Automatic generation of secure multidimensional code for Data Warehouses by using QVT transformations: an MDA approach

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

[1] Dep. of Information Technologies and Systems. Escuela Superior de Informática
Alarcos Research Group Institute of Information Technologies and Systems
University of Castilla-La Mancha
Paseo de la Universidad, 4. 13071. Ciudad Real, Spain
{Carlos.Blanco, Ignacio.GRodriguez, Eduardo.Fdezmedina, Mario.Piattini}@uclm.es

[2] Dep. of Software and Computing Systems. LUCENTIA Research Group
University of Alicante. San Vicente s/n. 03690. Alicante, Spain
jtrujillo@dlsi.ua.es

Abstract. Data Warehouses manage vital information for the decision making process, which may be discovered by unauthorized users if we do not establish security measures in all the stages of the development process. We have proposed an MDA architecture to develop secure Data Warehouses which allows them to be modeled at different abstraction levels (business, conceptual, logical and code level). We take into consideration security constraints in all models and we automatically transform models through QVT rules. This paper presents a set of QVT transformations focused on obtaining structural aspects (of cubes, dimensions or hierarchies) and security measures defined at a conceptual level in the final secure code for a specific OLAP tool.

1 Introduction

Data Warehouses (DWs) manage highly important information which is used by enterprises to make strategic decisions. It is possible for unauthorized users discover this information through On-Line Analytical Processing (OLAP) tools by using queries involving OLAP operations (roll-up, drill down, slice and dice) or inferences if security measures are not defined. Therefore, information security and confidentiality are vital aspects for the survival of organizations [1] which must be defined from the early stages of the development process [2] and finally taken into account in OLAP tools.

On the other hand, Model Driven Architecture (MDA) [3] is a model-orientated approach for software development based on the separation that exists between the specification of the system functionality and its implementation using specific platforms. It supports metamodel definition at different abstraction levels and transformations between them by using several proposals [4]. Since any MDA-based approach uses models as first order citizens it is important to use a suitable mechanism to deal with these models. MDA proposes to use *model transformations* as a mechanism to go from one level of abstraction to another, merge and wave models, look for matchings and so on. It is possible to find many languages for model transformations, such as ATL [5, 6], BOTL [7], KERMETA [8], Sitra [9]. Nonetheless OMG (*Object-Management Group*) proposes a new standard for model transformation based on the MOF standard (Meta-Object Facility) [10]: QVT [11] (*Query/Views/Transformations*). This language is aimed to define model transformation in an intuitive way.

We have applied this MDA approach to the development of secure DWs [12] which will be described in greater detail in the following sections. Our proposal allows us both to define models at different abstraction levels, and the security measures over them, but this proposal does not deal with their implementation in OLAP tools, although this problem has been studied in [13] which analyses SQL Server Analysis Services (SSAS) as a target platform and present an initial approach of a methodology with which to automatically obtain secure code in this tool. We have developed a set of vertical [14] QVT transformations with which to automatically obtain secure code in SSAS from conceptual models defined with our secure multidimensional PIM. This paper extends our previous works and presents this set of transformations which is focused on obtaining: structural issues (as cubes, dimensions, bases, attributes or hierarchies), security configuration into role-based access control (RBAC) policy used by SSAS and security constraints established at a conceptual level over multidimensional elements.

The remainder of the paper is organized as follows: in Section 2 we will describe the architecture which we used to develop secure DWs focusing on source and target models used to propose the set of QVT transformations that will be defined in Section 3. Later, in Section 4 we will present our conclusions and future work.

2 An MDA approach for Secure DWs

Our model driven (MDA) approach to develop secure DWs [12] considers security from the early stages of the development process and allows us to define security constraints at different levels of abstraction using business (CIM), conceptual (PIM), logical (PSM) and code models. Several works have been proposed to consider security constraints in these models: an i* extension at business level [15]; a UML profile developed specifically for the conceptual modeling of DW (called SECDW) [16] with security issues based on an access control and audit model [17] which considers several access control policies: MAC, DAC and RBAC, audit and constraints definition; a security extension of the CWM (Common Warehouse Metamodel) for logical modeling (called SECRDW) [18] using a relational approach (ROLAP). The final implementation in OLAP tools is dealt with in this work by using the previous steps given in [13]. We propose direct transformations from conceptual models (PIM) represented by using the SECDW metamodel to secure code at high level which represent the previous step towards obtaining secure code in SQL Server Analysis Services (SSAS) platform. We do not obtain PSM models because the target platform manages multidimensional code y we can directly transform multidimensional elements from PIM level. The following subsections focus on describing these source and target models in greater detail.

2.1 SMD PIM Metamodel (SECDW)

SECDW [16] is a secure multidimensional metamodel (SMD PIM) defined at the conceptual level by using an extension of a UML profile for DW [19] with an Access Control and Audit (ACA) model [17] which allows us to represent the main security requirements for the conceptual modeling of DWs, the SECDW metamodel. This UML profile has been designed for DWs and includes their main characteristics such as many-to-many relations, degenerated dimensions, multiple classifications or the alternative path of hierarchies. On the other hand, our Access Control and Audit model [17] considers a combination of mandatory (MAC) and role-base access control (RBAC) based on the classification of subjects and objects in the system, and allows us to define security constraints over multidimensional elements: SecureFact, SecureDegenerateFact, SecureDimension, SecureBase, SecureDegenerateDimension, SecureFactAttribute, SecureDescriptor, SecureOID and SecureDimensionAttribute. In addition, our authorization subjects can be classified from three points of view: in security levels (SecurityLevels), user categories (SecurityCompartment) or user roles (SecurityRoles), and we can also use security constraints (SConstraints).

Furthermore, we can define three kinds of rules in our ACA model by using OCL expressions and UML notes associated with the corresponding class: security rules (SIAR), authorization rules (AUR) and audit rules (AU). Sensitive Information Assignment Rules (SIAR) specify multilevel security policies using levels, compartments and roles, and allow us to define sensitivity information for each element in the MD model. Authorization Rules (AUR) permit or deny specified users access to objects. Auditing rules (AR) establish log conditions on certain objects to ensure that authorized users do not misuse their privileges.

2.2 SMD Code Metamodel for SSAS

In this subsection we present our target metamodel at high level code which represents both the structural aspects of DWs and the security measures defined at upper abstraction levels. This multidimensional secure code metamodel correspond with SQL Server Analysis Services as OLAP tool and we have considered the previous research steps given in [13] in which we analysed the security capabilities of SSAS and proposed how to obtain the final code including security measures defined in conceptual models.

In order to represent DWs code we have focused on structural and security issues and we have defined several metamodels: a security configuration metamodel, a cube metamodel and a dimension metamodel. These metamodels are the previous step towards obtaining the final secure multidimensional code in SSAS which can be automatically generated from these code metamodels.

Security configuration is composed of an XML file for each role in which a list of members for that role is represented. The metamodel is shown in Figure 1, and due to the fact that SSAS only uses a role-based access control policy and our ACA model considers levels, compartments and roles to set up security constraints, we will have to create new roles to define this additional information.

Figure 2 presents a cube metamodel which allows us to set up structural aspects such as cubes, measures, or hierarchies and security with cube permissions at cube, dimension or cell levels. Finally, a dimension metamodel is presented in Figure 3. This also defines both structural aspects such as dimensions, bases, attributes and hierarchies, and security constraints with permissions over dimensions or attributes.

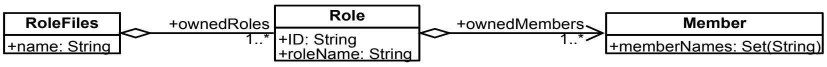


Fig. 1. Security Configuration Metamodel

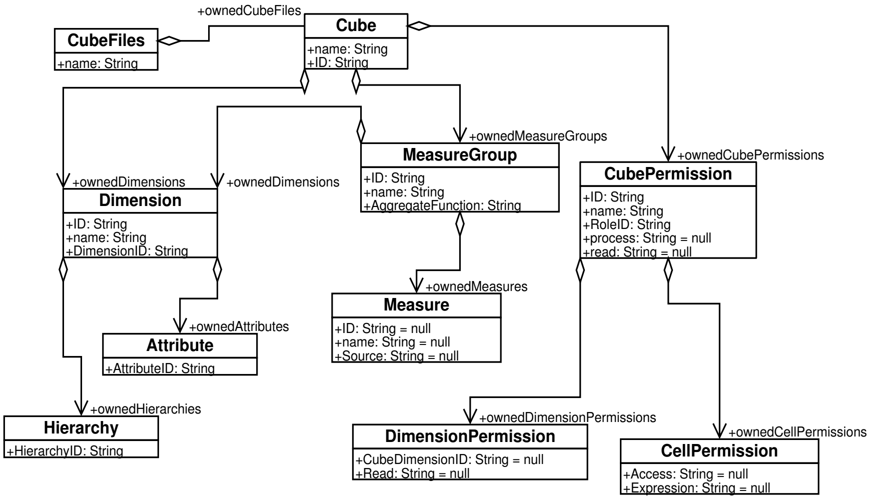


Fig. 2. Cube Metamodel

3 PIM to Code transformations

In this section, the set of proposed QVT transformations to generate the structural aspects and security measures is presented. As it could be seen in previous sections, the information of the code files is represented (in the context of MDA) by means of suitable metamodels (see Section 2.2). Each kind of file is represented by a different metamodel, thus a different QVT transformation has been created for each one. The proposed transformations are: (1) the source metamodel, representing the secure multidimensional metamodel; (2) the three target metamodels, representing the different kinds of code files; and (3) the transformations in charge of taking the information of the source metamodel and transform it into models fulfilling the target metamodels.

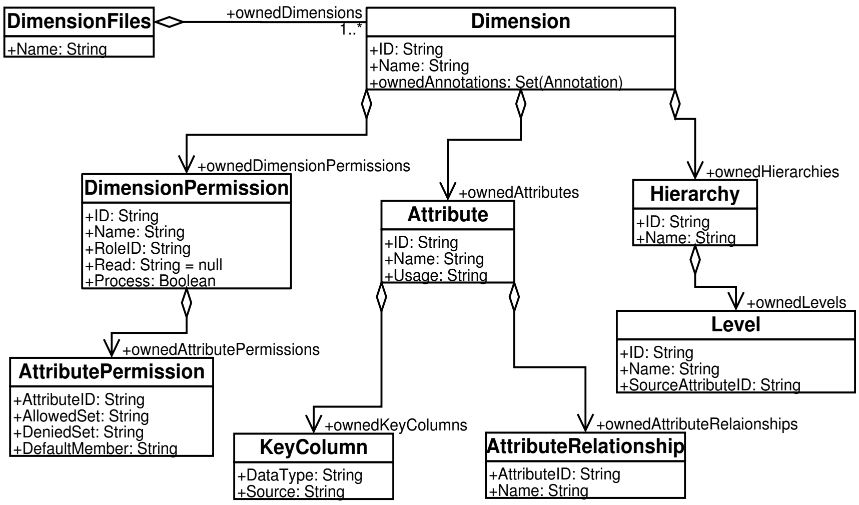


Fig. 3. Dimension Metamodel

In the following subsections the three transformations will be presented. For the SECDW2Cube and SECDW2Dimension transformations the structural and security rules will be shown. Since the transformations are quite verbose and the available space is limited only the "*signatures*" of the QVT rules will be shown, but not the QVT primitives implementing them. The obtention of the models representing the XML code files (that is, the PSM models) is divided into three transformations:

- *SECDW2Role*: Generates the security configuration models, one for each *SRole*, *SLevel* and *SCompartment* in the SECDW model (that is, the PIM model).
- *SECDW2Cube*: Generates the cube files representing Cubes derived from the *SFacts* in the SECDW model.
- *SECDW2Dimension*: Generates the dimension files, representing all the dimensions included in the SECDW model.

It is important to pinpoint that each of the proposed QVT transformations has two main objetives: firstly each transformation identifies the structural issues (which are spread in the SECDW model) related with roles, cubes and dimensions; secondly, each transformation improves/updates the models with security rules also extracted from the SECDW. The latest objective takes advantages from the security stereotypes included in the SECDW model. The extensibility mechanisms of UML2 [20] (to include and manage the stereotypes in models) allow to the proposed transformations the access to this security information, and thus, the inclusion of them in the PSM models.

The elements of the target models (that is, the PSM models) do not exist in the SECDW model (the source model). Thus, both the relations dealing with structural and security aspects reflects the knowledge and the required "know-how" to extract the information to build the *Dimensions* from the SECDW model (that could be understood as an independent platform independent representation of a secure DW). Then, the transformations can be seen as the implementation of an algorithm that specifies, step by step, how the target models must be built. The improvement of this algorithm is easy to carry out, thus, its adaptation to other kind of PSM models could be undertaken without much effort.

3.1 Security configuration transformations

Considering the SECDW (our Metamodel at conceptual level) and the Security Configuration metamodel (Figure 1), the *SECDW2Role* QVT transformation (see Figure 4) has been developed. From this transformation only the signatures of the relations or rules implementing it are shown. This transformation takes the SECDW model and generates a different model for each existing role. This models contains the information required to directly generate the XML code files. The *Package2RoleFiles* relation, which starts with the reserved word ”top”, is the main relation and is the first one to be executed. Other relations are triggered by this one. Since textual transformation could be difficult to understand, the graphical syntax of QVT has been used to show some relations from the transformation of the Figure 4. Figure 5 shows how the *Package* class from the source metamodels is transformated into the *RoleFiles* class.

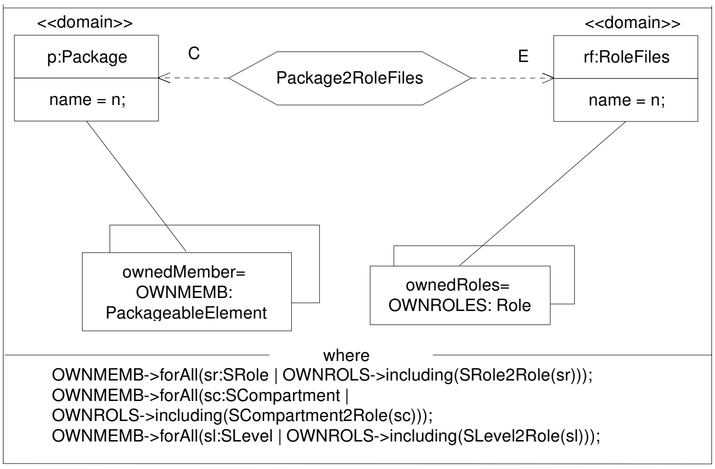
```
transformation SECDW2Role(SECDW pim, Role psm){
   top relation Package2RoleFiles{...}
   relation SRole2Role{...}
   relation SCompartment2Role{...}
   relation SLevel2Role{...}}
```

Fig. 4. SECDW2Role summarized transformation

3.2 Structural and security transformations of Cubes

Figure 6 shows the QVT transformation in charge of transforming the *SFact* classes from the source model into *Cube* classes of the target model (see Figure 3). As it was stated, this transformation has a double purpose: (1) it firstly execute a set of rules to generate the structure o the *Cube* from the SECDW model, and (2) it secondly improves the structural representation of the target model with all the security issues included into the SECDW model.

The set of relations in charge of dealing with the structural issues are the following: *SFact2Cube*, *SDimension2Dimension*, *Property2Attribute*, *ProcessSBase*, *CreateOwnedHierarchies*, *CreateMessageGroups*, *Property2Measure*. Using the elements from the source model these relations (or rules) enforce the creation of other elements, such as the *Cube* concept from the *SFact* (by means of the *SFact2Cube* relation). On the other hand, the relations in charge of dealing with the structural issues are: *SFact2Cube*, *SCompartment2CubePermission*, *SRole2CubePermission*, *SLevel2CubePermission*. For example, the relation *SRole2CubePermission* creates a *CubePermission* in the target model for each Security Role in the source model.



```
OWNMEMB->forAll(sr:SRole | OWNROLS->including(SRole2Role(sr)));
OWNMEMB->forAll(sc:SCompartment |
OWNROLS->including(SCompartment2Role(sc)));
OWNMEMB->forAll(sl:SLevel | OWNROLS->including(SLevel2Role(sl)));
```

Fig. 5. Package2RoleFiles relation

```
transformation SECDW2Cube (SECDW psm, Cube pim){
   top relation Package2CubeFiles{...}
   \\Structural rules
   relation SFact2Cube{...}
   relation SDimension2Dimension{...}
   relation Property2Attribute{...}
   relation ProcessSBase{...}
   relation CreateOwnedHierarchies{...}
   relation CreateMessageGroups{...}
   relation Property2Measure{...}
   \\Security rules
   relation SCompartment2CubePermission{...}
   relation SRole2CubePermission{...}
   relation SLevel2CubePermission{...}}
```

Fig. 6. SECDW2Cube summarized transformation

Figure 7 shows the graphical representation of the *SRole2CubePermission* relation. As it could be seen, the input element (or domain) is the *SRole*. From this input *SRole*, two elements are created/updated in the target model: firstly a *CubePermission* (with the information of the SRole) is created; secondly the *Cube* class (also in the target model) is updated, with the inclusion of the new CubePermission, in its collection of *CubePermissions*. That is to say, the cube is annotated with a new permissions corresponding to the roles existing in the source model.

3.3 Structural and security transformations of Dimensions and Bases

Finally, this subsection presents the *SECDW2Dimension* QVT transformation which includes the structural and security issues related with the creation of the *Dimension* models from the *SECDW* model (Figure 8). Starting from the *SECDW* model this transformation generates a set of models, each of them represents a *Dimension* code file. Obviously the model represents the elements related with the Dimension files, but not the

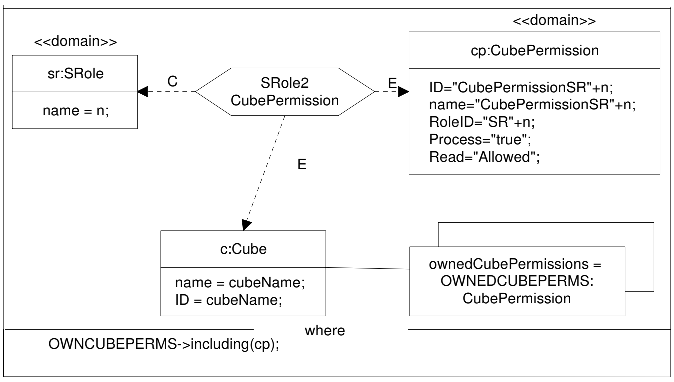


Fig. 7. SRole2CubePermission relation

code. Nonetheless, the code generation is a trivial task that can be carried out with an MDA tool or code generator. The QVT relations regarding the structural aspects are the following: *Package2DimensionFiles*, *SDimension2Dimension*, *KeyProperty2Attribute*, *NonKeyProperty*, *SBase2Attribute*.

```
transformation SECDW2Dimension(SECDW psm, Dimensions pim){
\\Structural rules
 top relation Package2DimensionFiles{...}
 relation SDimension2Dimension{}
 relation KeyProperty2Attribute{}
 relation NonKeyProperty{}
 relation SBase2Attribute{}
\\Security rules
 relation processSecureProperty{...}
 relation createNegativeAttributePermissions{...}
 relation createPositiveAttributePermisions{...}
 relation createDimensionSIARForSLevel{...}
 relation createDimensionSIARForSRole{...}
 relation createDimensionSIARForSCompartment{...}
 relation authorizeSLevel{...}
 relation authorizeSRole{...}
 relation authorizeSCompartment{...}}
```

Fig. 8. SECDW2Dimension summarized transformation

This relations creates in the *Dimension* model elements such as *Dimension Permissions*, *Attribute Permissions*, *Hierarchies*, and so on. On the other hand, the relations in charge of generating/updating the target model with the security aspects are the following: *processSecureProperty*, *createNegativeAttributePermissions*, *createPositiveAttributePermisions*, *createDimensionSIARForSLevel*, *createDimensionSIARForSRole*, *createDimensionSIARForSCompartment*, *authorizeSLevel*, *authorizeSRole*. The relation of Figure 9 (*createNegativeAttributePermissions*), expressed in the graphical syntax, process a given *SecurityProperty* (belonging to a *SDimension* in the source model) and produces a negative access rule.

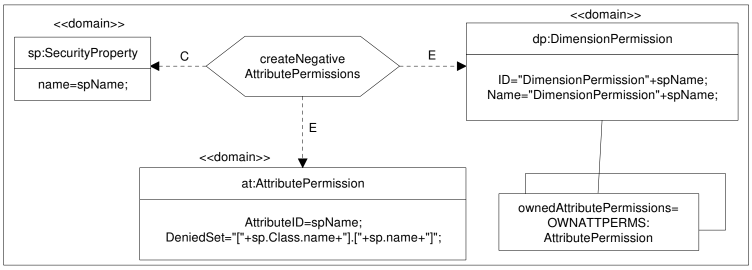

<<domain>>
sp:SecurityProperty
name=spName;
C
createNegative AttributePermissions
E
<<domain>>
dp:DimensionPermission
ID="DimensionPermission"+spName;
Name="DimensionPermission"+spName;
E
<<domain>>
at:AttributePermission
AttributeID=spName;
DeniedSet="["+sp.Class.name+"].["+sp.name+"]";
ownedAttributePermissions=
OWNATTPERMS:
AttributePermission

Fig. 9. createNegativeAttributePermissions relation

4 Conclusions

We have previously commented on the benefits of using an MDA approach to develop secure DWs and the necessity of finally translating both structural and security issues defined at upper abstraction levels into OLAP code. Therefore, in this work, we have dealt with this problem by using SSAS as a target platform which manages multidimensional elements such as cubes, dimensions or bases and allows us to set up security measures over them. However, our access control and audit model is richer than SSAS which only uses RBAC as an access control policy. This is a problem that we have solved by including new roles to represent this extra information of security roles, compartments and levels defined in the conceptual model in our security configuration in SSAS. In this work, we have presented a secure multidimensional code metamodel for SSAS at high level and we have also defined the necessary QVT transformations in order to obtain this code. We can obtain complete code for SSAS from our secure multidimensional PIM (SECDW). These transformations are composed of several sets of rules: firstly we have obtained a security configuration of the system by translating our security roles, levels and compartments defined at a conceptual level into an RBAC policy; we have then defined a set of rules for the structural aspects of the DW such as cubes, measures, dimensions, bases, attributes and hierarchies; and finally, we have translated security constraints established over multidimensional elements of the conceptual models.

In future works we will extend this approach by analysing advanced security measures defined with OCL expressions in conceptual models, and we will present a metamodel and transformations with which to obtain their code automatically. Furthermore, we will complete our MDA arquitecture by adding transformations from secure multidimensional PIM to others OLAP platforms such as Pentaho and Oracle. We must also improve our metamodels at upper abstraction levels and our access control and audit model to include new security constraints for detected security problems which are directly related to OLAP operations, such as navigations with roll-up or drill-down.

References

1. Dhillon, G., Backhouse, J.: Information system security management in the new millennium. Communications of the ACM **43**(7) (2000) 125–128
2. Mouratidis, H., Giorgini, P.: An introduction. In: Integrating Security and Software Engineering: Advances and Future Visions. Idea Group Publishing (2006)
3. MDA, O.M.G.: Model Driven Architecture guide. (2003)
4. Czarnecki, K., Helsen, S.: Classification of model transformation approaches. (2003)
5. Jouault, F., Kurtev, I.: Transforming models with ATL. In: International Workshop on Model Transformations in Practice (MTiP 2005). (2005)
6. Bézivin, J., Jouault, F., Valduriez, P.: An eclipse-based IDE for the ATL model transformation language. Technical Report n 04.08, University of Nantes (2005)
7. Braun, P., Marschall, F.: The bidirectional object oriented transformation language. Technical Report TUM-INFO-05-I0307-0/1.-FI, Institut für Informatik der Technischen Universität München (May 2003 2003)
8. Falleri, J., Huchard, M., Nebut, C.: Towards a traceability framework for model transformations in KERMETA. In: European Conference on Model-Driven Architecture Traceability Workshop (ECMDA-TW 2006), Bilbao, Spain (2006) 31–40
9. Akehurst, D.H., Bordbar, B., Evans, M.J., Howells, W.G.J., McDonald-Maier, K.D.: SITRA: Simple transformations in Java. In: 9th International Conference on Model Driven Engineering Languages and Systems. Volume LNCS 4199., Genova, Italy, Springer (2006) 351–364
10. OMG: Meta Object Facility (MOF) specification (2002)
11. OMG: QVT final adopted specification (2005)
12. Fernández-Medina, E., Trujillo, J., Piattini, M.: Model Driven multidimensional modeling of secure data warehouses. European Journal of Information Systems **16** (2007) 374–389
13. Blanco, C., Fernández-Medina, E., Trujillo, J., Piattini, M.: Implementing multidimensional security into OLAP tools. In: Third International Workshop ”Dependability Aspects on Data WArehousing and Mining applications” (DAWAM 2008), Barcelona, Spain, IEEE Computer Society (2008) 1248–1253
14. Mens, T., Van Gorp, P.: A taxonomy of model transformations. Electronic Notes in Theoretical Computer Science **152** (2006) 125–142
15. Soler, E., Stefanov, V., Mazón, J.N., Trujillo, J., Fernández-Medina, E., Piattini, M.: Towards comprehensive requirement analysis for data warehouses: Considering security requirements. In: Proccedings of The Third International Conference on Availability, Reliability and Security (ARES), Barcelona, Spain, IEEE Computer Society (2008) 104–111
16. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Developing secure data warehouses with a UML extension. Information Systems **32**(6) (2007) 826–856
17. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Access Control and Audit model for the multidimensional modeling of data warehouses. Decision Support Systems **42**(3) (2006) 1270–1289
18. Soler, E., Trujillo, J., Fernández-Medina, E., Piattini, M.: SECRDW: An extension of the relational package from CWM for representing secure data warehouses at the logical level. In: International Workshop on Security in Information Systems, Funchal, Madeira, Portugal (2007)
19. Lujan-Mora, S., Trujillo, J., Song, I.Y.: A UML profile for multidimensional modeling in data warehouses. Data & Knowledge Engineering **59**(3) (2006) 725–769
20. OMG: Unified Modeling Language: Superstructure. versin 2.0 (2005)