Journal of Universal Computer Science, vol. 20, no. 2 (2014), 79-106
submitted: 9/7/12, accepted: 30/1/14, appeared: 1/2/14 © *J.UCS*

Showing the Benefits of Applying a Model Driven Architecture for Developing Secure OLAP Applications

Carlos Blanco
(GSyA Research Group. Dep. of Mathematics, Statistics and Computer Science
Faculty of Sciences, University of Cantabria, Santander, Spain
Carlos.Blanco@unican.es)

Ignacio García-Rodríguez de Guzmán
(Alarcos Research Group – Institute of Information Technologies and Systems
Dep. of Information Technologies and Systems, Escuela Superior de Informática
University of Castilla-La Mancha, Ciudad Real, Spain
Ignacio.GRodriguez@uclm.es)

Eduardo Fernández-Medina
(GSyA Research Group – Institute of Information Technologies and Systems
Dep. of Information Technologies and Systems, Escuela Superior de Informática
University of Castilla-La Mancha, Ciudad Real, Spain
Eduardo.Fdezmedina@uclm.es)

Juan Trujillo
(Lucentia Research Group. Department of Information Languages and Systems
Facultad de Informática, University of Alicante, Alicante, Spain
jtrujillo@dlsi.ua.es)

Abstract: Data Warehouses (DW) manage enterprise information that is queried for decision making purposes by using On-Line Analytical Processing (OLAP) tools. The establishment of security constraints in all development stages and operations of the DW is highly important since otherwise, unauthorized users may discover vital business information.
The final users of OLAP tools access and analyze the information from the corporate DW by using specific views or cubes based on the multidimensional modelling containing the facts and dimensions (with the corresponding classification hierarchies) that a decision maker or group of decision makers are interested in. Thus, it is important that security constraints will be also established over this metadata layer that connects the DW's repository with the decision makers, that is, directly over the multidimensional structures that final users manage. In doing so, we will not have to define specific security constraints for every particular user, thereby reducing the developing time and costs for secure OLAP applications.
In order to achieve this goal, a model driven architecture to automatically develop secure OLAP applications from models has been defined. This paper shows the benefits of this architecture by applying it to a case study in which an OLAP application for an airport DW is automatically developed from models. The architecture is composed of: (1) the secure conceptual modelling by using a UML profile; (2) the secure logical modelling for OLAP applications by using an extension of CWM; (3) the secure implementation into a specific OLAP tool, SQL Server Analysis Services (SSAS); and (4) the transformations needed to automatically generate logical models from conceptual models and the final secure implementation.

Keywords: Security, Confidentiality, OLAP, Data Warehouses, Model Driven, MDA, Transformations, SSAS, Case Study.
Categories: H.2, H.2.1, H.2.2, K.6.5, L.4.0

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 the organizations and furthermore is used to include private data of individuals [Thuraisingham et al., 2007]. In this sense, information confidentiality is the main problem related to security to be tackled in the access to the data warehouse, due to final users will only carry out reading operations [Priebe and Pernul, 2001][Blanco et al., 2009][Trujillo et al., 2009].

Security has been traditionally 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 security requirements are taken into account to make design decisions and the system can accommodate them in a more natural way [Fernández-Medina et al., 2009][Blanco et al., 2011].

On the other hand, given that the data warehouse design process involves the traditional development stages in which the system is modeled at business, conceptual, logical level and eventually the final solution is implemented, model driven engineering approach can be applied [OMG, 2003][Inmon, 2008][Mundy et al., 2011].

There are proposal for developing secure data warehouses based on the automated development of the software by means of models definition and transformations between models, by reducing as a consequence the development times and costs [Fernández-Medina et al., 2007]. However, data warehouses used to be partially replicated in departmental data warehouses, which are focused on different users and business goals and examined by users with OLAP applications by carrying out analysis sessions where certain information is queried and grouped by different detail levels.

Despite of security measures 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 to the data warehouse are located. This is due to security specifications carried out on the data warehouse repository do not use related concepts about OLAP technology (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 warehouses accessed by users. Therefore, it is necessary an approach for OLAP applications which supports the definition of security constraints about the same views and multidimensional elements which will be managed by the final users when querying the data warehouse and which consider how these users interact with the data warehouse by means of OLAP operations [Thuraisingham et al., 2007].

This paper uses a case study to present an approach for the development of secure data warehouses focused on the OLAP technology, by means of including the security in an intermediate layer to be used by OLAP applications to access to the data warehouse information. This proposal is integrated into a previous architecture which

adopts the model driven engineering paradigm to model the DW repository and automatically generate its secure implementation according to a relational approach.

In order to achieve this goal the models and transformations needed to develop secure OLAP applications have been defined: (i) A previous UML profile for the conceptual modeling of secure data warehouses has been used and improved; (ii) A new logical metamodel for secure OLAP application has been defined based on CWM; (iii) The connection between conceptual and logical models has been achieved by developing QVT transformations; and (iv) The eventually secure implementation into SQL Server Analysis Services (SSAS) from logical models has been automated developing MOFScript rules.

The rest of this paper is organized as follows: Section 2 will present the related work on developing secure DWs and OLAP applications; Section 3 will briefly show our MDA approach for developing secure OLAP applications; Sections 4, 5 and 6 will describe our proposal by using a case study of an airport DW: conceptual modeling (Section 4); transformation to logical models for OLAP (Section 5); and transformation to SSAS implementation (Section 6); Section 7 will describe the lessons learned after carrying out this case study; and Section 8 will finally present our conclusions and future work.

2 Related Work

Firstly, several relevant works concerning with a complete secure development of information systems can be found. UMLsec [Jurjens, 2004][Jurjens and Schmidt, 2011] uses UML to define and evaluate security specifications using formal semantics. TROPOS is a methodology for software development based on the intentional goals of agents which provides an extension called Secure TROPOS [Giorgini et al., 2006] that allows us to model and analyze security requirements within functional requirement. Model Driven Security (MDS) [Basin et al., 2006] uses the MDA approach to include security properties in high-level system models and to automatically generate secure system architectures. Within the context of MDS, SecureUML [Lodderstedt et al., 2002] is proposed as an extension of UML for modeling a generalized role based access control. Mokum [van de Riet, 2008] which is an active object oriented knowledge based system for modeling which permits the specification of security and integrity constraints, and the automatic code generation. On the other hand there are processes for building security systems which are based on security models and standards. For instance the process PSSS (Process to Support Software Security) [Barreto et al., 2010] which is based on the activities derived from SSE-CMM [SSE-CMM, 2003], ISO/IEC 15408 [ISO/IEC, 2005], ISO/IEC 27002 [ISO/IEC, 2005] and OCTAVE [Alberts and Dorofee, 2002]. Nevertheless, although these are relevant contributions on secure information systems development they do not are specifically focused on DWs and their specific security problems.

A typical DW's architecture is composed of several layers (Figure 1): 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 mainly dealt with read operations over sensitive information used for the decision making, the main security problem related with

DWs is information confidentiality and should be taken into account in all layers and operations of the DW [Thuraisingham et al., 2007].

Each layer of this architecture presents specific security concerns which are following described.

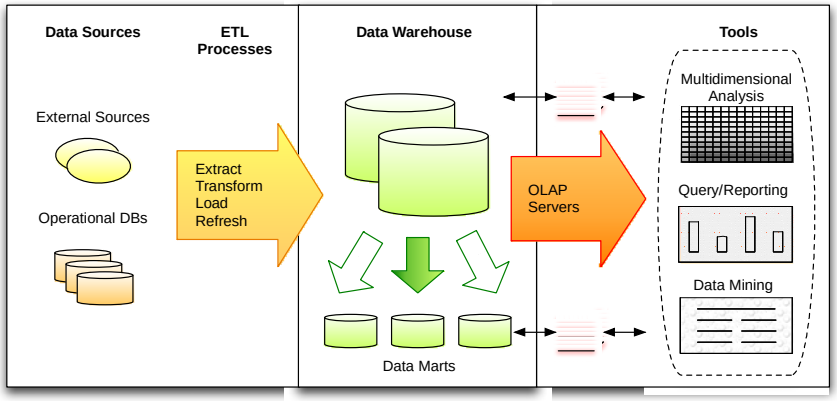

Figure 1: A Data Warehouse Architecture

2.1 Data Sources and ETL Processes

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

There are sensitive data (for instance, according to legal regulations) which should be assured regardless to final users (maybe different from those of data sources), to define an integrated security policy is an interesting starting point to establish security constraints in the DW. There several works related with the integration of different security policies in FIS [Thuraisingham, 1994][Jajodia and Wijesekera, 2001]. Saltor et al. [Saltor et al., 2002] use this parallelism to adapt a design architecture for FIS to DWs, and also to improve it with security capabilities supporting the integration of MAC policies.

On the other hand, since ETL processes extract and transform information from data sources which is finally loaded into the DW, it is important that ETL processes take also security information into account. However, although exist proposals for conceptual modeling ETL processes with an own notation [Simitsis and Vassiliadis, 2007] or by using a UML approach [Trujillo and Luján-Mora, 2003], nowadays they do not support security issues.

2.2 Data Warehouse's Repository

Concerning with a complete secure DWs development we solely found the methodology of Priebe and Pernul [Priebe and Pernul, 2001] in which the authors analyze security requirements and their implementation into commercial tools by

hiding multidimensional elements such as cubes, measures, slices and levels. They extend their proposal with a DW's representation at conceptual level with ADAPTed UML, but do not establish the connection between models in order to allow automatic transformations.

On the other hand, there are several works focused on the secure modeling for DWs at certain abstraction levels. At business level there are proposals based on ontologies, business process, UML, etc. but solely Paim and Castro [Paim and Castro, 2003] include security requirements, however they do not offer any formal metamodel.

At the conceptual level there are interesting works for modeling DWs considering their special characteristics by using extensions of the ER model, UML or an own notation, but they do not include security capabilities [Golfarelli et al., 1998][Sapia et al., 1998][Tryfona et al., 1999][Binh et al., 2000][Abelló et al., 2006][Luján-Mora et al., 2006][Prat et al., 2006]. The conceptual modeling of security issues is solely considered by the AdaptedUML of Priebe and Pernul [Priebe and Pernul, 2001].

Traditionally, the multidimensional modeling at logical level has depended of the DBMS used and, in this way can be mainly classified in online analytical processing over a relational (ROLAP), multidimensional (MOLAP) and hybrid (HOLAP) approaches. There are many modeling proposals which do not consider security but solely CWM [CWM, 2003] provides a formal metamodel with relational and multidimensional packages.

2.3 OLAP Applications

Final tools have also to consider security constraints in order to avoid unauthorized accesses. Research efforts have been traditionally carried out in this way but focused on the final stage of development without including security issues in the whole development process. For instance, Kirkgoze et al. propose to define a virtual cube for each subject [Kirkgoze et al., 1997], or Weippl et al. which define an access control model for DWs and OLAP which allows to define the OLAP operations authorized for each user [Weippl et al., 2001].

Nevertheless, nowadays the inference problem is still a challenge in DW security and an important research branch [Thuraisingham et al., 2007]. Since DWs store sensitive data by using different aggregation levels with different confidentiality requirements (for example, the average salary may be Unclassified and individual salaries Secret), the inference problem is similar to the previously studied problem for statistical databases which store summarized data such as sum or averages [Shoshani, 1997]. Some works have dealt with inference proposing query control systems [Wang et al., 2004][Liu et al., 2006][Sung et al., 2006].

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

Our previous works were focused on developing the DW repository including security aspects from the requirement model to its implementation in DBMS by using a relational approach [Soler et al., 2009].

Nevertheless, users access the data warehouse by using OLAP tools which manage concepts associated with the OLAP technology (cubes, aggregation levels, roll-up and drill-down operations, etc.). This paper presents an approach for developing secure OLAP applications based on models and transformations, which compliments and has been integrated with our previous architecture. This approach allows us to define security constraints associated with the same views and multidimensional elements which will be managed by the final users.

3 An Overview of our MDA architecture for Developing Secure OLAP Applications

This section briefly describes our model-based approach for developing secure OLAP applications (Figure 2). This approach follows the OMG proposal for the implementation of model driven development, MDA [OMG, 2003], defining a platform independent model (PIM), a platform specific model (PSM) and transformations that automate the final implementation.

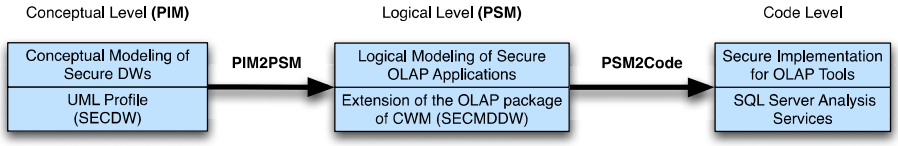

Figure 2: MDA architecture for developing Secure OLAP Applications

Firstly, a UML profile for the conceptual modeling of secure DWs is used as a platform independent model (PIM). This UML profile is called SECDW [Fernández-Medina et al., 2007] and allows us to define the multidimensional model of the DW (facts, dimensions, hierarchies, measures, attributes, etc.) and to associate security constraints to multidimensional elements. The security capabilities that can be specified in the conceptual models are provided by an access control and audit model for DW [Fernández-Medina et al., 2006]. It permits to classify objects and subjects into security levels, roles and compartments, and to specify security rules to establish the security privileges needed to access certain information (sensitive information assignment rules, SIAR), to define authorizations (AUR) or auditing (AR).

At the logical level, the system is modeled for a specific platform (PSM). A metamodel called SECMDDW provides the elements needed to model the DW focusing on the OLAP technology. SECMDDW is based on the OLAP package of CWM [OMG, 2003] for representing the structural elements (cubes, measures, hierarchies, dimensions, etc.) and permits the association of security permissions to cubes, dimensions and attributes.

Finally, the OLAP application is implemented in a certain OLAP tool, adapting the information represented in the logical model to the mechanisms provided by the target OLAP tool.

The process of developing secure OLAP applications by using this approach, has been automated by defining transformations that, in a first step obtain logical models from conceptual models (model to model transformations specified in QVT), and next, generate the secure implementation into SQL Server Analysis Services from logical models (model to text transformations specified in MOFScript).

This paper following describes our architecture for developing secure OLAP application by using a case study.

4 Conceptual Model for Secure DWs

The case study presented in this paper uses a DW for an airport, with several Data Marts for different purposes, such us Trips, Flights, Incidents or Multimedia information. This case study is focused on a Data Mart which manages trip information involving passengers who take flights in certain dates in order to reach destinations. This Data Mart is analyzed for airport staff, companies or passengers, and for instance can be used for companies to offer special prices for the top destinations of a passenger.

4.1 Conceptual Model

Figure 3 shows the conceptual model for this case study, defined according to SECDW. A central fact manages information about trips: price, purpose (which can be “tourist”, “business” or “military”), seat, distance, flight time and if the check-in and boarding procedures have been carried out or not. The information about trips can be organized according to several dimensions: passengers, baggage, flights and departure and arrival places and dates.

Passenger dimension class includes attributes with personal information about passengers (code, name and address) and extended security information, such us fingerprint, passport photo, criminal record, if it is considered a suspicious passenger and the estimated risk index (a number from 1 to 10). Baggage dimension class has several attributes with information of baggage items, codes, weight and if the baggage has been inspected and it is suspicious.

The remainder of the dimensions classes (Place, Date and Flight) has been associated with a set of base classes forming navigation hierarchies. In this way, the places can be aggregated by gates, terminals and airports; the dates by hours, days, months and years; and the flights by planes, aircraft types and companies.

4.2 Security Configuration

The security configuration of this case study uses a classification of users and objects in security compartments (SC), roles (SR) and levels (SL) (Figure 4). The security compartments are different airlines (company A, B and C). The security role (SR) hierarchy has a main system user “User” specialized into passengers and airport staff which is composed of security, flight and administration (specialized into boarding and baggaging) roles. The levels of security (SL) used are top secret (TS), secret (S), confidential (C) and undefined (U). The “UserProfile” class specifies the information

about the users which will be stored by the system: user code and name, and the security privileges associated (security level, roles and compartments).

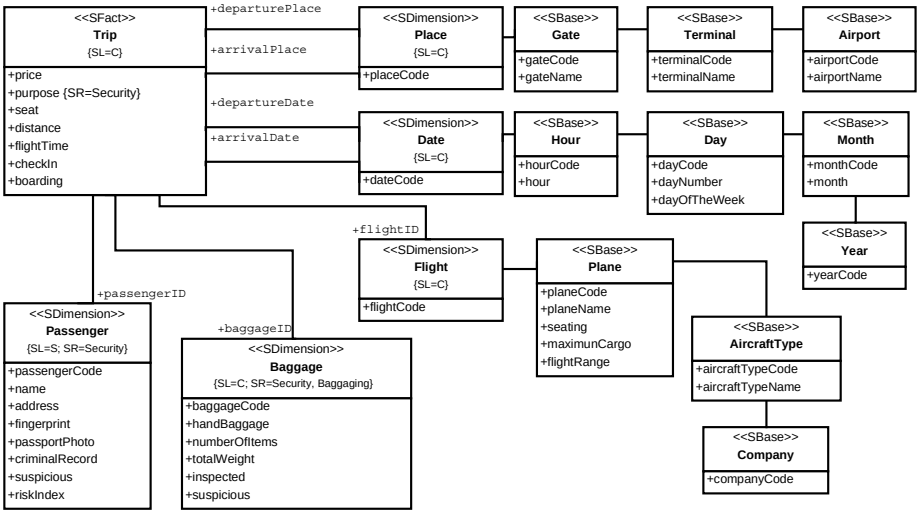

Figure 3: Conceptual Model

A set of sensitive information assignment rules (SIAR) has been furthermore defined over some classes and attributes by using stereotypes (Figure 3). “Trip” fact class can be accessed by users with confidential (or upper) security level; the dimensions “Place”, “Date” and “Flight” require a security level of confidential; “Passenger” dimension requires a security level of secret and a security role of “Security”; and “Baggage” dimension a security level of confidential and a security role of “Security” or “Baggaging”. A fine grain security constraint has been associated with the attribute “purpose” of the fact “Trip”, permitting its accesses to the security role “Security”.

4.3 Security Rules

More complex security (SIAR) and authorization (AUR) rules have been also defined (Figure 5). The “SIAR_TripPurpose” rule is associated to the “Trip” fact class and involves “Passenger” and “Flight” dimension classes. This rule increases the security requirements of the fact and the involved dimensions if the purpose of the trip is military (“purpose” attribute). In this case will be required a security level of “Secret” and a security role of “Security”.

The remainder of the SIAR rules (“SIAR_PassengerSuspicious” and “SIAR_BaggageSuspicious”) are associated to dimension classes and if the established conditions are satisfied also increase the security requirements needed to access them (that is the security level and role required). “SIAR_BaggageSuspicious” checks if the baggage is suspicious, whereas “SIAR_PassengerSuspicious” also checks the risk index of the passenger.

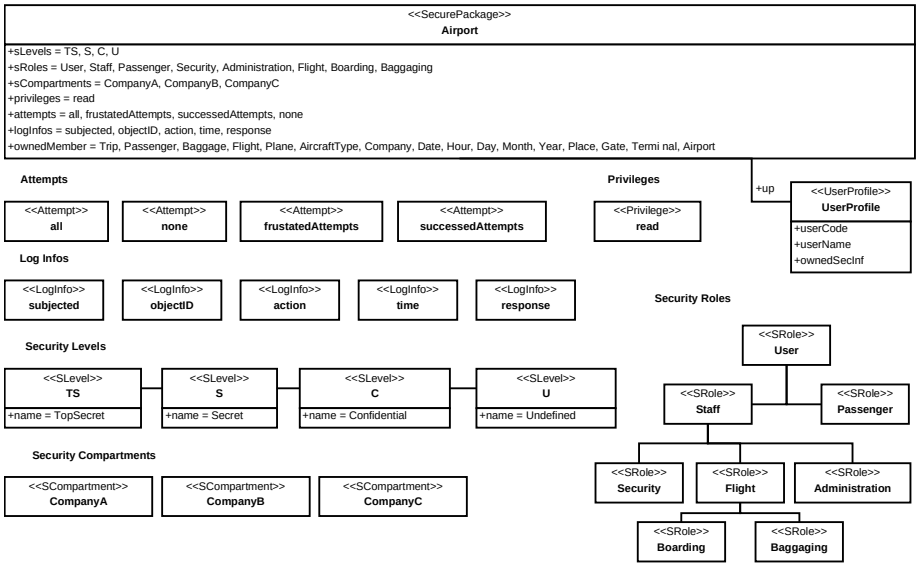

Figure 4: Conceptual Model: Security Configuration

Two authorization rules (AUR) have been furthermore defined. A negative authorization rule "AUR_Company" which checks the company of the user (security compartment) and denies accesses to information related with other companies (information about flights and its related base classes "Plane", "AircraftType" and "Company"). And a positive authorization rule "AUR_Passenger" which checks the user name ("name" attribute of the "UserProfile") and provides access to the basic information of the user (name, address and baggage id). Finally, an audit rule "ARfrustatedAttempts" stores log information about the frustrated attempts to access information about trips, passengers or baggages.

5 Logical Model for Secure OLAP Applications

In this section, the conceptual model is transformed into a logical model for secure OLAP applications, defined according to SECMDDW. This transformation has been automated by defining sets of QVT rules for obtaining the security configuration of the DW; cubes and dimensions within their related structural and security aspects; and security permissions which represents the security rules defined in the conceptual model. Next, each set of rules is described by applying it to the case study.

5.1 Security Configuration

Firstly, the transformation SECDW2Role obtains the security configuration defined at the conceptual level by using security levels, roles and compartments, and defines the security configuration at the logical level. Our logical model for secure OLAP applications establishes the security configuration by using roles, since OLAP tools

use a role-based (RBAC) security policy. Thus, new roles are created in the logical model for each level, compartment and role defined in the conceptual model.

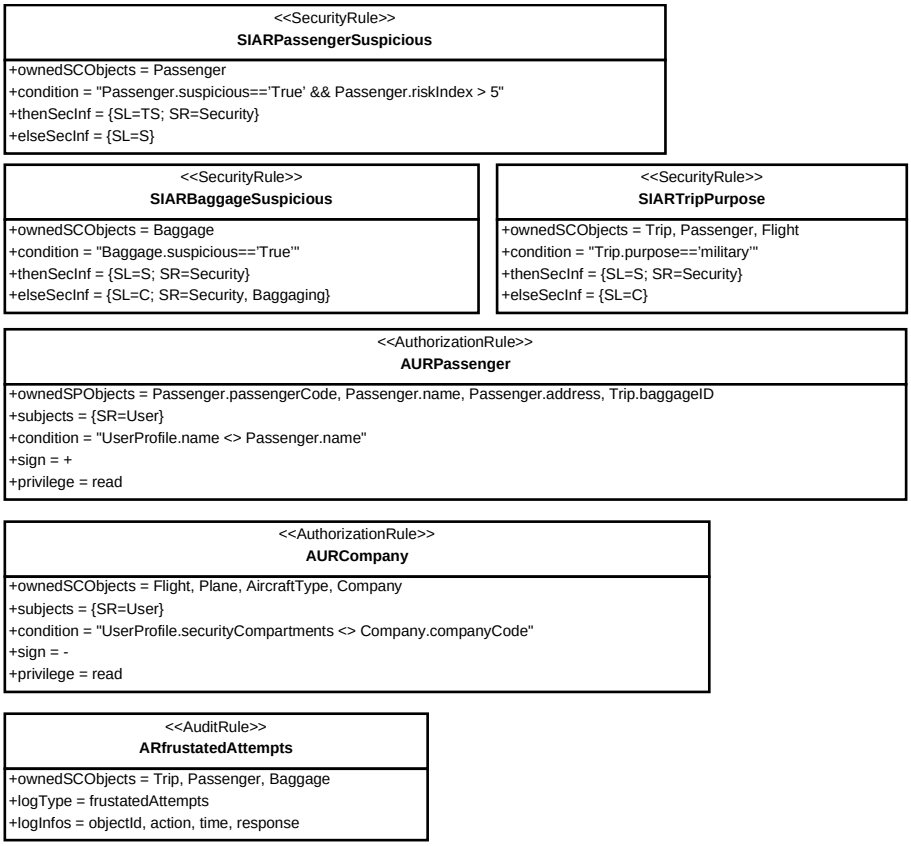

Figure 5: Conceptual Model: Security Rules

top relation **SPackage2RoleSchema:** Airport
relation **SRole2Role:** User, Passenger, Staff, Security, Flight, Administration, Boarding, Baggaging relation **SLevel2Role:** TS, S, C, U relation **SCompartment2Role:**CompanyA, CompanyB, CompanyC relation **AddMemberSRole2Role:** UserProfile relation **AddMemberSLevel2Role:** UserProfile relation **AddMemberSCompartment2Role:** UserProfile

Table 1: SECDW2Role transformation

Table 1 shows the rules that composed the transformation SECDW2Role and how they are applied to each role, level and compartment detected in the conceptual model, obtaining the logical model presented in Figure 6. Each rule creates specific roles which are called with the prefix SC, SL and SR according to their source (for

instance “SLTS” was the security level “TS” at conceptual level). Finally, members for each role are added by analyzing the user profiles.

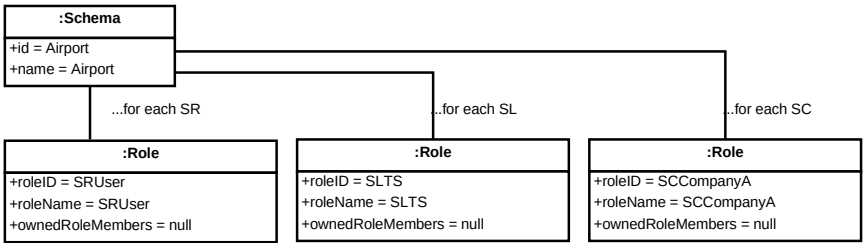

Figure 6: Logical Model: Security Configuration

5.2 Cubes

The transformation SECDW2Cube is composed of several structural rules which mainly create cubes, measures and hierarchies, and several security rules which transform security constraints of cubes into cube and cell permissions. Table 2 indicates what conceptual elements activate each rule, and the resulting logical model is shown in Figure 7.

Firstly, the “Trip” cube is created (“SFact2Cube” rule) with an associated measure group (“CreateMeasureGroups” rule) in which its properties are included as measures (“SProperty2Measure” rule): price, purpose, seat, distance, flight time, check in and boarding.

Next, security constraints defined directly over cubes and their attributes by using attached security information (secure compartments, roles and levels) are analyzed. This security information indicates sets of authorized and unauthorized roles: a specific security level (SL) restriction indicates that users with the same or upper security level are authorized; a security role (SR) restriction authorizes users with the same role and its descendents; and a security compartment (SC) restriction authorizes users with the same compartment.

top relation **SPackage2CubeSchema:**Airport
relation **SFact2Cube:** Trip relation **CreateMeasureGroups:** Trip relation **SProperty2Measure:** price, purpose, seat, distance, flightTime, checkIn, boarding
relation **SCompartmentClass2CubePermission:** Not thrown relation **SRoleClass2CubePermission:** Not thrown relation **SLevelClass2CubePermission:** (for Trip) C relation **SecureProperty2CellPermission:** Trip.purpose relation **SCompartmentAtt2CellPermission:** Not thrown relation **SRoleAtt2CellPermission:** (for Trip.purpose) Security relation **SLevelAtt2CellPermission:** Not thrown

Table 2: SECDW2Cube transformation

For each restriction established at cube level is created a set of positive cube permissions for the authorized roles and a set of negative cube permissions for the unauthorized roles denying cube’s measures (“SCompartmentClass2Cube

Permission", "SRoleClass2CubePermission" and "SLevelClass2CubePermission" rules). In this case study, the "Trip" cube has a security constraint of "Confidential" security level, which is represented in the logical model (Figure 7) as sets of positive cube permissions for the authorized roles ("Confidential", "Secret" and "TopSecret" security levels which are the roles "SLC", "SLS" and "SLTS" at the logical level) and negative cube permissions for the unauthorized roles ("Undefined" security level which is the "SLU" role).

The fine grain security constraints defined over cube properties are then dealt in a similar way, but in this case cell permissions related with the corresponding cube permissions are created ("SecureProperty2CellPermission", "SCompartmentAtt2 CellPermission", "SRoleAtt2CellPermission" and "SLevelAtt2CellPermission" rules). In this case study (Figure 7), the cube attribute "purpose" can be only accessed by users with the role "Security" which is represented in the logical model as security permissions: positive cell permissions related with the authorized cube permissions (role "Security" which is "SRSecurity" at the logical level) and negative cell permissions related with the unauthorized cube permissions (roles different from "Security") in which the denied property is specified by using a denied set expression.

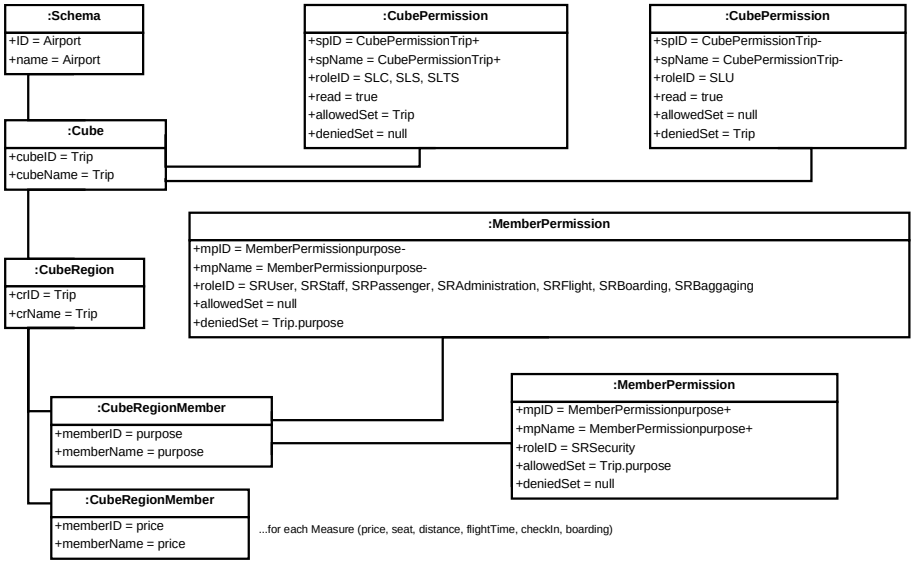

Figure 7: Logical Model: cube Trip

5.3 Dimensions

The transformation SECDW2Dimension is focused on analyzing dimensions creating at the logical level both structural aspects of dimensions, bases and their attributes and security permissions over dimensions and attributes. Table 3 shows how the relations that compose this transformation are activated by the elements defined in the conceptual model of the case study.

top relation **SPackage2DimensionSchema:**Airport
relation **SDimension2Dimension:** Place, Date, Flight, Passenger, Baggage relation **KeyProperty2KeyAttribute:** placeCode, date, flightCode, passengerCode, baggageCode relation **NonKeyProperty2Attribute:** name, address, fingerprint, passportPhoto, criminalRecord, suspicious, riskIndex, handBaggageCode, numberOfItems, totalWeight, inspected, suspicious relation **CreateOwnedHierarchies:** Place, Date, Flight relation **ProcessSBase:** Gate, Terminal, Airport, Hour, Day, Month, Year, Plane, Aircraft, Company relation **SBase2Attributes:** Gate, Terminal, Airport, Hour, Day, Month, Year, Plane, Aircraft, Company
relation **SCompartmentClass2DimensionPermission:** Not thrown relation **SRoleClass2DimensionPermission:** (for Baggage) Security, Baggaging and their descendants (for Passenger) Security and its descendants relation **SLevelClass2DimensionPermission:** (for Baggage) C, S, TS (for Passenger) S, TS relation **processSecureProperty:** Not thrown relation **createAttributePermission:** Not thrown relation **createNegativeSIARAttributePermissionForSCompartment:**Not thrown relation **createNegativeSIARAttributePermissionForSRole:** Not thrown relation **createNegativeSIARAttributePermissionForSLevel:**Not thrown

Table 3: SECDW2Dimension transformation

Dimension classes in the conceptual models are transformed into the corresponding Dimensions in the logical model ("SDimension2Dimension" rule). Then, the properties of the dimension are transformed into a key attribute ("KeyProperty2KeyAttribute" rule) and the remainder of dimension attributes ("NonKeyProperty2Attribute" rule).

Furthermore, for each base class directly related with a dimension is created a classification hierarchy ("CreateOwnedHierarchies" rule). The remainder base classes are transformed as the different aggregation levels of the hierarchies ("ProcessSBase" rule). Finally, each base class attribute is added as dimension attributes into the related dimension ("SBase2Attributes" rule) by using specific attribute names formed with the name of the base as prefix.

Figure 8 partially shows the logical model for our case study, focusing on "Place" dimension. The "Place" dimension has been represented together with its key attribute "placeCode" and its related base classes ("Gate", "Terminal" and "Airport") which have been transformed into the hierarchy "PlaceHierarchy0", several aggregation levels ("Gate", "Terminal" and "Airport" levels). Finally, the attributes of the base classes are included as attributes of the "Place" dimension (for instance "GategateCode" attribute).

The remainder of the dimension rules is focused on transforming security constraints directly defined over dimensions, bases and their attributes at the conceptual level. These security constraints indicate the security privileges (security compartment, role and level) which are needed to access the information.

When a security constraint is established involving a dimension, at the logical level positive and negative dimension permissions for the authorized and unauthorized roles are set up ("SCompartmentClass2DimensionPermission", "SRoleClass2DimensionPermission" and "SLevelClass2DimensionPermission" rules).

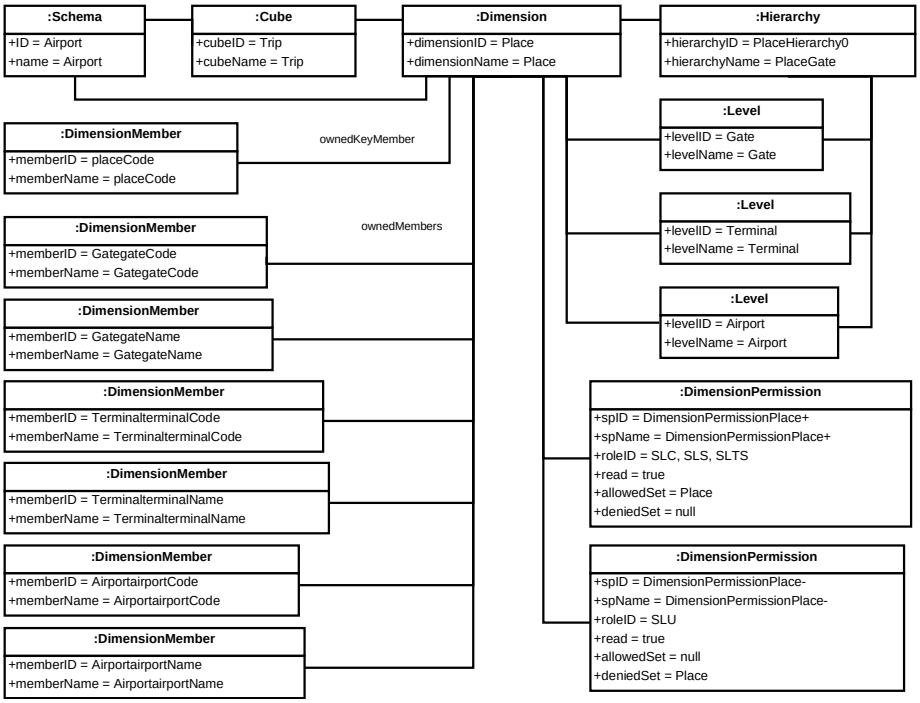

Figure 8: Logical Model: dimension Place

Figure 8 shows the dimension permissions created for “Place” dimension. “Place” dimension has a restriction of “Confidential” security level and have been therefore established positive permissions for users with the same or upper security level (“SLC”, “SLS” and “SLTS” roles) and negative permissions for users with lower security level (“SLU” role). These permissions include MDX expressions with indicate that “Place” is the information to be allowed or denied (“allowedSet” and “deniedSet” attributes).

For security constraints established over attributes, transformation rules create dimension and attribute permissions for authorized and unauthorized users, giving or denying access to the involved attributes (“processSecureProperty”, “createAttributePermissions”, “createNegativeSIARAttributePermissions ForSCompartment”, “createNegativeSIARAttributePermissionsForSRole” and “createNegativeSIARAttributePermissionsForSLevel” rules).

5.4 Security Rules

Complex security rules defined over cubes, dimensions, bases or their attributes are processed by the transformations SECDWSecurityRules2CubePermissions and SECDWSecurityRules2DimensionPermissions. These security rules can include conditions which depending of their evaluation establish different security requirements (SIAR) or authorize certain information (AUR).

top relation **processSecurityRules:**Airport
relation **processCubeSIAR:** SIARTripPurpose relation **processCubeAUR:** Not thrown relation **processDimensionSIAR:** SIARTripPurpose, SIARBaggageSuspicious, SIARPassengerSuspicious relation **processDimensionAUR:** AURPassenger, AURCompany relation **SCompartmentClass2CubePermission:** Not thrown relation **SRoleClass2CubePermission: (**SIARTripPurpose) Security relation **SLevelClass2CubePermission: (**SIARTripPurpose) TS, S relation **denySCompartmentClass2CubePermission:** Not thrown relation **denySRoleClass2CubePermission: (**SIARTripPurpose) <>Security relation **denySLevelClass2CubePermission: (**SIARTripPurpose) C, U relation **SLevelAtt2CellPermission:** Not thrown relation **SRoleAtt2CellPermission:** Not thrown relation **SCompartmentAtt2CellPermission:** Not thrown relation **denySCompartmentAtt2CellPermissionForSIAR:** Not thrown relation **denySRoleAtt2CellPermissionForSIAR:** Not thrown relation **denySLevelAtt2CellPermissionForSIAR:** Not thrown relation **createDimensionSIARForSCompartment:** Not thrown relation **createDimensionSIARForSRole: (**SIARTripPurpose) Security (SIARPassengerSuspicious) Security (SIARBaggageSuspicious) Security relation **createDimensionSIARForSLevel: (**SIARTripPurpose) TS, S, C (SIARPassengerSuspicious) TS, S, C (SIARBaggageSuspicious) TS, S, C relation **authorizeSCompartment:** Not thrown relation **authorizeSRole: (**SIARTripPurpose) Security (SIARPassengerSuspicious) Security (SIARBaggageSuspicious) Security, Baggaging relation **authorizeSLevel: (**SIARTripPurpose) TS, S, C (SIARPassengerSuspicious) TS, S (SIARBaggageSuspicious) TS, S, C relation **createAttributePermission:** Not thrown relation **createNegativeSIARAttributePermissions:** Not thrown relation **createDimensionAURForSCompartment:** Not thrown relation **createDimensionAURForSRole:** (AURCompany) User relation **createDimensionAURForSLevel:** Not thrown relation **authorizeSCompartmentForAUR:** Not thrown relation **authorizeSRoleForAUR:** (AURCompany) User relation **authorizeSLevelForAUR:** Not thrown relation **createAttributePermisionsForAUR:** (AURPassenger) User

Table 4: SECDWSecurityRules2CubePermissions transformation

For representing SIAR and AUR rules into the logical model, it is needed to create security permissions associated with cubes, dimensions, cells and attributes in order to authorize and deny accesses to certain users, but it is also necessary to evaluate the established conditions. These transformations serve from some auxiliary rules in order to create the security permissions needed.

In this way, each SIAR determines several set of users: (1) users who always can access the information (due to they always fulfill the requirements); (2) users who show the information filtered depending of the condition (which is included as denied set); and (3) users with low security privileges who cannot read the involved information in any case.

AUR rules are applied to the set of users which satisfy the security privileges specified in the AUR. Then, according to the AUR sign (positive or negative) permissions are set up authorizing or denying the read operation for the involved classes. Furthermore, the condition defined in the AUR is respectively included as allowed or denied set.

The application of this set of rules to our case study is shown in Table 4. Figure 9 shows how has been transformed a specific rule, "SIAR_TripPurpose" which involve the "Trip" cube and several dimensions. It evaluates the trip purpose and if it is "military" increases the security requirements from a "Confidential" security level to a "Secret" security level and "Security" ity role.

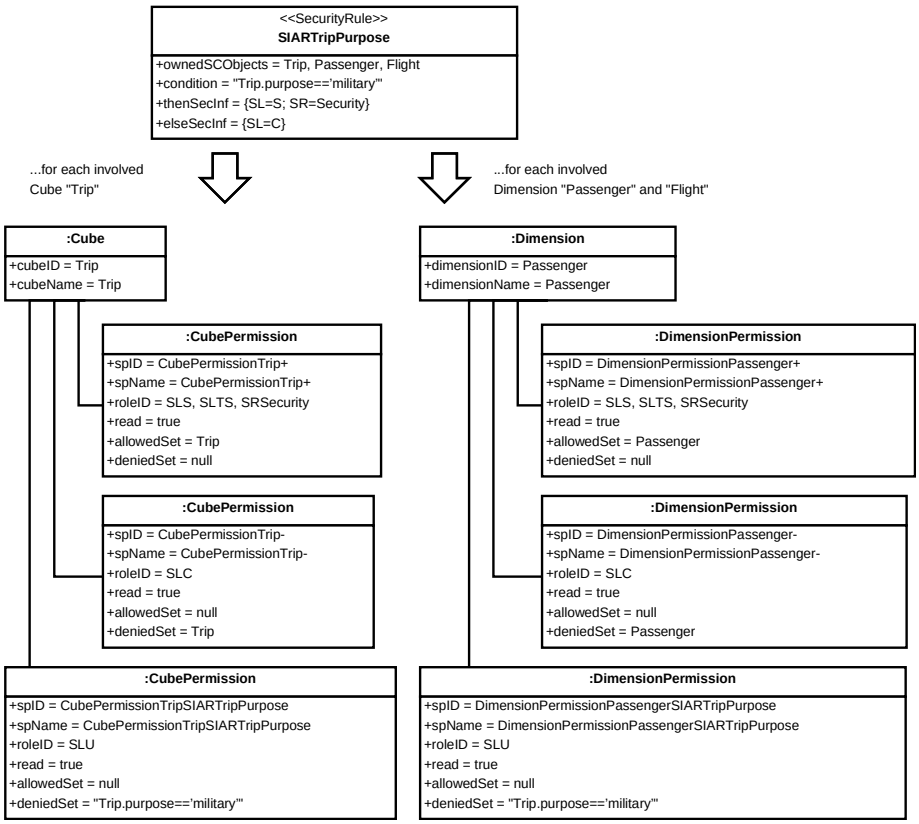

Figure 9: Logical Model: SIAR trip purpose

In the logical model are established three sets of cube permissions for the "Trip" cube and dimension permissions for the involved dimensions "Passenger" and "Flight". The first set authorizes users which always can access the information (SL >= S and SR = Security); the second one evaluates the condition for the remainder of the users (SL < S and SR <> Security) denying accesses if it is true (deniedSet =

"Trip.purpose == military"); and the last set deals with users who never could access the information (SL < C) denying the involved classes.

Figure 10 shows how the "SIAR_PassengerSuspicious" rule has been transformed at logical level. This rule evaluates if passengers are suspicious and their risk level, and if the condition is true the security requirements for "Passenger" dimension are increased from "Secret" security level to "Top Secret" security level and "Security" security role.

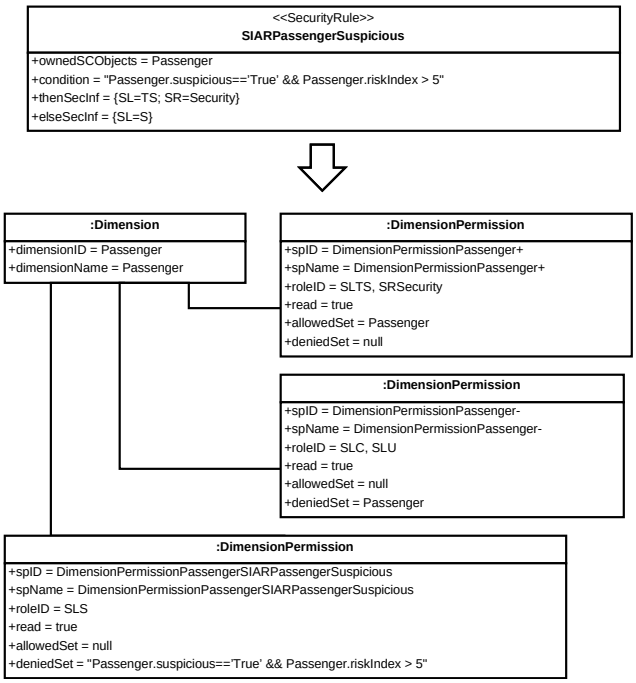

Figure 10: Logical Model: SIAR passenger suspicious

Three sets of dimension permissions are thus created: (1) authorizing users who always can access the information (SL >= TS and SR = Security); (2) denying the remainder of the users if they do not satisfy the condition (SL < TS and SR <> Security) by using the condition as denied set; and (3) denying users who never can access (SL < S) by using the dimension name as denied set.

On the other hand, the transformation of an example AUR rule has been also included in Figure 11. This rule, called "AUR_Company", is a negative authorization rule which denies accesses to flights information ("Flight" dimension class and their related base classes) of companies different from the security compartment of the user. A specific dimension permission for all the users is therefore created including the condition as denied set.

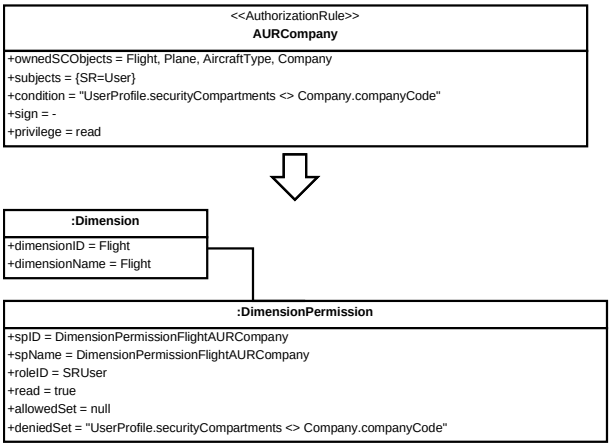

Figure 11: Logical Model: AUR company

6 Secure Implementation

Once the logical model for secure OLAP applications has been obtained, this model is transformed into a secure implementation for a specific OLAP tool. In this case, we have defined in our proposal a set of model to text transformations (in MOFScript) that allow us to automatically generate a secure implementation for SQL Server Analysis Services.

These transformation create the meta-information needed to implement the OLAP cubes into SSAS which uses three kinds of XML files: for security configuration (extension .role), cubes (extension .cube) and dimensions (extension .dim). Next, this section describes the application of these transformations to the case study.

6.1 Security Configuration

Firstly, the RBAC security policy represented in the logical model is transformed generating an XML file for each role. Table 5 shows the XML file for the role "SLTS" that represents the security level top secret.

```
<Role>
 <ID>SLTS</ID> <Name>SLTS</Name>
 <Description></Description>
<Members> <Member></Member> ... </Members>
</Role>
```

Table 5: SSAS implementation: security configuration

6.2 Cubes

Next, information about cubes is processed, creating an XML file for each cube in which both structural aspects and security permissions are specified.

Table 6 shows the XML file for the cube "Trip". It indicates information about the measures associated to the cube: price, purpose, etc. organized in a measure group. Then specifies the dimensions associated with the cube "Trip" (Passenger, Flight, etc.) and for each one defines its attributes and classification hierarchies.

```
<Cube> <ID>Trip</ID><Name>Trip</Name>
  <MeasureGroups>
    <MeasureGroup> <ID>Trip</ID><Name>Trip</Name>
      <Measures>
        <Measure><ID>price</ID><Name>price</Name></Measure>
        <Measure><ID>purpose</ID><Name>purpose</Name></Measure>
      ... </Measures>
    </MeasureGroup>  </MeasureGroups>
  <Dimensions>
  <Dimension>
    <ID>Passenger</ID><Name>Passenger</Name> <DimensionID>Passenger</DimensionID>
    <Attributes> <Attribute><AttributeID>passengerCode</AttributeID> </Attribute>
      <Attribute><AttributeID>name</AttributeID></Attribute>
    ... </Attributes>
   </Dimension>
  <Dimension> <ID>Flight</ID><Name>Flight</Name> <DimensionID>Flight</DimensionID>
    <Attributes>
      <Attribute><AttributeID>flightCode</AttributeID></Attribute> </Attributes>
    <Hierarchies>
      <Hierarchy> <HierarchyID>FlightHierarchy0</HierarchyID></Hierarchy>  </Hierarchies>
  </Dimension>... </Dimensions> </Cube>
```

Table 6: SSAS implementation: cubes

```
<Cube> <ID>Trip</ID> <Name>Trip</Name>
<CubePermissions>
  <CubePermission>
    <ID>CubePermissionTrip+SLTS</ID> <Name>CubePermissionTrip+</Name>
    <RoleID>SLTS</RoleID>
    <Process>true</Process><Read>Allowed</Read>
    <DimensionPermissions>
        <DimensionPermission> <CubeDimensionID>Trip</CubeDimensionID>
        <Read>Allowed</Read> </DimensionPermission> </DimensionPermissions>
  </CubePermission> ...
  <CubePermission>
    <ID>MemberPermissionpurpose+SRSecurity</ID>
    <Name>MemberPermissionpurpose+</Name>
    <RoleID>SRSecurity</RoleID>
    <Process>true</Process><Read>Allowed</Read>
    <DimensionPermissions>
      <DimensionPermission> <CubeDimensionID>Trip</CubeDimensionID>
      <Read>Allowed</Read> </DimensionPermission></DimensionPermissions>
    <CellPermissions><CellPermission> <Access>Read</Access>
      <Expression>Trip.purpose</Expression> </CellPermission></CellPermissions>
  </CubePermission> ... </CubePermissions> </Cube>
```

Table 7: SSAS implementation: cubes (security permissions)

All the security information needed to avoid unauthorized accesses is included in this file as security permissions for cubes or cells. Table 7 shows how several security permissions are associated to the cube "Trip", one of them authorizes accesses for the role "SLTS" (similar permissions for other roles have been omitted in this Table) and the last one is a fine grain security permission that authorizes accesses to the measure "purpose" for the role "SRSecurity".

6.3 Dimensions

Finally, the structural and security aspects related with dimensions are analyzed and transformed into a XML file for each dimension. Table 8 partially shows the implementation generated for the dimension "Flight". It indicates the attributes of the dimension (flightCode that is the key attribute, planeCode, etc.), and the classification hierarchies with the different aggregation levels (the hierarchy "FlightHierarchy0" with levels "Plane", "AircraftType" and "Company").

Then, the security constraints associated to dimensions or attribute dimensions are transformed into security permissions. Table 9 shows how for the dimension "Flight" are generated two sets of security permissions that authorize the roles that satisfy the security requirements (security roles "SLTS", "SLS", and "SLC") and hide information for unauthorized roles (security role "SLU").

```
<Dimension> <ID>Flight</ID><Name>Flight</Name>
 <Attributes>
  <Attribute><ID>flightCode</ID><Name>flightCode</Name><Usage>Key</Usage>
   <KeyColumns><KeyColumn><DataType>WChar</DataType></KeyColumn> </KeyColumns>
  </Attribute>
  <Attribute> <ID>planeCode</ID><Name>planeCode</Name>
   <KeyColumns> <KeyColumn><DataType>WChar</DataType></KeyColumn> </KeyColumns>
  </Attribute> ... </Attributes>
<AttributeRelationships>...</AttributeRelationships>
<Hierarchies> <Hierarchy> <ID>FlightHierarchy0</ID><Name>FlightPlane</Name>
<Levels>
   <Level><ID>Plane</ID><Name>Plane</Name>
   <SourceAttributeID>planeCode</SourceAttributeID> </Level> ...
</Levels> </Hierarchy> </Hierarchies> </Dimension>
```

Table 8: SSAS implementation: dimensions

```
<Dimension> <ID>Flight</ID><Name>Flight</Name>
<DimensionPermissions>
<DimensionPermission>
   <ID>DimensionPermissionFlight+SLTS</ID><Name>DimensionPermissionFlight+</Name>
   <RoleID>SLTS</RoleID><Process>true</Process><Read>Allowed</Read>
   <AllowedSet>Flight</AllowedSet><DeniedSet></DeniedSet>
   <AttributePermissions>...</AttributePermissions> </DimensionPermission> ...
<DimensionPermission>
   <ID>DimensionPermissionFlight-SLU</ID><Name>DimensionPermissionFlight-</Name>
   <RoleID>SLU</RoleID><Process>true</Process><Read>Allowed</Read>
   <AllowedSet></AllowedSet> <DeniedSet>Flight</DeniedSet>
   <AttributePermissions>...</AttributePermissions>
</DimensionPermission> ... </DimensionPermissions> </Dimension>
```

Table 9: SSAS implementation: dimensions (security permissions)

6.4 Queries in SSAS

The meta-information generated in these XML files is used by SSAS when users query the DW in order to provide authorized information. Figures 12 and 13 are screenshots of user queries that check the correct implementation of a security constraint "SIARBaggageSuspicious". This security rule was defined in the conceptual model and automatically transformed into SSAS code. It specifies that the security privileges required to access information about baggages is a security level of confidential and a security role of security or baggage, but those baggages identified as suspicious requires more restrictive security privileges: a security level of secret and a security role of security.

In Figure 12, a user with a security level of secret and a security role of security, queries information of baggages related with airports. This user satisfies the more restrictive security constraints and is authorized to show information about all the baggages (suspicious and not suspicious).

Nevertheless, in Figure 13, a user with lower security privileges try to query the same information. This user has a security level of confidential and a security role of baggaging. As can be shown in Figure 13 this user solely receives information of a subset of baggages which are the not suspicious baggages.

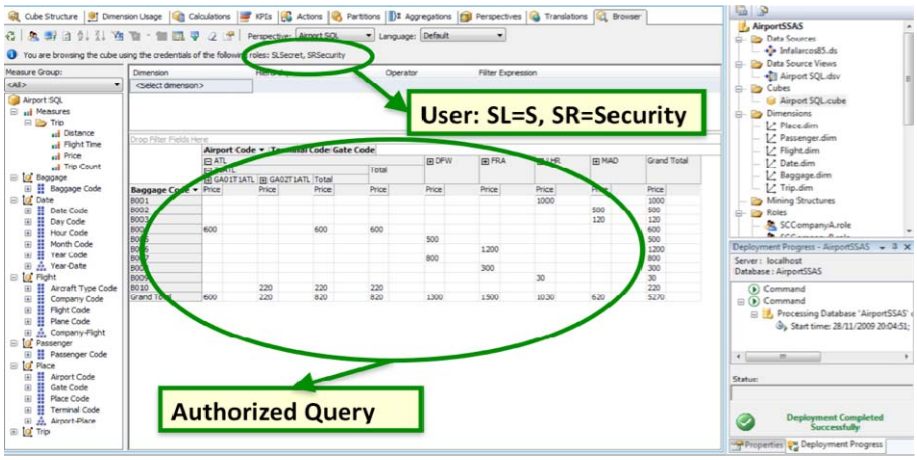

Figure 12: SSAS implementation: authorized query

7 Benefits of our Architecture

The proposal for developing secure data warehouses presented in this paper has been conceived as a model driven architecture [Mellor et al., 2003][Bézivin, 2004]. Model Driven Development is based on the definition of models which separate the specification of system functionalities and its implementation by using a specific technology. Furthermore, the development process can be automated by defining transformations from models towards the final implementation. This approach improves the development process reducing times and costs and also improves the

quality obtained in the final product [Fernández-Medina et al., 2009][Mouratidis, 2011].

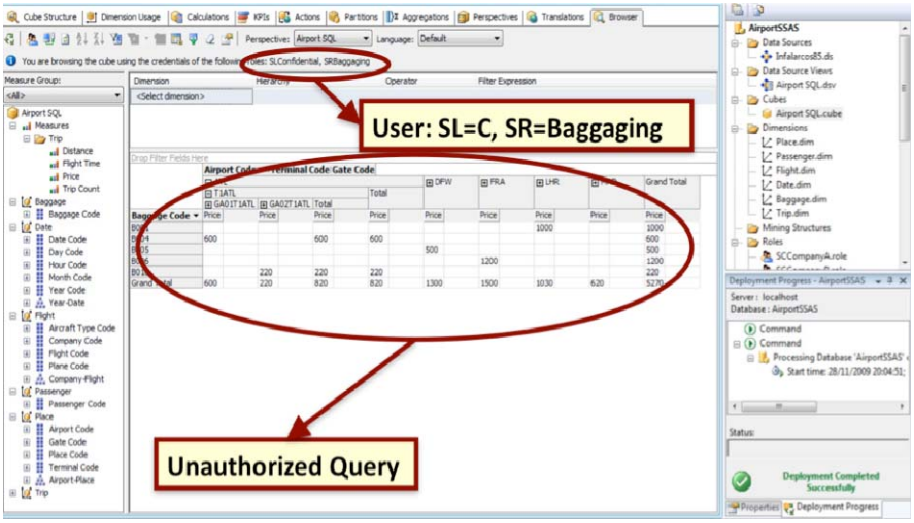

Figure 13: SSAS implementation: unauthorized query

The Model Driven approach has been successfully applied to different software development areas such as data bases [Vela et al., 2004][Li et al., 2005][Dubielewicz et al., 2007][Vara et al., 2007], data warehouses (Mazon & Trujillo, 2008; Vela et al., 2010; Fernandes et al., 2010), web services [Meliá and Gomez, 2006][Tan et al., 2006][Yu et al., 2006][Kraus et al., 2007], product lines [Braganca, 2007][Braganca and Machado, 2007], critical appications [Moebius et al., 2009] or real-time systems [Cuccuru et al., 2005][Lu et al., 2005][Wang et al., 2006].

In this work we have aligned the development stages of a data warehouse with a model driven architecture. Our proposal starts with the definition of a conceptual model that includes booth, structural and security aspects of the DW. Then, this model is automatically transformed into a logical model focused on the OLAP technology and finally into a secure implementation for an OLAP tool. In this way we save time and cost in the development process.

Moreover, the quality of the final solution obtained is also improved by using our approach which is focused on the security improvement. Our proposal allows designers to define conceptual models including security constraints within structural aspects. At the conceptual abstraction level designers model security restrictions in an easier and more understandable way than establishing security directly into the final implementation. The use of a more understandable model mitigates possible mistakes derived from managing a vast amount of code.

On the other hand, the identification and inclusion of security aspects into the models corresponding to early development stages improve the final product quality. If security constraints are early modelled, these security constraints are considered for

making design decisions when the system is automatically generated by using transformations. So, the security constraints are perfectly fit into the final solution.

The strategy that we have adopted to evaluate our architecture is guided by case studies. In a first stage our proposal was applied to several case studies such as data warehouses for managing hospital admissions or product sales. Next, in this work we have applied our architecture to a more complex case study in which we develop a data warehouse for an airport, managing information of flights, passengers and so on. The next step for validating our proposal is to apply it to industrial case studies with the participation of professional designers who could provide us useful feedback. It will be completed with a family of experiments for measuring how much our proposal improves the efficiency, understandability and security compared to the traditional development process.

The case study presented in this paper has been very useful to improve our model driven architecture for developing secure OLAP applications. Firstly, a reduced version of this case study was applied in order to evaluate the applicability of the first versions of our metamodels and transformations and improve them. Next, the case study was extended until its final version presented in this paper which allow us to evaluate the correct modelling and transformation of all the structural elements and security constraints. Now, we describe some examples of issues detected and improved after applying this case study.

Our proposal for the conceptual modelling of secure DWs was improved by adding new modelling elements that allows us to represent security constraints and complex security rules easier. For instance, a security element called "SecurityInformation" was introduced after detect that designers used to make errors when they assigned the same permission sets (security roles, compartments and levels) to different elements, since they had to repeat them manually. The new element (SecurityInformation) is a permission set that can be reused in order to mitigate these mistakes.

Furthermore, complex security rules (SIARs and AURs) were specified using notes with OCL expressions. These expressions are difficult to analyze and they were not automatically transformed toward the final implementation. This issue was improved, by including in the conceptual model new elements for representing security rules and defining the transformation rules needed to obtain security permissions in the logical model and their final implementation.

The logical model for secure OLAP applications is automatically obtained by applying transformations from the conceptual model and designer do not used to modify it. Nevertheless, some improvements were detected and incorporated to our proposal in order to achieve a better expressiveness. For instance, the logical model was initially composed of three metamodels (for security configuration, cubes and dimensions), and evolves to a unified model which was defined extending the OLAP package of CWM, searching for using standards proposed by OMG.

8 Conclusions

DWs manage vital business information which is very sensitive and has to be correctly assured in order to avoid unauthorized accesses. Because an early detection of security requirements has influence in the further design decisions providing better

security specifications and final products, security constraints should be considered in the whole development process from early stages to final tools. Furthermore, since users query the information by using OLAP tools which manage specific cubes or views from the corporative DW, security constraints should be also defined in this metadata layer by using the same multidimensional elements that will be managed by the final users.

Thanks to our proposal we are able to automatically develop secure OLAP applications from conceptual models, providing a complete model driven architecture that has been described in this paper by using a case study.

This architecture is composed of several security models and automatic transformations towards the final secure implementation: (i) a conceptual model for secure data warehouses (SECDW); (ii) a logical model for secure OLAP applications (SECMDDW) which is based on the OLAP package of CWM extended with security capabilities; (iii) a set of QVT transformations to automate the generation of logical models (PSM) from our conceptual models (PIM); (iv) the corresponding model-to-text transformations which allows the automatic code generation into a specific OLAP tool (SSAS) from the logical models (PSM).

As a further work we will complete the evaluation of our architecture by applying it to industrial case studies with the participation of professional designers. We will define a family of experiments in order to measure the improvement obtained in comparison to the traditional development process. On the other hand will improve this architecture in several lines: (i) including support for other PSM models (such as cloud) and final platforms (such as Pentaho); (ii) defining inverse transformations for allowing modernization processes; and (iii) including dynamic security models which complement the existing models dealing with the inference security problem.

Acknowledgements

This research is part of the following projects: SERENIDAD (PEII11- 037-7035) financed by the ”Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha” (Spain) and FEDER, and SIGMA-CC (TIN2012-36904) and GEODAS (TIN2012-37493-C03-01) financed by the ”Ministerio de Economía y Competitividad” (Spain).

References

[Abelló et al., 2006] Abelló, A., J. Samos and F. Saltor. "YAM2: a multidimensional conceptual model extending UML." Information Systems **31**(6): 668-677, 2006.

[Alberts and Dorofee, 2002] Alberts, C. and A. Dorofee. Managing information security risks: The OCTAVE (SM) approach, Boston: Addison Wesley. 2002.

[Barreto et al., 2010] Barreto, F. J., A. Dias and A. Bessa. Security Engineering Approach to Support Software Security. World Congress on Services**:** 48-55. 2010.

[Basin et al., 2006] Basin, D., J. Doser and T. Lodderstedt. "Model Driven Security: from UML Models to Access Control Infrastructures." ACM Transactions on Software Engineering and Methodology **15**(1): 39-91, 2006.

[Bézivin, 2004] Bézivin, J. "In Search of a Basic Principle for Model Driven Engineering." Upgrade **5**(2): 21-24, 2004.

[Binh et al., 2000] Binh, N. T., A. M. Tjoa and R. Wagner. An object oriented multidimensional data model for OLAP. Web-Age Information Management. **LNCS 1846:** 69-82. 2000.

[Blanco et al., 2009] Blanco, C., E. Fernández-Medina, J. Trujillo and M. Piattini (2009). Data Warehouse Security. Encyclopedia of Database Systems. L. Liu and M. T. Ozsu, Springer**:** 675-679.

[Blanco et al., 2011] Blanco, C., D. G. Rosado, C. Gutiérrez, A. Rodríguez, D. Mellado, E. Fernández-Medina, J. Trujillo and M. Piattini (2011). Security over the Information Systems Development Cycle. Software Engineering for Secure Systems: Industrial and Research Perspectives. H. Mouratidis. USA, IGI Group Inc**:** 113-154.

[Braganca, 2007] Braganca, A. (2007). Methodological Approaches and Techniques for Model Driven Development of Software Product Lines.

[Braganca and Machado, 2007] Braganca, A. and R. Machado. Model Driven Development of Software Product Lines. International Conference on the Quality of Information and Communications Technology, Lisbon, Portugal. 2007.

[Cuccuru et al., 2005] Cuccuru, A., R. De Simone, T. Saunier, G. Siegel and Y. Sorel. P2I: An Innovative MDA Methodology for Embedded Real-Time Systems. Euromicro Conference on Digital System Design. Porto, Portugal**:** 26-33. 2005.

[CWM, 2003] CWM, O. M. G. "Common Warehouse Metamodel (CWM)." 2003.

[Dubielewicz et al., 2007] Dubielewicz, I., B. Hnatkowska, Z. Huzar and L. Tuzinkiewicz. Evaluation of MDA-PSM database model quality in the context of selected non-functional requirements. International Conference on Dependability of Computer Systems. Szklarska Poreba, Poland**:** 19-26. 2007.

[Fernández-Medina et al., 2009] Fernández-Medina, E., J. Jurjens, J. Trujillo and S. Jajodia. "Model-Driven Development for secure information systems." Information and Software Technology **51**(5): 809-814, 2009.

[Fernández-Medina et al., 2006] Fernández-Medina, E., J. Trujillo, R. Villarroel and M. Piattini. "Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses." Decision Support Systems **42**: 1270-1289, 2006.

[Fernández-Medina et al., 2007] Fernández-Medina, E., J. Trujillo, R. Villarroel and M. Piattini. "Developing Secure Data Warehouses with a UML extension." Information Systems **32**(6): 826-856, 2007.

[Giorgini et al., 2006] Giorgini, P., H. Mouratidis and N. Zannone (2006). Modelling Security and Trust with Secure Tropos. Integrating Security and Software Engineering: Advances and Future Visions, Idea Group Publishing.

[Golfarelli et al., 1998] Golfarelli, M., D. Maio and S. Rizzi. "The Dimensional Fact Model: A Conceptual Model for Data Warehouses." International Journal of Cooperative Information Systems (IJCIS) **7.**(2-3): 215-247, 1998.

[Inmon, 2008] Inmon, W. H. 2.0 - architecture for the next generation of data warehousing, Morgan Kaufmann. 2008.

[ISO/IEC, 2005] ISO/IEC. ISO/IEC 15408 (Common Criteria v3.0) "Information Technology Security Techniques-Evaluation Criteria for IT Security". 2005.

[ISO/IEC, 2005] ISO/IEC. ISO/IEC 17799-27002 Code of Practice for Information Security Managament. 2005.

[Jajodia and Wijesekera, 2001] Jajodia, S. and D. Wijesekera. "Security in Federated Database Systems." Information Security Technical Report **6**(2): 69-79, 2001.

[Jurjens, 2004] Jurjens, J. Secure Systems Development with UML, Springer-Verlag. 2004.

[Jurjens and Schmidt, 2011] Jurjens, J. and H. Schmidt. UMLsec4UML2 - Adopting UMLsec to Support UML2. http://hdl.handle.net/2003/27602, Technical Reports in Computer Science. Technische Universitat Dortmund. 2011.

[Kirkgoze et al., 1997] Kirkgoze, R., N. Katic, M. Stolda and A. Min Tjoa. A Security Concept for OLAP. 8th International Workshop on Database and Expert System Applications (DEXA'97). Toulouse, France, IEEE Computer Society**:** 619-626. 1997.

[Kraus et al., 2007] Kraus, A., A. Knapp and N. Koch. Model-Driven Generation of Web Applications in UWE. International Workshop on Model-Driven Web Engineering. Como, Italy. 2007.

[Li et al., 2005] Li, B., S. Liu and Z. Yu. Applying MDA in traditional Database-based Application Development. International Conference on Computer Supported Cooperative Work in Design. Coventry, UK**:** 1038-1041. 2005.

[Liu et al., 2006] Liu, Y., S. Sung and H. Xiong. "A cubic-wise balance approach for privacy perservation in data cubes." Information Sciences **176**(9): 1215-1240, 2006.

[Lodderstedt et al., 2002] Lodderstedt, T., 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, Dresden, Germany, Springer. 2002.

[Lu et al., 2005] Lu, S., W. A. Halang and L. Zhang. A Component-based UML Profile to Model Embedded Real-Time Systems Designed by the MDA Approach. International Conference on Embedded and Real-Time Computing Systems and Applications. Hong Kong, China**:** 563-566. 2005.

[Luján-Mora et al., 2006] Luján-Mora, S., J. Trujillo and I. Y. Song. "A UML profile for multidimensional modeling in data warehouses." Data & Knowledge Engineering **59**(3): 725-769, 2006.

[Meliá and Gomez, 2006] Meliá, S. and J. Gomez. "The WebSA Approach: Applying Model Driven Engineering to Web Applications." Journal of Web Engineering **5**(2): 121-149, 2006.

[Mellor et al., 2003] Mellor, Clark and Futugami. "Model-driven development - Guest editor's introduction." IEEE Software **20**(5): 14-18, 2003.

[Moebius et al., 2009] Moebius, N., K. Stenzel and W. Reif. Generating formal specifications for security-critical applications - A model-driven approach**:** 68-74. 2009.

[Mouratidis, 2011] Mouratidis, H. Software Engineering for Secure Systems: Industrial and Research Perspectives, IGI Global. 2011.

[Mundy et al., 2011] Mundy, J., W. Thornthwaite and R. Kimball. The Microsoft Data Warehouse Toolkit: With SQL Server 2008 R2 and the Microsoft Business Intelligence Toolset, Wiley. 2011.

[OMG, 2003] OMG. CWM. Common Warehouse Metamodel. Version v1.1, http://www.omg.org/spec/CWM/1.1. 2003.

[OMG, 2003] OMG. MDA. Model Driven Architecture Guide Version 1.0.1. 2003.

[Paim and Castro, 2003] Paim, F. R. S. and J. Castro. DWARF: An approach for requirements definition and management of data warehouse systems. IEEE International Conference on Requirements Engineering**:** 75-84. 2003.

[Prat et al., 2006] Prat, N., J. Akoka and I. Comyn-Wattiau. "A UML-based data warehouse design method." Decision Support Systems **42**(3): 1449-1473, 2006.

[Priebe and Pernul, 2001] Priebe, T. and G. Pernul. A Pragmatic Approach to Conceptual Modeling of OLAP Security. 20th International Conference on Conceptual Modeling (ER 2001), Yokohama, Japan, Springer-Verlag. 2001.

[Priebe and Pernul, 2001] Priebe, T. and G. Pernul. A Pragmatic Approach to Conceptual Modeling of OLAP Security. 20th International Conference on Conceptual Modeling (ER 2001). Yokohama, Japan, Springer-Verlag. 2001.

[Saltor et al., 2002] Saltor, F., M. Oliva, A. Abelló and J. Samos. Building Secure Data Warehouse Schemas from Federated Information Systems. Heterogeneous Inf. Exchange and Organizational Hubs. H. Bestougeff, Dubois, J.E., Thuraisingham, B. Dordrecht, The Netherlands, Kluwer Academic Publisher**:** 123-134. 2002.

[Sapia et al., 1998] Sapia, C., M. Blaschka, G. Höfling and B. Dinter. Extending the E/R Model for the Multidimensional Paradigm. 1st International Workshop on Data Warehouse and Data Mining (DWDM'98), Singapore, Springer-Verlag. 1998.

[Shoshani, 1997] Shoshani, A. OLAP and Statistical Databases: Similarities and Differences. PODS 97, Tucson, AZ. 1997.

[Simitsis and Vassiliadis, 2007] Simitsis, A. and P. Vassiliadis. "A method for the mapping of conceptual designs to logical blueprints for ETL processes." Decision Support Systems, 2007.

[Soler et al., 2009] Soler, E., J. Trujillo, C. Blanco and E. Fernández-Medina. "Designing Secure Data Warehouses by Using MDA and QVT." Journal of Universal Computer Science (JUCS) **15**(8): 1608-1641, 2009.

[SSE-CMM, 2003] SSE-CMM. Systems Security Engineering - Capability Maturity Model. http://www.sse-cmm.org. 2003.

[Sung et al., 2006] Sung, Y., Y. Liu, H. Xiong and P. A. Xg. "Privacy preservation for data cubes." Knowledge Information Systems **9**(1): 38-61, 2006.

[Tan et al., 2006] Tan, W., L. Ma, J. Li and Z. Xiao. Application MDA in a Conception Design Environment. International Multi-Symposiums on Computer and Computational Sciences. Hangzhou, China**:** 702-704. 2006.

[Thuraisingham, 1994] Thuraisingham, B. "Security Issues for Federated Database Systems." Computer and Security **13**(6): 509-525, 1994.

[Thuraisingham et al., 2007] Thuraisingham, B., 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) **2**(4): 367-382, 2007.

[Trujillo and Luján-Mora, 2003] Trujillo, J. and S. Luján-Mora (2003). A UML Based Approach for Modeling ETL Processes in Data Warehouses. Conceptual Modeling - ER 2003. S. B. Heidelberg. Volume 2813/2003: 307-320.

[Trujillo et al., 2009] Trujillo, J., E. Soler, E. Fernández-Medina and M. Piattini. "An Engineering Process for Developing Secure Data Warehouses." Information and Software Technology **51**(6): 1033-1051, 2009.

[Tryfona et al., 1999] Tryfona, N., F. Busborg and J. Christiansen. starER: A Conceptual Model for Data Warehouse Design. ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP'99), Missouri, USA, ACM. 1999.

[van de Riet, 2008] van de Riet, R. P. "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.

[Vara et al., 2007] Vara, J. M., B. Vela, J. M. Cavero and E. Marcos. Model transformation for object-relational database development. ACM Symposium on Applied Computing. Seoul, Korea**:** 1012-1019. 2007.

[Vela et al., 2004] Vela, B., C. J. Acuna and E. Marcos. A Model Driven Approach for XML Database Development. International Conference on Conceptual Modeling. Shanghai, China**:** 780-794. 2004.

[Wang et al., 2004] Wang, L., S. Jajodia and D. Wijesekera. Securing OLAP Data Cubes Against Privacy Breaches. IEEE Symposium on Security and Privacy. Berkeley, California**:** 161-178. 2004.

[Wang et al., 2006] Wang, Y., X. Zhou, L. Liang and C. Peng. A MDA based SoC Modeling Approach using UML and SystemC. International Conference on Computer and Information Technology. Baridhara, Bangladesh**:** 245-251. 2006.

[Weippl et al., 2001] Weippl, E., O. Mangisengi, W. Essmayr, F. Lichtenberger and W. Winiwarter. An Authorization Model for Data Warehouses and OLAP. Workshop on Security in Distributed Data Warehousing. New Orleans, Louisiana, USA. 2001.

[Yu et al., 2006] Yu, B., C. Zhang and Y. Zhao. Transform from Models to Service Description Based on MDA. Asia-Pacific Conference on Services Computing. GuangZhou, China**:** 605-608. 2006.