Computer Science and Information Systems 12(2):541–565 DOI: 10.2298/CSIS140617007B

An MDA approach for developing Secure OLAP applications: metamodels and transformations

Carlos Blanco[1], Ignacio García-Rodríguez de Guzmán[2], Eduardo Fernández-Medina[3], and Juan Trujillo[4]

[1] GSyA and ISTR Research Groups. Dep. of Computer Science and Electronics. Faculty of Sciences. Univeristy of Cantabria. Av. de los Castros s/n. 39071. Santander. Spain. Carlos.Blanco@unican.es

[2] Alarcos Research Group. Institute of Information Technologies and Systems. Dep. of Information Technologies and Systems. Escuela Superior de Informática. Univ. of Castilla-La Mancha. Paseo de la Universidad, 4. 13071. Ciudad Real. Spain. Ignacio.GRodriguez@uclm.es

[3] GSyA Research Group. Institute of Information Technologies and Systems. Dep. of Information Technologies and Systems. Escuela Superior de Informática. University of Castilla-La Mancha. Paseo de la Universidad, 4. 13071. Ciudad Real, Spain. Eduardo.Fdezmedina@uclm.es

[4] Lucentia Research Group. Dep. of Information Languages and Systems. Facultad de Informática. University of Alicante. San Vicente s/n. 03690. Alicante, Spain. jtrujillo@dlsi.ua.es

Abstract. Decision makers query enterprise information stored in Data Warehouses (DW) by using tools (such as On-Line Analytical Processing (OLAP) tools) which employ specific views or cubes from the corporate DW or Data Marts, based on multidimensional modelling. Since the information managed is critical, security constraints have to be correctly established in order to avoid unauthorized access. In previous work we defined a Model-Driven based approach for developing a secure DW repository by following a relational approach. Nevertheless, it is also important to define security constraints in the metadata layer that connects the DW repository with the OLAP tools; that is, over the same multidimensional structures that end users manage. This paper incorporates a proposal for developing secure OLAP applications within our previous approach: it improves a UML profile for conceptual modelling; it defines a logical metamodel for OLAP applications; and it defines and implements transformations from conceptual to logical models, as well as from logical models to secure implementation in a specific OLAP tool (SQL Server Analysis Services).

Keywords: security, confidentiality, OLAP, data warehouses, MDD, MDA, QVT, MOFScript, SSAS.

1. Introduction

Data Warehouses manage a vast amount of sensitive information which has to be properly assured, since this information has a great strategic value for organizations and, what is more, it includes private data of individuals [57, 62]. In this sense, information confidentiality is the main security-related problem that has to be tackled in the access to the

data warehouse, since end users will carry out reading operations only [45, 65, 10]. Security has traditionally been considered in the final implementation of the data warehouse, but its inclusion in early development stages can produce more robust and higher-quality solutions, due to the fact that security requirements are taken into account in making design decisions, and also because the system can accommodate them in a more natural way [17, 1]. On the other hand, given that the data warehouse design process involves the traditional development stages in which the system is modelled at business, conceptual and logical level and then eventually the final solution is implemented, the model-driven engineering approach can be applied [41, 20, 38].

There are proposals for developing secure data warehouses based on the automated development of the software by means of model definition and transformations between models, reducing the development time and cost as a consequence. Despite the fact that security constraints can be applied on the central repository of the data warehouse, they can generate inconsistencies with the access layer, in which OLAP tools that enable users to access the data warehouse are located. This is because security specifications carried out on the data warehouse repository do not use OLAP technology-related concepts (such as cubes, aggregation levels, roll-up and drill-down operations, etc.) and are not specifically defined for each multidimensional view included in the departmental data marts accessed by users. From now on, we will refer to departmental data marts, by which we mean that we are working with a data warehouse focused on a departmental data mart.

All of the above considerations lead us to see that there is a need for an approach for OLAP applications which supports the definition of security constraints over the same multidimensional elements managed by end users [57]. These multidimensional elements are managed by the end users when querying the data warehouse and demonstrate how these users interact with the data warehouse by means of OLAP operations.

This paper presents an approach for the development of secure OLAP applications, by including security in the intermediate layer that connects OLAP applications with the DW repository. In order to achieve this goal, the models and transformations needed to develop secure OLAP applications have been defined. This proposal has been also integrated into previous architecture which adopts the model-driven paradigm to model the DW repository and automatically generate its secure implementation according to a relational approach.

The rest of this paper is organized as follows: Section 2 will describe related work; Section 3 will present our MDA approach for developing secure OLAP applications, giving (1) an introduction to our conceptual model (PIM), (2) the multidimensional logical metamodel (PSM) for secure OLAP applications defined in this paper as an extension of the OLAP package from CWM and (3) the transformations created in this paper to implement the secure OLAP application automatically in a specific OLAP tool (SSAS) from the conceptual model; Section 4 will describe the application of our proposal to a DW for a sales department; and finally, Section 5 will present our conclusions and future work.

2. Related Work

This section presents relevant works related to our proposal. It has been organized in several subsections that focus on proposals for secure information systems, secure DWs and secure OLAP applications.

2.1. Security on Information Systems

There are relevant contributions concerning a complete secure development of information systems. Although they do not focus specifically on DWs and their specific security problems, they present interesting ideas, such as proposals to include security constraints from the earliest development stages, to extend UML with security aspects, to use the model-driven approach, etc. Some of the most relevant proposals are described below.

TROPOS is a methodology for software development based on the intentional goals of agents; it provides an extension called Secure TROPOS [18, 32, 8]. Together, these include security concepts (constraints, secure goals, delegation of permissions, etc.) and activities (trust of permission modelling, delegation of permission modelling, etc.), presenting a framework that allows us to model and analyze security requirements within functional requirements. Furthermore, they provide a CASE tool (ST-Tool) that supports requirements analysis and verification.

Mokum [47], which is an active object-oriented knowledge-based system for modeling, permits the specification of security and integrity constraints, along with automatic code generation.

Some work proposes processes based on security models and standards for building security systems. Such is the case, for instance, with the process known as PSSS (Process to Support Software Security) [39] which is based on the activities derived from SSE-CMM, ISO/IEC 15408, ISO/IEC 27002 and OCTAVE.

UMLsec [22] defines and evaluates security specifications using formal semantics (labels, stereotypes, etc.). It focuses mainly on accessing control policies and in the specification of confidentiality and integrity requirements. UMsec uses the majority of UML diagrams and has been recently adapted to UML 2 [23].

The model-driven approach is based on the definition on different models that separate the specification of system functionalities and their implementation by using a specific technology. With this approach, the development process can be automated by defining transformations that are able to generate the final implementation from models. This approach has been successfully applied to different software development areas, improving the development process, thus reducing times and costs, as well as the quality obtained in the final product [16, 36]. Some examples of application are: data bases [61, 27, 11, 60], data warehouses [34, 62, 12], web services [35, 54, 67, 26], functional usability, product lines [7], data schema mappings using waving metamodels enriched with OCL constraints, critical applications or real-time systems [9, 30, 64].

Model-driven Security (MDS) [4] applies the model-driven approach to include security properties in high-level system models and to automatically generate secure system architectures. For the modelling of the system it proposes a UML extension called SecureUML [29] that permits the inclusion of access control aspects into the models. MDS has been applied to UMLsec [5], defining three abstraction levels (requirements, modelling and implementation) and providing tools for code generation, reverse engineering, verification and configuration [24]. Furthermore, some pieces of work are developing transformations between SecureUML and UMLsec [33].

2.2. Security on DWs

A typical DW architecture is composed of several layers: heterogeneous data sources; ETL (extraction/transformation/load) processes, which extract and transform data from

these data sources and load the information into the DW; the repository of the DW, where data are stored; and DBMS and OLAP tools which analyze data. Since DWs have mainly dealt with read operations on sensitive information used for decision making, the main security problem related to DWs is information confidentiality; this should be taken into account in all layers and operations of the DW. Each layer of this architecture presents specific security concerns, which are described in the rest of this section.

Since data sources are heterogeneous and can use different security policies (such as discretional access control - DAC, mandatory access control - MAC or role based access control - RBAC), the security problem concerning this layer is related to their integration into the DW design, and there is a similar problem in Federated Information Systems (FIS).

Security policies defined in data sources are specified bearing in mind that users of data sources could be different from the end users. We believe that the integration of these is not enough in and of itself to completely establish the DW security policy. There are indeed sensitive data which should be assured regardless of end users, however (for instance, to comply with legal regulations) and the integration of security constraints for these could be an interesting starting point for establishing a DW security policy.

There are several pieces of work that deal with the integration of different security policies in FIS [56, 21]. Saltor et al. [48] use this parallelism to adapt design architecture for FIS to DWs; they also set out to improve it with security capabilities supporting the integration of MAC policies. Since ETL processes extract and transform information from data sources which is finally loaded into the DW, it is important that ETL processes also take security information into account. However, although proposals for conceptual modeling of ETL processes exist, they do not support security issues [52, 51, 58, 37].

As regards a complete secure DW development, we found only the methodology of Priebe and Pernul [46], in which the authors analyze security requirements and their implementation in commercial tools by hiding multidimensional elements such as cubes, measures, slices and levels. They extend their proposal with a DW representation at conceptual level with ADAPTed UML, but they do not establish the connection between models so that automatic transformations can take place.

On the other hand, there are several pieces of work which focus on secure modeling for DWs at certain abstraction levels. At business level there are proposals based on ontologies, business process, UML, etc. but only Paim and Castro [43] include security requirements; they do not offer any formal metamodel, however.

At the conceptual level there is some interesting work for modeling DWs that considers their special characteristics by using extensions of the ER model, UML or a notation of their own, but they do not include security capabilities [19, 49, 59, 6, 31, 44]. The conceptual modeling of security issues is considered only by the AdaptedUML of Priebe and Pernul [46].

Traditionally, multidimensional modeling at logical level has depended on the DBMS used and in this way in online analytical processing can be classified mainly into relational (ROLAP), multidimensional (MOLAP) and hybrid (HOLAP) approaches. There are many modeling proposals which do not consider security; only CWM [40] provides a formal metamodel with relational and multidimensional packages.

A number of interesting proposals have come into being recently in the context of security in cloud data warehouses and big data [55]. These include proposals such as an

approach for securing cloud data warehouses by flexible verifiable secret sharing [3, 2], a security framework in Hadoop for big data computing across distributed Cloud data centres [68], or a solution for privacy that preserves medical data sharing in a cloud [66].

2.3. Security on OLAP applications

End-user tools also have to consider security constraints in their quest to avoid unauthorized access. Research efforts have traditionally been carried out in this way, but they have focused on the final stage of development without including security issues in the whole development process. For instance, Kirkgoze et al. propose a virtual cube for each subject [25], while Weippl et al. define an access control model for DWs; OLAP, for its part, makes it possible to define the OLAP operations authorized for each user [65].

Nevertheless, the problem of inference is still a challenge in present-day DW security, and it is an important research branch [57]. Since DWs store sensitive data by using different aggregation levels with a range of confidentiality requirements (for example, the average salary may be Unclassified, and individual salaries Secret), the inference problem is similar to the problem that has already been studied as regards statistical databases which store summarized data as sums or averages [50]. Some pieces of work have dealt with inference by proposing query control systems [63, 28, 53].

2.4. Discussion

Our research efforts focus on considering security in the whole DW development process, from the early stages of the development lifecycle. Our proposal defines several models which are improved with security capabilities, and aligns them with an MDA architecture, also providing the transformation rules needed to generate secure implementation automatically.

After analyzing the related work, we consider that the proposals described above present interesting ideas about how to include security in the whole development process of information systems, but none of them has data warehouses and OLAP as their focus of attention. This kind of systems presents specific structural characteristics (facts, dimensions, hierarchies, measures, etc.) as well as security aspects that have to be correctly specified for them.

Our analysis continued by describing some proposals for securing a specific DW development stage (data source integration, ETL and DW modelling). The most interesting one for the scope of this work was AdaptedUML, which we were able to use in our proposal for the conceptual modelling stage. Nevertheless, since AdaptedUML does not establish the connection between models in order to allow automatic transformations, we decided to use the SECDW UML profile that we had defined in previous work.

Finally, we focused on security models for a specific technology, OLAP. Since end users access the data warehouse by using OLAP tools, security constraints need to be be defined according to the concepts associated with the OLAP technology (cubes, aggregation levels, roll-up and drill-down operations, etc.). The proposals analysed center on the implementation stage, defining specific access control models, or extending languages used by the OLAP tools (such us MDX). Our work aims to define a model for OLAP, but at a high level of abstraction, separate from the implementation; in other words, our objective was to establish a common model that represents both the structural and the security

aspects, independently of the particular end-user OLAP tool to be used. This model can thus be employed to generate automatically the final implementation for a specific tool through transformations.

Taking all the above points into account, this paper presents a solution for developing secure OLAP applications that consider the whole development process, including security aspects from the earlier stages and allowing us to define security constraints associated with the same views, as well as multidimensional elements which will be managed by the end users. This proposal is based on models and transformations, defining a model-driven architecture able to generate the secure implementation from models automatically. Designers use our conceptual model to establish security constraints in a simple way and then the system is deployed accordingly. This is carried out in two stages; firstly, a logical model is automatically generated through transformations and then from this logical model we obtain automatically the final implementation that includes all the MDX expressions needed for each role. The system deployed is consistent with the conceptual model, and each change made in the model is deployed by following the same automatic process.

3. An MDA architecture for developing Secure OLAP Applications

This section presents our approach for developing secure OLAP applications, so that the reader may understand the context of this paper. Previous work did form a complete architecture (Figure 1) but focused solely on a relational approach and the final implementation in DBMS [13]. At the business level, a computational independent metamodel (CIM) supports an early definition of the security requirements by using a UML profile based on the i* framework.

A platform independent model (PIM) makes it possible for us to define the secure conceptual model of the DW. At this stage we use a UML profile called SECDW. It is based on a UML profile created specifically for conceptual modeling of DWs, complemented with an Access Control and Audit (ACA) model. It permits a classification of subjects and objects into hierarchies of roles (SR), horizontal classifications of compartments (SC) and clearance levels (SL). Furthermore, the ACA model enables there to be a definition of security rules: sensitive information assignment rules (SIAR); authorization rules (AUR); and audit rules (AR).

Following a relational approach at logical level (ROLAP), a Specific Platform Metamodel (PSM) called SECRDW (Secure Relational Data Warehouse) has been defined as an extension of the relational package from the Common Warehouse Metamodel [40]. SECRDW models relational elements such as tables, columns, or keys, and expresses the security rules defined at conceptual level. Moreover, the automatic transformation from conceptual models to relational logical models has been implemented by using QVT; the eventual implementation into a DBMS has been dealt with by Oracle Label Security.

Since most DWs are queried by OLAP tools following a multidimensional approach, this paper completes our architecture by including a new logical multidimensional model for secure OLAP applications, called SECMDDW (Secure Multidimensional Data Warehouse) and the final implementation in an OLAP tool (SSAS). Furthermore, the transformations needed to integrate that model into our architecture (T1 and T2 in Figure 1) have been defined. All these elements will be explained in the following sections.

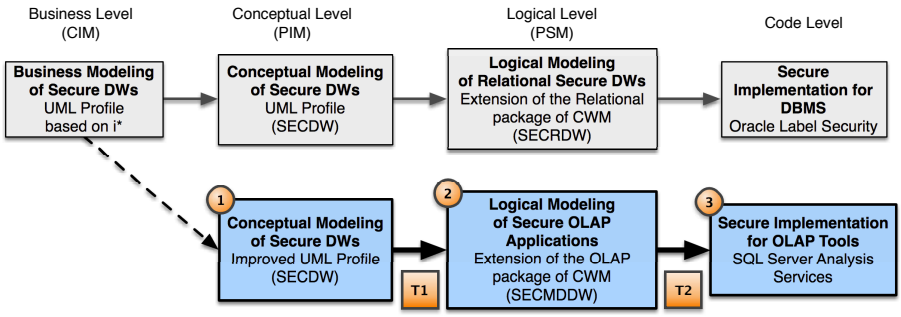

Fig. 1. Model driven architecture for developing Secure DWs.

The process of applying our approach is composed of several stages.

- Firstly, designers use our conceptual model to establish security constraints in a simple way by using our UML profile. They model both the structural and the security aspects of the DW.
- The system is automatically deployed in accord with this model and it is consistent with the conceptual model. That is, each change made in the conceptual model is deployed by following the same automatic process. The deployment process is achieved by applying the sets of transformation rules defined in our proposal. This transformation is composed of two steps:
  - The transformation from conceptual to logical model. This is a model-to-model transformation that generates a logical OLAP model that is independent from the end-user tool used. This transformation has been defined by using QVT rules. These allow us to establish relational mappings between models, in which each mapping or rule defines what particular elements we wish to check in the source model, which element we create or modify in the target model, and how that mapping has to be done.
  - Code generation from the logical model. This is a model-to-text transformation that generates the final implementation for a specific OLAP tool. This transformation has been defined by using MOFScript transformations that indicate mappings between the source model and target text files.

3.1. An introduction to the SECDW UML profile

SECDW [15] is a UML profile which has been defined previously in order to allow secure conceptual modeling of the DW. It is based on a UML profile created specifically for conceptual modeling of DWs,complemented with an Access Control and Audit (ACA) model [14] which includes security capabilities in the conceptual design by considering several security policies (Discretional Access Control, DAC; Mandatory Access Control, MAC; and Role-Based Access Control, RBAC). The SECDW has been improved in this paper so as to include support for the definition and transformation of complex security rules.

Figure 2 shows the SECDW profile. It permits the conceptual modeling of DW structural aspects by using packages (SecurePackage metaclass); classes (SecureClass) for facts (SFact), dimensions (SDimension) and bases (SBase); properties (SecureProperty). The security configuration of the system which we wish to model is defined by using three points of view: a hierarchical structure of Security Roles (SRole); a list of Security Levels (SLevel) with the clearance levels of the users; and a set of horizontal Security Compartments or groups (SCompartment). Once this configuration has been established, sets of certain security configurations composed of roles, levels and compartments can be defined as instances of secure information (SecureInformation). The security configuration for the elements of our conceptual model (packages, classes, attributes, etc.) is thus established by associating them with specific secure information instances. In addition, since the user profile (UserProfile) defines all the properties that the systems manage from users, it also has security information associated with it.

The ACA model enables three kinds of security rules to be defined (SConstraint) for the different elements of the DW by using Object Constraint Language (OCL) notes: the definition of sensitive information for multidimensional elements by sensitive information assignment rules (SIAR); the authorization or denegation of certain elements to specific subjects by authorization rules (AUR); and the inclusion of audit rules (AR) to ensure that authorized users do not misuse their privileges.

These rules could be complex, involving information about subjects, objects, conditions, security information, privileges, log types, etc. They were represented in the SECDW profile by using OCL notes associated with a certain multidimensional element, but these OCL expressions are difficult to analyze and transform in an automatic way. It is for that reason that in this paper the SECDW has been improved to provide a better representation and management of complex security rules, by including the information needed to support their transformation. This improvement does not affect our complete MDA architecture and our previously-defined transformations.

Security rules have been included as subclasses of SConstraint. SIAR security rules therefore use a SecurityRule metaclass representing conditions with boolean expressions and the secure information that will be assigned, whether the condition is satisfied or not. AUR rules use an AuthorizationRule metaclass with information about the security information associated, the sign of the authorization (positive or negative), the privilege (read, insert, update, delete, all) and a boolean expression for conditions. Finally, AR rules are specified by an AuditRule metaclass defining the access attempt and the logged information (subject, object, action, time and response).

Finally, our ACA model also includes a process of conflict resolution. Each structural element can have its security labels associated ; this operation is carried out by designers who could very well assign incorrect security privileges. When aggregations are carried out, the security labels defined are considered but this conflict resolution process is also applied.

3.2. Logical Metamodel for Secure OLAP Applications

This section presents a multidimensional logical metamodel (PSM), called SECMDDW (Secure Multidimensional Data Warehouse), which enables there to be a specification of both the structural and the security aspects that are closest to OLAP applications. The SECMDDW metamodel has been developed based on the CWM metamodels [40], which

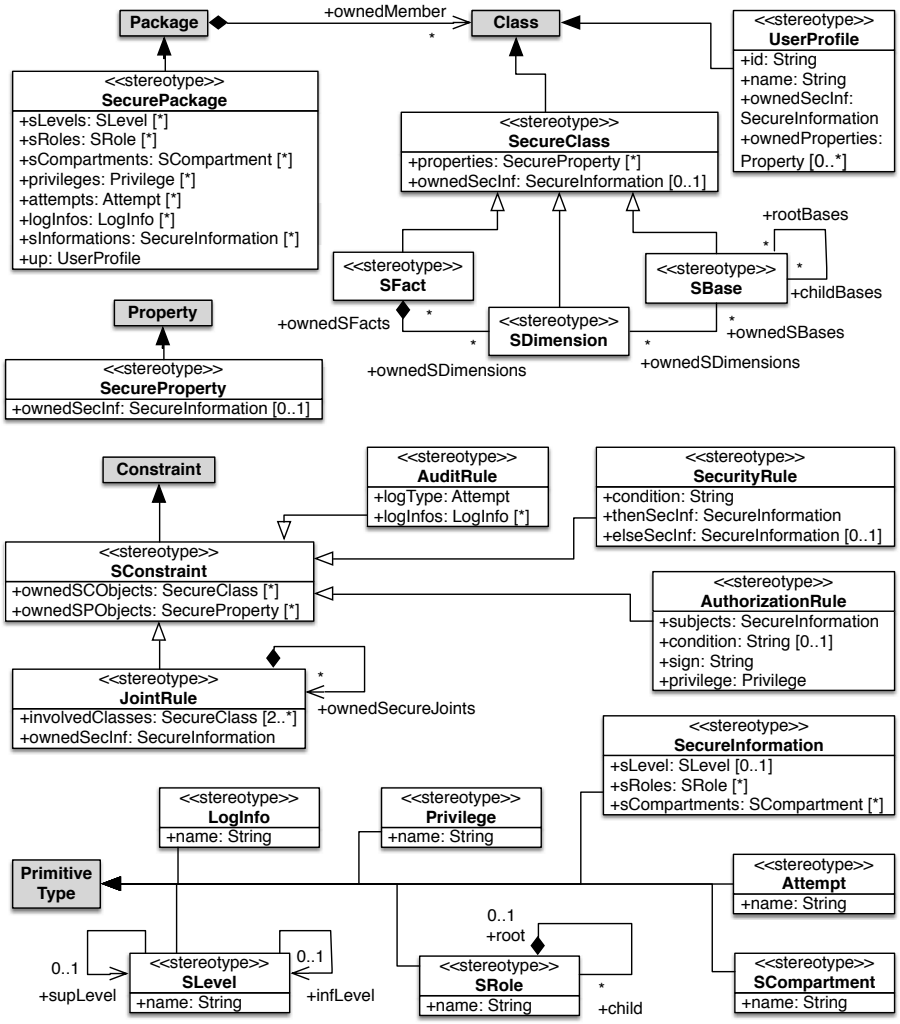

Fig. 2. Profile for conceptual modelling of Secure DWs (SECDW)

is the OMG proposal for representing and interchanging metadata for data warehousing and business intelligence. SECMDDW extends the OLAP package of CWM which focuses on data analysis and defines a metamodel of essential OLAP concepts common to most OLAP applications.

Figure 3 shows the logical metamodel for the secure OLAP applications developed. Three main parts can be identified: security configuration (the upper part of the Figure 3), cubes (middle part) and dimensions (bottom part). The security configuration of the system is represented by considering a role-based access control (RBAC) policy which is used by most of the OLAP tools. Since conceptual models are more abstract than logical models, there is a semantic loss when we move towards lower levels. At the conceptual

level, the SECDW considers the definition of the security configuration by using roles, levels and compartments, but at the logical level our metamodel focuses only on RBAC and security levels, and compartments should be adapted as security roles. That is, the security configuration is defined as a set of security roles (Role metaclass) with relationships between them, creating a hierarchy (child and parent properties).

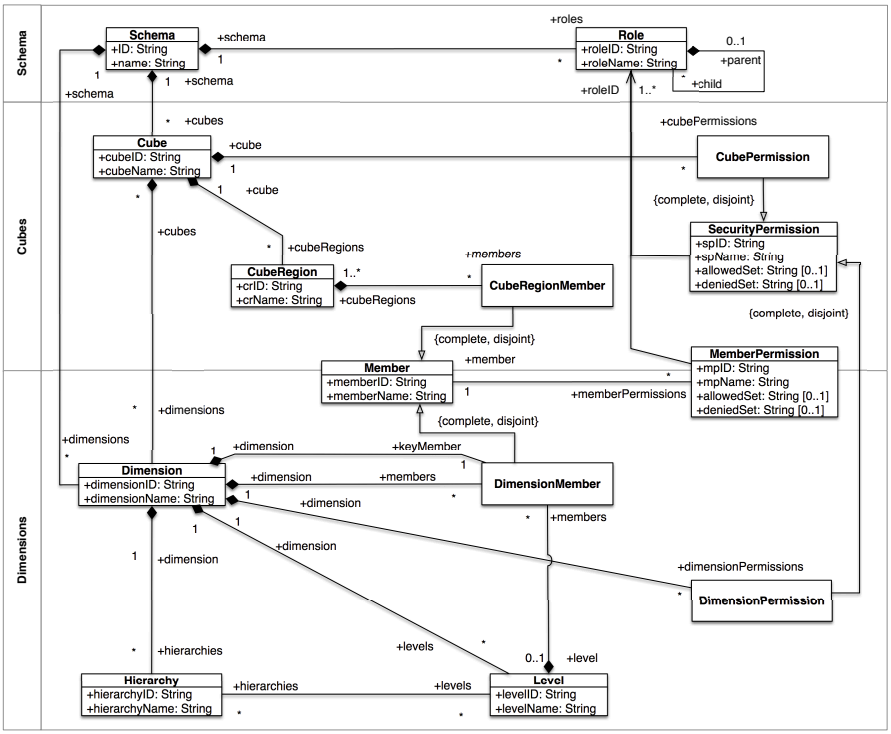

Fig. 3. Logical Metamodel for Secure OLAP applications (SECMDDW).

The remaining parts of our metamodel define both structural and security aspects for cubes and dimensions. Fact classes from our conceptual model are defined at the logical level as cubes (Cube metaclass), and their related attributes are defined as measures (CubeRegion and CubeRegionMember metaclasses). In addition, conceptual dimension classes are defined as dimensions (Dimension metaclass) with a set of attributes and an identifier (DimensionMember metaclass). Base classes from conceptual models are specified as attributes related to the corresponding dimension, therefore creating the necessary hierarchies (Hierarchy and Level metaclasses).

Considering the abstraction gap between the conceptual and logical levels, security rules (SIAR and AUR) from conceptual models (including complex security rules defined with OCL expressions) have been represented at the logical level by using sets of security permissions associated with cubes, cells, dimensions and attributes. Nevertheless, since

OLAP platforms provide specific auditing tools which are directly managed by administrators, audit rules (AR) have not been transformed into the logical model.

The SIAR and AUR security rules specified in conceptual models related to fact classes are defined in logical models as permissions associated with the corresponding cubes (CubePermission metaclass). Each cube permission is related to a certain security role (roleID attribute) and uses expressions to establish the information which has to be shown or hidden for that role (allowedSet and deniedSet attributes). These expressions are defined in MDX syntax, and once the system is deployed they are used to evaluate at run time the information that has to be provided to the user. Since our model allows security rules to be established in a positive or negative way, two sets have been defined (allowedSet for expressions that grant access and deniedSet for denying access). In this way, each permission can be defined by one set (allowed or denied) or by the combination of both.

This metamodel also makes it possible to establish fine grain permissions over cube measures by using member permissions associated with the corresponding measure (MemberPermission metaclass).

In addition, security constraints established in conceptual models involving dimension and base classes are defined at the logical level as permissions associated with dimensions (DimensionPermission metaclass) by including information about the security role (roleID attribute) and conditions with the information which can be accessed or not for that role (allowedSet and deniedSet attributes). The definition of fine grain security constraints is also permitted, with the use of attribute permissions associated with the corresponding attributes (MemberPermission metaclass).

3.3. Generating a Secure OLAP Implementation

This section shows the transformations which have been developed in order to automatically generate the secure OLAP implementation from conceptual models. This process is composed of two stages: a model-to-model transformation from conceptual models to logical models; and a model-to-text transformation from logical model to the final implementation into a specific OLAP tool, SSAS. Although it is possible to find many languages to implement transformations (QVT [42], ATL, MOFScript, JET, Xpand, etc.), in this paper we use the standards proposed by the OMG, QVT and MOFScript.

Transformation from Conceptual Models into Logical Models for OLAP Applications At the first stage we specify the transformation rules needed for the generation of logical models for secure OLAP applications (according to SECMDDW) from conceptual models (according to the SECDW). The mapping between conceptual and logical models has been organized into five main QVT transformations for processing (i) the security configuration, (ii) facts, (iii) dimensions and (iv and v) security rules. Each transformation is composed of a top relation that starts the execution of several auxiliary rules (relations).

Processing Security Configuration. Firstly, the transformation SECDW2Role generates the security configuration for the OLAP system by using a role-based access control (RBAC) policy. Our conceptual model is richer than our logical model, allowing the definition of security roles (SR), levels (SL) and compartments (SC). All this information has to be adapted to role hierarchies inferring the necessary security roles.

The top relation SecurePackage2RoleSchema explores the source model (the conceptual model), looking for the security information associated with the DW (that is composed of security levels, roles and compartments). For each SecurePackage found, a Schema element is created in the target model (the logical model). Then the relations SCompartment2Role, SLevel2Role and SRole2Role generate roles for each security compartment, level and role defined in the secure package. Furthermore, they associate these roles in hierarchies, in order to represent the structures defined in the conceptual model (a sequence of levels, a hierarchy of roles and a set of compartments).

Processing Facts. The transformation SECDW2Cube analyzes both the structural aspects and the security information associated with fact classes, creating in the logical model cubes, measures and security permissions attached to these multidimensional elements.

This transformation is composed of a top relation SPackage2CubeSchema, as well as several auxiliary rules for structural aspects and security permissions for cubes and measures. Firstly, structural rules create for each secure fact class a cube with a measure group, composed of several measures (fact attributes in the conceptual model). After this, security relations process the security information (roles, levels and compartments) associated with fact classes and their attributes, in order to generate security permissions associated with cubes and their measures (cube cells) in the logical model. These security permissions are adapted to the RBAC policy used in the logical level, including also the multidimensional expressions (MDX) needed to specify the information that have to be shown and hidden with respect to each role.

Using the QVT graphical syntax Figure 4 shows a detailed view of the SFact2Cube relation. The way in which the security information associated with SFact classes and their attributes are processed by other relations can be shown in the *where* clause of the relation (see bottom side of the Figure 4). Firstly, security compartments, roles and levels defined over SFact classes are analyzed, creating security permissions that affect the Cube as a whole (CubePermission). For instance, the relation SCompartmentClass2CubePermission focuses on the security compartments, and creates cube permissions authorizing the access to the cube for the roles involved (which are the representation of these compartments as roles in the logical model). The relation DenySCompartmentClass2CubePermission then creates cube permissions, denying access to the remaining security compartments.

Processing Dimensions and Bases.

The secure dimension and base classes defined in the conceptual model are processed by the transformation SECDW2Dimension, which generates dimensions, bases, attributes, hierarchies and security permissions in the logical model, defined over dimensions and attributes.

In the logical model, this creates dimensions with a key attribute and the remaining attributes. In addition, the remaining structural relations analyze the secure base classes, defining classification hierarchies, aggregation levels and attributes. The remaining relations then go on to process the security information (roles, levels and compartments) associated with dimension classes, base classes or their attributes. These relations transform the security constraints to the semantics used in the logical model; that is, they define security permissions using an RBAC security policy and MDX expressions to show or hide certain multidimensional OLAP elements (a dimension, an aggregation level, etc.).

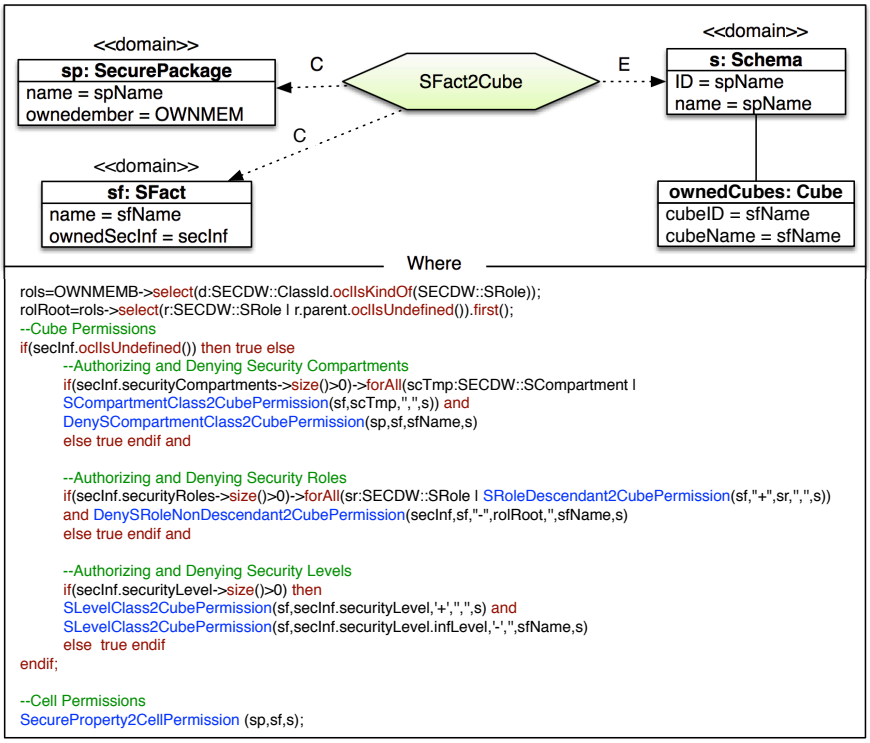

Fig. 4. Graphical notation for the relation SFact2Cube.

Processing Security Rules. Finally, the security rules (SIAR and AUR) defined in the conceptual model are processed. Since these rules can involve several multidimensional elements and the evaluation of conditions, it is necessary to create additional security permissions with MDX expressions, which are associated to cubes and dimensions (or to cube cells and dimension attributes if fine grain security permissions are needed). The transformation SecurityRule2CubePermission processes security rules attached to secure fact classes and their attributes, while SecurityRule2DimensionPermission processes those associated with dimension and base classes and their attributes.

SIAR rules are transformed by analyzing the security information expressed in the SIAR and generating security permissions for the authorized and unauthorized security roles at the logical level (which represent security compartments, roles and levels from the conceptual model). These security permissions include MDX expressions with the condition established in the SIAR, in order to hide the instances that satisfy the condition. AUR rules are transformed into a set of positive or negative permissions, depending on the AUR sign. The set of permissions created authorizes or denies all the roles that are affected by the rule.

Generating a Secure SSAS Implementation from Logical Models Logical models defined according to SECMDDW are very close to OLAP platforms. They define the structure and security of the DW by using multidimensional OLAP elements; the automatic transformation towards a secure implementation for different OLAP tools can thus be easily achieved by adapting this information to the syntax of each OLAP tool. To incorporate support for a specific OLAP tool in our architecture it is necessary to define a model-to-text transformation, from the logical model to the specific syntax of the end-user tool.

In this paper, we have developed the set of MOFScript transformations needed to obtain a secure implementation for SSAS automatically. SSAS uses several kinds of XML files (with role, cube and dim file extensions) to manage information about security roles, cubes and dimension. The MOFScript transformation developed to generate the secure SSAS implementation from logical models has therefore been grouped into three sets, namely security configuration, cubes and dimensions.

In the first place, the security configuration established in the logical model by using a RBAC policy is processed. For each role, an XML file (with role file extension) with information about its members is created. Structural and security aspects related to cubes are then analyzed. This creates a cube file (cube file extension) for each cube detected in the logical model, including information about this cube and its measures. Cube files are then fulfilled by other transformations which include the remaining information about hierarchies and security permissions defined for cubes and cells.

Finally, dimensions are processed by the transformation shown in Listing 1.1. That creates the dimension files needed (dim file extension), as well as the structural aspects related to dimensions. After that, the security permissions defined for this dimension, or their attributes, are analyzed; the security information needed is included in the corresponding dimension file: information about processing and reading privileges, MDX expressions defining the sets which are denied and allowed for each role, etc. (see Listing 1.1).

Listing 1.1. MOFScript transformation: dimensions

```
texttransformation dimensionPermissions (in psm: SECMDDW ){
psm.Schema::main(){
 self.cubes->forEach(c:psm.Cube){
  c.dimensions->forEach(d:psm.Dimension){
     file dimfile(d.dimensionID +    .dim );
     dimfile.print(   <DimensionPermissions>   );
     d.dimensionPermissions
     ->forEach(dp:psm.DimensionPermission){
      dimfile.print(   <DimensionPermission>   );
      dimfile.print(   <ID>   +dp.dpID+   </ID>   );
      dimfile.print(   <Name>   +dp.dpName+   </Name>   );
      dimfile.print(   <RoleID>   +dp.roleID+   </RoleID>   );
      dimfile.print(   <Process>   +dp.process+   </Process>   );
      dimfile.print(   <Read>   +dp.read+   </Read>   );
      dimfile.print(   <AllowedSet>   +dp.allowedSet+   </AllowedSet>   );
      dimfile.print(   <DeniedSet>   +dp.deniedSet+   </DeniedSet>   );
      dimfile.print(   <AttributePermissions>   );
      dp.attributePermissions
      ->forEach(ap:psm.AttributePermission){
       dimfile.print(   <AttributePermission>   );
       dimfile.print(   <AttributeID>   +ap.attributeID+   </AttributeID>   );
       dimfile.print(   <AllowedSet>   +ap.allowedSet+   </AllowedSet>   );
       dimfile.print(   <DeniedSet>   +ap.deniedSet+   </DeniedSet>   );
       dimfile.print(   </AttributePermission>   );}
      dimfile.print(   </AttributePermissions>   );
```

```
dimfile.print(    </DimensionPermission>    );}
dimfile.print(    </DimensionPermissions>   );} } }
```

4. Application Example

This section describes the application of our proposal for the development of an OLAP application for a sales department. The DW used in this example analyzes sales according to different perspectives (products, dates, customers and stores), and also considers several security constraints. First of all, the system is modeled at a conceptual level. The secure logical model for OLAP applications is then obtained automatically (by applying the QVT transformations defined). Finally, the secure implementation for SSAS is obtained from the logical model automatically (by applying the MOFScript rules developed).

4.1. Conceptual Model

Figure 5 shows the conceptual model for the Sales DW, defined according to the SECDW. This example is composed of a central fact Sale (secure fact class) with measures amount and quantity, which can be classified by using different dimensions Product, Store, Date and Customer (secure dimension classes). Furthermore, different aggregation levels have been defined (by using secure base classes) for Products which can be grouped by Category and by Customers, which can in turn be grouped by City.

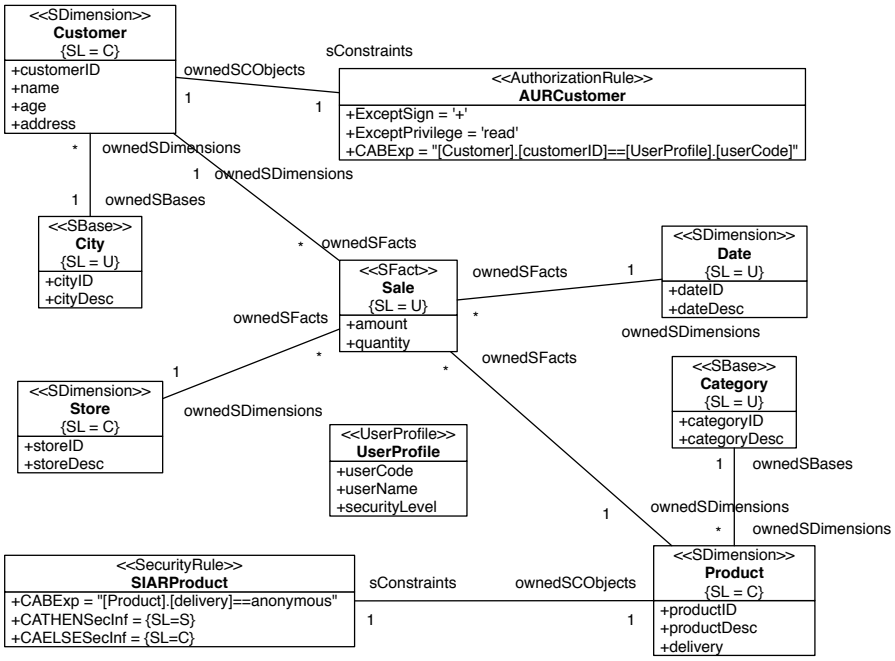

Fig. 5. Example: Conceptual Model (PIM).

In addition, the security configuration is established. Although our proposal allows us to specify security roles, levels and compartments, this example considers only the following set of security levels: secret (S), confidential (C) and undefined (U). The user profile therefore stores the security level associated with the user, as well as information about their identification and name. Security constraints have been established for multidimensional elements by using this security configuration, that is, by using security information sets composed of a specific security level requiredThe security privileges needed to access fact, dimension and base classes are defined first of all (as tagged values): a security level of C is required to access Product, Store and Customer dimensions; and a level of U for the fact Sale, dimension Date and bases Category and City.

Several security rules complement the model, moreover. A security rule (SIAR) is attached to the Product dimension, which increases the security level needed to access sales information grouped by Product from C to S if the kind of delivery is anonymous (the delivery property from the Product dimension). An authorization rule (AUR) attached to Customer allows each user to access their own customers information (although the users security level was lower than the one required for Customer, which is C).

4.2. Logical OLAP Model

Once the conceptual model has been defined, the transformation rules developed are applied, in order to obtain a logical model for secure OLAP applications (according to SECMDDW metamodel). The resulting logical model is shown, split into several figures. The security configuration defined is analyzed first of all, by the transformation SECDW2Role. In this case, as can be seen in Figure 6, several roles (SLS, SLC and SLU) are created for each security level from the conceptual model (S, C and U).

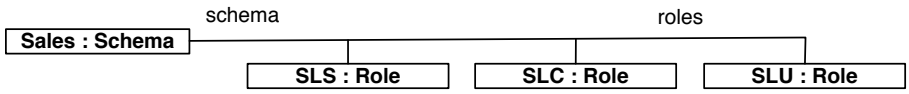

Fig. 6. Example: Logical Model (PSM). Security Configuration.

Secure fact classes defined in the conceptual model are processed by SECDW2Cube transformation. The secure fact class Sale generates a cube (Cube) with a measure group (MeasureGroup) composed of two measures (Measure): amount and quantity (see Figure 7). The security level required for access to the Sale secure fact class (a U security level) is represented in the logical model as a set of positive security cube permissions which grant access (process and read attributes) to the cube Sale (allowedSet attribute) for the role SLU (roleID attribute), as well as for the roles SLC and SLC, since these represent users with higher security privileges (upper security level).

After this, both the structural and the security aspects related to dimension and base classes are analyzed by the transformation SECDW2Dimension. As Figure 8 shows, each secure dimension class from the conceptual model generates a dimension (Dimension) in the logical model, with associated attributes and a key attribute for each dimension (Member). The different classification hierarchies defined for each dimension with secure

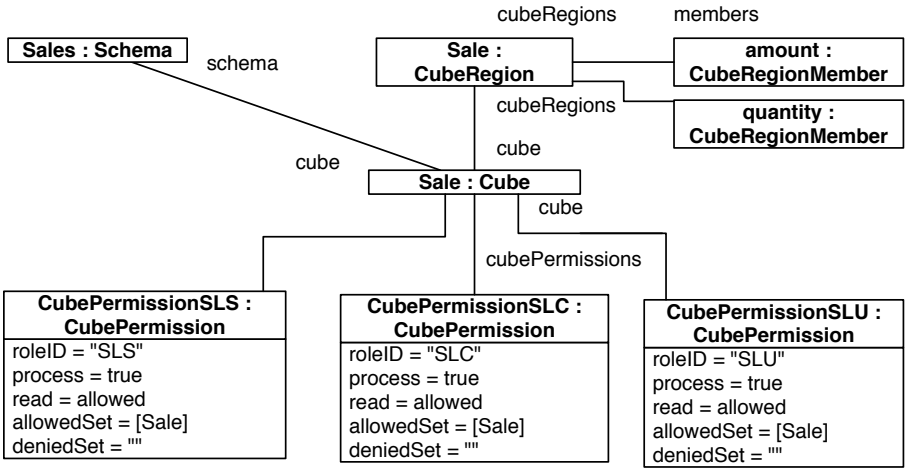

Fig. 7. Example: Logical Model (PSM). Cubes.

base classes in the conceptual level are specified in the logical model as hierarchies (Hierarchy) and different aggregation levels (Level) associated with dimensions (for instance, Customer can be grouped by City). The properties of the base are represented as dimension attributes in the logical level (for instance, cityID and cityDesc from the base City are now attributes of the Customer dimension).

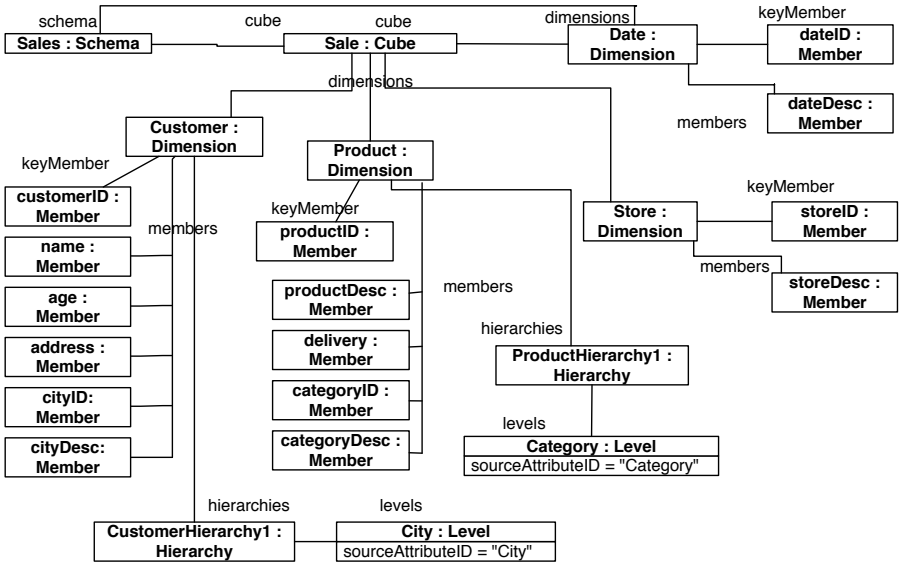

Fig. 8. Example: Logical Model (PSM). Dimensions Structure.

The security privileges needed to access dimensions and bases are modeled at the logical level as dimension permissions (as can be shown in Figure 9). Since a security level of C is required to access the dimension Store, three security dimension permissions have been defined: two of them to grant access for users with roles SLS and SLC (security levels S and C), and one of them to deny access for users with an SLU role (security level U). The remainder of the dimension permissions in Figure 9 is associated with the dimension Date and these are obtained in a similar way.

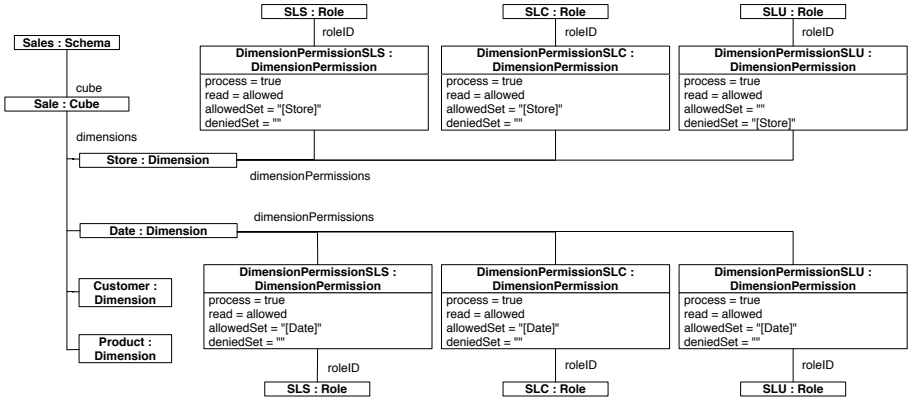

Fig. 9. Example: Logical Model (PSM). Dimensions Security.

The dimension and attribute permissions needed at the logical level to represent the security constraints that affect the Product dimension and their associated base class Category are shown in Figure 10. First, dimension permissions are created for each role (SLS, SLC and SLU). The security level required to access Product is C, but there is also a security rule (SIAR) that increases this to S when the kind of delivery is anonymous. The transformation SecurityRules2DimensionPermission processes this SIAR, generating dimension permissions that represent this situation by using allowed and denied sets: SLS role can access all products, with anonymous delivery or not (thus, the allowedSet is set to [Product]); SLC role can only access products with non- anonymous delivery (deniedSet set to [Product].[delivery] == anonymous); and SLU role cannot access any product (deniedSet set to [Product]). Nevertheless, the last dimension permission denies all accesses to products for the SLU role, and in the conceptual model, the security level required for the base class Category is U. In order to provide access to Category information (categoryID and categoryName properties) for the role SLU, positive attribute permissions for the attributes CategorycategoryID and CategorycategoryName are created in the logical model attached to the dimension permission corresponding to the role SLU.

Dimension permissions associated with the Customer dimension are shown in Figure 11. In the conceptual model a security level of C for Customer dimension was established, and U was determined for the City base. Security dimension permissions are created in the logical model for each role: granting access for SLS and SLC, and denying access for SLU. Furthermore, since users with a U security level cab see City information, positive attribute permissions for their attributes (CitycityID and CitycityName) are

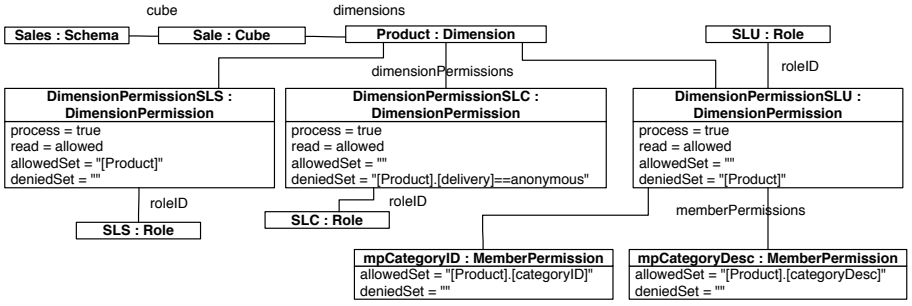

Fig. 10. Example: Logical Model (PSM). Security Rules.

attached to the dimension permission for the role SLU. Nevertheless, an authorization rule (AUR) allows users to access their own customer information. The transformation SecurityRule2DimensionPermission changes the information needed to represent this constraint in the logical model. In this case, since SLS and SLC roles can access all Customer information, only the dimension permission for the SLU role has to be modified, by setting the allowed set to the condition [Customer].[customerID]==[UserProfile].[userCode].

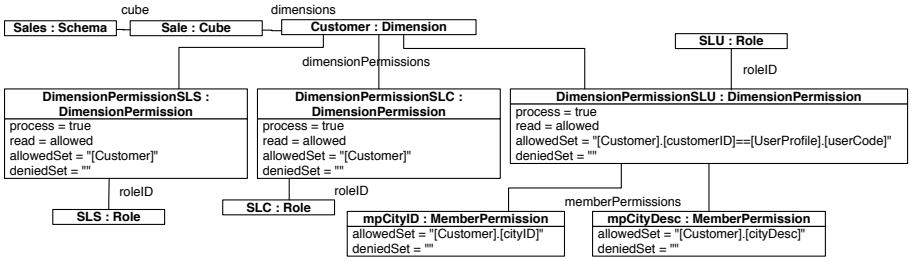

Fig. 11. Example: Logical Model (PSM). Authorization Rules.

4.3. Implementation into SSAS

The logical model obtained in the previous section is now transformed into secure code for SSAS by applying the MOFScript rules defined. This transformations generate different kind of files: for each security role (files SLS.role, SLC.role and SLU.role); for the Sales cube (Sales.cube file); and for each dimension (files Product.dim, Customer.dim, etc.). Listing 1.2 shows a piece of code from the Product.dim file, in which it can be shown how the dimension permission for the role SLU has been represented for SSAS by using its XML syntax and MDX expressions for the allowed and denied sets.

Listing 1.2. MOFScript transformation: dimensions

```
<Dimension>
<ID>Product </ID> <Name>Product </Name>
```

```
<!--Structural aspects have been omitted-->
<DimensionPermissions>
<!--Dimension Permissions for SLS and SLC have been omitted-->
<DimensionPermission>
 <ID>DimensionPermissionSLU </ID>
 <Name>DimensionPermissionSLU </Name>
 <RoleID>SLU</RoleID>
 <Process>true </Process>
 <Read>Allowed </Read>
 <AllowedSet></AllowedSet>
 <DeniedSet>[Product]</DeniedSet>
 <AttributePermissions>
  <AttributePermission>
  <AttributeID>CategorycategoryID </AttributeID>
  <AllowedSet>[Product].[CategorycategoryID]</AllowedSet>
  <DeniedSet></DeniedSet>
   </AttributePermission>
   <AttributePermission>
  <AttributeID>CategorycategoryDesc </AttributeID>
  <AllowedSet>[Product].[CategorycategoryDesc]</AllowedSet>
  <DeniedSet></DeniedSet>
   </AttributePermission>
 </AttributePermissions>
</DimensionPermission>
</DimensionPermissions>
</Dimension>
```

5. Conclusions

DWs manage vital business information which is very sensitive and which has to be correctly assured in order to avoid unauthorized access. Furthermore, since DWs are queried by OLAP tools which managed specific views from the corporative DW, security constraints should also be defined in this metadata layer by using the same multidimensional elements that will be managed by the end users of OLAP tools.

Thanks to our proposal we are able to develop secure OLAP applications, providing a complete MDA architecture composed of several security models and automatic transformations towards final secure implementation. That is: (i) a new logical metamodel (SECMDDW) for secure OLAP applications, based on the OLAP package of CWM; (ii) a set of model-to-model transformations from our conceptual models (PIM) to the new OLAP logical models (PSM); and (iii) the model-to-text transformations for generating code for a specific OLAP tool (SSAS) from the new OLAP logical models (PSM).

Furthermore, an application example has been presented to validate our proposal, an example in which a conceptual model is defined; transformation rules are applied to generate the secure multidimensional logical model and the eventual implementation in SSAS. The validation of our proposal has been planned in several stages. In this paper, we have presented an application example that helped us carry out an early evaluation. This application example includes a great variety of the structural and security elements which allow us to evaluate the applicability of the proposed models and transformations, enabling us to improve them. Nevertheless, as future work we consider it necessary to include a subsequent stage to complete the evaluation of our architecture by applying it to industrial case studies with the participation of professional designers. We will also define a family of experiments in order to measure the improvement obtained in comparison to the traditional development process.

Our further work will improve this architecture in several lines: (i) including new PSM models, giving support to other final platforms (such as Pentaho); (ii) defining inverse transformations to allow modernization processes; and (iii) including dynamic security models which complement the existing models that deal with the security problem of inference.

Acknowledgments. This research is part of the following projects: SIGMA-CC (TIN2012-36904), GEODAS-BC (TIN2012-37493-C01) and GEODAS-BI (TIN2012-37493-C03) funded by the Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER. SERENIDAD (PEII11-037-7035) and MOTERO (PEII11- 0399-9449) funded by the Consejería de Educacin, Ciencia y Cultura de la Junta de Comunidades de Castilla La Mancha, and Fondo Europeo de Desarrollo Regional FEDER.

References

1. Abraham, A., Lloret Mauri, J., Buford, J., Suzuki, J., Thampi, S., Khajaria, K., Kumar, M.: Evaluation of Approaches for Modeling of Security in Data Warehouses, vol. 191, pp. 9–18. Springer Berlin Heidelberg (2011)
2. Attasena, V., Harbi, N., Darmont, J.: Sharing-based privacy and availability of cloud data warehouses. In: Marcel, P. (ed.) Actes des 9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA 2013, Blois, France, Juin 13-14, 2013. RNTI, vol. B-9, pp. 17–32. Hermann (2013), http://editions-rnti.fr/?inprocid=1001892
3. Attasena, V., Harbi, N., Darmont, J.: fvss: A new secure and cost-efficient scheme for cloud data warehouses. In: Song, I., Simitsis, A., Cuzzocrea, A. (eds.) Proceedings of the 17th International Workshop on Data Warehousing and OLAP, DOLAP 2014, Shanghai, China, November 3-7, 2014. pp. 81–90. ACM (2014), http://doi.acm.org/10.1145/2666158.2666173
4. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: from uml models to access control infrastructures. ACM Transactions on Software Engineering and Methodology 15(1), 39–91 (2006)
5. Best, B., Jurjens, J., Nuseibeh, B.: Model-based security engineering of distributed information systems using umlsec. In: International Conference on Software Engineering. pp. 581–590. IEEE, Minneapolis, MN, USA (2007)
6. Binh, N.T., Tjoa, A.M., Wagner, R.: An object oriented multidimensional data model for olap. In Proc. of 1st Int. Conf. on Web-Age Information Management (WAIM) (2000)
7. Braganca, A., Machado, R.: Model driven development of software product lines. In: International Conference on the Quality of Information and Communications Technology. pp. 199–203. Lisbon, Portugal (2007)
8. Compagna, L., Khoury, P., Krausová, A., Massacci, F., Zannone, N.: How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. Artif. Intell. Law 17(1), 1–30 (2009)
9. Cuccuru, A., De Simone, R., Saunier, T., Siegel, G., Sorel, Y.: P2i: An innovative mda methodology for embedded real-time systems. In: Euromicro Conference on Digital System Design. pp. 26–33. IEEE, Porto, Portugal (2005)
10. Cuzzocrea, A., Russo, V.: Privacy Preserving OLAP and OLAP Security, pp. 1575–1581. IGI Global, Hershey, PA, USA (2009)
11. Dubielewicz, I., Hnatkowska, B., Huzar, Z., Tuzinkiewicz, L.: Evaluation of mda-psm database model quality in the context of selected non-functional requirements. In: International Conference on Dependability of Computer Systems. pp. 19–26. IEEE, Szklarska Poreba, Poland (2007)

12. Fernandes, L., Neto, B., Fagundes, V., Zimbrao, G., de Souza, J., Salvador, R.: Model-driven architecture approach for data warehouse. Sixth International Conference on Autonomic and Autonomous Systems (ICAS) (2010)
13. Fernández-Medina, E., Trujillo, J., Piattini, M.: Model driven multidimensional modeling of secure data warehouses. European Journal of Information Systems 16(4), 374–389 (2007)
14. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Access control and audit model for the multidimensional modeling of data warehouses. Decision Support Systems 42, 1270–1289 (2006)
15. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Developing secure data warehouses with a uml extension. Information Systems 32(6), 826–856 (2007)
16. Fernández-Medina, E., Jurjens, J., Trujillo, J., Jajodia, S.: Model-driven development for secure information systems. Information and Software Technology 51(5), 809–814 (2009), 0950-5849 doi: DOI: 10.1016/j.infsof.2008.05.010
17. Fischer-Hübner, S., Katsikas, S., Quirchmayr, G., Salem, A., Triki, S., Ben-Abdallah, H., Harbi, N., Boussaid, O.: Verification of Security Coherence in Data Warehouse Designs, vol. 7449, pp. 207–213. Springer Berlin Heidelberg (2012)
18. Giorgini, P., Mouratidis, H., Zannone, N.: Modelling security and trust with secure tropos. In: Integrating Security and Software Engineering: Advances and Future Visions. Idea Group Publishing (2006)
19. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill Osborne Media (2009)
20. Inmon, W.: 2.0 - architecture for the next generation of data warehousing. Morgan Kaufman Series in Data Management Systems, Morgan Kaufmann (2008)
21. Jajodia, S., Samarati, P., Sapino, M., Subrahmanian, V.: Flexible support for multiple access control policies. ACM Transactions on Database Systems 26, 214–260 (2001)
22. Jurjens, J.: Secure Systems Development with UML. Springer-Verlag (2004)
23. Jurjens, J., Schmidt, H.: Umlsec4uml2 - adopting umlsec to support uml2. Tech. rep., Technical Reports in Computer Science. Technische Universitat Dortmund, http://hdl.handle.net/2003/27602 (2011)
24. Jurjens, J., Shabalin, P.: Tools for secure systems development with uml. Invited submission to the FASE 2004/05 special issue of the International Journal on Software Tools for Technology Transfer 9(5-6), 527–544 (2007)
25. Kirkgoze, R., Katic, N., Stolda, M., Min Tjoa, A.: A security concept for olap. In: 8th International Workshop on Database and Expert System Applications (DEXA'97). pp. 619–626. IEEE Computer Society, Toulouse, France (1997)
26. Kraus, A., Knapp, A., Koch, N.: Model-driven generation of web applications in uwe. In: International Workshop on Model-Driven Web Engineering. Como, Italy (2007)
27. Li, B., Liu, S., Yu, Z.: Applying mda in traditional database-based application development. In: International Conference on Computer Supported Cooperative Work in Design. pp. 1038–1041. IEEE, Coventry, UK (2005)
28. Liu, Y., Sung, S., Xiong, H.: A cubic-wise balance approach for privacy perservation in data cubes. Information Sciences 176(9), 1215–1240 (2006)
29. Lodderstedt, T., Basin, D., Doser, J.: Secureuml: A uml-based modeling language for model-driven security. In: UML 2002. The Unified Modeling Language. Model Engineering, Languages Concepts, and Tools. 5th International Conference. vol. LNCS 2460, pp. 426–441. Springer, Dresden, Germany (2002)
30. Lu, S., Halang, W.A., Zhang, L.: A component-based uml profile to model embedded real-time systems designed by the mda approach. In: International Conference on Embedded and Real-Time Computing Systems and Applications. pp. 563–566. IEEE, Hong Kong, China (2005)
31. Luján-Mora, S., Trujillo, J., Song, I.Y.: A uml profile for multidimensional modeling in data warehouses. Data & Knowledge Engineering 59(3), 725–769 (2006)

32. Massacci, F., Mylopoulos, J., Zannone, N.: Computer-aided support for secure tropos. Automated Software Engineering 14(3), 341–364 (2007)
33. Matulevicius, R., Dumas, M.: Towards model transformation between secureuml and umlsec for role-based access control. In: Proceeding of the 2011 conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010. pp. 339–352. IOS Press, Amsterdam, The Netherlands, The Netherlands (2011), http://portal.acm.org/citation.cfm?id=1940590.1940622
34. Mazón, J.N., Trujillo, J.: An mda approach for the development of data warehouses. Decision Support Systems 45(1), 41–58 (2008)
35. Meliá, S., Gomez, J.: The websa approach: Applying model driven engineering to web applications. Journal of Web Engineering 5(2), 121–149 (2006)
36. Mouratidis, H.: Software Engineering for Secure Systems: Industrial and Research Perspectives. IGI Global (2011)
37. Muñoz, L., Mazón, J.N., Trujillo, J.: Automatic generation of etl processes from conceptual models. In: Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP. pp. 33–40. DOLAP '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1651291.1651298
38. Mundy, J., Thornthwaite, W., Kimball, R.: The Microsoft Data Warehouse Toolkit: With SQL Server 2008 R2 and the Microsoft Business Intelligence Toolset. Wiley (2011)
39. Nunes, F.J.B., Belchior, A.D., Albuquerque, A.B.: Security engineering approach to support software security. Services, IEEE Congress on 0, 48–55 (2010)
40. OMG: Cwm. common warehouse metamodel. version v1.1. http://www.omg.org/spec/CWM/1.1 (2003)
41. OMG: Mda. model driven architecture guide version 1.0.1. http://www.omg.org/cgi-bin/doc?omg/03-06-01 (2003)
42. OMG: Qvt. meta object facility 2.0 query/view/transformation specification. http://www.omg.org/spec/QVT/1.1 (2011)
43. Paim, F.R.S., Castro, J.: Dwarf: An approach for requirements definition and management of data warehouse systems. 11th IEEE International Requirements Engineering Conference, 2003. Proceedings. (2003)
44. Prat, N., Akoka, J., Comyn-Wattiau, I.: A uml-based data warehouse design method. Decision Support Systems 42(3), 1449–1473 (2006)
45. Priebe, T., Pernul, G.: Towards olap security design - survey and research issues. In: 3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00). pp. 33–40. Washington DC, USA (2000)
46. Priebe, T., Pernul, G.: A pragmatic approach to conceptual modeling of olap security. In: 20th International Conference on Conceptual Modeling (ER 2001). Springer-Verlag, Yokohama, Japan (2001)
47. van de Riet, R.: Twenty-five years of mokum: For 25 years of data and knowledge engineering: Correctness by design in relation to mde and correct protocols in cyberspace. Data & Knowledge Engineering 67(2), 293–329 (2008)
48. Saltor, F., Oliva, M., Abelló, A., Samos, J.: Building secure data warehouse schemas from federated information systems (2002)
49. Sapia, C., Blaschka, M., Hofling, G., Dinter, B.: Extending the e/r model for the multidimensional paradigm. In: 1st International Workshop on Data Warehouse and Data Mining (DWDM'98). vol. 1552, pp. 105–116. Springer-Verlag, Singapore (1998)
50. Shoshani, A.: Olap and statistical databases: Similarities and differences. In: PODS 97. Tucson, AZ (1997)
51. Simitsis, A., Sloutas, D., Castellanos, M.: Representation of conceptual etl designs in natural language using semantic web technology. Data & Knowledge Engineering 69(1), 96–115 (2010)

52. Simitsis, A., Vassiliadis, P.: A method for the mapping of conceptual designs to logical blueprints for etl processes. Decision Support Systems 45(1), 22–40 (2008)
53. Sung, Y., Liu, Y., Xiong, H., A. Xg, P.: Privacy preservation for data cubes. Knowledge Information Systems 9(1), 38–61 (2006)
54. Tan, W., Ma, L., Li, J., Xiao, Z.: Application mda in a conception design environment. In: International Multi-Symposiums on Computer and Computational Sciences. pp. 702–704. IEEE Computer Society, Hangzhou, China (2006)
55. Tankard, C.: Big data security. Network Security 2012(7), 5 – 8 (2012), http://www.sciencedirect.com/science/article/pii/S1353485812700636
56. Thuraisingham, B.: Security issues for federated database systems. Computer and Security 13(6), 509–525 (1994)
57. Thuraisingham, B., Kantarcioglu, M., Iyer, S.: Extended rbac-based design and implementation for a secure data warehouse. International Journal of Business Intelligence and Data Mining (IJBIDM) 2(4), 367–382 (2007)
58. Trujillo, J., Luján-Mora, S.: A uml based approach for modeling etl processes in data warehouses. In: Heidelberg, S.B.. (ed.) Conceptual Modeling - ER 2003, vol. Volume 2813/2003, pp. 307–320 (2003)
59. Tryfona, N., Busborg, F., Christiansen, J.: starer: A conceptual model for data warehouse design. In: ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP'99). pp. 3–8. ACM, Missouri, USA (1999)
60. Vara, J., Vela, B., Cavero, J., Marcos, E.: Model transformation for object-relational database development. In: ACM Symposium on Applied Computing. pp. 1012–1019. ACM, Seoul, Korea (2007)
61. Vela, B., Fernandez-Medina, E., Marcos, E., Piattini, M.: Model driven development of secure xml databases. ACM Sigmod Record 35(3), 22–27 (2006)
62. Vela, B., Mazón, J., Blanco, C., Fernández-Medina, E., Trujillo, J., Marcos, E.: Automatic generation of secure xml data warehouses by using qvt within an mda framework. Information and Software Technology (INFSOF) 55(9) (2013)
63. Wang, L., Jajodia, S., Wijesekera, D.: Securing olap data cubes against privacy breaches. In: IEEE Symposium on Security and Privacy. pp. 161–178. Berkeley, California (2004)
64. Wang, Y., Zhou, X., Liang, L., Peng, C.: A mda based soc modeling approach using uml and systemc. In: International Conference on Computer and Information Technology. pp. 245–251. IEEE, Baridhara, Bangladesh (2006)
65. Weippl, E., Mangisengi, O., Essmayr, W., Lichtenberger, F., Winiwarter, W.: An authorization model for data warehouses and olap. In: Workshop on Security in Distributed Data Warehousing. New Orleans, Louisiana, USA (2001)
66. Yang, J.J., Li, J.Q., Niu, Y.: A hybrid solution for privacy preserving medical data sharing in the cloud environment. Future Generation Computer Systems (0), – (2014), http://www.sciencedirect.com/science/article/pii/S0167739X14001253
67. Yu, B., Zhang, C., Zhao, Y.: Transform from models to service description based on mda. In: Asia-Pacific Conference on Services Computing. pp. 605–608. IEEE, GuangZhou, China (2006)
68. Zhao, J., Wang, L., Tao, J., Chen, J., Sun, W., Ranjan, R., Kołodziej, J., Streit, A., Georgakopoulos, D.: A security framework in g-hadoop for big data computing across distributed cloud data centres. Journal of Computer and System Sciences 80(5), 994 – 1007 (2014), http://www.sciencedirect.com/science/article/pii/S002200001400018X, special Issue on Dependable and Secure Computing The 9th {IEEE} International Conference on Dependable, Autonomic and Secure Computing

Carlos Blanco has an MSc and PhD in Computer Science from the University of Castilla-La Mancha (Spain). He is working as a lecturer at the Science Faculty at the University of

Cantabria (Spain) and is a member of the GSyA Research Group at the School of Computer Science at the University of Castilla-La Mancha (Spain). His research activity is in the field of Security for Information Systems and its specially focused on Data Warehouses, OLAP tools, MDD and Ontologies. He has published several communications, papers and book chapters related with these topics. He is author of several papers in international journals such as DSS, CSI, INFSOF, JUCS, JRPIT, IJBIDM. He is involved in the organization of several international workshop (WOSIS, WISSE, MoBiD) and has served as reviewer for international journals, conferences and workshops (CSI, ASE, IJITDM, JUCS, JITSE, JWE, ARES, DaWaK, SECRYPT, EssOs, etc.).

Ignacio García-Rodríguez de Guzmán has an MSc in Computer Science and a PhD from the University of Castilla-La Mancha (UCLM). He is currently a member of the Alarcos Research Group in the School of Computer Science of the UCLM and his research activity is in the field of MDA, Architecture-Driven Modernization, Reverse Engineering and Reengineering, SOA and Web Services.

Eduardo Fernández-Medina holds a PhD in Computer Science from the University of Castilla-La Mancha. He leads the GSyA Research Group of the Department of Computer Science at the University of Castilla-La Mancha. His research activity is in the field of security in databases, data warehouses, web services and information systems, and also in security metrics. Fernández-Medina is co-editor of several books and book chapters on these subjects and has presented several dozens of papers at national and international conferences (DEXA, CAISE, UML, ER, etc.). He is the author of several manuscripts in national and international journals (DSS, ACM Sigmod Record, IS, IST, C&S, ISS, etc.) and belongs to various professional and research associations (AEC, ISO, IFIP WG11.3, etc.).

Juan Trujillo is an Associate Professor at the Department of Software and Computing Systems in the University of Alicante (Spain), and he is the leader of the LUCENTIA Research Group at the same Department. His main research topics include bussiness intelligence applications, data warehouses development, data base conceptual modeling, multidimensional data bases, OLAP and data mining applications, object oriented analysis and design by using UML, MDA, data warehouses security and quality, etc. He has published more than a hundred research works in different national and international high impact conferences and journals, such as the ER, UML, ADBIS, CaiSE or WAIM, DKE, CS&I, DSS, ISOFT, IS, or JDBM. He has also participated as a PC member in different workshops, conferences, and JCR journals such us ER, DOLAP, DSS, y SCI, JDM, KAIS, ISOFT, JOD or DKE. Moreover, he has been Program Chair and Co-Chair in DOLAP, DAWAK, FP-UML and BP-UML.

Received: June 17, 2014; Accepted: March 26, 2015.