European Journal of Information Systems (2007) 16, 374–389
© 2007 Operational Research Society Ltd. All rights reserved 0960-085X/07 $30.00
www.palgrave-journals.com/ejis

Model-driven multidimensional modeling of secure data warehouses

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

[1]*Dept. Tecnologías y Sistemas de Información, Univ. Castilla-La Mancha, Ciudad Real, Spain;* [2]*Dept. Lenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante, Spain*

Correspondence: Eduardo Fernández-Medina, Dept. Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Paseo de la Universidad, 4, Ciudad Real 13071, Spain.
Tel: +34 926 295300 ext. 3744;
Fax: +34 926 295354;
E-mail: eduardo.fdezmedina@uclm.es

Received: 27 October 2006
Revised: 31 May 2007
Accepted: 27 July 2007

Abstract

Data Warehouses (DW), Multidimensional (MD) databases, and On-Line Analytical Processing (OLAP) applications provide companies with many years of historical information for the decision-making process. Owing to the relevant information managed by these systems, they should provide strong security and confidentiality measures from the early stages of a DW project in the MD modeling and enforce them. In the last years, there have been some proposals to accomplish the MD modeling at the conceptual level. Nevertheless, none of them considers security measures as an important element in their models, and therefore, they do not allow us to specify confidentiality constraints to be enforced by the applications that will use these MD models. In this paper, we present an Access Control and Audit (ACA) model for the conceptual MD modeling. Then, we extend the Unified Modeling Language (UML) with this ACA model, representing the security information (gathered in the ACA model) in the conceptual MD modeling, thereby allowing us to obtain secure MD models. Moreover, we use the OSCL (Object Security Constraint Language) to specify our ACA model constraints, avoiding in this way an arbitrary use of them. Furthermore, we align our approach with the Model-Driven Architecture, the Model-Driven Security and the Model-Driven Data Warehouse, offering a proposal highly compatible with the more recent technologies.
European Journal of Information Systems (2007) **16,** 374–389.
doi:10.1057/palgrave.ejis.3000687

Keywords: MDA; MDDW; MDS; data warehouses; secure multidimensional modeling; access control; audit; UML

Introduction

W. Inmon coined the term Data Warehouse (DW) in the early 1990s as: 'A subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decisions' (Inmon, 2002). A DW is populated from data gathered from (heterogeneous) data sources (legacy systems, relational databases, etc.) and the Extraction–Transformation–Loading (ETL) processes are responsible for the extraction of data from heterogeneous data sources, their transformation (conversion, cleaning, normalization, etc.) and their loading into DWs. These DWs are normally queried by using a variety of front-end tools such as the classical reporting tools, OLAP (On-Line Analytical Processing) tools or even data mining tools. OLAP tools are probably the most extended ones between managers due to their power in providing summarized data and the facility in using them by easy point-and-click operations (Thomsen, 1997).

It is widely accepted that these systems are based on the multidimensional (MD) modeling. The benefit of using this MD modeling is two-fold. On the one hand, the MD model is close to the way of thinking of data analyzers and, therefore, helps users understand data. On the other

hand, the MD model supports performance improvement as its simple structure allows us to predict final users intentions. MD modeling structures information into facts and dimensions. A fact represents interesting measures of a business process (sales, deliveries, etc.), whereas a dimension considers the context for analyzing a fact (product, customer, time, etc.).

In most cases, MD models also store information regarding private or personal aspects of individuals, like identification data, medical data or even religious beliefs, ideologies, or sexual tendencies. Many governments are very concerned about privacy, and they promulgate laws to protect individual privacy, such as the European Union Directive 95/46/CE of the European Parliament and Council on people protection regarding personal data management and free circulation of data (and its national adaptation, as for instance LOPD in Spain and BDSG in Germany), the European Union' Safe Harbour Law, the Unites States' HIPPA (Health Insurance Portability and Accountability Act), Gramm–Leach–Bliley Act, Sarbanes–Oxley Act, etc. According to these laws, organizations must provide exhaustive access control and complete audit trails for all accesses to personal data, and failure in complying with these laws tends to be very strict, imposing severe penalties.

Therefore, considering that the very survival of the organizations frequently depends on the correct management, security and confidentiality of information (Dhillon & Backhouse, 2000), and the extreme importance of the information that users can discover by using these kinds of applications, it is crucial to specify confidentiality measures in the MD modeling process, and enforce them. Indeed, as some authors remarked (Devanbu & Stubblebine, 2000; Ferrari & Thuraisingham, 2000; Toval *et al.*, 2002), information security is a serious requirement which must be carefully considered, not as an isolated aspect, but as an element presented in all stages of the development lifecycle, from the requirement analysis to implementation and maintenance.

We can consider three technologies that have been widely used to protect information against improper disclosure or modifications. Authentication, access control and audit altogether provide the foundation for information security (Sandhu & Samarati, 1997). Authentication establishes the identity of one party to another. Access control determines what one party will allow another one to do with respect to resources and objects mediated by the former. Access control usually requires authentication as a prerequisite. The Audit process gathers data about activities in the system and analyzes it to discover security violations or diagnose their cause. Authentication is a mechanism that is design-independent and relies more on the general security company policies that define its type (password-based authentication, token-based authentication, or biometric authentication) and the necessary technology. Therefore, authentication does not need to be considered during the conceptual design of DWs, and it is out of the scope of this paper. Nevertheless, access control and audit have an important component of design. In fact, access control and audit considerations should be taken into account throughout the design process from its early stages, and not just when the system is completely developed (Hall & Chapman, 2002). Therefore, we claim that these aspects should be included in the conceptual modeling of a system as the sooner we deal with security aspects, the more willing we are to consider all main security aspects in the final system implementation.

In this paper, we define an Access Control and Audit (ACA) model that allows us to specify access control and audit rules when accomplishing the conceptual MD modeling of DWs. Since there are some proposals of MD models and modeling processes for DWs, the ACA model should be independent, but easily adaptable to all these proposals. We have chosen a proposal, based on UML (Trujillo *et al.*, 2001; Luján-Mora *et al.*, 2006), that easily allows us to model all main MD properties at the conceptual level. Then, we have defined a UML extension that allows us to specify all concepts that are previously defined in the ACA model following the previously-presented UML approach. As we will describe in the next section, our complete approach for designing secure DWs follows the Model-Driven Architecture (MDA) (Kleppe *et al.*, 2003; OMG, 2004b), the Model-Driven Security (MDS) (Basin *et al.*, 2003), and the Model-Driven Data Warehouse (MDDW) (Poole, 2003). To the best of our knowledge, this is the first formal approach for the conceptual design of DWs in considering security and audit aspects as part of the conceptual MD model.

The remainder of this paper is structured as follows: The next section presents the MDA, MDS and MDDW compliant architecture of our approach. The subsequent section briefly summarizes the conceptual approach for MD modeling on which we base our work on. The fourth section presents our ACA model. Further section briefly introduces the UML extension for the conceptual modeling of secure DWs we use. The sixth section presents a case study and applies our ACA model and UML extension for secure MD modeling. The penultimate section discusses related work. The last section presents the main conclusions and sketches our immediate future work. Owing to the number of acronyms that appears in this paper as well as all the abbreviations we define in our ACA model, we have collected all of them at the end of the paper in order to facilitate its rapid meaning location throughout the whole paper.

An MDA, MDS and MDDW compliant approach

MDA (OMG, 2004b) is an Object Management Group (OMG) standard that addresses the complete lifecycle of designing, deploying, integrating, and managing applications. MDA separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. Thus, MDA encourages specifying a Platform Independent Model (PIM) by using any specification

language – typically the Unified Modeling Language (UML). Then, this PIM can be transformed into multiple Platform Specific Models (PSM) in order to be executed on a concrete platform by transforming this PSM into the corresponding Code. In Figure 1, we can see the general overview of the MDA architecture, in which one PIM can be transformed into different PSMs, and one PSM can be transformed into different Codes according to the concrete platform where the system is to be implemented. Alternatively, horizontal transformations (normally called bridges) can also be defined in order to transform models at the same level.

We have aligned our approach for the conceptual modeling of DWs with the MDA approach. Thus, as we will show throughout the paper (see Figure 2), the conceptual modeling of the DW itself is accomplished by using a UML profile without considering any implementation aspect on a concrete target platform. Then, the resulting PIM can be transformed into any logical model representing the multidimensionality of data, and finally this logical model can be transformed into a particular DBMS. This multidimensionality of data is normally represented at the logical level in relational platforms by using the well-known star schema (and its variants snowflakes and fact constellations). The star schema represents facts and dimensions into fact tables and dimension tables, respectively. In order to summarize what a star schema is, let us say that the primary key of a fact table is composed by the foreign keys to all the dimension tables to which the fact table is related. It is called a star schema as it is usually drawn as a *star*, placing the fact table in the center and each dimension table being a vertex of the star (see Figure 3). In this paper, we have focused on the definition of models at PIM level of the Figure 2. Details about the definition of the PSM of our MDA architecture without considering security issues can be found in Mazón *et al.* (2007), adding security issues in Soler *et al.* (2007a), and the QVT relations to transform PIM to PSM in the design of secure DWs can also be found in Soler *et al.* (2007b).

On the other hand, MDS is a new approach (Basin *et al.*, 2003) for integrating security into the information systems design. This approach considers design models and security models, which are combined, leading to a new kind of models that is called security design model.

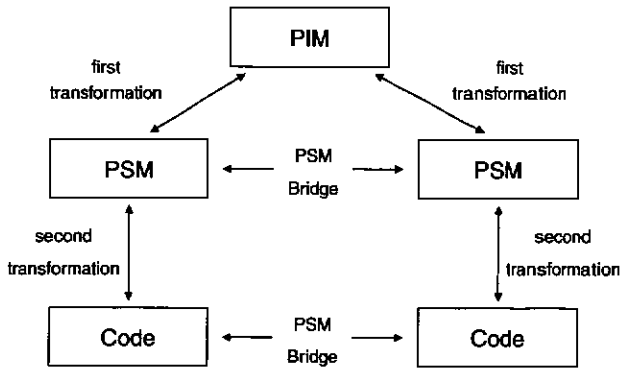

Figure 1 General overview of the Model-Driven Architecture (Kleppe *et al.*, 2003).

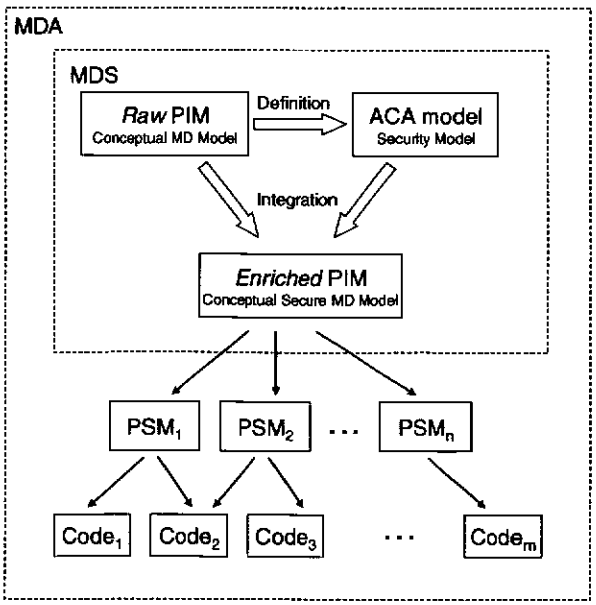

Figure 2 Merging the UML profile and the ACA Model into one PIM.

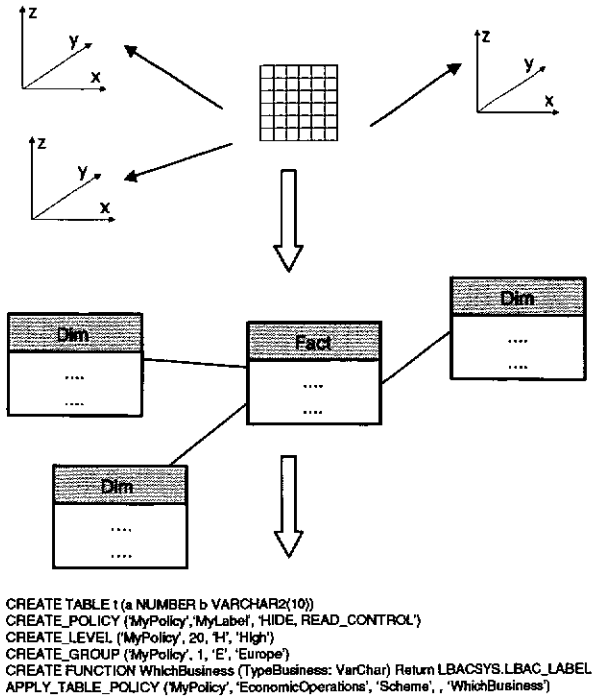

Figure 3 Transformation of a Secure MD schema into a Relational schema (based on the star schema) and then into OLS code.

Our approach has also been aligned with the general idea of MDS (see Figure 2). We have considered a *Raw* PIM for the conceptual modeling of DWs, which is the design model. This model does not contain security details, and as previously commented, we consider an extension of UML as notation for the conceptual modeling of DWs. Moreover, we have defined our ACA model, which is completely independent from the design model (and of course it is independent from the target platform, so in some sense is another PIM) and, in MDS terminology, is the security model. This independence is highly relevant, because we could use this security model together with other DW conceptual models. Combining both the design model and the security model, an *Enriched* PIM is performed. This security design model is a DW conceptual model that also contains the security details that can be specified with our ACA model. The design model must usually be extended to be able to represent the security details. In this case, we have formally defined an extension of UML for designing secure DWs (Fernández-Medina *et al.*, 2004), summarized in the fifth section. Therefore, this *enriched* PIM with all the security information is the model that will participate in the MDA architecture in the upcoming transformations.

A model transformation is the process of converting one model to another one. In France & Bieman (2001), model transformations are categorized along vertical (a source model is transformed into a target model at a different level of abstraction) and horizontal (a source model is transformed into a target model that is at the same level of abstraction) dimensions. MDA also provides different kinds of mappings to transform one model into another such as the type or instance mapping, marking models or even the metamodel transformations (Frankel, 2003).[1]

We have developed an algorithm that, from the MD schemas accomplished by using our UML profile and the ACA model altogether, generates through a vertical transformation, a relational PSM based on the star schema – relational is the most common representation for logical MD models (Kimball & Ross, 2002) – and then, the code in Oracle Label Security (OLS) (Levinger, 2002) is automatically generated. Nevertheless, our architecture would allow us to transform our *enriched* PIM into any other logical model,[2] and any PSM into the code of one concrete platform that is able to implement secure databases or DWs, such as DB2, Microsoft SQL Server 2000, or for certain OLAP tools such as Microstrategy, Cognos Powerplay or Oracle Discoverer. In Figure 3, we show a high-level view of the transformation process from a secure MD model to the relational model by using the star schema, and from this one into the code of OLS platform structures according to the modeling elements.

[1]It is out of the scope of this paper to provide further detail on all these transformations.

[2]See Abelló *et al.* (2001) for a summary of the most relevant logical models proposed for MD modeling.

Model-Driven Data Warehouse (MDDW)

The latest approach of the OMG for aligning the design of DWs with the general MDA paradigm is called MDDW (Poole, 2003). The MDDW also proposes the specification of a PIM, then transform it into a PSM, and finally into a specific Code in a target platform (see the second section for further detail).

The peculiarity of this MDDW is that instead of using models at every level, it proposes the use of metamodels. These metamodels are based on the standard Common Warehouse Metamodel (CWM) (Poole *et al.*, 2002; OMG, 2004a). The CWM provides a set of metamodels that are comprehensive enough to model an entire DW including data sources, ETL processes, MD modeling, relational implementation of a DW and so on.

However, the CWM (and their metamodels) was conceived for interchanging metadata about DWs between different platforms and tools, thereby providing a standard for the common representation of metadata at any level.

Nevertheless, our approach differs in both the CWM and the MDDW approaches in which we consider CWM metamodels (i) are too generic to represent all peculiarities of MD modeling (Medina & Trujillo, 2002a) and (ii) are too complex to be handled for conceptual modeling by both final users and designers. For these reasons, instead of using metamodels for specifying *PIMs* or *PSMs*, we prefer to use models such as the UML extension for secure DWs and the ACA model presented in this paper. If we would further need to interchange DW metadata between different applications or platforms, we could represent our approach for secure DWs in CWM by extending the corresponding CWM metamodel as both approaches are based on the standard UML. Actually, we have already shown that these transformations are feasible as in Medina & Trujillo (2002a, b) we represented our previous UML approach for conceptual MD model (Trujillo *et al.*, 2001) with no security rules in the CWM metamodel for MD modeling.

Object-oriented MD modeling

In this section, we outline the approach, based on the UML, we use for DW conceptual modeling (Trujillo *et al.*, 2001; Luján-Mora *et al.*, 2006). This approach has been specified by means of a UML profile that contains the necessary stereotypes in order to carry out the MD modeling at the conceptual level successfully (Gogolla & Henderson-Sellers, 2002). The main features of MD modeling considered are the relationships *many-to-many* between facts and one specific dimension, degenerated dimensions, multiple classification and alternative path hierarchies, and non-strict and complete hierarchies. In this approach, structural properties of MD modeling are represented by means of a UML class diagram in which the information is clearly organized into facts (items of interest for an enterprise) and dimensions (context in which facts have to be analyzed).

Facts and dimensions are represented by means of fact classes (stereotype Fact) and dimension classes (stereotype Dimension), respectively. Fact classes are defined as composite classes in shared aggregation relationships of *n* dimension classes. The minimum multiplicity in the role of the dimension classes is 1 (all the facts must always be related to all dimensions). The relations *many-to-many* between a fact and a specific dimension are specified by means of the multiplicity 1..* in the role of the corresponding dimension class.

Our *Admission* example will allow us to find out the profitability of a patient (through the defined measures *income, cost* and *benefit*) by considering the diagnosis that was made, the patient to whom the diagnosis was made, and the date it was made. In Figure 4, we can see how the *Admission* fact class has a many-to-one relationship with all dimensions classes (*diagnosis, patient* and *time*).

A fact is composed of measures or fact attributes. By default, all measures in the fact class are considered to be additive. For non-additive measures, additive rules are defined as constrains and are included in the fact class. Furthermore, derived measures can also be explicitly represented (indicated by /) and their derivation rules are placed between braces near the fact class. See the *benefit* attribute and its corresponding derived rule in the *Admission* fact class in Figure 4.

This approach also allows the definition of identifying attributes in the fact class (stereotype OID). In this way *degenerated dimensions* can be considered (Kimball, 1996), thereby representing other fact features in addition to the measures for analysis. For example, we could store the bill number (*bill_number*) as a degenerate dimension (see *Admission fact* in Figure 4), allowing us to consider another interesting fact feature different from the usual measures.

With respect to dimensions, each level of a classification hierarchy is specified by a base class (stereotype Base). An association of base classes specifies the relationship between two levels of a classification hierarchy. The only prerequisite is that these classes must define a Directed Acyclic Graph (DAG) rooted in the dimension class (DAG constraint is defined in the stereotype Dimension). The

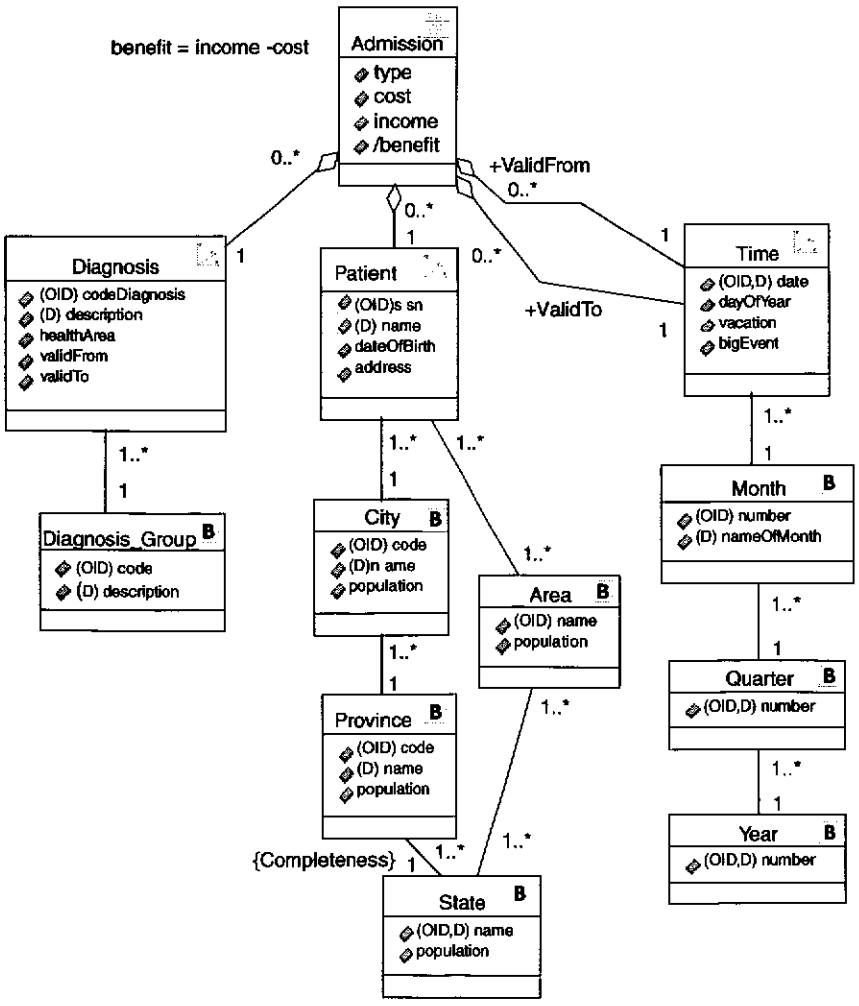

Figure 4 Multidimensional modeling using the UML.

DAG structure can represent both multiple and alternative path hierarchies. Every base class must also contain an identifying attribute (OID) and a descriptor attribute[3] (stereotype D). These attributes are necessary for an automatic generation process into commercial OLAP tools, as these tools store this information on their metadata.

Owing to the flexibility of UML, we can also consider non-strict hierarchies (an object at a hierarchy's lower level belongs to more than one higher-level object) and complete hierarchies (all members belong to one higher-class object and that object consists of those members only). These features are specified by means of the multiplicity of the roles of the associations and defining the constraint {completeness} in the target associated class role respectively. See *Patient* dimension in Figure 4 for an example of all kinds of classification hierarchies. Lastly, the categorization of dimensions is considered by means of the generalization/specialization relationships of UML.

Security considerations

As can be easily seen from our example in Figure 4, we have to deal with sensitive information such as the diagnosis made to a patient or even other serious family illnesses. Therefore, a final DW solution for this example should consider the final users who can access to certain specific information. In real-world DW implementations, this is normally accomplished by limiting the OLAP operations that certain final users can apply on specific data. However, we deeply believe that these security aspects should also be considered together with data in the corresponding conceptual MD model. In this way, we will consider DW security aspects from the early stages of a DW and enforce them in the further implementation.

ACA model

Access control is not a complete solution for securing a system (Sandhu & Samarati, 1997) as it must be coupled with auditing. Auditing requires the recording of all user requests and activities for their later analysis. Therefore, in our approach, we consider both concepts to be integrated in the conceptual MD modeling design.

Although there are many authorization models that allow a flexible and easy specification of authorizations, they rely on the particular properties of the underlying data model (Jajodia *et al.*, 2001). As a result, these authorization models cannot be easily extended to other data models, such as the MD modeling.

Access control models are typically composed of a set of authorization rules that regulate access to objects. Each authorization rule usually specifies the *subject* to which the rule applies, the *object* to which the authorization refers, the *action* to which the rule refers, and the *sign* describing whether the rule states a permission or a denial for the access.

In order to regulate access to objects in a MD model, we have considered a read-only version of the Mandatory Access Control (MAC) model, in a simplified way, Role-Based Access Control (RBAC), and a set of authorization rules, that represent exceptions to the general multilevel rules. We define our ACA model as a composition of sensitivity information assignment rules, where the designer define the security information for all elements of the MD model (facts, dimensions, etc.), a set of authorization rules, where the designer can specify different situations in which the multilevel rules should be reinforced, and finally, a set of audit rules, which represent the audit requirements that the designer considers.

In the next subsections we introduce all details of the ACA model: The access control model that has been considered are authorization subjects, authorization objects, actions, sensitivity information assignment rules, authorization rules, audit rules, and conflict resolution.

Access control model

The access control model that we have considered as a basis for the MD modeling has been the MAC model. MAC has been widely studied (Sandhu & Chen, 1998; Samarati & De Capitani Di Vimercati, 2000), and many vulnerabilities has been detected, such as its lack of flexibility, the polyinstantiation, etc. Nevertheless, most of these troubles are provoked by the necessity of considering both read and write operations into the system. Fortunately, we can initially consider that the sole operation to be used by the final users in decision support systems is *read*, so MAC is absolutely suitable. Moreover, MAC is being integrated in some of the most important DBMS, such as Oracle*9i* Label Security (Levinger, 2002) and DB2 Universal Database (UDB) (Cota, 2004). This is important because MD models could be implemented by some of these DBMSs.

In our model, we have considered three different but compatible ways of classifying users, by their security level, the user role they play and the user compartments they belong to:

- *Security levels*: It indicates the clearance level of the user.
- *Security user roles*: Used by a company to organize users in a hierarchical role structure, according to the responsibilities of each type of work. Each user can play more than one role. In some sense, RBAC (Sandhu *et al.*, 1996, 2000) is also considered in this model in a simplistic way.
- *Security user compartments*: Used by an organization also to classify users into a set of horizontal compartments or groups, such as geographical, working area, etc. Each user can belong to one or more compartments.

Therefore, for each object in the model, the user access requirements (security level, user roles, and user compartments) can be defined, and thereby specifying with high

[3]A descriptor attribute will be used as the default label in the data analysis in OLAP tools.

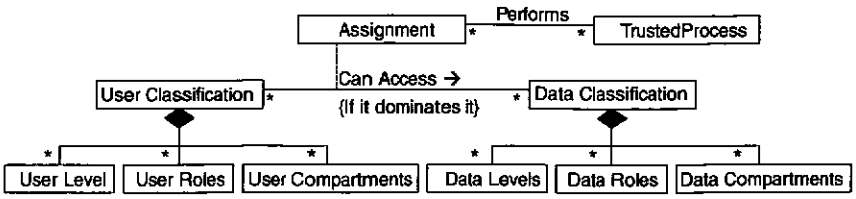

Figure 5 Class model for the mandatory access control.

accuracy which users can access to each object. The MAC rule for read operations depends on a dominance rule. A user can access data only if the user classification dominates the data classification. This happen only if (a) the security level of the user is greater than or equal to the security level of the data, (b) all the user compartments that have been defined for the data are defined for the user, and (c) at least one of the user roles (or a descendent in the user role hierarchy) that the data has defined, is played by the user. A trusted process is in charge of assigning the user classification for each user, and the data classification for each data. Figure 5 shows an extension of the pattern that has been presented in (Fernandez & Pan, 2001), which clearly explains this access control model. The RBAC is also integrated in this combined model, since the user role hierarchy is involved in the dominance rule we have previously presented.

Authorization subjects

The specification of subjects[4] in access control rules has often two apparently contrasting requirements (Samarati & De Capitani Di Vimercati, 2000). On the one side a subject reference must be simple, to allow for efficient access control and for exploiting possible relationships between subjects in resolving conflicts between the authorizations (e.g., most specific relationships between roles and sub-roles). On the other side, one would like to see more expressiveness than the simple reference to user identities or user roles, compartments or security levels, providing support of profile-dependent authorizations whose validity depends on properties associated with users (e.g., age, citizenships, or field-of-specification). Our solution nicely encounters both requirements by supporting both user classification concepts (security levels, roles and compartments) and user profiles. The profile can be modeled as a structured object that represents all the relevant properties of each subject. Therefore, the *subject* component of our ACA model includes two parts:

- An *identity*, that can be composed of one or more of the following sub-attributes:
  - *Id*: The user identifier.
  - *Role-id*: The identifiers of one or more user roles.
  - *Compartment-id*: The identifiers of one or more user compartments.
  - *Level-id*: The name of one security level or the interval of security levels.
- A *subject expression* which is an OCL expression (Warmer & Kleppe, 2003) on users' profiles.

The grammar we consider for the definition of elements in the ACA model uses EBNF (Extended Backus Naur Form) syntax, in which | means a choice, ? means optionally, * means zero or more times, and + means one or more times. Authorization subjects are then defined as illustrated in Table 1(a).

For instance, *subject* element '***RID** administrative **AND** **CID** U.S.A. **AND** **CID** financial services **COND** profile. age>18*'[5] denotes all subjects that play *administrative* role, who has defined the compartments *U.S.A.* and *financial services*, and also satisfy the condition.

Authorization objects

According to the description of MD models in the fourth section, we identify these kinds of protection objects (equivalent concepts can be found in other MD modeling approaches): fact classes, dimension classes, base classes, attributes and instances. The *object* component of our ACA model includes two parts:

- An *identity*, that can be one of the following sub-attributes:
  - *Class-Id*: The class identifier. It can be related to a fact, dimension or base class.
  - *Attribute-id*: The attribute identifier.
- An *object expression* which is an OCL expression on the class model that represents the MD model. This condition selects some instances of a class.

Authorization objects are defined as illustrated in Table 1(b). For instance, *object* element '***CL** diagnosis **COND** diagnosis.type=AIDS*' denotes all instances of the *Diagnosis* dimension class whose *type* attribute has the value *AIDS*.

Authorization actions

For the sake of simplicity, in this ACA model, we have only considered the *read* action. Other database-oriented actions relevant for ETL processes such as insert, delete, update, and maybe other OLAP-oriented actions such as

[4]In this paper we will indistinctly refer to subject and user.

[5]In this paper we will indistinctly refer to subject and user

Table 1 ACA grammar

a)	Subjects := subjectIdentification subjectExpression subjectIdentification := subjectIdentifier (logicalOperator subjectIdentifier)* subjectIdentifier := ("ALLSUBJECTS" | "ID" userID) | ("RID" roleID) | ("CID" compartmentID) | ("SL" securityLevel) logicalOperator := "AND" | "OR" subjectExpression := ("COND" OCLExpression[9]) *
b)	Objects := objectIdentifier objectExpression objectIdentifier := ("CL" className) | ("ATT" className"."attributeName) objectExpression := ("COND" OCLExpression)*
c)	Actions := action (logicalOperator action)* Action := "READ"
d)	SIAR := "OBJECTS" Objects ("INVCLASSES" involvedClasses)? ("SECINF" securityInformation | "COND" conditionAssignment) involvedClasses := Objects (logicalOperator Objects)* securityInformation := ("SL" securityLevel)? ("SR" userRole+)? ("SC" userCompartment+)? conditionAssignment := "IF" booleanExpression "THEN" (securityInformation | conditionAssignment) ("ELSE" (securityInformation | conditionAssignment))? "ENDIF"
e)	AUR := "SUBJECTS" Subjects "OBJECTS" Objects "ACTIONS" Actions "SIGN" Sign ("INVCLASSES" involvedClasses)? Sign := "+" | "-"
f)	AR := "OBJECTS" Objects "LOGTYPE" logType "LOGINFO" logInformation logType := "none" | "all" | "frustratedAttempts" | "successfulAttempts" logInformation := subjectID? objectID? action? Time? response?

[9]It represents a condition based on the user profile and on the conceptual MD model (in our case, the class diagram, considering elements such as fact classes, dimension classes, etc.)

drilling-through, drilling-down, rolling-up, slice, dice, and so on, are out of the scope this work, and will be considered in future works. Therefore, the action component of our model has only one element that identifies the type of action (see Table 1(c)).

Sensitivity information assignment rules

Owing to the fact that the access control that we have considered is MAC, we have to specify the sensitivity information of each element in the multilevel model. Therefore, for each Sensitivity Information Assignment Rule (SIAR), we need to specify the following concepts:

- *Objects* to which the rule is applicable.
- *Security information* that is assigned. It represents the access class for the object, which, as previously mentioned, can be composed of security levels, user roles and user compartments.
- *Involved classes* in the query. In MD models, the sensitivity of the information may depend on the classes that are involved in the query. Some information may not be particularly confidential if it is consulted in isolation, but the same information might be highly confidential if it is associated with other data. For instance, a list of illnesses is not very confidential, but if this information is associated with patients, it is transformed into confidential information. By default, there is always a MD involved class to which the rule refers, and which it is not necessary to specify. This element is also necessary as not all dimensions are *sliced* or *diced*: instead we may need to specify a security rule on a dimension that is not explicitly a constraint.
- *A condition* specified with OCL. Different instances of a class in our MD model can have different access classes depending on the value of some attributes. If a condition is defined in one of these rules, the specific access class of each instance of this object will depend on the evaluation of this condition. If we do not define a condition in a rule, all specified objects will have the same access class.

Sensitivity information assignment rules are defined as illustrated in Table 1(d). We can consider the following example of SIAR: '***OBJECTS CL*** *diagnosis* ***COND IF*** *diagnosis.healthArea = oncology* ***THEN SL*** *Secret* ***ELSE SL*** *Confidential* ***ENDIF***'. This SIAR defines the security level for each instance of the class *diagnosis* depending on the evaluation of that condition.

Authorization rules

There are different types of AUthorization Rules (AURs). The most important are implicit, explicit, positive, negative, weak and strong (Ferrari & Thuraisingham, 2000). Moreover, access control models can be open or closed (Samarati & De Capitani Di Vimercati, 2000). In a multilevel system, we can consider that all objects have sensitivity information (the less restrictive by default), so we can consider that our system is closed. Nevertheless, in our system both positive and negative AURs can coexist. Implicit and explicit AURs can also be defined, but we will not explicitly consider weak and strong authorizations. For each AUR, we will specify the following concepts:

- *Subjects* to which the rule is applicable.
- *Objects* to which the rule is applicable.
- *Actions* considered. As previously commented, only the *read* action is considered.
- *Sign*, that defines if the authorization is positive or negative.
- *Involved classes in* the query. It specifies the classes that have to be involved in the query in order to be applicable this AUR.

It is important to mention that if the AUR is positive, the rule will be evaluated against all users that are not able to access the information, according to the multilevel criteria (see section 'Conflict resolution'). On the other hand, if the AUR is negative, the rule will be evaluated against all users that have access to the information. Authorization rules are defined as illustrated in Table 1(e).

We can consider the AUR1 as follows: '***SUBJECTS RID*** *administrative* ***AND CID*** *U.S.A.* ***AND CID*** *financial services* ***COND*** *profile.age > 18* ***OBJECTS CL*** *diagnosis* ***COND*** *diagnosis.type = AIDS* ***ACTIONS*** *READ* ***SIGN*** *+*'.

AUR1 is a positive authorization rule that allows all subjects that play the administrative role, born in U.S.A. and having the financial services compartment, and which age is greater than 18, to read all the instances of diagnosis class if the type of the diagnosis is AIDS.

Audit rules

Audit controls are useful both as deterrent against misbehavior as well as a means to analyze the user behavior in using the system to find out possible attempted or actual violations. Additionally, auditing is essential to ensure that authorized users do not misuse their privileges (Sandhu & Samarati, 1997). Audit Rules (ARs) can be specified considering the following concepts:

- *Objects* to which the rule is applicable.
- *Log type.* It specifies whether the access has to be recorded. Values can be none, all access, only frustrated accesses, or only successful accesses.
- *Information* to be logged. Depending on the situation, we can record information such as the subject requesting the access, the object to be accessed, the operation requested, the time of the request, and the response of the access control model.

Audit rules are defined as illustrated in Table 1(f).

AR: '***OBJECTS CL*** *bankAccount* ***LOGTYPE*** *frustrated Attempts* ***LOGINFO*** *subjectID objectID action time*' is an example of audit rule, in which frustrated access attempts to the bankAccount classes should be controlled by audit trails where the subject and object identification, the action and the time in which the access is performed have to be recorded.

Conflict resolution

When a user executes a query on the DW, different security rules can be applicable, thereby appearing conflicts between several AURs, several SIARs, and even between AURs and SIARs. We solve these conflicts considering the rules included in Table 2.

UML extension for secure MD modeling

In this section, we sketch our UML extension (*profile*[6]) to apply our ACA model to the conceptual MD modeling of DWs. Basically, we have reuse the previous profile defined in Luján-Mora *et al.* (2006), which allows us to design DWs from a conceptual perspective as described in the third section, and we have added the required elements that we need to specify the security aspects (Subjects, Objects, Actions, Sensitive Information Assignment Rules, Authorization Rules, and Audit Rules) considered in our ACA model.

According to Conallen (2000), an extension to the UML begins with a brief description and then lists and describes all the stereotypes, tagged values, and constraints of the extension. In addition to these elements, an extension contains a set of well-formedness rules. These rules are used to determine whether a model is semantically consistent with itself. According to this quote, we define our UML extension for secure conceptual MD modeling following the schema composed of these elements: *description* (a little description of the extension in natural language), *prerequisite extensions* (it indicates whether the current extension needs the existence of previous extensions), *stereotypes/tagged values* (the definition of the stereotypes and/or tagged values), *well-formedness rules* (the static semantics of the metaclasses are defined both in natural language and as a set of invariants defined by means of OCL expressions), and *comments* (any additional comment, decision or example, usually written in natural language).

In the following, we will only present main aspects of our profile by providing the profile description. This will help us place any new defined element by specifying the UML element where the new element is inherited. We refer

[6] A *profile* is a set of improvements that extend an existing UML type of diagram for a different use. These improvements are specified by means of the extendibility mechanisms provided by UML (stereotypes, properties and restrictions) in order to be able to adapt it to a new method or model.

Table 2 Conflict resolution

	Positive AUR	Negative AUR	SIAR
Positive AUR	No conflict	"Most specific" rule: • An AUR that refers to an individual user has more preference than any other AUR that refers to a set of users. • An AUR that refers to a particular role *r* has more preference than any other AUR that refers to an ascendant of *r* in the user role tree. • If two AURs have the same preference, we will select the negative one.	The set of users that will be able to access the information will be composed by users that fulfill the SIAR and users that do not fulfill the SIAR but fulfill the condition of the AUR. Example: AUR2 and SIAR4 in Table 3.
Negative AUR		No conflict	The set of users that will not be able to access the information will be composed by users that do not fulfill the SIAR and the users that fulfill the SIAR but fulfill the condition of the AUR. Example: AUR1 and SIAR1 in Table 3.
SIAR			The security information for each instance, will be the result of applying the most restrictive SIAR. Example: SIAR1 and SIAR2 in Table 3.

the reader to Fernández-Medina *et al.* (2004) for a complete description of the UML profile for designing secure MD models, as this paper is focused on the ACA model.

Description

This UML extension reuses a set of stereotypes previously defined in Luján-Mora *et al.* (2006), and defines a set of tagged values, stereotypes, and constraints, which enables us to create secure MD models. The 20 tagged values we have defined are applied to certain objects that are specially particular to MD modeling, allowing us to represent them in the same model and on the same diagrams that describe the rest of the system. These tagged values will represent the sensitivity information of the different objects of the MD modeling (fact class, dimension class, base class, attributes, etc.) and they will allow us to specify security constraints depending on this security information and on the value of attributes of the model. A set of inherent constraints are specified in order to define well-formedness rules. The correct use of our extension is assured by the definition of constraints in both natural language and OSCL (Fernández-Medina & Piattini, 2004).

Thus, we have defined seven new stereotypes: one specializes in the Class model element, two specialize in the Primitive model element and four specialize in the Enumeration model element. In Figure 6, we have represented portions of the UML metamodel[7] to show where our stereotypes fit. We have only represented the specialization hierarchies, as the most important fact about a stereotype is the base class that the stereotype specializes. In this figure, new stereotypes are colored in dark gray, whereas stereotypes we reuse from the previous profile (Luján-Mora *et al.*, 2006) are in light gray and classes from the UML metamodel remain white.

A case study applying our extension for secure MD modeling

In this section, we apply our ACA model and UML extension for the conceptual design of a secure MD model in the context of a typical healthcare system. Regarding the example of Figure 4 (in section 'Object-oriented multidimensional modeling'), we have only considered a reduced example in order to focus our attention on the main security specifications. Figure 7(a) shows the simplified hierarchy of user roles of the system, and (b) shows the security levels that have been defined.[8] For the sake of simplicity, in this example, compartments have not been defined.

Considering a submodel of the MD model that is shown in Figure 4 that is composed of *Admission* fact class, *Diagnosis* and *Patient* dimension classes, and *Diagnosis_group* and *City* base classes, we could define its ACA model by defining the SIARs, AURs and ARs that are specified in Table 3.

Figure 8 shows an MD model that includes all classes previously described, and an additional class (*UserProfile*). *UserProfile* class (stereotype *UserProfile*) contains the information of all users who will have access to this MD model. We can observe in Figure 8 that we use several tagged values to allow us to model all our rules of the ACA model. SIAR1 is represented in the model by

[7] All the metaclasses come from the *Core Package*, a subpackage of the *Foundation Package*. We based our extension on the UML 1.5 as this is the current accepted standard. To the best of our knowledge, the current UML 2.0 is not the final accepted standard yet.

[8] As can be seen in Figure 6, Level is a new data type inherited from the *UML enumeration* data type.

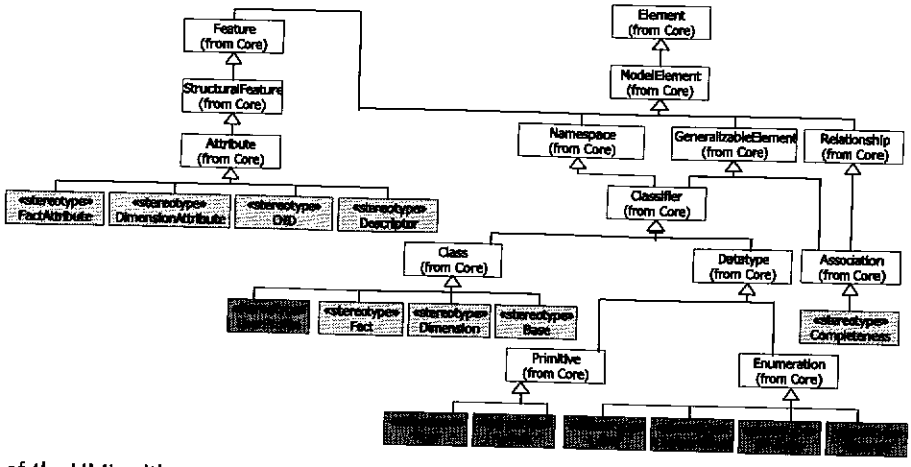

Figure 6 Extension of the UML with stereotypes.

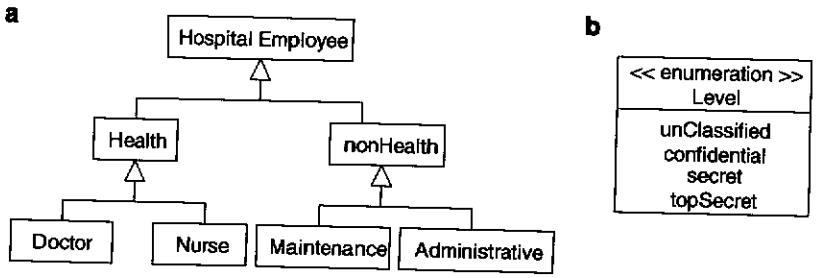

Figure 7 (a) User role hierarchy, (b) Security levels.

defining *SL* (security level) and *SR* (security roles) tagged values in *Admission* fact class. The same strategy is used to model SIAR 2–7. SIAR 8 and 9 are modeled with UML notes (labeled with numbers 1 and 2 respectively) that are associated with the *Admission* class. In these notes, we include tagged values that allow us to represent all important concepts that have been previously identified in the ACA model. We model these constraints as notes for the sake of the model readability. AUR 1 and 2 are modeled with notes 4 and 5 respectively, and finally, AR 1 is modeled with note 3. We can observe that each note is associated with the class that is represented as the *object* of the AUR and AR.

Related work

As this paper deals with several different research topics, we can organize the related work as follows.

MD modeling

Lately, several MD data models have been proposed. Some of them fall into the logical level (such as the well-known star-schema (Kimball & Ross, 2002). Others may be considered as formal models as they provide a formalism to consider main MD properties. A review of the most relevant logical and formal models can be found in Blaschka *et al.* (1998) and Abelló *et al.* (2001).

In this section, we will only make brief reference to the most relevant models that we consider 'pure' conceptual MD models. These models provide a high level of abstraction for the main MD modeling properties at the conceptual level and are totally independent from implementation issues. One outstanding feature provided by these models is that they provide a set of graphical notations (such as the classical and well-known Extended Entity–Relationship model) that facilitates their use and reading. These are as follows: *The Dimensional-Fact (DF) Model* (Golfarelli *et al.*, 1998; Golfarelli & Rizzi, 1998), *The Multidimensional/ER (M/ER) Model* (Sapia *et al.*, 1998; Sapia, 1999), *The starER Model* (Tryfona *et al.*, 1999), the Model proposed by Husemann *et al.* (2000), and *The Yet Another Multi-dimensional Model (YAM²)* (Abelló *et al.*, 2002). Unfortunately, none of them has been accepted as a standard for the conceptual modeling of DWs. Recently, another approach (Trujillo *et al.*, 2001; Luján-Mora *et al.*, 2006) has been proponed as an object-oriented (OO) conceptual MD modeling approach. This proposal is a profile of the UML (OMG, 2004c), which use the standard extension mechanisms (stereotypes, tagged values and constraints) provided by the UML.

However, none of these approaches for MD modeling considers security as an important issue of their conceptual models, and therefore, they do not solve the problem of modeling security from the early stages of a DW project.

Table 3 ACA model

SIAR 1	For each instance of the *Admission* fact class, the security level will be at least *Secret*, and the security roles will be *Health* and *Admin*.
	OBJECTS CL Admission **SECINF SL** Secret **SR** (Health, Admin)
SIAR 2	The security roles for the *cost* attribute of the *Admission* fact class will be only *Admin*
	OBJECTS ATT Admission.cost **SECINF SR** Admin
SIAR 3	For each instance of *Diagnosis* class dimension, the security level will be at least *Secret*, and the security role will be *Health* or *Admin*
	OBJECTS CL Diagnosis **SECINF SL** Secret **SR** Health
SIAR 4	For each instance of the *Patient* class dimension, the security level will be at least *Secret* and the security role will be *Health*
	OBJECTS CL Patient **SECINF SL** Secret **SR** (Health, Admin)
SIAR 5	The security role for the *address* attribute of the *Patient* dimension class will be only *Admin*
	OBJECTS ATT Patient.address **SECINF SR** Admin
SIAR 6	For each instance of the *Diagnosis_group* base class, the security level will be at least *Confidential*
	OBJECTS CL Diagnosis_group **SECINF SL** Confidential
SIAR 7	For each instance of *City* base class, the security level will be at least *Confidential*
	OBJECTS CL City **SECINF SL** Confidential
SIAR 8	For each instance of *Admission* fact class, if the description of the diagnosis_group of its diagnosis is specially sensitive (cancer or AIDS), then its security level will be *topSecret*, otherwise it will be *Secret*.
	OBJECTS CL Admission **INVCLASSES CL** Diagnosis **AND CL** Diagnosis_group **AND CL** Patient **COND IF** self.Diagnosis.Diagnosis_group.description='Cancer' or self.Disgnosis.Diagnosis_group.description='AIDS' **THEN SL** topSecret **ELSE SL** Secret **ENDIF**
SIAR 9	For each instance of *Admission* fact class, if its cost is greater than $10000, then its security level will be *topSecret*, otherwise it will be *Secret*
	OBJECTS CL Admission **INVCLASSES CL** Patient **COND IF** self.cost>10000 **THEN SL** topSecret **ELSE SL** Secret **ENDIF**
AUR 1	If a query involves *Diagnosis*, *Diagnosis_group* and *Patient* classes, the information that can be collected about patients is very sensitive. So, we will allow access information only members of the *Health* security role, and if their working area is the same that the health area of the patients.
	SUBJECTS RID Health **COND** userProfile.workingArea<>self.diagnosis.healthArea **OBJECTS CL** Admission **ACTION** READ **SIGN** - **INVCLASSES CL** Diagnosis **AND CL** Diagnosis_group **AND CL** Patient
AUR 2	Patients will be special users of the system as we would like they could access their own data.
	SUBJECTS ALLSUBJECTS **COND** userProfile.name=self.name **OBJECTS CL** Patient **ACTION** READ **SIGN** +
AR 1	We wish to record the subject, object and time for all frustrated access attempts, in order to analyze who tries to access illegally to the information of our data warehouse
	OBJECTS CL Admission **LOGTYPE** frustratedAttempts **LOGINFO** subjectID ObjectID time

Security integration into the design process

There are a few proposals that try to integrate security into conceptual modeling such as the Semantic Data Model for Security (Smith, 1991) and the Multilevel Object Modeling Technique (Marks *et al.*, 1996), but they are partial (since they do not cover the complete development process). More recent proposals are UMLSec (Jürjens, 2002) and SecureUML (Lodderstedt *et al.*, 2002) where UML is extended to develop secure systems. These approaches are very interesting, but they only deal with

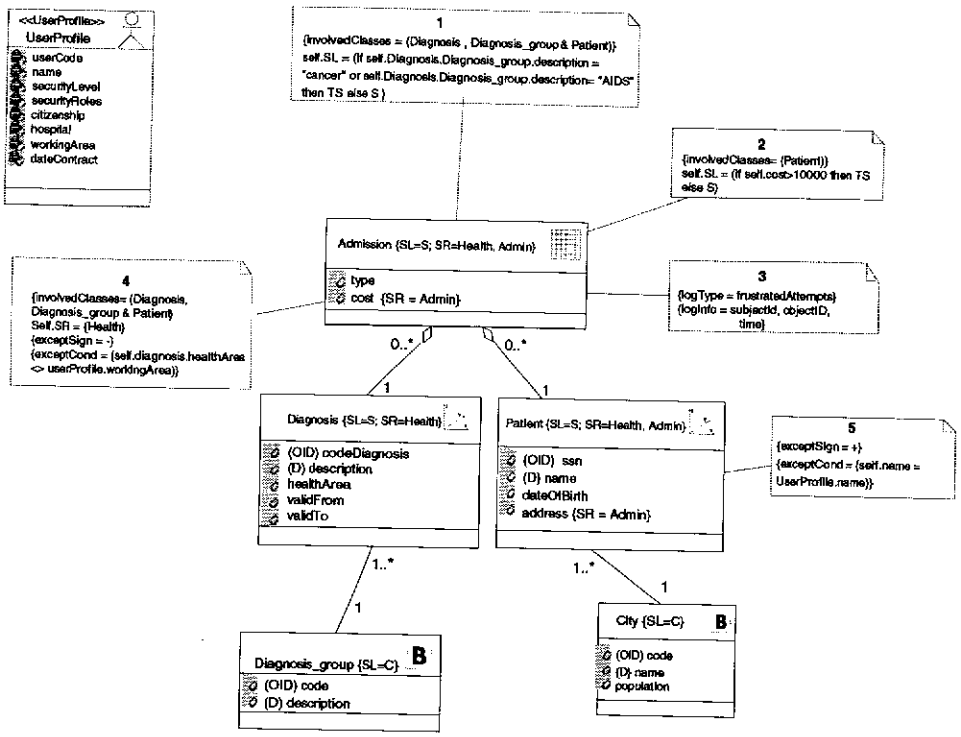

Figure 8 Example of multidimensional model with security information and constraints.

information systems (IS) in general, while conceptual database and DW design are not considered. Moreover, a methodology and a set of models have been proposed (Fernández-Medina & Piattini, 2003) in order to design secure databases to be implemented with OLS. This approach, based on the UML, is important because it considers security aspects in all stages of the database development process, from requirement gathering to implementation. Together with the previous methodology, the proposed Object Security Constraint Language (OSCL) (Piattini & Fernández-Medina, 2001), based on the Object Constraint Language (OCL) (Warmer & Kleppe, 2003) of UML, allows us to specify security constraints in the conceptual and logical database design process, and to implement these constraints in a concrete database management system (DBMS), OLS. Nevertheless, the previous methodology and models do not consider the design of secure MD models for DWs, and therefore, are not adequate to represent the peculiarities of DWs.

In the literature, we can find several initiatives to include security in DW (Kirkgöze *et al.*, 1997; Katic *et al.*, 1998; Priebe & Pernul, 2000; Rosenthal & Sciore, 2000). Many of them are focused on interesting aspects related to access control, multilevel security, its applications to federated databases, applications using commercial tools and so on. These initiatives refer to specific aspects that allow us to improve DW security in acquisition, storage, and access aspects. However, none of them considers the security aspects into all stages of the system development cycle nor considers the introduction of security in conceptual MD design.

Access control models

Many proposals have been developed in order to protect information against improper disclosure or modifications. All of them exploit the particularities of the systems they deal with, such as the types of objects, subjects, privileges, signs, conflict resolutions, etc. For instance, there are authorization models for data archives (Bonatti *et al.*, 2001), database systems (Rabitti *et al.*, 1991; Bertino *et al.*, 1999), XML documents (Damiani *et al.*, 2002b), and even for multimedia documents (Damiani *et al.*, 2002a).

On the other hand, there are some interesting proposals that try to define an authorization model for DWs (Kirkgöze *et al.*, 1997; Katic *et al.*, 1998; Weippl *et al.*, 2001; Wang *et al.*, 2004), but they mainly deal with OLAP operations accomplished with OLAP tools. Therefore, they are not conceived to be integrated in the MD modeling as part of the DW design process. Furthermore, from our point of view, we should consider basic security aspects of DWs by means of a conceptual model from the early stages of a DW project, and then, more specific security rules can be defined for particular groups of users by OLAP tools or any other analysis tool, but being consistent with the main general security rules defined for the DW they are to be queried. Finally, these above-presented proposals only consider access control, but not audit.

Conclusions and future work

In this paper, we have presented an ACA model and an extension of the UML that allows us to specify and

represent main confidentiality aspects in the conceptual modeling of secure DWs. The ACA model allows us to define rules in order to specify the security information of each element in the MD model (SIARs), rules for representing authorization rules (AURs), which works together with SIARs, and rules that allow us to specify audit requirements (ARs). The UML extension contains the needed stereotypes, tagged values and constraints for a complete and powerful secure MD modeling. These new elements allow us to specify security aspects such as security levels on data, compartments and user roles on the main elements of a MD model such as facts, dimensions, classification hierarchies and so on. We have used the OSCL to specify the ACA rules and the constraints attached to these new defined elements, thereby avoiding an arbitrary use of them.

One of the key advantages of our approach is that we base on well-known and fashion standards such as UML. We have also shown that our approach is aligned with MDS, and also with the MDA and MDDW approaches proposed by the OMG. Considering that DW, MD Databases, and OLAP applications are used as very powerful mechanisms for discovering crucial business information in strategic decision-making processes, this proposal provides interesting advances for improving the security of decision support systems and protecting sensitive information that these systems usually manage.

Our immediate future work is to improve the ACA model, extending the set of privileges considered in this paper (i.e. read) to allow us to specify security aspects in the crucial ETL processes for DWs, thereby considering other operations such as delete, insert and update. Moreover, we also plan to consider implementation issues to use the considered security aspects when querying a MD model from OLAP tools, which includes a deep study on the different combinations of dimensions, measures, hierarchies and so on involved in classical OLAP operations such as roll-up, drill-down, slicing, dicing and so on. We also plan to provide a metamodel transformation based on the CWM for the enriched secure UML profile and ACA model proposed in this paper. Finally, we will work on completing our MDA architecture with the definition and integration of a Computation Independent Model.

Acknowledgements

This research is part of the following projects: METASIGN (TIN2004-00779), and ESFINGE (TIN2006-15175-C05-05), projects from the Spanish Ministry of Education and Science; DIMENSIONS (PBC-05-012-1), DADS (PBC-05-012-2), and MISTICO (PBC06-0082) projects partiality supported by the FEDER and the 'Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha', Spain.

Acronyms

ACA: Access Control and Audit
AR: Audit Rule
AUR: Authorization Rule
CWM: Common Warehouse Metamodel
DAG: Directed Acyclic Graph
DBMS: Database Management System
DW: Data Warehouse
EBNF: Extended Backus Naur Form
ETL: Extraction–Transformation–Loading
MAC: Mandatory Access Control
MD: Multidimensional
MDA: Model-Driven Architecture
MDDW: Model-Driven Data Warehouse
MDS: Model-Driven Security
MOF: Meta Object Facility
OCL: Object Constraint Language
OID: Object Identifier
OLAP: On-Line Analytical Processing
OLS: Oracle Label Security
OMG: Object Management Group
OSCL: Object Security Constraint Language
PIM: Platform Independent Model
PSM: Platform Specific Model
RBAC: Role Based Access Control
SIAR: Security Information Assignment Rule
UML: Unified Modeling Language
XMI: eXtensible Markup Interchange
XML: eXtensible Markup Language

Abbreviations used in the ACA grammar

ATT: Attribute
CID: Compartment Identification
CL: Class
COND: Condition
ID: User Identification
INVCLASSES: Involved Classes
LOGINFO: Log Information
RID: Role Identification
SC: Security Compartments
SECINF: Security Information
SL: Security Level
SL: Security Levels
SR: Security Roles

About the authors

Eduardo Fernández-Medina is Ph.D. and M.Sc. in Computer Science from the University of Sevilla. He is member of the ALARCOS research group, and Associated Professor at the Escuela Superior de Informática of the Universidad de Castilla-La Mancha at Ciudad Real (Spain). His research activities include security in databases, data warehouses, business processes, web services and information systems, security

metrics, etc. His e-mail address is: eduardo.fdezmedina@uclm.es.

Juan Trujillo is an associate professor at the Computer Science School at the University of Alicante, Spain. Trujillo received a Ph.D. in Computer Science from the University of Alicante (Spain) in 2001. His research interests include database modeling, conceptual design of data warehouses, multidimensional databases, OLAP, and object-oriented analysis and design with UML. His e-mail address is jtrujillo@dlsi.ua.es.

Mario Piattini is M.Sc. and Ph.D. in Computer Science by the Politechnical University of Madrid. He is Full Professor at the Escuela Superior de Informática of the Castilla-La Mancha University (Spain). He leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. His research interests are advanced database design, database quality, software metrics, object-oriented metrics, software maintenance. His e-mail address is mpiattin@inf-cr.uclm.es

References

ABELLÓ A, SAMOS J and SALTOR F (2001) A framework for the classification and description of multidimensional data models. *12th International Conference on Database and Expert Systems Applications (DEXA'01)* Lecture Notes in Computer Science, Vol. 2113, pp 668–677 Springer, Berlin.

ABELLÓ A, SAMOS J and SALTOR F (2002) YAM2 (Yet Another Multidimensional Model): an extension of UML. In *International Database Engineering & Applications Symposium (IDEAS 2002)* (NASCIMENTO MA, TAMER ÖZSU M and ZAÏANE OR, Eds), pp 172–181, IEEE Computer Society Edmonton, Canada.

BASIN DA, DOSER J and LODDERSTEDT T (2006) Model driven security: from UML models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology* **15(1)**, 39–91.

BERTINO E, JAJODIA S and SAMARATI P (1999) A flexible authorization mechanism for relational data management systems. *ACM Transactions on Information Systems* **17**, 101–140.

BLASCHKA M, SAPIA C, HÖFLING G and DINTER B (1998) Finding your way through multidimensional data models. In *Proceedings of the Ninth International Conference on Database and Expert Systems Applications (DEXA'98)* Lecture Notes in Computer Science, Vol. 1460, pp 198–203, Springer-Verlag, Vienna, Austria.

BONATTI P, DAMIANI E, DE CAPITANI DI VIMERCATI S and SAMARATI P (2001) An access control model for data archives. In *Proceedings of the IFIP-TC11 International Conference on Information Security*, Paris, France.

CONALLEN J (2000) *Building Web Applications with UML. Object Technology Series.* Addison-Wesley, Reading MA.

COTA S (2004) For certain eyes only. *DB2 Magazine* **9(1)**, 40–45.

DAMIANI E, DE CAPITANI DI VIMERCATI S, FERNANDEZ-MEDINA E and SAMARATI P (2002a) An access control system for SVG documents. In *Research Directions in Data and Applications Security* (GUDES E and SHENOI S, Eds), pp 219–230, Kluwer Academic Publisher, Boston.

DAMIANI E, DE CAPITANI DI VIMERCATI S, PARABOSCHI S and SAMARATI P (2002b) A fined-grained access control system for XML documents. *ACM Transactions on Information and Systems Security* **5**, 169–202.

DEVANBU P and STUBBLEBINE S (2000) Software engineering for security: a roadmap. In *The Future of Software Engineering* (FINKELSTEIN A, Ed), pp 227–239, ACM Press, New York.

DHILLON G and BACKHOUSE J (2000) Information system security management in the new millennium. *Communications of the ACM* **43(7)**, 125–128.

Directive 95/46/CE of the European Parliament and Council, dated 24 October, about People protection regarding the personal data management and the free circulation of these data. DOCE no. L281, 23/11/1995, P.0031-0050, 1995.

FERNANDEZ EB and PAN RY (2001) A pattern language for security models. In *Proceedings of the 8th Conference on Patterns Languages of Programs (PLOP 2001)*,Illinois, USA.

FERNÁNDEZ-MEDINA E and PIATTINI M (2003) Designing secure database for OLS. In *Database and Expert Systems Applications: 14th International Conference (DEXA 2003)* Lecture Notes in Computer Science, Vol. 2736, Prague, Czech Republic (MARIK V, RETSCHITZEGGER W and STEPANKOVA O, Eds), pp 886–895, Springer, Berlin.

FERNÁNDEZ-MEDINA E and PIATTINI M (2004) Extending OCL for secure database design. In *Proceedings of the International Conference on the Unified Modeling Language (UML 2004)*, Lecture Notes in Computer Science, Lisbon, Portugal Springer-Verlag, Berlin.

FERNÁNDEZ-MEDINA E, TRUJILLO J, VILLARROEL R and PIATTINI M (2004) Extending the UML for designing secure data warehouses. In *Proceedings of the International Conference on Conceptual Modeling (ER 2004)*. Springer-Verlag, Shangai, China.

FERRARI E and THURAISINGHAM B (2000) Secure database systems. In *Advanced Databases: Technology Design* (PIATTINI M and DÍA O, Eds) Artech House, London.

FRANCE R and BIEMAN J (2001) Multi-view software evolution: a UML-based framework for evolving object-oriented software. In *Proceedings of the International Conference on Software Maintenance*, Florence, Italy, pp 386–397.

FRANKEL DS (2003) *Model Driven Architecture. Applying MDA to Enterprise Computing.* Indiana Wiley, Indianapolis.

GOGOLLA M and HENDERSON-SELLERS B (2002) Analysis of UML Stereotypes within the UML metamodel. In *Proceedings of the 5th International Conference on the Unified Modeling Language – The Language and its Applications.* Lecture Notes in Computer Science, Vol. 2460, Dresden, Germany, pp 84–99, Springer, Berlin.

GOLFARELLI M, MAIO D and RIZZI S (1998) The dimensional fact model: a conceptual model for data warehouses. *International Journal of Cooperative Information Systems* **7(2–3)**, 215–247.

GOLFARELLI M and RIZZI S (1998) A methodological framework for data warehouse design. In *Proceedings of the 1st International Workshop on Data Warehousing and OLAP (DOLAP'98)*, Maryland, USA, pp 3–9.

HALL A and CHAPMAN R (2002) Correctness by construction: developing a commercial secure system. *IEEE Software* **19(1)**, 18–25.

HUSEMANN B, LECHTENBORGER J and VOSSEN G (2000) Conceptual data warehouse design. In *Proceedings of the 2nd. International Workshop on Design and Management of Data Warehouses (DMDW'2000).* Technical University of Aachen (RWTH). Stockholm, Sweden, pp 3–9.

INMON H (2002) *Building the Data Warehouse*, 3rd edn, John Wiley & Sons, USA.

JAJODIA S, SAMARATI P, SAPINO ML and SUBRAHMANIAN VS (2001) Flexible support for multiple access control policies. *ACM Transactions on Database Systems* **26**, 214–260.

JÜRJENS J (2002) UMLsec: extending UML for secure systems development. In *UML 2002 – The Unified Modeling Language, Model Engineering, Concepts and Tools.* Lecture Notes in Computer Science, Vol. 2460. Dresden, Germany (JÉZÉQUEL USSMANN H and COOK S, Eds), pp 412–425, Springer, Berlin.

KATIC N, QUIRCHMAYR G, SCHIEFER J, STOLBA M and MIN TJOA A (1998) A prototype model for data warehouse security based on metadata. In *Proceedings of the 9th International Workshop on Database and Expert Systems Applications (DEXA'98)*, pp 300–308, IEEE Computer Society, Vienna, Austria.

KIMBALL R (1996) *The Data Warehousing Toolkit*, John Wiley, New York, USA.

KIMBALL R and ROSS M (2002) *The Data Warehousing Toolkit*, John Wiley, New York, USA.

KIRKGÖZE R, KATIC N, STOLDA M and MIN TJOA A (1997) A security concept for OLAP. In *Proceedings of the 8th International Workshop on Database and Expert System Applications (DEXA'97)*, pp 619–626, IEEE Computer Society, Toulouse, France.

KLEPPE A, WARMER J and BAST W (2003) *MDA Explained; The Model Driven Architecture: Practice and Promise*. Addison-Wesley, Reading, MA.

LEVINGER J (2002) Oracle label security. Administrator's guide. Release 2 (9.2). http://www.csis.gvsu.edu/GeneralInfo/Oracle/network.920/a96578.pdf.

LODDERSTEDT T, BASIN D and DOSER J (2002) SecureUML: a UML-based modeling language for model-driven security. In *Proceedings of the UML 2002*. The Unified Modeling Language. Model Engineering, Languages Concepts, and Tools. 5th International Conference, pp 426–441, Springer, Dresden, Germany.

LUJÁN-MORA S, TRUJILLO J and SONG IY (2006) A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering* **59(3)**, 725–769.

MARKS D, SELL P and THURAISINGHAM B (1996) MOMT: a multi-level object modeling technique for designing secure database applications. *Journal of Object-Oriented Programming* **9(4)**, 22–29.

MAZÓN JN, TRUJILLO J and LECHTENBÖRGUER J (2007) An MDA approach for the development of data warehouses. *Decision Support Systems*, Accepted for publication. Available online. doi:10.1016/j.dss.2006.12.003.

MEDINA E and TRUJILLO J (2002a) Representing conceptual multidimensional properties using the common warehouse metamodel (CWM). In *Proceedings of the Advances in Web-Age Information Management, 3rd International Conference, WAIM 2002*. Lecture Notes in Computer Science, Vol. 2419, Beijing, China, pp 259–270, Springer, Berlin.

MEDINA E and TRUJILLO J (2002b) A standard for representing multidimensional properties: the common warehouse metamodel (CWM). In *Proceedings of the Advances in Databases and Information Systems, 6th East European Conference, (ADBIS 2002)*. Lecture Notes in Computer Science, Vol. 2435, Bratislava, Slovakia, pp 232–247, Springer, Berlin.

OMG (2004a) Object Management Group. Common Warehouse Metamodel Specification, V1.1.

OMG (2004b) Object Management Group. Model Driven Architecture (MDA).

OMG (2004c) Object Management Group: Unified Modeling Language Specification 1.5.

PIATTINI M and FERNÁNDEZ-MEDINA E (2001) Specification of security constraint in UML. In *Proceedings of the 35th Annual 2001 IEEE International Carnahan Conference on Security Technology (ICCST 2001)*, pp 163–171, London, Great Britain.

POOLE J (2003) *Model-Driven Data Warehousing*. Burlingame, CA.

POOLE J, CHANG D, TOLBERT D and MELLOR D (2002) *Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration*. John Wiley, New York, USA.

PRIEBE T and PERNUL G (2000) Towards OLAP security design – survey and research issues. In *Proceedings of the 3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00)*, pp 33–40, Washington DC, USA.

RABITTI F, BERTINO E, KIM W and WOELK D (1991) A model of authorization for next-generation database systems. *ACM Transactions on Database Systems* **16(1)**, 88–131.

ROSENTHAL A and SCIORE E (2000) View security as the basic for data warehouse security. In *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouse (DMDW'00)*, pp 8.1–8.8, Sweden.

SAMARATI P and DE CAPITANI DI VIMERCATI S (2000) Access control: policies, models, and mechanisms. In *Foundations of Security Analysis and Design* (FOCARDI R and GORRIERI R, Eds), pp 137–196, Springer Bertinoro, Italy.

SANDHU R, COYNE E, FEINSTEIN H and YOUMAN C (1996) Role-based access control models. *IEEE Computer* **29(2)**, 38–47.

SANDHU R and CHEN F (1998) The multilevel relational data model. *ACM Transactions on Information and Systems Security (TISSEC)* **1(1)**, 93–132.

SANDHU R, FERRAIOLO D and KUHN R (2000) The NIST model for role-based access control: towards a unified standard. In *Proceedings of the Fifth ACM Workshop on Role-Based Access Control*, pp 47–63, Berlin, Germany.

SANDHU R and SAMARATI L (1997) Authentication, access control, and intrusion detection. In *CRC Handbook of Computer Science and Engineering* (TUCKER A, Ed) CRC Press Inc, Boca Raton, FL.

SAPIA C (1999) On modeling and predicting query behavior in OLAP systems. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99)*, pp 1–10, Heidelberg, Germany.

SAPIA C, BLASCHKA M, HÖFLING G and DINTER B (1998) Extending the E/R model for the multidimensional paradigm. In *Proceedings of the 1st International Workshop on Data Warehouse and Data Mining (DWDM'98)*, pp 105–116, Springer-Verlag, Singapore.

SMITH GW (1991) Modeling security-relevant data semantics. *IEEE Transactions on Software Engineering* **17(11)**, 1195–1203.

SOLER E, TRUJILLO J, FERNANDEZ-MEDINA E and PIATTINI M (2007a) SECRDW: an extension of the relational package from CWM for representing secure data warehouses at the logical level. In *Proceedings of the Fifth International Workshop on Security in Information Systems (WOSIS 2007)*, pp 245–256, Accepted, Insticc Press, Funchal, Madeira, Portugal.

SOLER E, TRUJILLO J, FERNANDEZ-MEDINA E and PIATTINI M (2007b) A set of QVT relations to transform PIM to PSM in the design of secure data warehouses. In *Proceedings of the IEEE Second International Symposium on Frontiers in Availability, Reliability and Security (FARES 2007)*, pp 644–654, Vienna, Austria.

THOMSEN E (1997) *OLAP Solutions*. John Wiley & Sons, Inc., New York, USA.

TOVAL A, NICOLÁS J, MOROS B and GARCÍA F (2002) Requirement reuse for improving information systems security: a practitioner's approach. *Requirement Engineering Journal* **6(4)**, 205–219.

TRUJILLO J, PALOMAR M, GÓMEZ J and SONG IY (2001) Designing data warehouses with OO conceptual models. *IEEE Computer, special issue on Data Warehouses* **12(34)**, 66–75.

TRYFONA N, BUSBORG F and CHRISTIANSEN J (1999) starER: a conceptual model for data warehouse design. In *Proceedings of the ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP'99)*, pp 3–8, ACM, Missouri, USA.

WANG L, JAJODIA S and WIJESEKERA D (2004) Securing OLAP data cubes against privacy breaches. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp 161–178, Berkeley, California.

WARMER J and KLEPPE A (2003) *The Object Constraint Language Second Edition. Getting Your Models Ready for MDA*. Addison Wesley, Reading, MA.

WEIPPL E, MANGISENGI O, ESSMAYR W, LICHTENBERGER F and WINIWARTER W (2001) An authorization model for data warehouses and OLAP. In *Proceedings of the Workshop on Security in Distributed Data Warehousing New Orleans*, Louisiana, USA.