Transforming Conceptual Model into Logical Model for Temporal Data Warehouse Security: A Case Study

Marwa S.Farhan
Information Systems Dep,
Faculty of Computers and Information, Helwan University,
Cairo, Egypt

Mohamed E. Marie
Faculty of Computers and Information, Helwan University,
Cairo, Egypt

Laila M. El-Fangary & Yehia K. Helmy
Faculty of Computers and Information, Helwan University,
Cairo, Egypt

***Abstract*—Extraction–transformation–loading (ETL) processes are responsible for the extraction of data from several sources, their cleansing, customization and insertion into a data warehouse. Data warehouse often store historical information which is extracted from multiple, heterogeneous, autonomous and distributed data sources, thereby, the survival of the organizations depends on the correct management, security and confidentiality of the information. In this paper, we are using the Model Driven Architecture (MDA) approach to represent logical model requirements for secure Temporal Data Warehouses (TDW). We use the Platform-Independent Model (PIM) which does not include information about specific platforms and technologies. Nowadays, the most crucial issue in MDA is the transformation between a PIM and Platform Specific Models PSM. Thus, OMG defines use the Query/View/Transformation (QVT) language, an approach for expressing these MDA transformations. This paper proposes a set of rules to transform PIM model for secure temporal data warehouse (TDW) to PSM model, we apply the QVT language to the development of a secure data warehouse by means of a case study.**

Keywords- ETL; temporal data warehouse; Data warehouse security; MDA; QVT; PIM; PSM.

I. INTRODUCTION

Data warehouse often store historical information which is extracted from multiple, heterogeneous, autonomous and distributed data sources, thereby; the survival of the organizations depends on the correct management, security and confidentiality of the information. The application of the Model Driven Architecture (MDA) [1] in the secure modeling of DWs allows obtaining the secure logical scheme from the conceptual model. In this work we apply a set of QVT [2] relations to the development of a secure DW. Various approaches for the conceptual design of the DW repository have been proposed in [4, 5, 6, 7]. These proposals are twofold; on the one hand they try to represent the main MD properties at the conceptual level by abstracting away details of the target database platform where the DW will be implemented. On the other hand, they also define how to derive a logical representation tailored to a specific database technology (relational or multidimensional). These approaches are lacking in formal mechanisms to univocally and automatically obtain the logical representation of the conceptual model [3].

In order to overcome this limitation, in previous proposals, we have described a model driven framework for the development of DWs, based on the MDA standards. In this paper we will propose a set of rules to transform the proposed conceptual model in [8] to logical model, our model include two stages (1): ETL stage we use UML class diagram to represent ETL processes in the logical model. (2):DW stage we use the Query/View/Transformation (QVT) language [2] to the MD modeling of the DW repository within our MDA framework. The PIM model can be translated into: (1) one or more Platform Specific Models (PSM) with information about the specific technology used; or (2) other PIMs with a different level of abstraction. Each PSM can then be translated into a code that can be executed in the specific platform. The proposed model focuses on the logical modeling for temporal Data warehouse (TDW) considering the DW security issues. This paper is organized as the following: section 2 presents the related work. Section 3 presents ETL logical model considerations, Section 4 describes MDA and QVT features and an example is provided in this section to show how to apply MDA and QVT transformation rules. Finally, section 5 points out our conclusions and future works.

II. RELATED WORK

This section divides the related work according to two main research topics covered by this paper: *ETL modeling*, and *data warehouse modeling*. The paper focuses on the logical modeling specifications in these topics.

A. ETL Modeling

The modeling and optimization of ETL processes at the logical level is presented in [9], [10]. The authors of [11] proposed a design method that includes an algorithmic transformation of conceptual to logical models for ETL processes. The conceptual modeling of the ETL processes is discussed in [12]. In [13, 14] the authors focus on the dynamic [13] and static [14] modeling of the ETL processes. There are few research papers on DW security and OLAP (Online Analytical Processing) security [15, 16, 17, 18]. But none of the above mentioned papers discuss an integrated security model for ETL processes. In this paper, we propose the security aspects and temporal aspects that have to be considered in the

analysis and design phases of ETL processes and DW. Supporting our proposal the lack of security issues in ETL are mentioned in [19, 20].

B. Data warehouse Modeling

There are interesting contributions in the field of information systems security but they do not deal with DWs in the context of their specific security issues. One of the most relevant proposals that integrate security through the use of UML is UMLsec [21], which can be used to specify and evaluate UML security specifications using formal semantics. Furthermore, Model-Driven Security (MDS) [22] extends MDA to build secure information systems. Its designers specify the inclusion of security properties in high-level system models and use tools to automatically generate secure system architectures. Within the context of MDS, the same authors propose an extension of UML for modeling a generalized Role-Based Access Control (RBAC) called SecureUML [23].

Data Warehouses (DWs) present specific characteristics and security challenges related to all their layers and operations [24]. Proposals for DWs at conceptual and logical levels, which consider special characteristics of DWs, also exist, but they do not support security issues. The most interesting proposal is [25] in which the authors define a methodology to analyze security requirements, to represent them at the conceptual level .However; they do not define the transformation between levels. Another important proposals are [20,26,27,30,32]which provides security models at different abstraction levels and has been aligned with an MDA architecture in which security models are embedded and scattered throughout the high-level system models, which are transformed towards the final implementation according to the MDA strategy. However these models are consider the read operation in DW and didn't consider the temporal DW requirements. Our research efforts are thus applied to the development of secure DWs considering confidentiality issues during the whole development process, from an early development stage to the logical model. Our scope in this paper is how to transform conceptual model to logical model considering DW security and temporal issues.

III. ETL Logical Model Considerations

During the ETL process, data is extracted from an OLTP databases, transformed to match the data warehouse schema, and loaded into the data warehouse database. Fig. 1 shows the general framework for ETL processes. In the bottom layer we depict the data stores that are involved in the overall process. On the left side, we can observe the original data providers (typically, relational databases and files). The data from these sources are extracted (as shown in the upper left part of Fig. 1) by extraction routines, which provide either complete snapshots or differentials of the data sources. Then, these data are propagated to the *Data Staging Area* (DSA) where they are transformed and cleaned before being loaded to the data warehouse. The data warehouse is depicted in the right part of Fig. 1 and comprises the target data stores, i.e., fact tables and dimension tables. Eventually, the loading of the central warehouse is performed through the loading activities depicted on the upper right part of the figure [12].

The ETL process is not a one-time event. As data sources change the data warehouse will periodically updated. Also, as business changes the DW system needs to change in order to maintain its value as a tool for decision makers, as a result of that the ETL also changes and evolves. The ETL processes must be designed for ease of modification. A solid, well-designed, and documented ETL system is necessary for the success of a data warehouse project. As fig.1 shows An ETL system consists of three consecutive functional steps: extraction, transformation, and loading, in the following sections we will explain the ETL stage of our PSM. Our model transforms the PIM model which we proposed in [8] to PSM.

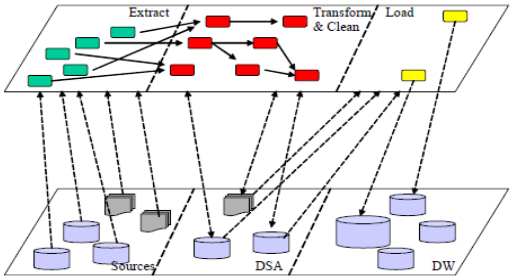

Figure 1.The environment of ETL processes

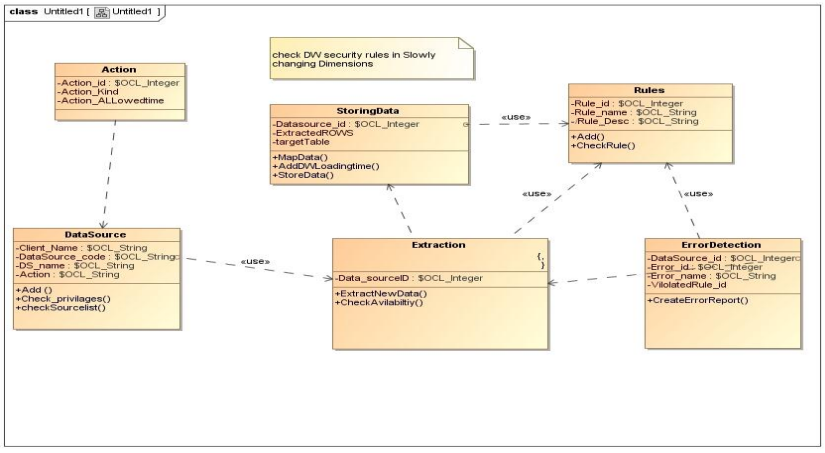

Figure 2. ETL logical model (ETL PSM)

We will use the UML class diagram because this is the most standardize way to express the ETL activities in the logicallevel.Fig.2 shows the ETL PSM model for the whole ETL processes as follows

A. Extraction

The first step in any ETL scenario is data extraction; in this step we extract data from their sources. We have 3 classes in this step:

1) DataSource class and Action Class (in parallel)

These two classes are working in parallel; the scenario is graphically depicted in Fig.2 and involves the following transformations.

1-first, we check the data source in the data source metadata and in the same time we check the allowed privileges in the action class for each data source.

2. Second, if we find any invalid privileges to any data source we reject the data extraction from this data source. We do this by comparing the required action in DS class and allowed action in Action class.

2) Extraction Class

3. After checking data source availability we use Extraction class. This class is responsible for extracting the modified and new data from data sources not extract all data. We do this by comparing the incoming data with the data in old Trans table which we stored in it the last refresh cycle data to extract the new data from the last DW refresh cycle, this table is stored in ETL metadata.

B. Transformation

The transformation step tends to make some cleaning and conforming on the incoming data to gain accurate data which is correct, complete, consistent, and unambiguous. This process includes data cleaning, transformation, and integration. *This stage includes three classes (Extraction class, Rule Class, Error detection), these classes are working in parallel*

Extraction class: as we explained earlier we extracted the modified data, in the transformation step we check these data against rules in Rule class.
Rule Class: we check data against set of privileges, these privileges contains (1)Security privileges which considers security rules which defined in (my paper);(2)Access privileges which includes:

a) ***Read access:*** *is the normal read operation it means just query data is allowed.*

b) ***Write access****: includes (insert) for temporal and non temporal data. insert, update or delete in case of temporal(Slowly Changing Dimension(SCD),these security constraints is explained in details in[8]*

Error detection class: this class is responsible for generate report with the invalid data using attributes and operations in Rule class to describe data source, the violated rules, error description, error time …etc.

C. Loading

Loading data to the target multidimensional structure is the final ETL step. In this step, extracted and transformed data is written into the dimensional structures actually accessed by the end users and application systems. In the load process we check the loaded data against DW security constraints, if there's any invalid data it can be added to error report as we explained earlier. If data is valid we add the data warehouse loading time(DWLT) to the SCD to represent the time when the new data is loaded in DW; this DWLT represents the (Start_time and end_time) of each row in SCD.

IV. MDA AND QVT FEATURES

QVT is an essential part of the MDA standard as a means of defining formal and automatic transformations between models. The QVT is a standard approach for defining formal relations between MOF compliant models.QVT consists of two parts: declarative and imperative. (1)Declarative part: provides mechanisms to define transformation as a set of relations that must hold between the model elements of a set of candidate models (source and target models).(2)Imperative part: defines operational mappings that extend the declarative part with imperative implementations when it is difficult to provide a purely declarative specification of a relation.

The proposed model focuses on the declarative part of QVT because the scope of the paper is the logical model requirements. This paper focuses on the relational layer of QVT which supports the specification of relationships that must hold between MOF models by means of a relations language. A relation is defined by the following elements:

-Two or more domains: each domain is a set of elements of a source or a target model. The kind of relation between domains must be specified: checkonly (C), i.e., it is only checked if the relation holds or not; and enforced(E), i.e., the target model can be modified to satisfy the relation.

– When clause: it specifies the conditions under which the relation needs to hold (i.e. precondition).

– Where clause: it specifies the condition that must be satisfied by all model elements participating in the relation (i.e. postcondition).

Defining relations by using the QVT language has the following advantages : (i) it is a standard language, (ii) relations are formally established and automatically performed, and (iii) relations can be easily integrated in an MDA approach[28].

A. Transformation from PIM model to PSM model

This sections explains the proposed rules to transform from PIM model to PSM, we will base on the approaches which was presented in [29, 30].Fig.3 illustrates the Secure Multidimensional MDA architecture [1]. On the left hand side the Secure Multidimensional conceptual scheme, i.e., SMD PIM is presented. By means of the transformation T1 we obtain the relational logical scheme, i.e., SMD PSM, represented in the centre of Figure 3. If we choose a SGBD that implements security aspects, then SMD PSM is transformed according to T2 into code for the target platform. This code is called the Secure Multidimensional Code (SMD Code). The Figure illustrates how the security constraint defined by means of a security Rule (represented as an UML note) is transformed

from the conceptual level to the logical level by employing T1, and later transformed into code with the T2 transformation.

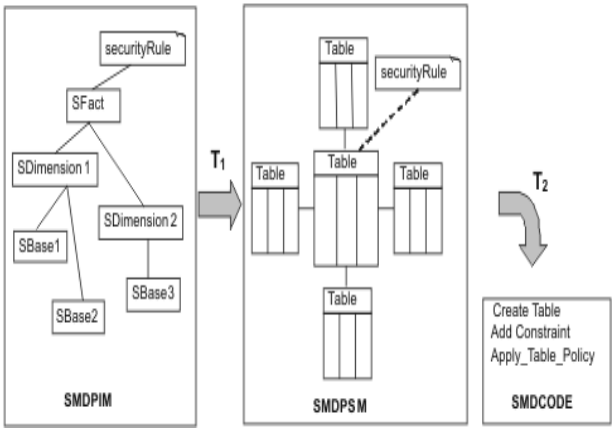

Figure 3.General Transformation Schema

B. QVT relations to obtain the PSM

In multidimensional modeling, the logical level is designed according to the specific properties of the SGBD (Relational Online Analytical Processing, ROLAP, Multidimensional Online Analytical Processing, MOLAP or Hybrid Online Analytical Processing, HOLAP). Still, Kimball [31] assures that the most common representation is on relational platforms, i.e., on ROLAP systems. The SMD PSM allows us to represent at the relational level the security requirements that were represented in the conceptual modeling of the DW. In this model we can represent tables, columns, primary and foreign keys, etc. Thus, we can establish security in attributes and tables. We express the security constraints that were modeled at the conceptual level by means of UML notes. We apply QVT declarative approach based on the proposed models in[27,29**]** **.**Fig.4 shows the case study which used in[29],we will use the same case study and we will use the QVT to represent the security requirements of TDW which was not considered in the QVT before this model.

Fig. 4 shows a secure MD model that includes a fact class (Admission), three dimensions classes (Diagnosis, Patient and Time), five base classes (DataD, Diagnosis_Group, DataP, City, and DataT), and a UserProfile class. The Admission fact class –SFact stereotype- contains all the individual admissions of patients in one or more hospitals, and can be acceded by all the users who have security levels secret or topSecret -labeled value SecurityLevels (SL)-, and perform health or administrative roles -tagged value SecurityRoles (SR)-. Be observed how the attribute cost only can be acceded by users whom play administratively role - tagged value SR- The class base DataP contains the information of the patients of the hospital and can be acceded by all the users who have security level secret - tagged value SL-, and play health or administrative roles -t value SR-.

The Address attribute can be only accessed by users who have an administrative role -tagged value SR of attributes-. City base class contains the information of cities, and it allows us to group patients by cities. City base class can be accessed by all users who have confidential security level -tagged value SL-. DataD base contains the information of each user diagnosis, and can be accessed by users who have a health role - tagged value SR-, and have secret security level -tagged value SL-. Finally Diagnosis_group contains a set of general groups of diagnosis. Each group can be related to several diagnoses, but a diagnosis will be always related to a group. Diagnosis_group can be accessed by all users who have confidential security level -tagged value SL-. Some security constraints have been specified by using the previously defined constraints.

1) Transforming SFacts into STables

The first relation in executing is SecureDW2SSchema, with it, all levels of security: confidential, secret and topsecret, as well as all the hierarchical roles tree is transformed into their equivalent ones of SSchema.

The UserProfile2RUserProfile relation transforms the UserProfile class into a table belonging to SSchema that will have the same name of UserProfile. The relation that follows, i.e., SFact2STable is shown in its graphical notation in Fig. 5, by means of this relation each SFact jointly with its security properties is transformed into a table that will contain the same information of security.

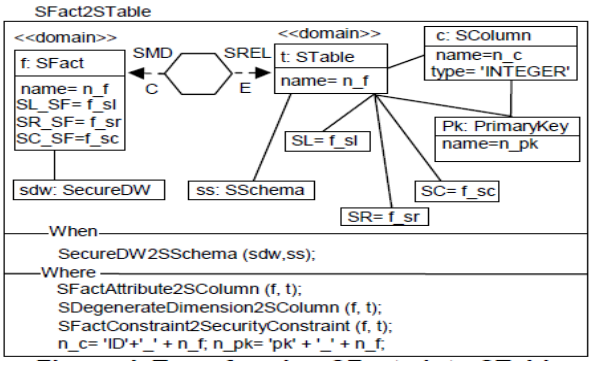

Figure 5 Transforming SFacts into STables

In Fig.6, we are going to show how the attributes of SFact are transformed into SColumns of the table that represents the SFact, so that, each column will contain the security information of its corresponding attribute in the SFact.

In Figure 7 we show the result of applying the SFact2STable relation to our case study. The SFact Admission is transformed into a table of the model SMD PSM, i.e., in the Admission table, that will have a primary key, as well as the security properties securityLevel and securityRole.

Fig. 8 shows the result of applying the SFactAttribute2SColumn relation, as a consequence, the Admission table will contain the columns respectively type and cost of type string and float. The column cost will have associated the security property securityRole. Also in Fig. 8, the associated requirements of security to the Admission table are modeled in the heading of the table, according to the SECRDW metamodel.

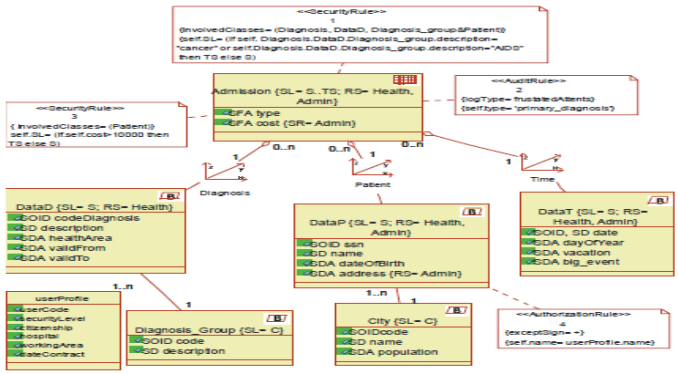

Figure 4: Example of secure multidimensional modeling

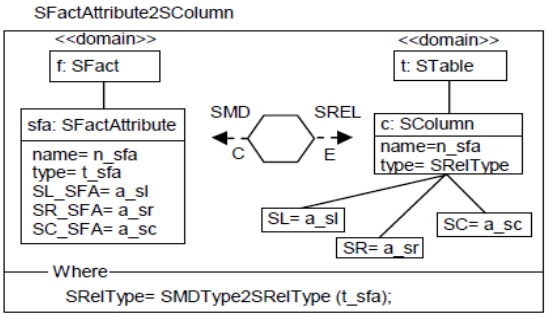

Figure 6 Transforming SFact attributes into Scolumn

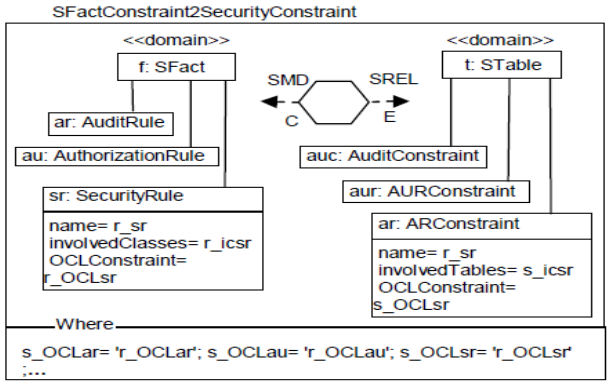

Figure 9 SFactConstraint2SecurityConstraint

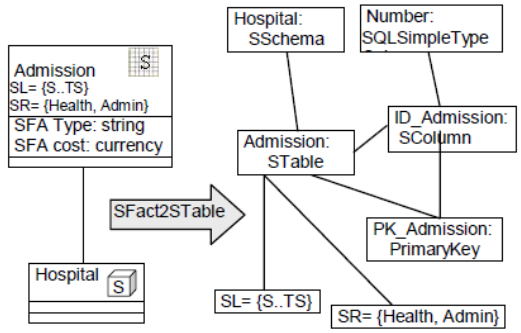

Figure 7 Applying SFact2Stable

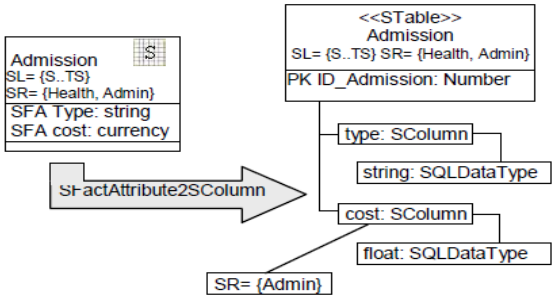

Figure 8 SFactConstraint2SecurityConstraint

Fig. 9 presents the definition of the SFactConstraint2SecurityConstraint relation, which guarantees that all the constraints associated with the SFact are transformed in constraints associated with the table, just as it can be seen in Fig. 10.

2) Transforming SDimension into Stable

In this section we will explain how to represent temporal and non-temporal dimensions.

Fig.10 we show the definition of the SDimension2STable relation. In multidimensional modeling the dimension do not have attributes [10]. For this reason, when the SDimension2STable relation is executed, a table is created whose name is merged with the names from dimension and rootBase respectively.

The rootBase is the only SBase associated with the SDimension. All the associated security information with the rootBase is transformed in security properties of the table and by means of the execution of the relations that appear in the clause *where* of the SDimension2STable relation, is guaranteed that all the attributes of the rootBase are going to conform the columns of the table.

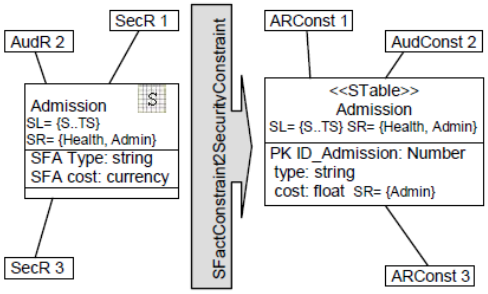

Figure 10 Applying SFactConstraint2SecurityConstraint

In the Fig. 11 we show the definition of the SBase2STable relation. This relation creates a table with a primary key, as well as a foreign key in the table that receives as parameter when it is invoked; logically the primary key and the foreign key will be associated for guaranteeing that the tables form a part of a relation one-to-many between the SBases. In the clause *where* this relation is called again, as well as the SpecializedSBase2STable relation to assure us that we cover the whole hierarchy of bases that conforms the dimension.

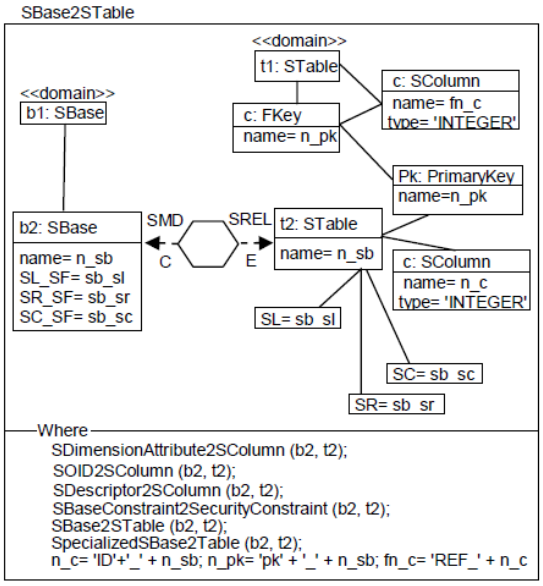

Figure 11 Transforming SBase into STable

In Fig.12 we illustrate the application of the SBase2STable relation to our case study. When the relation is executed, the City table is created with the primary key PK_City. This primary key will be associated with the foreign key that is also created in the STable Patient_Data. The City table will have associate the security property defined by means of securityLevel with confidential value. The final result is that the tables City and Patient_DataP are related.

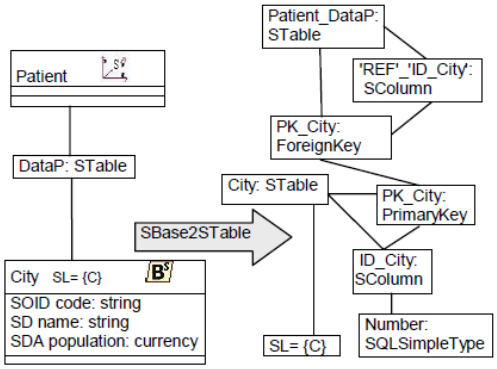

Figure 12 Applying SBase2STable

a) Dimension

In Fig. 13, fig. 14 we illustrate the applying of the SDimension2STable relation, as a result of applying this relation, the Patient_DataP table is created, with all the security properties that has associated the rootBase, in this case the security level secret and the user roles health and admin.Several of the relations that appear in the clause *where* from the SDimension2STable relation keep certain similarity with the defined ones for the SFact2STable relation, for that reason; next we are going to define the SBase2STable relation.

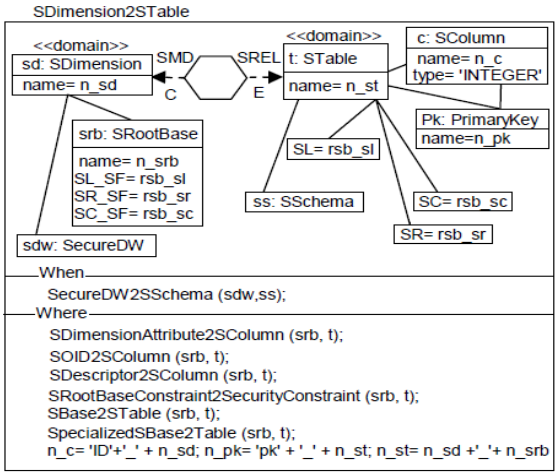

Figure 13 Transforming SDimension into STable

a) Slowly Changing Dimension2Stable

In Fig.15 we assumes that we have SCD in the schema, we illustrate how to apply QVT rules in case of we have SCD, we here assume that patient dimension is SCD,as a result of applying this relation, the Patient_DataP table is created, with all the security properties that has associated the rootBase, in this case the security level secret and the user roles health and admin.

We focus on temporal attributes as Valid time attribute (Stat_time,End_time) and dimension security privileges(DSP) that we explained earlier.

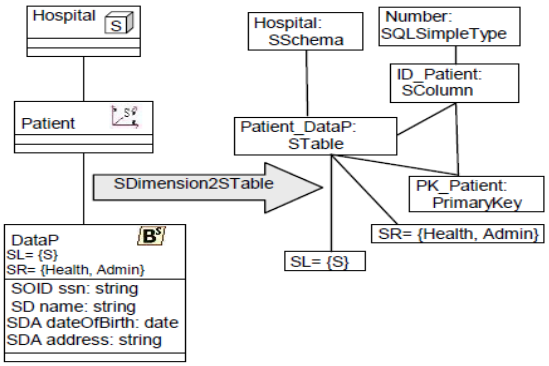

Figure 14 Applying SDimension2Stable

The dimension access security privileges (Insert,Upate,Delete) must be equal to BaseRoot access Privileges (Bsp).

In sake of the space we just clarify the SCD features and all the other features are the same in fig.13,fig.14.The first four relations that appears in the clause where of the SBase2STable relation guarantee the transformation of all SBase attributes in columns of the table that represents the SBase, as well as the transformation of all the constraints associated to the SBase in constraints associated to the table that represents the SBase. The calls to the SBase2Stable and SpecializedBase2STable relations permit to cover recursively through all the hierarchy of bases that conforms the dimension. The SpecializedBase2Stable relation has certain similarity with the SBase2Stable relation, for that reason we are not going to define it.

In Fig. 16 we have omitted the attributes in some tables, as well as the primary keys and the foreign key to make the scheme snowflake more understandable. Be observed how the security constraints have been modelled at the logical level. To complete the case study only remains apply the AssocSF_SD2FKey, AssocSDF_SD2FKey and AssocSDF_SF2FKey relations, which enable to establish relationships between SFact and the SDimensions, between the SDegenerateFact and the SDimensions and between the SFact and the SDegenerateFact.

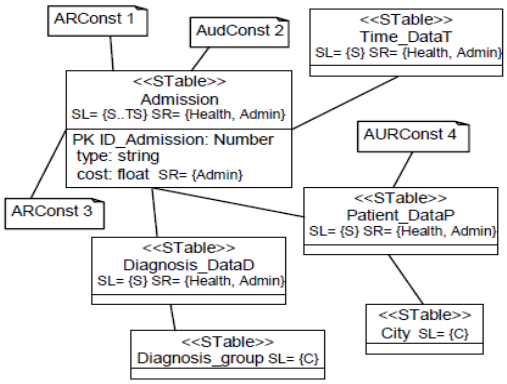

Figure 16 Snowflake schema representing a instance of the SMD PSM

In our case only proceeds to establish relations between the SFact Admission and the SDimensions, therefore we do not have SDegenerateFact. As consequence, when the AssocSF_SD2FKey relation is applied, then three foreign keys are created the Admission table. These keys enable the relationships between the Admission table with the Diagnosis_DateP, Patient_DataP and Time_DataT tables.

V. CONCLUSION

We have accomplished an important step towards completing our MDA architecture for developing secure Data Warehouses with presenting rules to transform the proposed conceptual model in [8] to logical model in this paper. The proposed model considers DW temporal and security requirements; the model is divided to two stages ETL stage and DW stage. We extend our approach by defining QVT relations in order to automatically transform the MD conceptual model into logical models that are closer to the relational implementation.

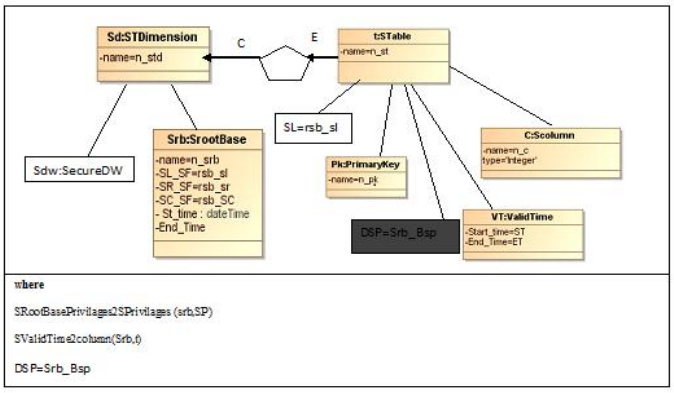

Figure 15 Slowly Changing Dimension(SCDSDimension)2STable

Our MDA architecture for developing secure DWs allows us to define security requirements and temporal issues in DW but should be extended with possibility to use the multidimensional capabilities in any commercial tools as SQL

Server Analysis Services to complete our MDA framework. Our immediate future work is to translate the security measures defined in our secure multidimensional model at conceptual model and logical model into secure multidimensional code for any commercial tool.

REFERENCES

[1] Object Management Group: MDA Guide 1.0.1, http://www.omg.org/cgi-bin/doc?omg/03-06-01,2003

[2] Object Management Group: MOF 2.0 Query/Views/Transformations, http://www.omg.org/cgi-bin/doc?ptc/2005-11-01,2005

[3] .NORBERTO MAZÓN, J.PARDILLO AND J. TRUJILLO,"APPLYING TRANSFORMATIONS TO MODEL DRIVEN DATA WAREHOUSES", DISCOVERY LECTURE, VOL 4081, PP.13-22,2006

[4] A. ABELL´O, , J. SAMOS AND F. SALTOR,"A FRAMEWORK FOR THE CLASSIFICATION AND DESCRIPTION OF MULTIDIMENSIONAL DATA MODELS". IN: DEXA.LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER VOL. M2113, PP. 668–677,2001

[5] M. Golfarelli, S. Rizzi," Methodological framework for data warehouse design". In: DOLAP, ACM (1998) 3–9

[6] N. Tryfona, F. Busborg and J.G.B. Christiansen, "starER: A conceptual model for data warehouse design", **DOLAP '99:** Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP, 1999

[7] S. Luj´an-Mora, J. Trujillo and I.Y.Song, "A UML profile for multidimensional modeling in data warehouses". Data& Knowledge Engineering ,Vol. 59, Issue 3, PP. 725–769, December 2006.

[8] M.S. Farhan, M.E. Marie,L.M.E Fangary and Y.K, Helmy, "An Integrated Conceptual Model for Temporal Data Warehouse Security". Computer and Information Science, Vol 4, No 4, pp.46-57, 2011

[9] A. Simitsis, P. Vassiliadis, and T. K. Sellis." Optimizing ETL processes in data warehouses". In Proc. ICDE,pp. 564–575, 2005.

[10] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, and S. Skiadopoulos. "A generic and customizable framework for the design of ETL scenarios". Information Systems,vol.30(7),pp.492–525, 2005.

[11] A. Simitsis. "Mapping conceptual to logical models for ETL processes" In Proc. DOLAP,2005.

[12] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos." Conceptual modeling for ETL processes". In Proc. DOLAP, pp. 14–21, 2002

[13] M. Bouzeghoub, F. Fabret, and M. Matulovic. "Modeling data warehouse refreshment process as a workflow application". In Proc. DMDW, 1999.

[14] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Information integration:" Conceptual modeling and reasoning support". In Proc. CoopIS, pp. 280–291, 1998.

[15] R. Kirkg¨oze, N. Katic, M. Stolda, and A. M. Tjoa. "A security concept for OLAP". In Proc. DEXA, pp 619–626, 1997.

[16] S. Jajodia and D. Wijesekera. "Securing OLAP data cubes against privacy breaches". IEEE Symposium on Security and Privacy Proceedings, pp. 161–178, 2004.

[17] T. Priebe and G. Pernul. "A pragmatic approach to conceptual modeling of OLAP security". In Proc. ER,pp. 311–324, 2000.

[18] E. Fernandez-Medina, J. Trujillo, R. Villaroel, and M. Piattini. "Extending UML for designing secure data warehouses". In Decision Support Systems, 2006.

[19] S. Rizzi, A. Abell´, J.Lechtenb¨orger, and J. Trujillo, "Research in Data Warehouse Modeling and Design: Dead or Alive? "In Proc. of DOLAP'06,2006

[20] M. Mrunalini, T.V. Suresh Kumar and K. Rajani Kanth, "Simulating Secure Data Extraction in Extraction Transformation Loading (ETL) Processes," Third UKSim European Symposium on Computer Modeling and Simulation, 2009

[21] J. Jürjens, "Secure Systems Development with UML", Springer-Verlag, USA,2004

[22] D. Basin, J. Doser, and T. Lodderstedt, 'Model driven security: from UML models to access control infrastructures', *ACM Transactions on Software Engineering and Methodology*, Vol. 15, No. 1, pp.39–91,2006

[23] T. Lodderstedt, D. Basin and J. Doser,"SecureUML: a UML-based modeling language for model-driven security', UML 2002, The Unified Modeling Language, Model Engineering, Languages Concepts, and Tools, 5th International Conference, Springer, Dresden, Germany,2002

[24] [24] B. Thuraisingham, M. Kantarcioglu and S. Iyer, 'Extended RBAC-based design and implementation for a secure data warehouse', International Journal of Business Intelligence and Data Mining (IJBIDM), Vol. 2, No. 4, pp.367–382,2007

[25] T. Priebe and G. Pernul "A pragmatic approach to conceptual modeling of OLAP security", 20th International Conference on Conceptual Modeling (ER 2001), Springer-Verlag, Yokohama, Japan, 2001

[26] E. Fernández-Medina, J. Trujillo, R. Villarroel and M. Piattini, "Developing secure data warehouses with a UML extension", Information Systems, Vol. 32, No. 6, pp.826–856,2007

[27] C. Blanco,I. García-Rodríguez de Guzmán, I., Rosado, D.G., E. Fernandez-Medina, J. Trujillo, "Applying QVT in order to implement secure data warehouses in SQL server analysis services", Journal of Research and Practice in Information Technology, Vol. 41, No. 2, pp.119–138..2009

[28] J.Norberto Mazón, J.Trujillo and J. Lechtenbörger." A Set of QVT Relations to Assure the Correctness of Data Warehouses by Using Multidimensional Normal Forms.", 25th International Conference on Conceptual Modeling, Tucson, AZ, USA, November 6-9, 2006, Proceedings. Volume 4215 of Lecture Notes in Computer Science, pages 385-398, Springer, 2006.

[29] E. Soler, , J. Trujillo, E. Fernandez-Medina and M.Piattini,"Application of QVT for the Development of Secure Data Warehouses: A case study" The Second International Conference on Availability, Reliability and Security, ARES 2007.

[30] J.Pardillo, J. Mazón, "Designing OLAP schemata for data warehouses from conceptual models with MDA", 2010

[31] R. Kimball and M. Ross, *The Data Warehousing Toolkit*, 2nd edition:John Wiley, 2002

[32] C. Blanco,I. García-Rodríguez de Guzmán and E.Fernández-Medina "Defining and Transforming Security Rules in an MDA Approach for DWs". *Int. J. Business Intelligence and Data Mining, Vol. 5, No. 2, 2010*