in Figure 2 since they are going to be used in this work. The OLAP cube metamodel presented in Figure 2(a), is structured into a schema class that owns all elements of an OLAP model, i.e., dimensions and cubes. In the OLAP metamodel, each dimension is a collection of hierarchies and levels. In addition, cubes are used to store measures and they are related to the dimensions. These latter can also contain multiple and diverse hierarchical arrangements including specialized hierarchies that support ordering levels by hierarchy levels. The work of Prat *et al.* [19] contains a more complete version.

According to Kimball and Ross [20], the most common representation for multidimensional models is relational; thus, this work assumes that the second PSM is the relational model. Figure 2(b) illustrates a simplified CWM [21] relational metamodel in which a schema is composed of tables with columns and keys. Each key is a column which can be a primary or foreign key. Each fact table contains numerous foreign keys derived from its dimensions. The fact table also has numerical columns to represent measures and each dimension can have columns to represent dimension attributes. For simplicity reasons, this study uses the simplified relational metamodel shown in Figure 2(b).

### 3.3. Transformation process

With the multidimensional model as the starting point and the production of the implementation code as the finish line, this work's proposal aims to model data warehouses using the MDA approach as shown in Figure 3. The authors specify the required metamodels and the main model transformations for each MDA level [22]. The authors first created ECORE [23] models for the source and target meta-models. Many metamodels require multiple model transformations to be created. From these created metamodels, two transformations techniques are needed: model to model (M2M) and model to text (M2T). The M2M transformation approach, by using the QVT operational language [24], is utilized to produce two PSM models from one PIM model. Next, the M2T transformation is executed with Acceleo language [25] to automatically generate the SQL and XML [26] codes required for creating OLAP cubes and relational data warehouses from PSM models.

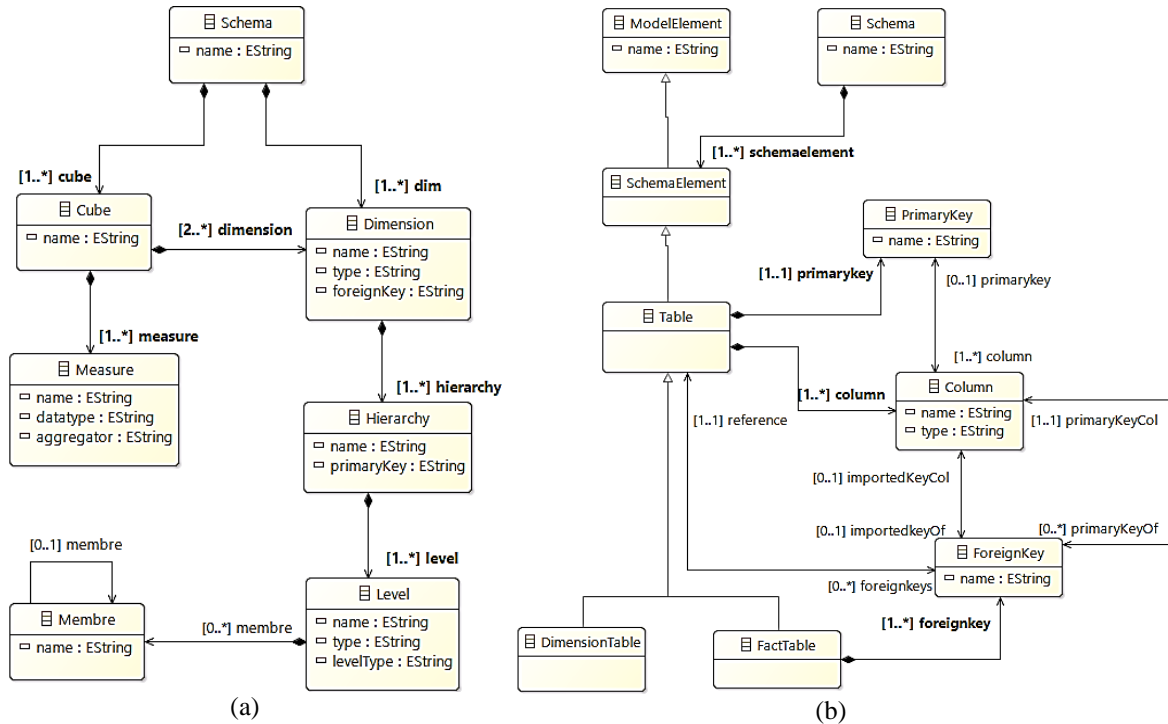


Figure 2. Simplified data warehouse relational (a) OLAP cube target metamodels and (b) a simplified CWM

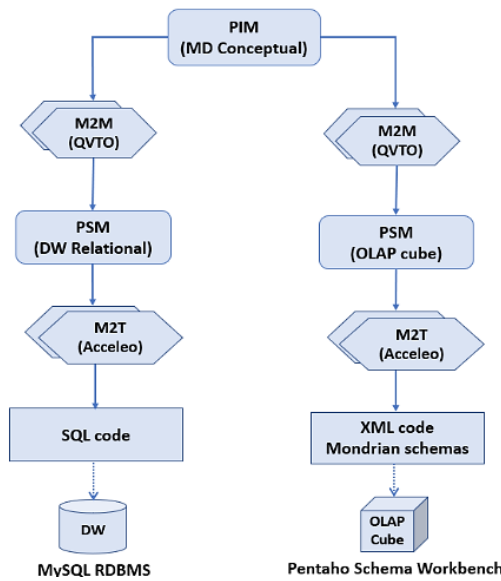


Figure 3. The proposed transformation process in MDA approach

### 3.3.1. The M2M transformation rules (PIM to PSM)

The first transformation uses, in entry, a model of multidimensional type, and in output a model of OLAP cube type. The first rule of transformation defines the mapping between all the multidimensional package's elements and the OLAP schema's elements. The purpose of the second rule is to transform each fact into a cube by creating the measures and dimensions for each cube. It is a matter of transforming each dimension to an OLAP dimension, which also entails the generation of hierarchies and levels. To accomplish this, the authors define a function that creates hierarchies by looping over the associations of type rolls-up-to, without forgetting to give the various elements names and types. Figure 4 presents the principal part of the M2M transformation rules using the QVT language.

The second transformation uses a multidimensional model in the input and generates a ROLAP model in the output. However, the authors proposed just below the main rules to obtaining a model of the data warehouse relational, from the conceptual model. A summary of these rules is presented in Figure 5.

```

MD2OLAP.qvto ×
modeltype MDM uses "http://www.example.org/mdmm";
modeltype OLAPMM uses "http://www.example.org/olapmm";
transformation MD2OLAP(in mdModel:MDMM, out olapModel:OLAPMM);
main() {
  mdModel.objects()[Package]->map MDPackage2SchemaOLAP();
}
mapping Package::MDPackage2SchemaOLAP() : Schema {
  name:=self.name;
  cube:=self.facts->map fact2cube();
}
mapping Fact::fact2cube() : Cube {
  name:=self.name.toLower();
  measure:=self.measures->map factAtt2measure();
  dimension:=self.dimensions->map Dim2Dim();
}

```

Figure 4. QVT code of M2M transformation from multidimensional to OLAP model

```

MD2ROLAP.qvto ×
transformation MD2ROLAP(in mdModel:MDMM, out rolapModel:ROLAPMM);
main() {
  mdModel.objects()[Package]->map MDPackage2SchemaROLAP();
}
mapping Package::MDPackage2SchemaROLAP() : Schema {
  name:=self.name+"_ROLAP";
  schemaelement:=self.dimensions->map dim2Table();
  schemaelement+=self.facts->map fact2Table();
}

```

Figure 5. QVT code of M2M transformation from multidimensional to data warehouse relational model

### 3.3.2. The M2T transformation rules

Unlike M2M, Aceleo language is used to implement M2T transformations. The transformation rules towards the SQL and XML codes are generated to respectively build the data warehouse and its equivalent OLAP cubes as shown in Figure 6. First, the database is created via the SQL code generator. The subsequent step is the generation of the different dimension's tables correspondingly to the relational model. Finally, a fact table is created that contains the foreign keys referencing primary keys in dimensions tables already generated. The SQL code to create all the tables corresponds to a "Create Table". Figure 6(a) bellow presents the transformation rules with Aceleo to generate an SQL file.

Then, the process of creating an OLAP cube still depends on the platform used and requires a thorough understanding of its tools. In fact, the multidimensional structures are defined by the OLAP model. Therefore, the proposed approach aims to automatically generate OLAP cube from the code. The most popular BI platforms were compared, and the results showed that each platform had different cube code extensions and templates. In this paper, the authors explain the automatic generation of XML code for the Mondrian schemas. A Mondrian schema is a structure of a multidimensional database defined by an XML file and Pentaho schema workbench platform can be used to create Mondrian schemas. It's interesting to note that the relational data warehouse must already exist to generate the OLAP cube. Figure 6(b) presents the transformation rules with Aceleo to generate an XML file.

```

ROLAP2SQL.mtl ×
[comment encoding = UTF-8 /]
[module ROLAP2SQL('http://www.example.org/rolapmm')]
[template public ROLAP2SQL(aSchema : Schema)]
[comment @main/]
[file (aSchema.name.concat('.sql'), false, 'UTF-8')]
CREATE DATABASE [aSchema.name];
USE [aSchema.name];

[for (dim : DimensionTable | aSchema.schemaelement)]
[dim2SQL(dim)]
[/for]
[for (fact : FactTable | aSchema.schemaelement)]
[fact2SQL(fact)]
[/for]
[/file]
[/template]
[template public dim2SQL(dim : DimensionTable)]
CREATE TABLE [dim.name/] (
[for (c : Column | dim.column separator (',\n')) [c.name/] [c.type/] [/for]
PRIMARY KEY ([dim.primarykey.name/])
);
[/template]

```

(a)

```

OLAP2XML.mtl ×
[comment encoding = UTF-8 /]
[module OLAP2XML('http://www.example.org/olapmm')]
[template public OLAP2XML(aSchema : Schema)]
[comment @main/]
[file (aSchema.name.concat('.xml'), false, 'UTF-8')]
<Schema name="[aSchema.name/]">
[for (cube : Cube | aSchema.cube)]
[Cube2XML(cube)]
[/for]
</Schema>
[/file]
[/template]
[template public cube2XML(cube : Cube)]
<Cube name="[cube.name/]" visible="true" cache="true" enabled="true">
<Table name="[cube.name/]">
</Table>
[for (dim : Dimension | cube.dimension)]
[dim2XML(dim)]
[/for]
[for (measure : Measure | cube.measure)]
[measure2XML(measure)]
[/for]
</Cube>
[/template]

```

(b)

Figure 6. M2T transformation with Aceleo to generate a (a) SQL and (b) XML code

**4. CASE STUDY**

To validate the proposed approach, several tests were conducted. For example, the authors considered the UML class diagram model representing a business, among its activities, the sale of products to its customers; these products are classified under categories, and several subcategories form a category. The UML model instance of the company information system is represented in Ecore file. This file is composed of the element package composed by a fact SalesOrderDetail which contains measures of sales like sum of sales and quantities. The package also contains four dimensions, Customer, Product, ProductCategory, and ProductSubCategory that each contains attributes which characterize them. Figure 7 presents the model instance of class diagram. This model is provided as input for the QVT transformation. As output of this transformation, a file in XMI format is automatically generated. It represents the OLAP cube and the company data warehouse.

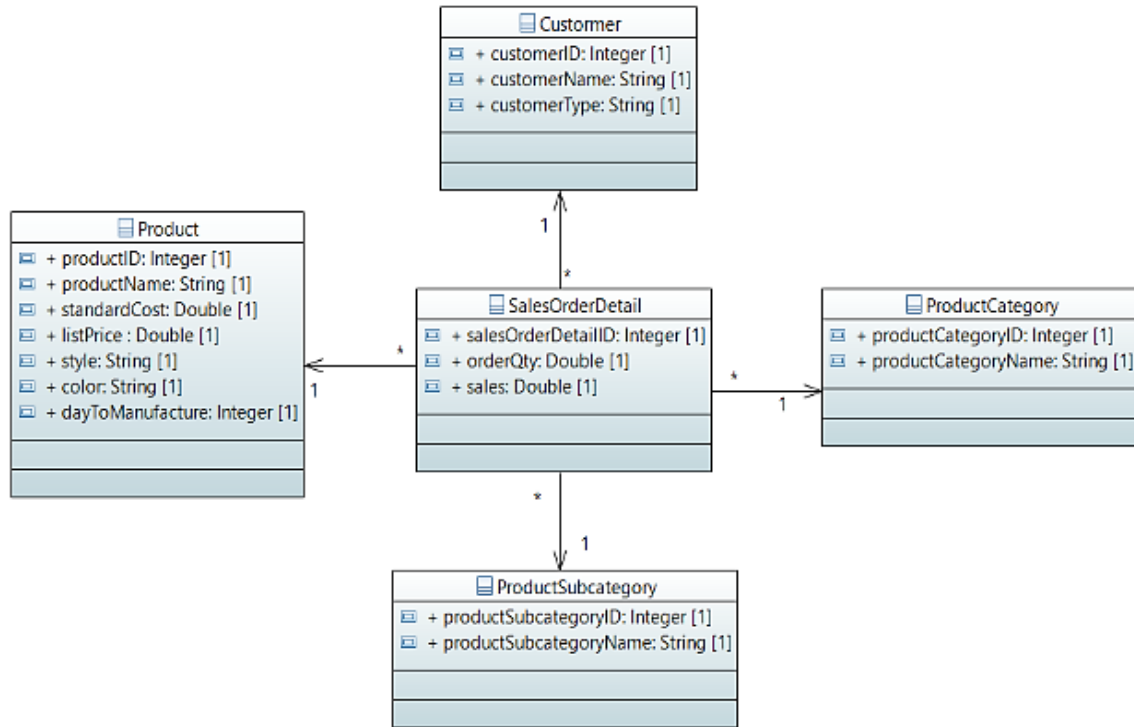


Figure 7. UML class diagram source model

**5. RESULTS AND DISCUSSION**

The transformation process is implemented using Eclipse platform. After defining the source and target metamodels (Ecore files), then the transformation rules are written to automatically generate the relational and OLAP models from the multidimensional one. The objective consists in applying the MDA approach to develop a data warehouse, in fact, there are different kinds of model transformations are defined in MDA, in this case, the authors are interested in transformations M2M (PIM to PSM) using QVT language and M2T (PSM to Code) using Acceleo language, in the first case a source UML model was transformed into two targets: OLAP cube and data warehouse relational model. Figure 8 shows the results after applying the M2M transformation rules. The first PSM result as shown in Figure 8(a) is an OLAP schema that is composed of a cube table named SalesOrderDetail surrounded by dimensions: Customer, Product, Category, SubCategory. The second PSM result as shown in Figure 8(b) is a relational that is composed of a table fact named SalesOrderDetail surrounded by other tables dimensions: Customer, Product, Category, SubCategory.

The next step is to define the transformation rules M2T using Acceleo to generate the SQL and XML code. This transformation uses as input the OLAP and data warehouse relational resulting models. In practice, writing the M2T transformation presents no difficulties; it simply entails creating a text file containing the transformation rules. Figures 9 and 10 illustrate the results after applying the M2T transformation rules. The first application generates SQL code for a sales management database as shown in Figures 9 and 11 shows the resulting OLAP cube after executing the obtained XML file on Pentaho schema workbench and processing it. The second application generates an XML file of Mondrian schema from an OLAP cube.



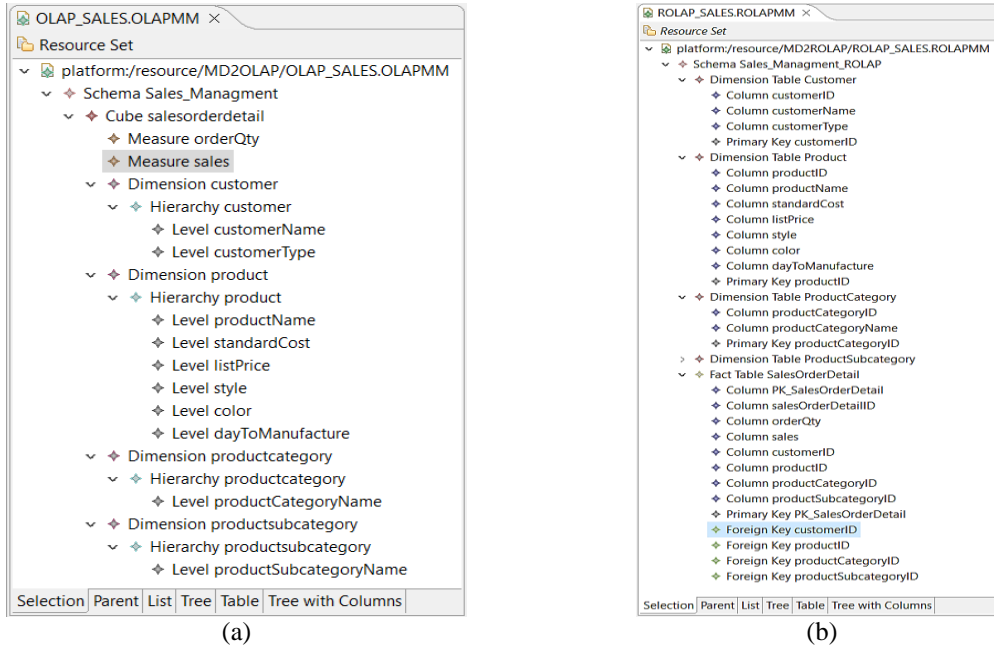


Figure 8. PSM results (a) is an OLAP schema that is composed of a cube table named SalesOrderDetail and (b) relational that is composed of a Table Fact named SalesOrderDetail

```

CREATE DATABASE Sales_Management_ROLAP;
USE Sales_Management_ROLAP;

CREATE TABLE Customer (
    customerID Integer, customerName VARCHAR(30), customerType VARCHAR(30), PRIMARY KEY (customerID)
);
CREATE TABLE Product (
    productID Integer, productName VARCHAR(30), standardCost NUMERIC, listPrice NUMERIC,
    style VARCHAR(30), color VARCHAR(30), dayToManufacture NUMERIC, PRIMARY KEY (productID)
);
CREATE TABLE ProductCategory (
    productCategoryID Integer, productCategoryName VARCHAR(30), PRIMARY KEY (productCategoryID)
);
CREATE TABLE ProductSubcategory (
    productSubcategoryID Integer, productSubcategoryName VARCHAR(30), PRIMARY KEY (productSubcategoryID)
);
CREATE TABLE SalesOrderDetail (
    PK_SalesOrderDetail Integer, salesOrderDetailID Double, orderQty Double, sales Double, customerID Integer,
    productID Integer, productCategoryID Integer, productSubcategoryID Integer, PRIMARY KEY (PK_SalesOrderDetail),
    FOREIGN KEY (customerID) REFERENCES Customer(customerID),
    FOREIGN KEY (productID) REFERENCES Product(productID),
    FOREIGN KEY (productCategoryID) REFERENCES ProductCategory(productCategoryID),
    FOREIGN KEY (productSubcategoryID) REFERENCES ProductSubcategory(productSubcategoryID)
);
    
```

Figure 9. SQL file generated for building data warehouse

```

<Schema name="Sales_Management">
  <Cube name="salesorderdetail" visible="true" cache="true" enabled="true">
    <Table name="salesorderdetail" />
    <Dimension type="StandardDimension" visible="true" foreignKey="customerID" highCardinality="false" name="customer">
      ...
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="productID" highCardinality="false" name="product">
      ...
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="productCategoryID" highCardinality="false" name="productcategory">
      ...
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="productSubcategoryID" highCardinality="false" name="productsubcategory">
      ...
    </Dimension>
    <Measure name="orderQty" column="orderQty" datatype="Numeric" aggregator="sum" visible="true" />
    <Measure name="sales" column="sales" datatype="Numeric" aggregator="sum" visible="true" />
  </Cube>
</Schema>
    
```

Figure 10. XML file generated of OLAP cube

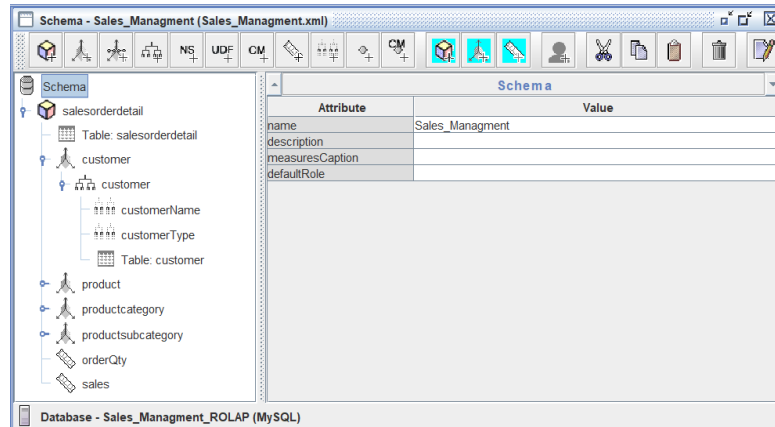


Figure 11. The generated OLAP cube on Pentaho schema workbench

## 6. CONCLUSION

In this paper, an MDA approach is proposed to transform a UML class diagram representing a multidimensional conceptual as a PIM model into a data warehouse repository and its OLAP cube as PSMs models. The main advantage of integrating MDA to develop a data warehouse is improving productivity and decreasing implementation cost. On the other hand, the metamodels used in the process of data warehouse development, are proposed just for PIM and PSM. The transformations rules were developed using QVT to transform the class diagram into data warehouse relational and OLAP cube models, and then the SQL and XML codes were generated in order using Acceleo to create the multidimensional structures of data warehouses and OLAP cubes.

The main contribution of this work is to provide the required metamodels and thorough transformation rules to obtain the relational data warehouse and OLAP cube from the conceptual model. After generating the relational and OLAP models, the cube implementation code is automatically created. In addition, this paper aims to rethink and to complete the studies presented in the work by using the standards QVT and Acceleo to write transformation rules aimed at automatically generating SQL code to create a data warehouse relational on a MySQL database and an XML Mondrian code for the Pentaho schema workbench platform, where the deployment of the cube that was automatically implemented.

For the future works, this study can be extended to include the CIM metamodel as the first level of MDA to design the user requirements and complete the transformation engine, from data warehouse requirements analysis to code generation and deployment of OLAP cubes and data warehouse relational. Furthermore, this work aims to develop a tool to gather the whole elements of the approach and be able to use it efficiently.

## REFERENCES




- [1] W. H. Inmon, "Building the Data Warehouse", 4th ed., John Wiley & Sons, Oct. 2005.
- [2] A. Al-Badarnah and O. Al-Badarnah, "Challenges and interesting research directions in model driven architecture and data warehousing: a survey," *International Journal of Computer Science and Information Security*, vol. 14, no. 3, pp. 364-398, Mar. 2016, doi: 10.6084/m9.figshare.3154030.
- [3] O. Al-Badarnah and A. El Sheikh, "Developing model-driven quality-aware data warehouses with a UML profile," *International Journal of Computer Science and Information Security*, vol. 14, no. 5, pp. 456-474, May 2016.
- [4] A. Gosain and J. Singh, "Conceptual multidimensional modeling for data warehouses: a survey," *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, Nov. 2015, pp. 305-316, doi: 10.1007/978-3-319-23201-0\_10.
- [5] J. Miller and J. Mukerji "Technical guide to model driven architecture: MDA Guide Version 1.0. 1.," *Object Management Group (OMG)*, 2003.
- [6] O. E. Beggar, K. Lettrache, and M. Ramdani, "Towards an MDA-Oriented uml profiles for data warehouses design and development," *Proceedings of the 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Oct. 2016, pp. 1-6, doi: 10.1109/SITA.2016.7772270.
- [7] K. Lettrache, O. E. Beggar, and M. Ramdani, "The automatic creation of OLAP cube using an MDA approach," *Software: Practice and Experience*, vol. 47, no. 12, pp. 1887-1903, Jun. 2017, doi: 10.1002/spe.2512.
- [8] A. B. Batoure, "Using model-driven engineering for decision support systems modelling, implementation and powering," *International Journal of Data Engineering (IJDE)*, vol. 8, no. 1, pp. 1-12, Aug. 2020, doi: 10.3991/ijoe.v15i12.10720.
- [9] A. Sraï, F. Guerouate, N. Berbiche, and H. Drissi, "An MDA approach for the development of data warehouses from relational databases using ATL transformation language," *International Journal of Applied Engineering Research*, vol. 12, no. 12, pp. 3532-3538, Jan. 2017.
- [10] M. Hanine, M. Lachgar, S. Elmahfoudi, and O. Boutkhom, "MDA approach for designing and developing data warehouses: a systematic review and proposal," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, no. 10, pp. 99-110, Oct. 2021, doi: 10.3991/ijoe.v17i10.24667.






- [11] S. Luján-Mora, J. Trujillo, and I.-Y. Song, "A UML profile for multidimensional modeling in data warehouses," *Data and Knowledge Engineering*, vol. 59, no. 3, pp. 725-769, Dec. 2006, doi: 10.1016/j.datak.2005.11.004.
- [12] J.-N. Mazon, J. Trujillo, M. Serrano, and M. Piattini, "Applying MDA to the development of data warehouses," *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP (DOLAP 05)*, Association for Computing Machinery, Nov. 2005, pp. 57-66, doi: 10.1145/1097002.1097012.
- [13] J. Mazón and J. Trujillo, "Data warehousing meets MDA: A case study for multidimensional modeling," *New Trends in Data Warehousing and Data Analysis*, vol. 3, no.1, pp. 51-70, Oct. 2009.
- [14] O. E. Beggar, K. Letrache, and M. Ramdani, "DAREF: MDA framework for modelling data warehouse requirements and deducing the multidimensional schema," *Requirements Engineering*, vol. 26, no. 2, pp. 143-165, Jun. 2021, doi : 10.1007/s00766-020-00339-9.
- [15] C. G. Neil, M. E. D. Vincenzi, and C. F. Pons, "Design method for a historical data warehouse, explicit valid time in multidimensional models," *Ingeniare Revista Chilena de Ingeniería*, vol. 22, no. 2, pp. 218-232, Apr. 2014, doi: 10.4067/S0718-33052014000200008.
- [16] C. Blanco, I. G. de Guzmán, E. Fernández-Medina, and J. Trujillo, "An architecture for automatically developing secure OLAP applications from models," *Information and Software Technology*, vol. 59, no. 1, pp. 1-16, Mar. 2015, doi: 10.1016/j.infsof.2014.10.008.
- [17] N. E. Moukhi, I. E. Azami, A. Mouloudi, and A. Elmounadi, "Towards a new automatic data warehouse design method," *Electronic Journal of Information Technology*, vol.11, no.1, pp. 1-21, Nov. 2018.
- [18] "UML 2.0 superstructure final adopted specification" Object Management Group (OMG), Dec. 2017.
- [19] N. Prat, J. Akoka, and I. Comyn-Wattiau, "A UML-based data warehouse design method," *Decision Support Systems*, vol. 42, no. 3, pp. 1449-1473, Dec. 2006, doi: 10.1016/j.dss.2005.12.001.
- [20] R. Kimball and M. Ross, *The Kimball group reader: relentlessly practical tools for data warehousing and business intelligence remastered collection*, 2nd ed., Wiley, Dec. 2015.
- [21] "Common warehouse metamodel (CWM) specification," Object Management Group (OMG), Vers. 1.1, Mar. 2003.
- [22] R. Esbai, F. Elotmani, and F. Z. Belkadi, "Toward automatic generation of column-oriented NoSQL databases in big data context," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 15, no. 9, pp. 4-16, June 2019, doi: 10.3991/ijoe.v15i09.10433.
- [23] "Meta object facility (MOF) core specification," Object Management Group (OMG), Vers. 2.5.1, Oct. 2019.
- [24] "2.0 query/view/transformation (QVT) specification," Object Management Group (OMG), Jun. 2016.
- [25] J. Musset *et al.*, "Acceleo user guide," Obeosoft, 2006.
- [26] "XML metadata interchange (XMI) specification," Object Management Group (OMG), Vers. 2.5.1, Jun. 2015.

## BIOGRAPHIES OF AUTHORS






**Redouane Esbai**    teaches the concept of Information System at Mohammed First University. He got his thesis of national doctorate in 2012. He got a degree of an engineer in Computer Sciences from the National School of Applied Sciences at Oujda. His activities of research in the MASI Laboratory (Applied Mathematics and Information System) focusing on MDA integrating new technologies XML, Spring, Struts, and GWT. He can be contacted at email: es.redouane@gmail.com.



**Soufiane Hakkou**    is Associate Ph.D. Student. He got his specialized master's degree in software quality from Mohammed First University at Oujda. He is a researcher studying the business process model and notation and their applications in Electronic Document Management using MDA at MASI Laboratory in Mohammed First University, Morocco. He can be contacted at email: soufianehakkou@yahoo.fr.



**Mohamed Achraf Habri**    is Associate Ph.D. Student. He got his specialized master's degree in software quality from Abdel Malek Essaadi University at Tangier. He is a researcher studying the business process model and notation and their applications in MDA at MASI Laboratory in Mohammed First University, Morocco. He can be contacted at email: hmohamedachraf@gmail.com.