A UML 2.0/OCL Extension for Designing Secure Data Warehouses

Rodolfo Villarroel

Departamento de Computación e Informática. Universidad Católica del Maule
Avenida San Miguel 3605
Talca (Chile)
rvillarr@spock.ucm.cl

Eduardo Fernández-Medina and Mario Piattini

Departamento de Informática. Universidad de Castilla-La Mancha
Paseo de la Universidad, 4
13071 Ciudad Real (Spain)
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Juan Trujillo

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante
C/San Vicente S/N
03690 Alicante (Spain)
jtrujillo@dlsi.ua.es

At present, it is very difficult to develop a methodology that fulfills all criteria and comprises all security constraints in the successful design of data warehouses. If that methodology were developed, its complexity would hinder its success. The solution, therefore, would be an approach in which techniques and models defined by the most accepted model standards were extended by integrating the necessary security aspects that at this moment in time are not covered by the existing methodologies. In this paper, we will focus on solving confidentiality problems in the conceptual modelling of data warehouses by defining a profile using the UML 2.0 extensibility mechanisms. In addition, we define an OCL extension that allows us to specify the security constraints of the elements in conceptual modelling of data warehouses and we apply this profile to an example.

Keywords: Secure Data Warehouse, UML profile, OCL, security, confidentiality.

ACM Classification : D2.2 (Design Tools and Techniques), K6.5 (Security and Protection)

1. INTRODUCTION

Security, and specifically confidentiality, is a very important aspect for data warehouses, due to the fact that the constant changes of user requests and data sources force them not only to be more flexible but also to control confidentiality of information more effectively. A very important aspect of data warehouses that should be considered, and which makes them different from operational systems, is that information is not treated statically, but rather the evolution of this information, in

Manuscript received: 12 April 2005
Communicating Editor: Julio Cesar Hernandez

other words, its history (Inmon, 2002), becomes more important as time goes by. For this reason, mechanisms allowing confidentiality of such a great quantity of information must be established. Indeed, the very survival of organizations depends on the correct management, security and confidentiality of information (Dhillon and Backhouse, 2000). In fact, as some authors have remarked (Devanbu and Stubblebine, 2000; Ferrari and Thuraisingham, 2000), security of information is a serious requirement which must be given careful thought to, not as an isolated aspect, but as an element present in all stages of the development lifecycle, from requirement analysis to implementation and maintenance. Chung, Nixon, Yu and Mylopoulos (2000) also insist on the need to integrate security requirements into design, by providing designers with models specifying aspects of security. They do not deal with data warehouse issues, however.

In the past few years, various approaches have been proposed for representing the main multidimensional (MD) properties at the conceptual level (Abelló, Samos and Saltor, 2002; Golfarelli, Maio and Rizzi, 1998; Husemann, Lechtenborger and Vossen, 2000; Sapia, Blaschka, Höfling and Dinter, 1998; Trujillo, Palomar, Gómez and Song, 2001; Tryfona, Busborg and Christiansen, 1999). Nonetheless, none of these approaches for MD modelling, considers security to be an important issue in their conceptual models, so they do not solve the problems arising from this question in these kinds of systems. It is true that, in the relevant literature, we can find several initiatives for the inclusion of security in data warehouses (Katic, Quirchmayr, Schiefer, Stolba and Min Tjoa, 1998; Kirkgöze, Katic, Stolda and Min Tjoa, 1997; Priebe and Pernul, 2000; Rosenthal and Sciore, 2000). Many of these focus on interesting aspects related to access control, multilevel security, their applications to federated databases, applications using commercial tools and so on. However, none of them considers security aspects which incorporate all stages of the system development cycle, nor the introduction of security into MD conceptual design.

We believe that our solution would be an approach in which techniques and models defined by the most accepted model standards were extended by integrating the necessary security aspects that, at present, are not covered by the existing methodologies. Taking this into account, we see that the UML offers us two different approaches for extending its metamodel (Fuentes and Vallecillo, 2004). The first one provides us with the possibility of defining a new modelling language by using MOF (Meta Object Facility) in which there are not any restrictions regarding what can be done with a metamodel. For example, metaclasses and relationships can be added and removed according to our needs. We have not chosen this option, because the new language will not respect the UML semantics and consequently we will not be able to use commercial tools based on UML. Moreover, the purpose of our proposal is to be able to generate a secure conceptual modelling with ease and precision, applied to a specific dominion, in this case, to data warehouses. This fact fits perfectly with the concept of profile which corresponds to the second approach provided by the UML for the extension of a metamodel.

A UML 2.0 profile is defined as a UML package stereotyped "profile", that can extend either a metamodel or another profile (OMG, 2003). A profile is used to extend an existing metamodel by using three basic mechanisms provided by the UML: stereotypes, tagged values and constraints, to adapt it to a dominion, platform or specific method. In our case, we will use the mechanisms indicated to incorporate security aspects into conceptual modelling of data warehouses.

The remainder of this paper is structured as follows. Section 2 will present the UML 2.0/OCL profile for designing secure data warehouses. In Section 3, an example of modelling using the proposed extensibility mechanisms will be set out. Finally, Section 4 will put forward our main conclusions and will introduce our work for the immediate future.

2. UML 2.0/OCL PROFILE FOR DESIGNING SECURE DATA WAREHOUSES

In this section, we present the main aspects of our profile for the design of secure data warehouses. 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 this extension. Basically, we have reused the profile defined previously in Luján-Mora, Trujillo and Song (2002), which allows us to design data warehouses from a conceptual perspective, then adding the elements required for the generating of the profile (a set of tagged values, stereotypes, and constraints), thus enabling us to create secure MD models. Furthermore, an extension is formed by a set of well-formedness rules that will ensure correct static semantics of the multidimensional model.

The goal of this UML profile is to be able to design an MD conceptual model, but at the same time classifying information, in order to define which properties the user has to possess in order to be entitled to gain access to information. Therefore, our aim is to classify the security information that will be used in our conceptual modelling of data warehouses. We can define, for each element of the model (fact class, dimension class, fact attribute, etc.), its security information, specifying a sequence of security levels, a set of user compartments and a set of user roles. We can also specify security constraints considering these security attributes. The security information and these constraints indicate the security properties that users have to have to be able to access information. We have adapted OCL (Warmer and Kleppe, 2003) to be coherent with our UML 2.0 profile.

2.1 General Description

Our profile will be called SECDW (Secure Data Warehouses) and will be represented as a UML package. This profile will not only inherit all properties from the UML metamodel but it will also incorporate new data types, stereotypes, tagged values and constraints. In Figure 1, a high-level view of our SECDW profile is provided. The package SECDW and the OCL are imported from the SECDW profile. Therefore, SECDW data types and OCL types will be used as valid types for the stereotypes of our profile.

2.2 Data Types

We need the definition of some new data types to be used in the tagged value definitions of the new stereotypes. In Table 1, we will provide the new data type definitions we have specified.

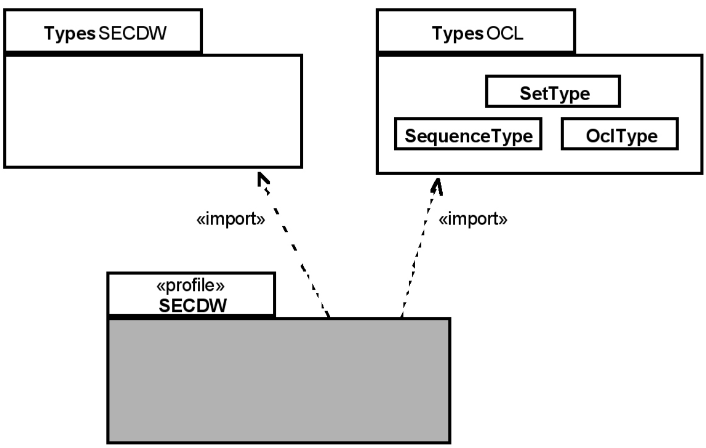


Figure 1: High level view of our SECDW profile

Name	Base class	Description
Level	Enumeration	The type Level will be an ordered enumeration composed of all security levels that have been considered.
Levels	Primitive	The type Levels will be an interval of levels composed of a lower level and an upper level.
Role	Primitive	The type Role will represent the hierarchy of user roles that can be defined for the organization.
Compartment	Enumeration	The type Compartment is the enumeration composed of all user compartments that have been considered for the organization.
Privilege	Enumeration	The type Privilege will be an ordered enumeration composed of all different privileges that have been considered.
AccessAttempt	Enumeration	The type Attempt will be an ordered enumeration composed of all different access attempts that have been considered.

Table 1: New Data Types

All the information considered in these new data types has to be defined for each specific secure conceptual database model, depending on its confidentiality properties, and on the number of users and complexity of the organization in which the data warehouse will be operative.

In Figure 2, we can observe the values associated to each one of the necessary types. Security levels, roles and organizational compartments can be defined according to the needs of the

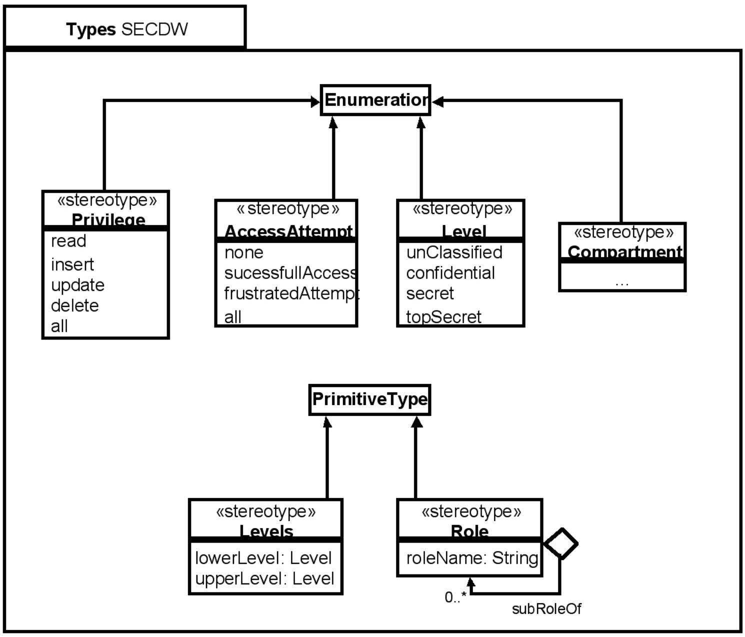


Figure 2: Values associated to new data types

organization. However, for this figure to be better understood, we have considered within the "Level" data type, the typical values associated to security levels.

2.3 Stereotypes

We have defined a package that includes all the stereotypes that will be necessary in our profile (see Figure 3). This profile contains four types of stereotypes:

- Secure class and secure data warehouses stereotypes (and stereotypes inheriting information from them) that contain tagged values associated to attributes (model or class attributes), security levels, user roles and organizational compartments.
- Attribute stereotypes (and stereotypes inheriting information from attributes) and instances, which have tagged values associated to security levels, user roles and organizational compartments.
- Stereotypes that allow us to represent security constraints, authorization rules and audit rules.
- UserProfile stereotype, which is necessary to specify constraints depending on particular information of a user or a group of users.

In Figure 3, we can see the tagged values associated to each one of the stereotypes. For example, 'SecureDW' stereotype has the following values associated: Classes, SecurityLevels, SecurityRoles and SecurityCompartments. In Table 2, we will show the description of each one of the stereotypes.

2.4 Tagged Values

The tagged values we have defined are applied to certain components that are especially particular to MD modelling, allowing us to represent them in the same model and in the same diagrams that

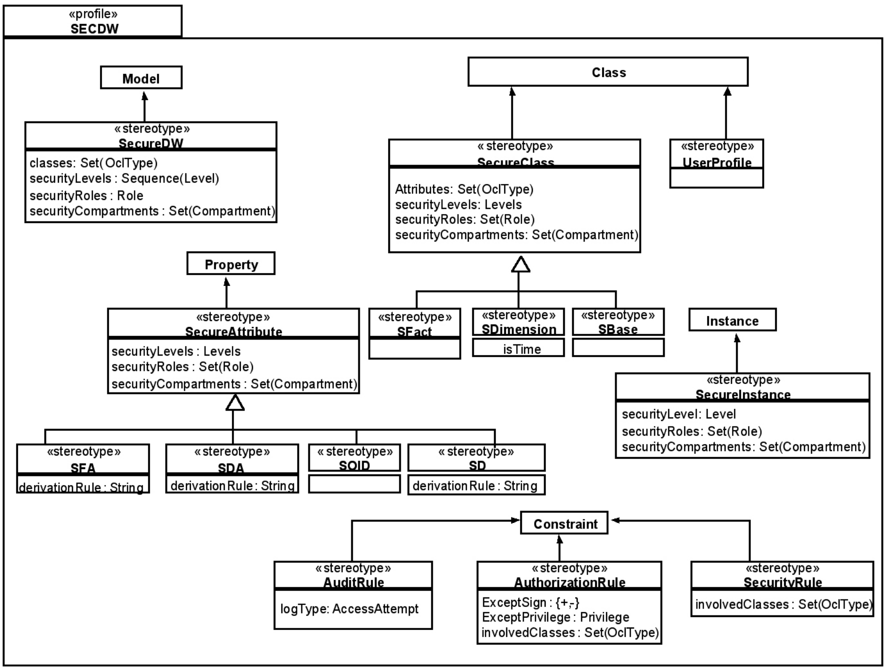
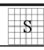


Figure 3: New stereotypes

Name	SecureDW	Icon	
Description	Instances of this data warehouse model will allow us to define security information and constraints regarding its elements.		
Name	UserProfile	Icon	
Description	Classes of this stereotype contain all the properties that the systems manage from users.		
Name	Secure Class	Icon	
Description	This type of class can have sensitivity information associated. We can therefore classify these according to their own confidentiality properties.		
Name	SecureFact	Icon	
Description	They represent facts within a multidimensional model. They inherit tagged values from SecureClass.		
Name	SecureDimension	Icon	
Description	They represent dimensions within a multidimensional model. They inherit tagged values from SecureClass.		
Name	SecureBase	Icon	
Description	They represent dimension hierarchy levels within a multidimensional model. They inherit tagged values from SecureClass.		
Name	SecureAttribute	Icon	
Description	This type of attributes can have sensitivity information associated. We can therefore classify these attributes according to their own confidentiality properties.		
Name	SecureFactAttribute	Icon	**SFA**
Description	They represent Fact class attributes within a multidimensional model and inherit tagged values from SecureAttribute.		
Name	SecureDimensionAttribute	Icon	**SDA**
Description	They represent Dimension or Base class attributes within a multidimensional model and inherit tagged values from SecureAttribute.		
Name	SecureOID	Icon	**SOID**
Description	They represent OID attributes (Identifier attribute) of Fact, Dimension or Base classes within a multidimensional model and inherit security aspects from SecureAttribute.		
Name	SecureDescriptor	Icon	**SD**
Description	They represent descriptor attributes of Dimension or Base classes within a multidimensional model and inherit security aspects from SecureAttribute.		
Name	SecureInstance	Icon	
Description	This type of instances can have sensitivity information associated. We can therefore classify these instances according to their own confidentiality properties.		

Name	AuditRule	Icon	
Description	This type of rules can contain information to analyze the user behaviour when using the system. Therefore, they will specify whether access must be registered.		
Name	AuthorizationRule	Icon	
Description	This type of rules can contain information to permit or deny access. Therefore, they will specify if authorization is positive or negative and the privileges necessary for access.		
Name	SecurityRule	Icon	
Description	This type of rules can have sensitivity information associated. Therefore, they will specify if security information is necessary.		

Table 2: Stereotypes

describe the rest of the system. In Table 3, the necessary tagged values in our profile are shown. These tagged values will represent the sensitivity information of the different elements of the MD modelling (fact class, dimension class, base class, etc.), and they will allow us to specify security constraints depending on this security information and on the value of attributes of the model.

2.5 Well-Formedness Rules

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 Object Constraint Language (OCL). We will identify and specify some well-formedness rules needed for the correct use of the new elements specified in this profile. These rules are grouped as follows:

- Correct value of tagged values. For example; the security levels defined for each class of the model and for each attribute of each class has to belong to the sequence of security levels that has been defined for the model.
- Security information of instances. For example, the security level of the instance of a class has to be included in the ranking of security levels that has been defined for the class.
- Relationship between security information of classes and their attributes. The security levels defined for an attribute have to be equal to, or more restrictive than, the security levels defined for its class.
- Categorization of dimensions. When a dimension class is specialized in several base classes, the security levels of the subclasses have to be equal to, or more restrictive than, the security levels of the superclass.
- Classification hierarchies. As a general rule, we can consider that the more specific the information is, the more restrictive its access is.
- Derived Attribute. The security levels of a derived attribute have to be equal or more restrictive than the attributes which this attribute is based on.
- Combination of dimensions. For example, a query that involves the combination of several dimension classes, as well as the fact class, has to consider the combination of the security information of all classes. The security levels of the combination will be the most restrictive of the security levels of all classes considered in the query.

Name	Type	Description	Default Value
Classes	Set(Ocltype)	It specifies all classes of the model. This new tagged value is useful in order to navigate through all classes of the model.	Empty set
Attributes	Set(OclType)	It specifies all attributes of the class. This new tagged value is useful in order to navigate through all attributes of the model.	Empty set
Security-Levels	Levels	It specifies the interval of possible security level values that an instance of this class can receive.	The lowest level (if we consider traditional levels, should be 'Unclassified')
Security-Roles	Set(Role)	It specifies a set of user roles. Each role is the root of a subtree of the general user role hierarchy defined for the organization.	The set composed of one role that is the role hierarchy defined for the model
Security-Compartments	Set (Compartment)	It specifies a set of compartments. All instances of this class can have the same user compartments, or a subset of them.	Empty set of compartments
LogType	AccessAttempt	It specifies whether the access has to be recorded: none, all access, only frustrated accesses, or only successful accesses.	None
Involved-Classes	Set(OclType)	It specifies the classes that have to be involved in a query to be enforced in an exception.	Empty
ExceptSign	{+,-}	It specifies if an exception permits (+) or denies (-) access to instances of this class to a user or a group of users.	+
Except-Privilege	Set(Privilege)	It specifies the privileges the user can receive or remove.	Read
isTime	Boolean	It indicates whether dimension represents a time dimension or not.	False
derivationRule	String	If the attribute is derived, this tagged value represents the derivation rule.	Empty

Table 3: Tagged values

For example, we can consider the following rule, related to the correct value of the tagged values, and express it using OCL: 'The set of user roles defined for each class and attribute of the model has to be a subtree of the roles tree that has been defined for the model'.

```
context Model
inv self.classes-> forAll(c | c.Roles-> forAll( r | self.Role->includesAll(r)))
inv self.classes-> forAll(c | c.attributes-> forAll(a | a.Roles-> forAll (r |
self.Role-> includesAll(r))))
```

2.6 OCL Extension

We will need some syntactic definitions that are not considered in standard OCL. Besides Set, OrderedSet, Bag and Sequence, we will need the *Tree* type. *Tree* type will be defined as a collection containing a root and a tree sequence. This type will be necessary to represent the user roles hierarchy. Consequently, the tree type will be able to use the operations of this collection defined by OCL and also the two new operations that are described below:

- Root:This will indicate the tree root.
- Subtree(n): This will indicate the n subtree (starting from the left side) of the sequence of subtrees of a tree.

Trees can be described using complex OCL structures. However, we consider that there is a simpler representational way to define a new type of data collection. The new data type *tree* will not be used for modelling but it will be necessary later, during the implementation of an automated tool that allows us to check OCL sentences.

This profile provides us with a series of aspects that will facilitate the use of our OCL extension. For example, it will be possible:

- To navigate, using the tagged values, in an intuitive way. This is possible due to the fact that tagged values are considered as attributes.
- To establish constraints by using UserProfile stereotype attributes. In this way, we will not only be able to refer to a contextual instance (writing "Self" first) but also to a contextual user (writing "UserProfile" first) thus limiting information depending on the characteristics of the user that is requesting that information.
- To model dynamic constraints, using security rules, authorization rules and audit rules. The context keyword will introduce the context of the expression, and the keywords secRule, auditRule and authRule denote, respectively, the stereotype «securityRule», «AuditRule», and «AuthorizationRule» of the constraint.

3. AN EXAMPLE APPLYING OUR PROFILE

We have considered a small-scale example in order to focus our attention on security specifications. Our SecureModel, named 'Hospital' is based on a typical health-care system. Given SECDW profile, Figure 4 shows us how this profile has been applied to the package 'Hospital'. Applying SECDW profile means that it is allowed, but not necessarily required, to apply the stereotypes that are defined as part of the profile.

Figure 5 shows us the secure multidimensional model *Hospital* whose patient admission is composed of a fact class named *Admission*, dimension classes called *Diagnosis, Patient* and *Time*, and base classes named *Diagnosis_group* of Patient Dimension. Additionally, in this modelling, an additional class called *UserProfile* is considered (stereotype *UserProfile*), that will contain

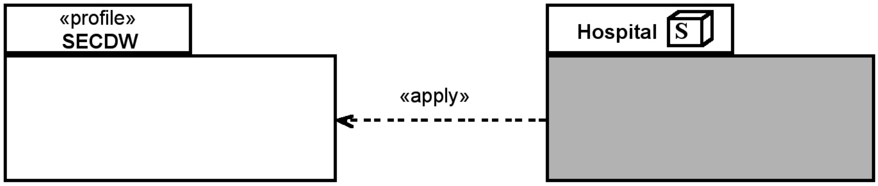


Figure 4: SECDW profile applied to a Hospital package

information of all users entitled to access to this multidimensional model (it will be possible to be use this class as a contextual user in the specification of our constraints with OCL).

We have used the following security levels: *Confidential*, *Secret* and *topSecret*. User roles *Health* (including *Doctor* and *Nurse* subroles) and *NonHealth* (including *Maintenance* and *Administrative* subroles) have been defined. The root of this hierarchical roles tree is *HospitalEmployee*. In this example, we have not considered organizational compartments.

In Figure 5, we can see that, in our model, we use the classes stereotypes inherited from the proposal stated in Luján-Mora, Trujillo *et al* (2002), into which we have added security aspects (*secureFact*, *secureDimension*, *secureBase* representing them with the same icons but adding to them a letter "S", indicating that it is a secure class). At the same time, all our constraints (*AuditRule*, *AuthorizationRule* and *SecurityRule*) will be modelled using UML notes. The number of each numbered paragraph corresponds to the number of each note in Figure 5.

1. The security level of each instance of *Admission* is defined by a security constraint specified in the model. If the value of the *description* attribute of the *Diagnosis_group* to which *diagnosis*

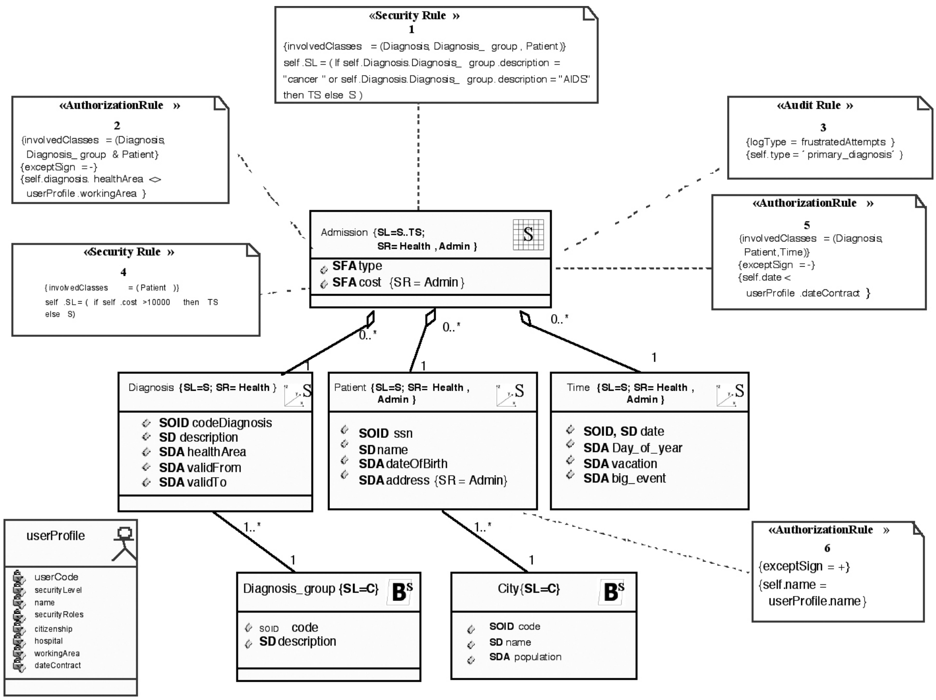


Figure 5: Example of secure multidimensional modelling

belongs is *cancer* or *AIDS*, the security level –tagged value *SL*- of this admission will be *top secret*, otherwise *secret*. This constraint is only applied if the user makes a query whose information comes from *Diagnosis* dimension or *Diagnosis_group* base classes, together with *Patient* dimension –tagged value *involvedClasses*-. Therefore, a user who has *secret* security level could obtain the number of patients with *cancer* for each city, but never if information of *Patient* dimension appears in the query.

2. For confidentiality reasons, we could deny access to admission information to users whose working area is different than the area of a particular admission instance. This is specified by another exception in *Admission* fact class, considering a condition and the tagged values *involvedClasses*, *exceptSign*.
3. The tagged value *logType* has been defined for *Admission* class, specifying the value *frustratedAttempts*. This stereotype specifies that the system has to record, for future audit, the situation in which a user tries to access information whose type is ‘primary diagnosis’ of this fact class, and so where the system denies it because of lack of permission.
4. The security level –tagged value *SL*- of each instance of *Admission* can also depend on the value of *cost* attribute, which indicates the price of the admission service. In this case, the constraint is only applicable to queries that contain information of the *Patient* dimension –tagged value *involvedClasses*-.
5. Users can be denied access to data of patients who have been treated before the date of initial contract of the staff in the health area. This stereotype is specified with an exception in the *Admission* class, considering a condition and *InvolvedClasses* and *ExceptSign* tagged values.
6. Patients could be special users of the system. In this case, it could be possible that patients access their own information as patients (for instance, for querying their personal data). This constraint is specified by using the *exceptSign* tagged value in the *Patient* class.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a UML 2.0/OCL profile that allows us to represent the main security aspects in the conceptual modelling of data warehouses. This extension contains the necessary stereotypes, tagged values and constraints for a complete and powerful secure MD modelling. 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 modelling such as facts, dimensions and classification hierarchies. We have used the OCL to specify the constraints attached to these new defined elements, thereby avoiding an arbitrary use of these.

Taking into account that data warehouses are used for discovering crucial business information in the strategic decision-making process, this proposal provides as with interesting advances in improving security in decision-support systems, as well as protection of sensitive information, which these systems generally manage.

Our work for the immediate future consists of developing an automated tool that allows us not only to model data warehouses in a secure way, using our profile, but also to translate as well as validate all our OCL sentences specified in the modelling. Furthermore, our proposal will be tested in a real environment in order to acquire empirical experience, and to obtain results of its efficiency.

ACKNOWLEDGEMENTS

This research is part of the RETISTIC (TIC2002-12487-E), MESSENGER (PCC-03-003-1) and the DIMENSIONS (PBC-05-012-2) projects, supported by the Dirección General de Investigación of the Ministerio de Ciencia y Tecnología, the network VII-J.RITOS2, financed by CYTED and the METASIGN project (TIN2004-00799), supported by the CICYT.

REFERENCES

ABELLÓ, A., SAMOS, J. and SALTOR, F. (2002): YAM2 (Yet another multidimensional model): An extension of UML. International Database Engineering & Applications Symposium (IDEAS 2002). Edmonton, Canada. *IEEE Computer Society*: 172-181.

CHUNG, L., NIXON, B., YU, E. and MYLOPOULOS, J. (2000): Non-functional requirements in software engineering. Boston/Dordrecht/London. Kluwer Academic Publishers.

CONALLEN, J. (2000): Building web applications with UML. Addison-Wesley.

DEVANBU, P. and STUBBLEBINE, S. (2000): Software engineering for security: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*, Limerick, Ireland, ACM Press.

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

FERRARI, E. and THURAISINGHAM, B. (2000): Secure database systems. Advanced databases: Technology design. PIATTINI, M. and DÍAZ, O. London. Artech House.

FUENTES, L. and VALLECILLO, A. (2004): An Introduction to UML Profiles. *UPGRADE* 2(2): 6-13.

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

HUSEMANN, B., LECHTENBORGER, J. and VOSSEN, G. (2000): Conceptual data warehouse design. *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses (DMDW'2000)*. Stockholm, Sweden. Technical University of Aachen (RWTH). 28: 3-9.

INMON, H. (2002): Building the data warehouse. USA. John Wiley & Sons.

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

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

LUJÁN-MORA, S., TRUJILLO, J. and SONG, I. Y. (2002): Extending the UML for multidimensional modeling. *5th International Conference on the Unified Modeling Language (UML 2002)*, Dresden, Germany, Springer-Verlag. LNCS 2460.

OMG (2003): UML 2.0 Infraestructure Specification, OMG Document pct/03-09-5. http://www.uml.org

PRIEBE, T. and PERNUL, G. (2000): Towards OLAP security design – Survey and research Issues. *3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00)*, Washington DC, USA.

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

SAPIA, C., BLASCHKA, M., HÖFLING, G. and DINTER, B. (1998): Extending the E/R model for the multidimensional Paradigm. *1st International Workshop on Data Warehouse and Data Mining (DWDM'98)*, Singapore, Springer-Verlag LNCS 1552.

TRUJILLO, J., PALOMAR, M., GÓMEZ, J. and SONG, I. Y. (2001): Designing Data Warehouses with OO Conceptual Models. *IEEE Computer*, special issue on Data Warehouses(34): 66-75.

TRYFONA, N., BUSBORG, F. and CHRISTIANSEN, J. (1999): starER: A conceptual model for data warehouse design. *ACM 2nd International Workshop on Data Warehousing and OLAP (DOLAP'99)*, Missouri, USA, ACM.

WARMER, J. and KLEPPE, A. (2003): The object constraint language. Getting your models ready for MDA. Second Edition. Addison Wesley.

BIOGRAPHICAL NOTES

Rodolfo Villarroel

Rodolfo Villarroel has an MSc in Computer Science from the Universidad Técnica Federico Santa María (Chile), and is currently a PhD student at the Escuela Superior de Informática of the University of Castilla-La Mancha in Ciudad Real (Spain). Assistant Professor at the Computer Science Department of the Universidad Católica del Maule (Chile), his research activity is in the field of security in data warehouses and information systems, and of software process improvement. Author of several papers on data warehouse security and improvement of software configuration management process, Villarroel belongs to the Chilean Computer Science Society (SCCC) *and the Software Process Improvement Network (SPIN-Chile). His e-mail is rvillarr@spock.ucm.cl*

Eduardo Fernández-Medina holds a PhD and an MSc in Computer Science. He is Assistant Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha at Ciudad Real (Spain), his research activity being in the field of security in databases, datawarehouses, web services and information systems, and also in security metrics. Fernández-Medina is co-editor of several books and chapter books on these subjects, and has several dozen papers in national and international conferences. He is a member of the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha. He belongs to various professional and research associations (ATI, AEC, ISO, IFIP WG11.3 etc.). Eduardo's e-mail is eduardo.fdezmedina@uclm.es

Eduardo Fernández-Medina

Juan Trujillo is an associated 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 modelling, conceptual design of data warehouses, multidimensional databases, OLAP, as well as object-oriented analysis and design with UML. With papers published in international conferences and journals such as ER, UML, ADBIS, CAiSE, WAIM, Journal of Database Management (JDM) and IEEE Computer, Trujillo has served as Program Committee member of several workshops and conferences such as ER, DOLAP, DSS, and SCI and has also spent some time as a reviewer of several journals such as JDM, KAIS, ISOFT and JODS. His e-mail is jtrujillo@dlsi.ua.es

Juan Trujillo

Mario Piattini has an MSc and a PhD in Computer Science from the Politechnical University of Madrid. He is a Certified Information System Auditor from the ISACA (Information System Audit and Control Association). Full Professor at the Escuela Superior de Informática of the Castilla-La Mancha University (Spain) and author of several books and papers on databases, software engineering and information systems, Piattini 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 and software maintenance. His e-mail address is Mario.Piattini@uclm.es

Mario Piattini