SECRDW: An Extension of the Relational Package from CWM for Representing Secure Data Warehouses at the Logical Level

Emilio Soler[1], Juan Trujillo[2], Eduardo Fernández-Medina[3] and Mario Piattini[3]

[1] Departamento de Informática. University of Matanzas
Autopista de Varadero km 3. Matanzas, Cuba

[2] Departamento de Lenguajes y Sistemas Informáticos. University of Alicante
C/ San Vicente S/N 03690 Alicante, Spain

[3] Grupo ALARCOS, Departamento de Tecnologías y Sistemas de Información
Centro Mixto de Investigación y Desarrollo de Software UCLM-Soluziona
University of Castilla-La Mancha
Paseo de la Universidad, 4 - 13071 Ciudad Real, Spain

Abstract. Data Warehouses (DWs) constitute a valuable support to store extensive volumes of historical data for the decision making process. For this reason, it is vital to incorporate security requirements from the early stages of the DWs projects and enforce them in the further design phases. Very few approaches specify security and audit measures in the conceptual modeling of DWs. Furthermore, these security measures are specified in the final implementation on top of commercial systems as there is not a standard relational representation of security measures for DWs. On the other hand, the Common Warehouse Metamodel (CWM) has been accepted as the standard for the exchange and the interoperability of the metadata. Nevertheless, it does not allow us to specify security measures for DWs. In this paper, we make use of the own extension mechanisms provided by the CWM to extend the relational package to specify at the logical level the security and audit rules captured during the conceptual modeling phase of the DWs design. Finally, in order to show the benefits of our extension, we apply it to a case study related to the management of the pharmacies consortium businesses.

1 Introduction

According to the current development of the digital technology, the organizations began to adopt more and more computerized information systems, which rely upon databases and DWs. Therefore, the very survival of the organization depends on the appropriate manipulation of the security and confidentiality of the corresponding information [3]. Normally in the DWs projects, security aspects are implemented in the final stages of the design. However, the information security is a serious requirement which must be given careful thought to, not as an isolated aspect, but as an present element in all development lifecycle stages, from requirement analysis to implementation and maintenance [2]. The

Soler E., Trujillo J., Fernández-Medina E. and Piattini M. (2007).
SECRDW: An Extension of the Relational Package from CWM for Representing Secure Data Warehouses at the Logical Level.
In *Proceedings of the 5th International Workshop on Security in Information Systems*, pages 245-256
DOI: 10.5220/0002438002450256

above-mentioned justifies that is vital to incorporate confidentiality measures in the design of DWs and enforce them.

On the other hand, it is widely accepted that the DW design is based on the multidimensional (MD) modeling which structures the information into facts and dimensions. For the design of DWs we base our proposal on Model Driven Architecture (MDA) [14]. MDA proposes several models at different levels: at conceptual level the Platform Independent Model (PIM) and at the logical level the Platform Specific Model (PSM). In our context, the PIM corresponds the conceptual MD modeling based on the UML presented in the works [6, 5, 24], which extended the proposal based on UML [9], in order to incorporate security requirements in the conceptual design of DWs. The PSM corresponds with our extension of the CWM at the logical level.

The previous work presented in [11] employ MDA for the DWs development, choosing the relational metamodel from CWM [13]. The relational package of the CWM enables mediated interchange between relational DBs from the majority of relational commercial systems [18]. However, security and audit measures cannot be modeled in the CWM because it does not provide the modeling constructors for representing data security related to issues such as access rights, users or roles [12]. Most data access control approaches are based on the proprietary metadata structures of specific software products [17], thus, integrating security related to metadata into the CWM improve the security support and facilitate the establishment of a standardized access control mechanism for data warehouses [12]. According to MDA we do not need the metadata of a DBMS; we need a metamodel that allows us to represent security and audit measures at the logical level. Hence, is this paper we present an extension of the relational metamodel from CWM by using its own extensions mechanisms. By this way we represent, at the logical level, all the security and audit measures captured during the conceptual modeling phase of the DWs design.

The rest of the paper is structured as follows. The works related to our proposal are discussed in section 2. Secure multidimensional modeling is introduced in section 3. Section 4 shows an overview of the CWM. Section 5 presents our extension of the relational metamodel from CWM, next, in section 6 we show a case study in order to show the benefits to use our extension in the design of secure DWs. Finally, section 7 draws the main conclusions and outlines our immediate future work.

2 Related Work

Relevant literature on this subject comprises several initiatives to include security in the DW design. In [7] the authors describe a prototype model for DWs security based on metadata, which enable to define views of data for each group of users, however, it does not permit to specify complex restrictions of confidentiality. Rosenthal and Sciore [19], extend SQL grants and create a mechanism of inferences to establish the security. Another attempt is the architecture for both Federated Information Systems (FIS) and DWs that preserve MultiLevel security integration between FIS and DWs [20]. These approaches ([7, 19, 20]) are extractives but only focus on practical issues such as acquisition, storage and access control at the OLAP side. None of them examine the representation of security into both, at conceptual and logical stage.

On the other hand, there are more elaborated initiatives that propose models of authorization for the DWs design. For example, in [8] the authors propose a security concept for OLAP, which is a role based security model for data warehouses. Priebe and Pernul [17] propose a security design methodology similar to the classical database design methodology (requirement analysis, conceptual, logical, and physical design) covering requirements and concrete implementations in commercial systems. The same authors (Priebe and Pernul) in [16] extend the ADAPTed UML model for the previous conceptual phase, specifying a methodology and a MD security constraint language for conceptual modeling of OLAP security. In [4] the authors show that access privileges for DWs and OLAP can be expressed more intuitively than using SQL's grant statements, their access control model focus specifically on expressiveness and usability. These proposals ([8, 16, 17]) offer security models at the conceptual level by means of security constraints, but basically deal with OLAP operations. These proposals [17, 16] are one of the best references in this area. As a summary, these works implements the security rules considered in their conceptual approach in commercial database systems. On the other hand, we base our approach in the works [5, 6, 24] in which the authors claim for the design of the security rules in all stages of the DWs design, from conceptual to final implementation. And therefore, in this paper, we formally extend the CWM in order to allow us to automatically transform the security rules considered at the conceptual level in the logical representation of the DWs.

Numerous proposals exist that extends CWM with different objectives: for the modeling of logical object-oriented relational data storage and the corresponding ETL process [10], for universal data mining library that implements data mining methods and algorithms [23], for recording the trace information of metadata evolution and maintain consistency during metaclass evolution [25], for representing and integrate the metadata generated by data and metadata lineage implementation [21] and for providing quality information to DW client tools [1] and for building a conceptual model for data quality and cleaning, both applicable to operational and data warehousing context. However, none of the previous proposals extend the relational metamodel from CWM with security aspects. Only the work presented in [22] shows how the CWM could be adequate for representing security measures for DWs at the logical level. In this paper the CWM is not formally extended through the formal extension mechanisms.

3 Secure Multidimensional Modeling

The main properties of the MD modeling are represented by UML profile [9], which is based on OO conceptual modeling. In [6], the previous profile is reused in order to be able to design an MD conceptual model classifying both information and users in order to represent the main security aspects in the conceptual modeling of DWs. Therefore, the profile allows us to classify the security information that will be used in our conceptual modeling of data warehouses. For each element of the model (fact class, dimension class, fact attribute, etc.), is defined its security information, specifying a sequence of security levels, a set of user compartments and a set of user roles. Security constraint is considered to specify security in attributes. The security information and these constraints indicate the security properties that users have to be able to access information. The description of the profile is represented as a UML package. All the above constraints (AuditRule, AuthorizationRule and SecurityRule) are modeled using UML notes.

In the considered SMD modeling (Secure Multidimensional Modeling), the structural properties of MD modeling are represented by means of a UML class diagram in which the information is clearly organized into facts and dimensions. These facts and dimensions are represented by SFact and SDimension classes respectively, where S is the abbreviation of secure. With respect to SDimensions, each level of a classification hierarchy is specified by a SBase class. An association of SBase classes specifies the relationship between two levels of a classification hierarchy. Every SBase class must also contain an identifying SAttribute OID (SOID) and a SDescriptor attribute (SD). The class called UserProfile will contain information of all users entitled to access to the MD model. An example of secure MD modeling is shown in Fig. 4 of the section 6.

In the following section we present a general description of the CWM, emphasizing the different mechanisms for their extension.

4 An Overview of the CWM

The main purpose of the CWM [13] is to enable easy interchange of warehouse and business intelligence metadata between warehouse tools, warehouse platforms and warehouse metadata repositories in distributed heterogeneous environments. CWM is based on three key industry standards: i) UML, an OMG modeling standard, ii) MOF (Meta Object Facility), an OMG metamodeling and metadata repository standard, and iii) XMI (XML Metadata Interchange), an OMG metadata interchange standard.

The UML standard defines a rich object oriented modeling language that is supported by a range of graphical design tools. The MOF standard defines an extensible framework for defining models for metadata, and providing tools with programmatic interfaces to store and access metadata in a repository. The XMI standard allows metadata to be interchanged as streams or files with a standard format based on XML. CWM has been designed to conform to the ”MOF model”, it belongs to the M2 layer, we refer the reader to [13, 18] for further details on the different metamodel layers of the CWM.

4.1 Organization of the CWM

CWM is organized in 21 separate packages which are grouped into five stackable layers by means of similar roles[2]. We will mainly focus our work on the Resource layer and, more precisely, on the Relational package as a relational metamodel that describes the corresponding metadata of the relational data resources. The Resource layer describes the structure of data resources that act as either sources or targets of a CWM mediated interchange. The Relational package describes data accessible through a relational interface such as a native RDBMS, Object DB Connectivity, or Java DB Connectivity.

[2] For more details we refer the reader to [13]

4.2 CWM Extensibility Mechanism

CWM provides extension mechanisms to build specific metamodels. According to [13], there are two general techniques to extend CWM: Use of the general extension mechanisms provided by the UML Object Model, by means of tagged values and stereotypes. This approach is usually used for minor extensions (for example additional attributes to objects model) that are not significant enough to require the creation of a specific model. The second variant is non-normative model extensions or modeled extensions [18] documented as additional metamodel packages that extend the CWM metamodel. This proposal is used for more complex extensions, CWM itself is built following this extension type. To represent security aspects at the logical level we need to introduce new classes and associations, hence, the non-normative extension is the preferred mechanism, because it is not a simple extension [18].

In the next section, we use the non-normative extension mechanism to extend the Relational package, in order to represent security and audit rules at the logical level.

5 The SECRDW Extension

The extension of the relational package from CWM defines new classes to allow representing at the logical level all the security and audit requirements captured during the conceptual modeling phase of DWs design. This extension will be called SECure Relational Data Warehouses (SECRDW) metamodel, which depends on the following packages: Relational, Core and Data Types.

5.1 Inheritance

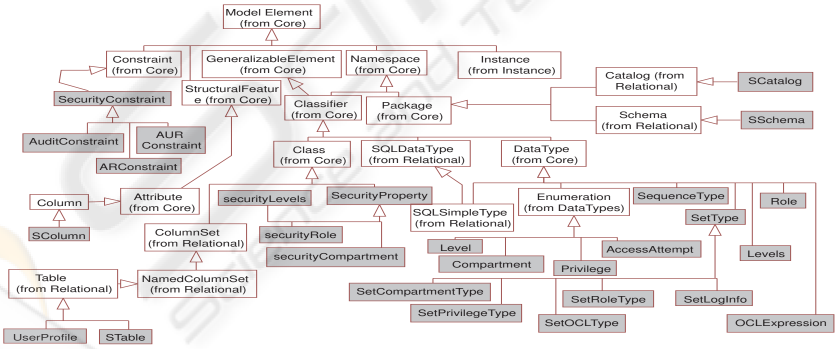


Fig. 1. SECRDW Package Inheritance.

In Fig.1 we show the new classes that conform the SECRDW package colored in grey, whereas classes from the CWM metamodel remain white. The SSchema (SCatalog) classes specialize the schema (catalog) classes to allow a secure schema (catalog). STable and UserProfile specializes to the Table metaclass. SColumn is specialized in the Column metaclass. The UserProfile table is a special table that store information of users with access to the systems, these rights are specified by SecurityProperty (securityLevel, securityCompartment and securityRole). STable and SColumn has associate security information by means of SecurityProperty (securityLevel, securityCompartment and securityRole). SecurityProperty specializes to the Class (from Core) metaclass, with it, we establish by means of securityLevel, securityCompartment and securityRole access properties over tables and columns that the user must be fulfilled to accede to the same ones. AuditConstraint is useful both as a deterrent against misbehavior as well for analyzing the user behavior by employing the system to find out possible attempted or actual violations. AuditConstraint is essential to record the access to tables and columns performed by users. ARConstraint allows to define rules for specifying multilevel security policies in tables and columns. AURConstraint, may coexist with ARConstraint, and enable to specify the access to the tables and columns, thus permitting us to specify security models which are much more elaborate. The SecurityConstraint class logically inherits properties of the Constraint class from Core. The data types are studied more in depth in the following section.

5.2 New Data Types

In general, the CWM packages only support data type attributes that are considered necessary for information interchange between systems [13]. To represent security and audit information at the logical level we need new data types. In Fig. 2 new classes appears that inherit from DataType or from Enumeration classes. The new classes that represent new data types appear with gray color in Fig. 2. These new data types are necessary to model the access properties (securityProperty) and the constraints (SConstraint) to STable, UserProfile and SColumn.

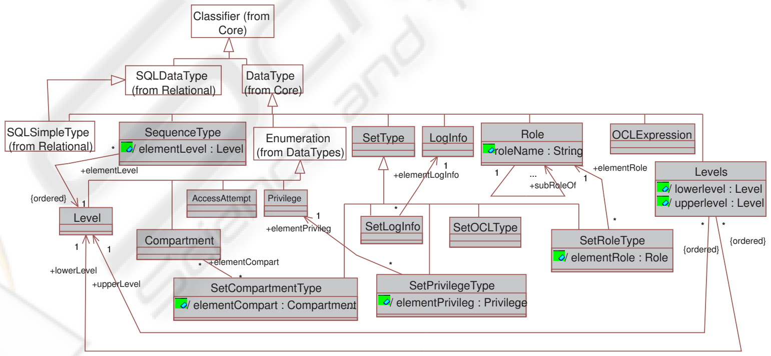


Fig. 2. New Data Types for SECRDW Package.

The SequenceType class represents a data type that allows specifying all the levels of security that can be used by the elements of the model (ordered from minor to the most restrictive). Level is an ordered enumeration composed of all security levels that have been considered (unclassified, confidential, secret and top Secret). Compartment is the enumeration composed of all user compartments that have been considered. Privilege will be an ordered enumeration composed of all different privileges that have been considered (read, insert, delete, update, all). Attempt will be an ordered enumeration composed of all different access attempts that have been considered (all, frustratedAttempt, successfullAccess, none). Levels will be an interval of levels composed of a lower level and an upper level. If the upper and lower security levels coincide, all instances will have the same security level; else, the specific level will be defined according a securityConstraint. OCLExpression specifies an Object Constraint Language (OCL) expression that fulfils some condition for the users of the system. Role will represent the hierarchy of user roles that can be defined. SetRoleType specifies a set of users, each role is the root a subtree of the hierarchy of user roles considered. SetCompartmentType represent a set of compartments. SetPrivilegeType specifies the privileges the user can receive or remove. SetOCLType specifies the tables involved in a query performed by the user, in order to establish new requirement for tables or columns by means of securityConstraint (ARConstraint or AURConstraint). SetLogInfo specifies the elements that we want to register for a future audit, usually refers to subject requesting the access (subjectID), tables or columns to be accessed (objectID), the operation requested (action), the time request (time) and the access control response (response).

5.3 New Secure Classes and Main Association

The SECRDW package define a container SCatalog and SSchema that are inherited from Schema and Catalog respectively. SCatalog is a local repository of meta data describing all databases maintained by the relational database engine. SSchema is a collection of STables and securityProperties and aim to security at the model level. A ColumnSet represents any form of relational data. A STable and userProfile are inherited from Table, which contains Columns. Be observed in Fig. 3 that the table userProfile contains columns to specify the access properties (securityProperty) that has the user. UserProfile unlike STable is only and does not have association with the rest of the tables of the system. A ForeignKey associates columns from one table with columns of another table. PrimaryKey class inherits from the UniqueConstraint. PrimaryKey and ForeignKey metaclasses are ownered by STable metaclass (see Fig. 3).

To represent security and audit measures in the new metamodel, we add some metaclasses. SecurityProperty metaclass inherits from the Class (from Core) metaclass and specializes as SecurityLevels, SecurityCompartments and SecurityRoles metaclasses. Furthermore, representing security constraints, authorization rules and audit rules in the metamodel we add AuditConstraint class, ARConstraint class and AURConstraint class, which inherit from SecurityConstraint. To specify constraints depending on particular information of a user or a group of users, we introduce the userProfile metaclass. Observe in Fig. 3 the new classes that we have added to the relational package relational from CWM, as well as the new associations between classes. The new classes contain attributes of each one of the types specified in Fig. 2, these attributes allow to represent all the security information captured during the conceptual modeling of the DWs design. Especially, the attribute objectCond refers to an additional condition imposed to

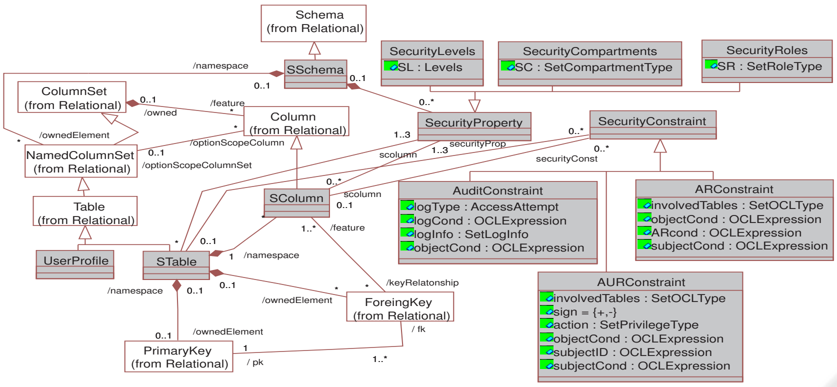


Fig. 3. New classes and associations.

the STable or SColumn object. The attribute subjectCond, allows to specify a condition for the users of the system.

In the following section, we are going to show how we do use the extension in the representation at the logical level of a secure MD model related to the management of the pharmacies consortium businesses.

6 A Case Study

In this section, we apply our extension of the CWM relational metamodel in the context of a pharmaceutical consortium. The consortium managers several pharmacies that offer different services types to the community and wishes to control everything relating to the sales of medicines by means of the prescription medical. To define a classification of data and users that is typical for this type of business (the most general is Pharmacy Employee, which is then specialized into the Pharmacist and nonPharmacist roles, and which are in turn specialized into the assistant and technicians roles in the former case, and into maintenance and administrative in the latter). As security levels, we have considered in this case confidential, secret and topSecret. Inside the company exists a pharmacovigilance group, that guards by the security use of certain medicaments and a committee that guards by the health of his clients, for it we have defined four security compartments: pharmacovigilanceCenter, generalCenter, healthOversightCenter and comercialManagerCenter.

6.1 Defining the PIM

In Fig. 4 we show an instance of the Secure Multidimensional Model, i.e., our SMD PIM, which illustrates a part of the DWs that is required to the previous problem. The SFact Sales_Prescription (stereotype SFact) contain all the sales information in one or more pharmacies, and can be acceded by users who have security levels secret or topSecret, play an Administrative or Pharmacist role and belong to pharmacovigilanceCenter, healthOversightCenter and comercialManagerCenter compartments. The sales attribute can be only accessed by users who perform the administrative role (tagged values SR of sales attribute) and belong to comercialManagerCenter compartment, and therefore the access to this attribute will be forbidden for others users (pharmacist and maintenance employees or belong to other different comercialManagerCenter compartment). The income attributes can be only accessed by users who perform the administrative role (tagged value SR of income attribute). Others static user classification for the classes of the conceptual model defined in Fig. 4 are: The SFact Sales_Prescription contain four dimensions (Pharmacy, Patient, Medication and Time), which contain SBase hierarchies. The access to these SBase hierarchies is established in the same way that was done with the SFact. UserProfile class contains the information of all users who will have access to this secure MD model. Each user has securityLevels (SL), securityRoles (SR) and securityCompartments (SC) associated.

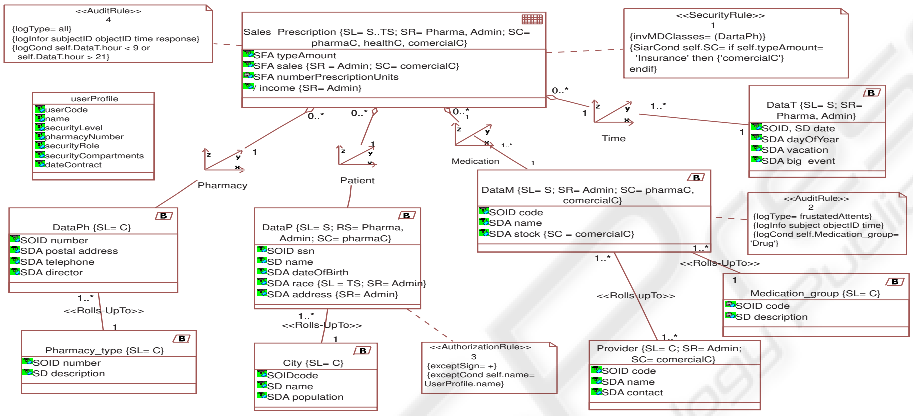


Fig. 4. Example of MD model with security information and constraint.

Several security constraints have been specified by using the previously defined constraints, stereotypes and tagged values.The following paragraphs correspond to notes 1 and 2 in Fig. 4:

1. For each instance of the fact class Sales_Prescription, if the type of payment is through an insurance the security compartment will be comercialManagerCenter (tagged value SC). This constraint is only applied if the user makes a query whose information comes from the DataPh.
2. We wish to record the subject, object and time for every frustrated access attempt to DataM (Data Medication) of the drug description.

6.2 Defining the PSM

Starting from the PIM in Fig. 4, we apply QVT relations [15] to achieve an instance of the SECRDW metamodel, i.e., our secure PSM, just as we show in Fig. 5, which represent a snowflake schema at the logical level. The PSM instance show in Fig. 5 correspond to an instance of the metamodel extended in subsection 5.3. With the extension of CWM we formalized the concepts for a relational platform, although they are closest to the MD despite when the used logical paradigm was not so similar to the relational model, then the transformation is very interesting from the MD to the relational model with respect to security.

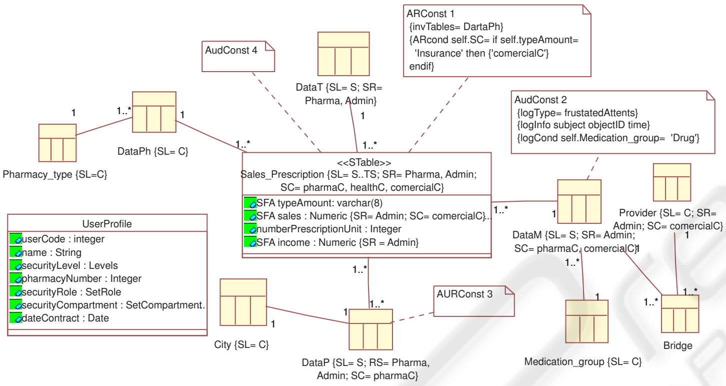
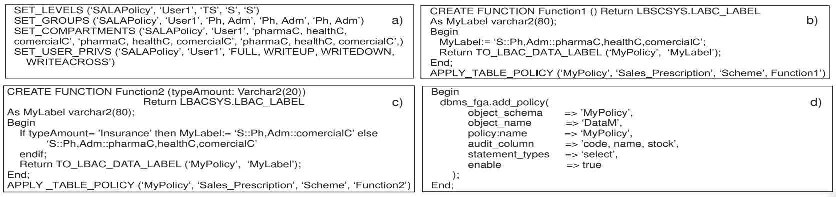


Fig. 5. A snowflake representing an instance of SECRDW metamodel at the logical level.

The fact Sales_Prescription is represented in Fig. 5 by means of the STable Sales_Prescription. In this table we represent all its columns, as well as all the information of associated security, that restrict the access to the own table and its columns. Each SBase is transformed into a STable, nevertheless, the class UserProfile is transformed into the UserProfile table. To represent the relation many-to-many between the tables DataM and Provider we have created a bridge table. The security information (SL, SR and SC) represented in the table Sales_Prescription of Fig. 5 constitute instances of the SecurityProperty class that appears in Fig. 3. This security information is modeled at the logical level in the headline of the own table (see Fig. 5). The security constraints SecurityRule 1, AuditRule 2, AuthorizationRule 3 and AuditRule 4 that appear in Fig. 4 are transformed into instances of the SecurityConstraint class that appears in Fig. 3. These instances are represented in Fig. 5 by means of UML's notes with the names ARConst 1, AudConst 2, AURConst 3 and AudConst 4, nevertheless, to make the Fig. 5 more understandable, we only have showed the attributes of the ARConst 1 and AudConst 2 classes, which constitute instances of classes that represent new data types introduced in Fig. 2.

6.3 Code Example in Oracle DBMS

To finish the case study we show some implementations of the security aspects modeled in the SECRDW metamodel that appears in Fig. 5. We have chosen version 10 of Oracle DBMS since it supports security and audit facilities by means of some of its components namely Oracle Label Security (OLS10g), Virtual Private Databases (VPD) and Oracle Fine-Grained Auditing (FGA). We are going to explain only the security aspects that contemplate our extension, for it first we created a security policy named ’MyPolicy’ and valid levels, compartments and hierarchical groups.



```
SET_LEVELS (‘SALAPolicy’, ‘User1’, ‘TS’, ‘S’, ‘S’)
SET_GROUPS (‘SALAPolicy’, ‘User1’, ‘Ph, Adm’, ‘Ph, Adm’, ‘Ph, Adm’)        a)
SET_COMPARTMENTS (‘SALAPolicy’, ‘User1’, ‘pharmaC, healthC,
comercialC’, ‘pharmaC, healthC, comercialC’, ‘pharmaC, healthC, comercialC’,)
SET_USER_PRIVS (‘SALAPolicy’, ‘User1’, ‘FULL, WRITEUP, WRITEDOWN,
   WRITEACROSS’)
```

```
CREATE FUNCTION Function1 () Return LBSCSYS.LABC_LABEL
As MyLabel varchar2(80);                                                    b)
Begin
  MyLabel:= ‘S::Ph,Adm::pharmaC,healthC,comercialC’;
  Return TO_LBAC_DATA_LABEL (‘MyPolicy’,  ‘MyLabel’);
End;
APPLY_TABLE_POLICY (‘MyPolicy’, ‘Sales_Prescription’, ‘Scheme’, Function1’)
```

```
CREATE FUNCTION Function2 (typeAmount: Varchar2(20))
                                 Return LBACSYS.LBAC_LABEL                  c)
As MyLabel varchar2(80);
Begin
   If typeAmount= ’Insurance’ then MyLabel:= ‘S::Ph,Adm::comercialC’ else
          ‘S::Ph,Adm::pharmaC,healthC,comercialC’
   endif;
   Return TO_LBAC_DATA_LABEL (‘MyPolicy’,  ‘MyLabel’);
End;
APPLY _TABLE_POLICY (‘MyPolicy’, ‘Sales_Prescription’, ‘Scheme’, ‘Function2’)
```

```
Begin
  dbms_fga.add_policy(                                                      d)
        object_schema      => ’MyPolicy’,
        object_name        => ‘DataM’,
        policy:name        => ‘MyPolicy’,
        audit_column       => ’code, name, stock’,
        statement_types    => ‘select’,
        enable             => true
    );
End;
```

Fig. 6. Implementing our constraints in Oracle 10g.

In Fig. 6 a) we show how the User1 satisfy the security properties for the table Sales_Prescription. Fig. 6 b) show how we define and establish the security information for the table Sales_Prescription through labeling functions from OLS, although is not possible to consider security at the column level. The ARConst 1 is implemented by means of the labeling function represented in Fig. 6 c). FGA allow us to define and implement the AudConst 2 (see Fig. 6d)). In AudConst 2 we can’t implement the logType and logCond 2 because FGA does not allow us, neither to choose the audit type (logType) nor the condition referring to columns from different tables (logCond).

7 Conclusions and Future Work

In this work, we have presented an extension of the relational package of the CWM to represent at the logical level all the captured security and audit rules during the conceptual modeling stage of DWs. This proposal is aligned with MDA, with it we contemplated security aspects in all design phases of the DWs, from the PIM, with the proposal of modeling conceptual based in UML, and its corresponding representation at the logical level based on this paper. In order to show the validity of our extension we have developed a case study to illustrate how we modeled at the logical level the security and audit requirements represented during the conceptual modeling stage. Our immediate future work consists in implementing an automatic transformation between the PSM and the implementation level and extending the i* proposal for DWs to incorporate security and audit aspects in the requirement analysis phase of DWs.

Acknowledgements

This work has been partially supported by the METASIGN project (TIN2004-00779) from the Spanish Ministry of Education and Science, by the DADASMECA project (GV05/220) from the Regional Government of Valencia, and by the DIMENSIONS (PBC-05-012-1) DADS project (PBC-05-012-2) from the Regional Science and Technology Ministry of Castilla-La Mancha (Spain).

References

1. G. C. M. Amaral and M. L. M. Campos, “AQUAWARE: A Data Quality Support Environment for Data Warehousing”, SBBD’04, Brasília, DF, Brasil (2004)
2. P. Devanbu and S. Stubblebine, “Software Engineering for Security: a Roadmap”, presented at The Future of Software Engineering, Limerick, Ireland (2000)
3. G. Dhillon and J. Backhouse, “Information Systems Security Management in the New Millenium”, Communications of the ACM, vol. 43 (7) (2000)
4. W. Essmayr, E. Weippl, F. Lichtenberger, W. Winiwarter, and O. Mangisengi, “An Authorization Model for DWs and OLAP”, Workshop On Security In Distributed DW, USA (2001)
5. E. Fernández-Medina, J. Trujillo, R. Villarroel, and M. Piattini, “Access Control and Audit Model for the Multidimensional Modeling of DWs”, DSS, vol. 42 (2006) 1270-1289
6. E. Fernández-Medina, J. Trujillo, R. Villarroel, and M. Piattini, “Developing Secure DWs with a UML Extension”, I. Systems, vol. Article In Press, Corrected Proof (2006)
7. N. Katic, G. Quirchmayr, J. Schiefer, M. Stolba, and A. M. Tjoa, “A Prototype Model for Data Warehouse Security Based on Metadata”, DEXA’98, Vienna, Austria (1998)
8. R. Kirkgöze, N. Katic, M. Stolda, and A. M. Tjoa, “A Security Concept for OLAP”, DEXA’97, Toulouse, France (1997)
9. S. Luján-Mora, J. Trujillo, and I. Y. Song., “A UML profile for multidimensional modeling in data warehouses”, Data& Knowledge Engineering (DKE), vol. 59, 725-769 (2006)
10. T. Maier, “A Formal Model of the ETL process for OLAP-Based Web Usage Analysis”, WebKDD’04, Seatle, Washington, USA (2004)
11. J.-N. Mazón, J. Trujillo, M. Serrano, and M. Piattini, “A MDA approach for the development of data warehouses”, Decis. Support Syst. (2007), doi:10.1016/j.dss.2006.12.003
12. F. Melchert, A. Schwinn, C. Herrmann, and R. Winter, “Using Reference Models for Data Warehouse Metadata Management”, AMCI’05, Omaha, NE, USA (2005)
13. OMG, ”Common Warehouse Metamodel Specification 1.1” (2003)
14. OMG, “MDA Guide Version 1.0.1”, J. M. a. J. Mukerji, Ed.: OMG (2003)
15. OMG, ”MOF 2.0 QVT Final Adopted Specification” (2005)
16. T. Priebe and G. Pernul, “A Pragmatic Approach to Conceptual Modeling of OLAP Security”, ER’01, Yokohama, Japan (2001)
17. T. Priebe and G. Pernul, “Towards OLAP Security Design - Survey and Research Issues”, DMDW’00, Sweden (2000)
18. J. Poole, D. Chang, D. Tolbert, and D. Mellor, Common Warehouse Metamodel Developers Guide. Indianapolis, Indiana: Wiley Publishing, Inc (2003)
19. A. Rosenthal and E. Sciore, “View Security as the Basic for Data Warehouse Security”, DMDW’00, Sweden (2000)
20. F. Saltor, M. Oliva, A. Abelló, and J. Samos, “Building Secure DW Schemas from FIS”, in Heterogeneous Information Exchange and Organizational Hubs., Ed.: KA 123-134 (2002)
21. A. S. Santana and A. M. d. C. Moura, “Metadata to Support Transformations and Data & Metadata Lineage in a Warehousing Environment”, DAWAK’04, Zaragoza, Spain (2004)
22. E. Soler, R. Villarroel, J. Trujillo, E. Fernández-Medina, and M. Piattini, “Representing Security and Audit Rules for DW at the Logical Level by using the CWM”,ARES’06, Vienna
23. M. Thess and M. Bolotnicov, “XELOPES Library Documentation Version 1.2.3”, Prudsys AG (2004)
24. R. Villarroel, E. Fernández-Medina, M. Piattini and J. Trujillo, “A UML 2.0/OCL Extension for Designing Secure DWs”, Journal of Research and Practice in IT, vol. 38 (2006)
25. X. Zhao and Z. Huang, “A Formal Framework for Reasoning on Metadata Based on CWM”, ER’06, Tucson, AZ, USA (2006)