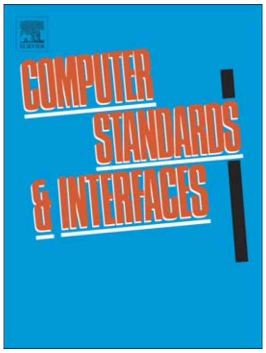


AUML2.0 Profile to define Security Requirements for DataWarehouses

Juan Trujillo, Emilio Soler, Eduardo Fernández-Medina, Mario Piattini

PII: S0920-5489(08)00147-5
DOI: doi: 10.1016/j.csi.2008.09.040
Reference: CSI 2641

To appear in: *Computer Standards & Interfaces*

Received date: 19 February 2008
Revised date: 6 August 2008
Accepted date: 28 September 2008

Please cite this article as: Juan Trujillo, Emilio Soler, Eduardo Fernández-Medina, Mario Piattini, AUML2.0 Profile to define Security Requirements for DataWarehouses, *Computer Standards & Interfaces* (2008), doi: 10.1016/j.csi.2008.09.040

A UML 2.0 Profile to define Security Requirements for Data Warehouses

Juan Trujillo [a,*], Emilio Soler [b], Eduardo Fernández-Medina [c], Mario Piattini [c]

[a] *LUCENTIA Research Group, Department of Language and Information Systems. University of Alicante. Apto correos 99. E-03080. Alicante, Spain*

[b] *Department of Computer Science. University of Matanzas. Autopista de Varadero km 3. Matanzas, Cuba*

[c] *ALARCOS Research Group, Information Systems and Technologies Department, UCLM Research and Development Institute, University of Castilla-La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, Spain*

Abstract

Many Data Warehouses (DWs) fail to provide the appropriate information because the users' requirements are not correctly modeled. In addition, the security requirements are considered in the final implementation, and do not take the users' necessities into consideration. However, as DWs store confidential and sensitive information, it is crucial to take security measures into account from early DWs design phases, and to enforce them. This paper proposes a profile which uses the Unified Modeling Language (UML) extensibility mechanisms. This profile allows us to define security requirements for DWs at the business level, taking into account the information requirements modeled with a previous profile. Our proposal is aligned with Model Driven Architecture (MDA), thus permitting the transformation of security requirements throughout the entire DWs life cycle. Finally, in order to show the benefits of our profile, we develop a case study related to the management of a pharmacy consortium business.

Key words: data warehouses; security requirements; UML profile

1. Introduction

In recent years many methods concerning how Data Warehouses (DWs) should be designed have been proposed [1]. Nevertheless, it is accepted that the development of DWs must be based on conceptual multidimensional (MD) modeling [14], which structures the information into facts and dimensions. Conceptual MD modeling produces a specification, which must be consistent with data sources and user needs, i.e, the needs of Decision makers. Decision makers' needs are usually captured during the requirements analysis phase of the DWs design by means of functional requirements.

Whereas many methods for functional requirements have been proposed, no appropriate methods for systematic definition, design, and development of non-functional requirements (NFRs) exist [7]. Security requirements such as NFRs are especially significant as they are not given the appropriate consideration during the early phase of the development process [21].

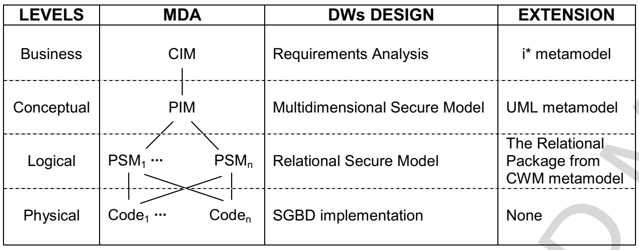
Within the context of DWs projects, the security aspects are normally implemented in the final phases of design [28]. However, security concerns must inform every phase of software development, from requirements engineering to design, implementation, testing and deployment [6].

The Model Driven Architecture (MDA) is an Object Management Group (OMG) standard from the which addresses the complete life cycle of the development of applications by using models in software development. MDA relies on the idea of separating the specification of a system's operations from the details of its platform [22]. MDA proposes several models at different levels: The Computation Independent Model (CIM), the Platform Independent Model (PIM), the Platform Specific Model (PSM) and Code. In the MDA framework the standard for defining transformations between models is the Query/Views/Transformation (QVT) [23]).

Fig. 1 shows the extensions proposed in order to accommodate DW development to the MDA approach. The CIM is based on an extension of the $i*$ framework [39] proposed in [19], which deals solely with information requirements (i.e, functional requirements) for DWs design at the

* Corresponding author. Phone: 34-965903772; fax: +34-965909326
Email addresses: jtrujillo@dlsi.ua.es (Juan Trujillo), emilio.soler@umcc.cu (Emilio Soler), eduardo.fdezmedina@uclm.es (Eduardo Fernández-Medina), mario.piattini@uclm.es (Mario Piattini).

business level. The PIM corresponds with an extension of the Unified Modeling Language (UML) profile presented in [36], which reuses the results of [18]. This profile allows us to consider the main properties of secure MD modeling at the conceptual level. The PSM corresponds with an extension of the Common Warehouse Metamodel (CWM) at the logical level [35], and Code with implementation at the physical level, i.e. with a Database Management System (DBMS). We have recently with which metamodels [36] and [35] in order to define QVT relations to transform PIM into PSM for Secure DWs design [34]. This set of QVT relations have been validated through the development of a case study [33]. As Fig. 1 shows, we cannot model security requirements at the Business level alone.



LEVELS	MDA	DWs DESIGN	EXTENSION
Business	CIM	Requirements Analysis	i* metamodel
Conceptual	PIM	Multidimensional Secure Model	UML metamodel
Logical	PSM_1 ··· PSM_n	Relational Secure Model	The Relational Package from CWM metamodel
Physical	$Code_1$ ··· $Code_n$	SGBD implementation	None

Fig. 1. Aligning the design of secure DWs with MDA

In a previous work [19] we have employed $i*$ modeling and the MDA framework in order to model goals and information requirements for DWs, i.e, functional requirements. This proposal defines a UML profile based on $i*$ modeling for the DWs design, which allows us to formalize $i*$ diagrams in order to model a CIM. However, the approach focuses solely on information requirements, i.e. it does not include security as a special non-functional requirements type. Therefore, in this paper we propose a new UML profile that reuses the above mentioned profile [19], whilst adding security requirements for DWs at the business level. Our proposal is coupled with both the MDA framework [20] and with our previously developed works, thus allowing a total integration of methods with which to develop a complete methodology to build secure DWs. The main benefits of our proposal are: (i) The adaptation of the $i*$ framework [39] to define and integrate both security and information as functional requirements into a secure CIM for DWs, (ii) a guarantee of consistency since the profile avoids the situation of having different definitions and properties for the same concept throughout a model, (iii) an attempt to create a proposal which is more understandable to both DW designers and final users.

The following section describes some of the most relevant proposals with regard to requirements engineering and security modeling in DWs. Section 3 explains previous UML profiles based on $i*$ framework for the modeling of functional requirements for DWs. Our extension is presented in Section 4, which first shows the models that serve as a basis for our profile and goes on to present the extended profile. The benefits of our proposal are shown in Section 5 through the use of a case study. Finally, Section 6 draws the main conclusions and outlines our immediate future work.

2. Related Work

This section on related work begins with the main approaches towards requirements engineering. We then study work related to the main approaches dealing with security in DWs.

2.1. *Requirements engineering*

The main approaches towards requirements engineering are grouped into two proposals: Goal-based and Scenarios-based requirements analysis, respectively. Goal-based requirements analysis comprises the following approaches: GBRAM (Goal-Based Requirements Analysis Method) [3], KAOS (Knowledge Acquisition in AutOmated Specification), $i*$ framework [39], Tropos methodology and Non-Functional Requirements (NFRs) [5].

GBRAM allows us to identify and to refine goals into operational requirements. This approach is supported by a set of heuristic rules and guidelines. It is essentially a document analysis technique. The proposal has been applied in the field of security and privacy. However, it does not address elicitation techniques since goals are not sufficient in the definition of access control models. KAOS is a goal based requirement acquisition and elaboration method. It offers an expressive conceptual modeling language to assist requirement engineers in the specification of requirements driven by high-level goals. KAOS has been extended to model security and privacy goals and anti-goals [16] but by focusing on the system rather than the organizational goal. Moreover, KAOS does not permit the modeling of complex access control policies. It is thus not appropriate to our needs.

The $i*$ framework is employed to model and itemize organizational contexts and rationales. It comprises two components: Strategic Dependency (SD) and Strategic Rationale (SR). The framework has been applied to support access control analysis [17]. However, it provides no guidance as to how roles and privileges are identified, from where they originate, or how privileges are assigned to these roles. We believe that it can be formalized with UML and extended for our purposes. Tropos methodology is an agent-oriented approach derived from the $i*$ framework. This methodology covers the software development phases, i.e. early and late requirements, and architectural and detailed design. A secure Tropos is introduced in [11]. The proposal is a formal extension of Tropos for the modeling and analysis of functional and security requirements. Secure Tropos deals with security issues in general but it does not take the possibility of defining an access control model for information systems into consideration .

NFRs is a goal-based requirement analysis method which systematically addresses non functional requirements in the early stages of system development. The *i** framework shares many concepts with the NFR framework. NFRs treats security goals as an overall business requirement, whilst *i** and Tropos [21] consider security goals within the context of the organizational actors who aim to achieve them. We believe that it would be extremely difficult to use NFRs for our purposes.

Scenario-based requirements analysis describes the software behavior of a system [37]. Scenarios are useful in the elicitation of possible occurrences and the corresponding assumptions, obstacles, and post-conditions. However, we believe that goals/softgoals usually refer to the organization or system's high-level objectives.

None of the approaches mentioned for requirements engineering allows us to define and elicit security requirements for DWs. Moreover, some must be adapted to our purposes. However, we wish to reuse proposal [19], which is based on the i* framework. Hence, we have based our proposal on Goal-Based Requirements Analysis, and specifically on the Goal-oriented Requirement Language (GRL) [12], which is based on the NFRs and *i** frameworks. We have adopted the GRL to perform our UML profile owing to its expressiveness and integration with NFRs.

2.2. *Security in Data Warehouses*

In recent years several initiatives to include security in DWs design have appeared. [13] describes a prototype model for DWs security based on metadata, whose main goal is to reduce user queries to only those data which are to be seen by that user. However, this does not allow us to specify complex restrictions of confidentiality such as deny-allow the access to a special user combining groups and security constraints. Rosenthal and Sciore [29] extend SQL grants and create a mechanism of inferences through which to establish the security of DWs, which gives permission to access the tables and views of the system. Another attempt is the architecture of both Federated Information Systems (FIS) and DWs which preserve MultiLevel security integration between FIS and DWs [30]. The authorization of the DWs scheme built takes into account the security policy of the federation itself. [15] defines a model based on the Discretionary Access Model (DAC) which proposes a security concept for OLAP, a role based security model for DWs. According to these security rules, a derived data cube is defined for each role. [38] shows how access privileges for DWs and OLAP can be expressed more intuitively than SQL's grant statements. This access control model focuses specifically on expressiveness and usability. These approaches [13,29,30,15,38] are attractive but only focus on practical issues such as acquisition, storage and access control on the OLAP side. None of them examine the representation of security at the Requirement Analysis stage.

On the other hand, more elaborate initiatives which propose authorization models for DWs design also exist. For example, Priebe and Pernul [26] propose a security design methodology similar to the classical database methodology (requirement analysis, conceptual, logical, and physical design) which covers requirements and concrete implementations in commercial systems. The same authors extend the ADAPTed UML (which uses ADAPT symbols as UML stereotypes) model for the aforementioned conceptual phase [27], and specify a methodology and an MD security constraint language for the conceptual modeling of OLAP security. These approaches [26,27] offer security models at the conceptual level by means of security constraints, but basically deal with OLAP operations. In short, these works implement the security rules considered in their conceptual approach to commercial database systems. We, however, base our approach on the work of [8] in which the authors propose a novel model for security and audit that can be used in the entire DWs life cycle, from requirement analysis to final implementation. Therefore, in this paper, we propose a profile which allows us to define security requirements for DWs at the requirement analysis stage, i.e, business level.

To this end, we argue that there is still room for investigation related to the definition and modeling of security requirements in the DWs development process.

3. Modeling Information Requirements for DWs

A requirement analysis stage for DWs aims to obtain our informational requirements from decision makers [25]. In this section we resume a previous proposal related to functional requirements for DWs and we explain how to adapt the i**f*ramework in order to elicit them.

In a previous work [19] the *i** framework [39] is adapted to the modeling of goals and information requirements for DWs. The *i** framework is an approach for modeling and reasoning about organizational environments and their information systems [39]. The main elements of the *i** framework are: *Actors*, *Goals*, *Softgoals*, *Tasks*, *Resources* *Means-Ends* links, *Decomposition* links and *Contribution* links [17]. An *Actor* (○) is used to refer generically to any unit to which intentional dependencies can be ascribed, e.g. "*inventory manager*". *Goals* (⬭) answer the question "*what does the actor want to achieve*", e.g. "*examine inventory levels* and "*study inventory movements*". *Tasks* (⬡) are used to represent the specific procedure to be performer by actors, e.g. "*analyze quantity to hand*". *Resource* (▭) is a physical or informational entity, about which the main concern is its availability (for example, "*provide information about inventory*". The *Tasks* are connected to the *Goals* through *means-ends* links (−▷). A *Goal* is satisfied if any of its *Tasks* is satisfied. A *Task* may be detailed into *Goals*, *Subtasks*, *Resources* and *Softgoals* through the *Decomposition* links (—+). High-level abstract *Goals* (*Softgoals*) are reduced into the lower-level, more specific *Goals* (*Sofgoals*) or operationalized in terms of *Task* through *Contribution* link (—>—).

Modeling by using the *i** framework comprises two main components. The *Strategic Dependency* (SD) model is a network of intentional dependencies (*Dependency* link, -D-). It describes the dependency relationships among various actors in an organizational context. The *Strategic Rationale* (SR) is used to describe stakeholder interests and concerns, and how they might be addressed by various configurations of systems and environments [39]. The SR model is obtained when the internal rationales of actors from the SD model are made explicit.

The adaptation of the *i** framework is based on two extensions of UML [24]: (i) a profile for *i**; and (ii) a profile which adapts *i** to the DW domain. In accordance with the UML specification [24], in Fig. 2 we show the packages which resume the elements contained in the proposal [19]. The profiles use two kinds of extending relationships: the *Extension* relationship (whose arrowhead is shown as a filled triangle) which points from stereotypes (the extending elements, labeled as `<<stereotype>>`) to metaclasses (the UML extended elements, labeled as `<<metaclass>>`), and the *Generalization* relationship (an arrowhead with a hollow triangle) between stereotypes. On the left hand side of Fig. 2 we have represented the *i** profile by means of various UML metaclasses (i.e. *Package*, *Class*, *AssociationClass*, and *Association*) and stereotypes (the *IElement*, *Argumentable*, and *IRelationship* stereotypes). These stereotypes permit the representation of SR and SD models belonging to the *i** framework.

On the right hand side of Fig. 2 we show the *i** profile for DWs, which is based on a classification of the different kind of goals that decision makers expect to fulfill with the DWs: (i) **Strategic goals** represent the highes level of abstraction. These are the main objectives of the business process)for example, "*increase sales*"); (ii) **Decision goals** represent the medium level of abstraction. They attempt to answer the question: "*how can a strategic goal be achieved?*" (for example, "*determine some kind of promotion*"); (iii) **Information goals** represent the lowest level of abstraction. They attemp to answer the question: "*how can decision goals be achieved in terms of information required*", for example, "*analyze customer purchases*" or "*examine stocks*"). The profile reuses the previous stereotype *Goal*, as we can see in Fig. 2.

For decision makers, every goal must be specified according to the classification of goals in terms of the strategic-decision-information hierarchy. Information requirements (*Requirement* as *Task* on the right hand side of Fig. 2) for decision makers are derived from information goals. The profile has added three MD elements as resources: the business process to be analyzed (*BusinessProcess* stereotype), process measures under analysis (*Measure stereotype*), and context of analysis (*Context* stereotype). These stereotypes are, therefore, derived from Resource (see right hand side of Fig. 2).

The *i** profile for DWs provides a mechanism with which to represent actors (*IActor*, ○) and their goals (*Goal*, ⬭). The information requirements of decision makers are considered as tasks (*Task*, ⬡), and the elements needed in the DW to provide such information are considered as resources (*Resource*, □). According to the kind of DWs element, these resources can be labeled as `<<BusinessProcess>>`, `<<Context>>`, or `<<Measure>>`. We furthermore model relationships such as means-end (*MeansEnd*, -▷) thus representing alternative means to fulfill goals, or tasks, i.e., the possible relationships are Goal-Goal and Goal-Task. Decomposition (*Decomposition*, —┼) represents the elements which are necessary if a task is to be performed. Additionally, the profile allows us to define aggregation relationships between context of analysis (for instance, the city context can be aggregated by the country context). In order to model these relationships, the we have used the (shared) aggregation relationship of UML (*Association* UML metaclass, represented as —◇).

4. A UML Profile to adapt the *i** framework to the modeling of security requirements

In this section we describe the UML 2.0 profile based on *i** modeling which is used to define security requirements for DWs. The extended profile reuses the previous extension explained in Section 3, which allows us to obtain functional requirements for DWs by using the *i** framework. The aim of the UML profile is to translate the concepts of the i* framework by extending its semantics and notation. This profile allows us to elicit security requirements in order to define an Access Control and Audit (ACA) model for DWs at the business level.

4.1. *Modeling Security*

Security requirements are requirements which are associated with the protection of valuable assets in the system. This protection requires that every access to a system and its resources be controlled and that all and only authorized access can take place, and is thus called Access Control (AC) [31]. The development of an access control system is usually carried out by access control policies (ACP), access control models and an access control mechanism [31], which constitutes different levels of abstraction. ACP are security requirements which describe how access is managed, what information can be accessed by whom, and under what conditions that information can be accessed [10], i.e., it defines high-level rules. Access control models provide a formal representation of the access control security policy, whereas the access control mechanism defines the low-level (software and hardware) functions that implement the controls imposed by the policies and are formally stated in the model [31].

ACP are grouped into three main classes: Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC) [31]. DAC policies

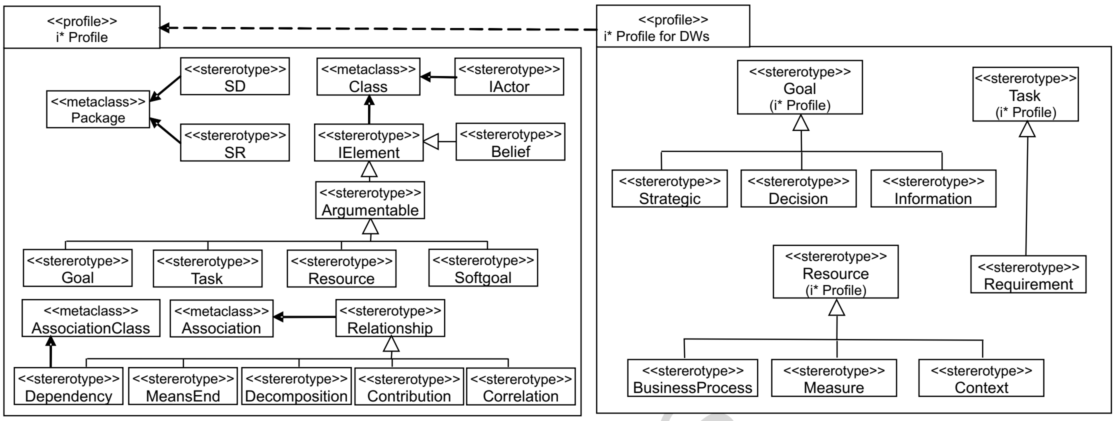

<<profile>>
i* Profile
<<profile>>
i* Profile for DWs
<<stereotype>> SD
<<metaclass>> Package
<<stereotype>> SR
<<metaclass>> Class
<<stereotype>> IActor
<<stereotype>> IElement
<<stereotype>> Belief
<<stereotype>> Argumentable
<<stereotype>> Goal
<<stereotype>> Task
<<stereotype>> Resource
<<stereotype>> Softgoal
<<metaclass>> AssociationClass
<<metaclass>> Association
<<stereotype>> Relationship
<<stereotype>> Dependency
<<stereotype>> MeansEnd
<<stereotype>> Decomposition
<<stereotype>> Contribution
<<stereotype>> Correlation
<<stereotype>> Goal (i* Profile)
<<stereotype>> Task (i* Profile)
<<stereotype>> Strategic
<<stereotype>> Decision
<<stereotype>> Information
<<stereotype>> Resource (i* Profile)
<<stereotype>> Requirement
<<stereotype>> BusinessProcess
<<stereotype>> Measure
<<stereotype>> Context

Fig. 2. UML profile for $i*$ in the context of DWs.

control access based on the identify of the requestor and on access rules which state what requestors are allowed to do. MAC policies control access based on mandated regulations determined by a central authority. RBAC policies control access depending on the roles that users have within the system and on rules stating what accesses are permitted to users in given roles.

In previous work we have defined an Access Control and Audit (ACA) model for DWs by specifying security rules at the conceptual level [8]. This approach is based on access control to guarantee confidentiality and audit, which are essential components for the DWs design. However, security includes other characteristics such as authentication, integrity, repudiation and availability, which constitute mechanisms that are design-independent and rely to a greater extent on company policies. They are not therefore taken into account by the ACA model. The ACA model allows us to represent the confidentiality and audit measures of DWs by classifying subjects and objects in the system [1] . The classification uses access classes on the basis of three different but compatible ways of classifying users: by their *security level*, by the *role*, and by the *compartments* to which they belong. The access class is one element of a partially ordered set of classes, in which an access class c_1 dominates an access class c_2 *if and only if* the security level of c_1 is greater than or equal to that of c_2, the compartments of c_1 include those of c_2, and at least one of the user roles of c_1 (or one of its ancestors) is defined for c_2 [8]. The following classes are described in order to permit us specify the ACA model:

Security user roles are used by a company to organize users into a hierarchical role structure, according to the responsibilities of each type of work. Each user can play more than one role.

Security levels indicate the clearance level of the user. This is usually an element of a hierarchically ordered set, such as Top Secret (TS), Secret (S), Confidential (C), and Unclassified (U), where $TS > S > C > U$.

Security user compartments are also used by an organization to classify users into a set of horizontal compartments or groups, such as geographical location, area of work, etc. Each user can belong to one or more compartments.

As was previously explained, the ACA model uses the classification for users and objects based levels, roles and compartments. Therefore, the ACA model combines the MAC and RBAC models. MAC models have been widely studied, and many vulnerabilities have been detected, such as their lack of flexibility, their polyinstantiation [32], etc. Nevertheless, most of these problems arise from the necessity of taking into consideration both read and write operations in the system. Fortunately, we consider that the sole operation that will be used by the final users in decision-support systems is read, so the MAC model is absolutely appropriate. In contrast to the MAC model, the RBAC model represents a promising direction and a useful paradigm for many commercial and governmental organizations [31].

On the other hand, the $i*$ modeling framework has been employed in the modeling of Access Control Policies (ACP), but focus on Role Based Access Control (RBAC) [17]. As we explained in Section 2.1, the proposal assumes that roles and privileges have been previously derived. This issue makes it very difficult to elicit and define security requirements in the $i*$ framework. Hence, we need to adapt the $i*$ framework by means of an extension of UML if we are to represent security requirements for DWs. We base our proposal on the ACA model, which is a special case of ACP.

4.2. *An overview of our profile*

We take the SR model proposed by the $i*$ framework. Starting from the SR models we develop three SR models: (1) *GOModel* which contains the stereotypes from the $i*$ profile for DWs [19], i.e. a CIM for DWs. (2) *SOModel* which is made up of the stereotypes defined by our i* profile for secure DWs, i.e. stereotypes for eliciting security

[1] The ACA model also allows us to define Sensitive Information Assignment Rules (SIARs) in order to specify the security information of each DWs element, rules for representing authorization rules (AURs), which work together with SIARs, and rules which allow us to specify audit requirements (ARs).

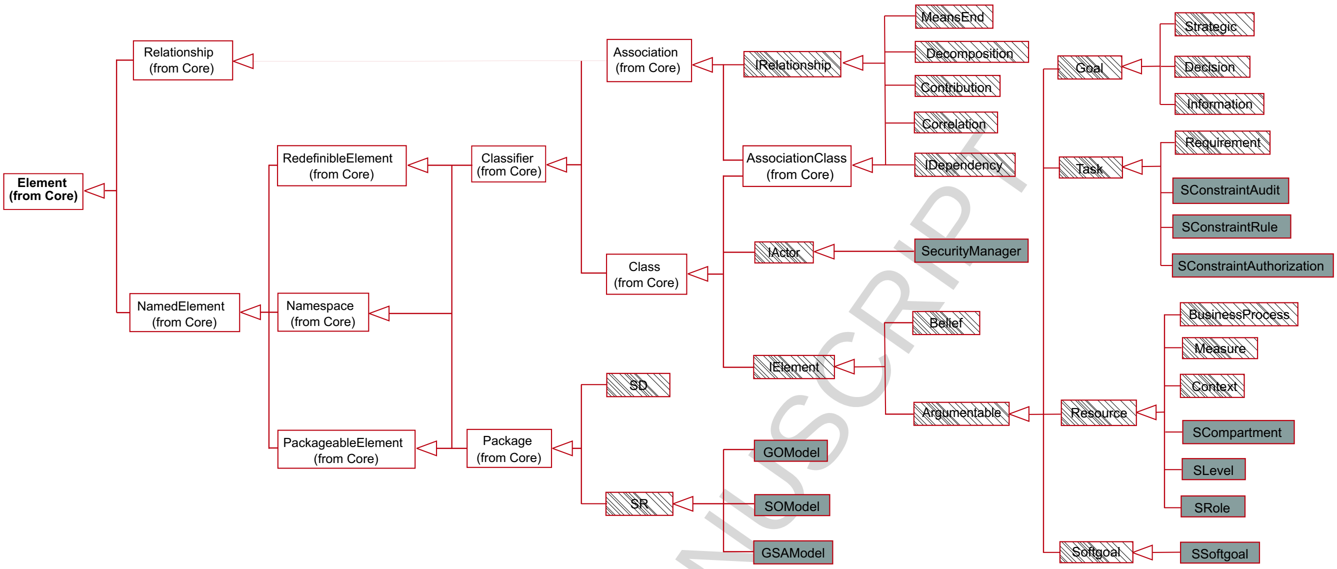


Fig. 4. Extension of the UML with stereotypes

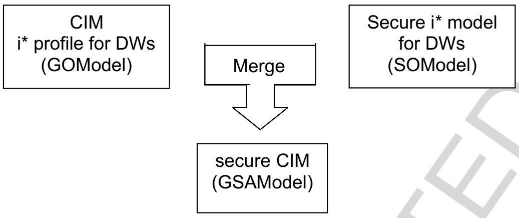


Fig. 3. Defining a secure CIM for DWs

requirements for DWs. (3) *GSAModel* which merges the above models, i.e. a secure CIM for secure DWs which constitutes an access control model for DWs at the business level. Fig. 3 portrays how the aforementioned models are merged in order to define a secure CIM.

If we are to model security requirements for DWs, then information requirements are necessary (functional requirements). These are modeled in our profile by the *GOModel* (Goal-organization-model) which creates a refinement process with the strategic, decision and information goals. This model is supported by the *i** profile for DWs represented on the right hand side of Fig. 2 in Section 3.

The security requirements for DWs are defined in the *SOModel* (Softgoal-organization-model) through the softgoals refinement process which is carried out by a special actor called the SecurityManager (the person in charge of the security within the organization). Levels, roles and compartments are discovered during this process. Therefore, new stereotypes are needed to represent security information at the business level, which are represented as resources associated with the softgoals refined.

Once the *GOModel* and *SOModel* models have been defined we have information requirements and security requirements. Hence, it is necessary to establish a relationship between them, which is achieved by the *GSAModel* (Goal-Softgoal-analysis-model). This model is used to associate MD elements from *GOModel* (*BusinessProcess*, *Context* and *Measures* resources) with the security information represented in the *SOModel* (levels, roles and compartment). We can also establish constraints with which to fulfill the softgoals refined, which are associated with resources from *GOmodel* through the dependency link from *i**. According to the ACA model we need three kinds of constraint rules (security information, authorization and audit). Unfortunately these rules cannot be defined in detail due to the granularity at this level. The *GSAModel* will represent our secure CIM for DWs design. According to Fig. 1 this model must be transformed throughout the DWs life cycle, from requirement analysis to the physical level.

4.3. *UML extension mechanism*

The UML Superstructure specification [24] defines two extension mechanisms for UML 2.0: (1) The profiles mechanism, which is not a first-class extension mechanism (i.e., it does not permit the modification of existing metamodels) and (2) the first-class extensibility which is handled through MOF, in which there are no restrictions on what you are allowed to do with a metamodel. It is possible to add and remove metaclasses and relationships as is necessary. Our proposal is based on a UML profile, which consists of *Stereotypes, Constraints* and *Tagged Values*. *Stereotypes* are specific metaclasses, which are either represented by a string between a pair of guillemets (<< >>) or are rendered as a new icon. *Tagged Values* are standard meta-attributes, and profiles are specific kinds of packages [24]. *Constraints* are applied to stereotypes in order to indicate restrictions.

Our Profile is defined following the proposal of [4], which states that "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". Taking this proposal into account, we have employed the schema: Description, Prerequisite extensions, Stereotypes, Tagged-

Table 1
Package stereotypes

Stereotypes	
Name	*GOModel*
BaseClass	Package (from SR)
Description	This Package contains the i* model for DWs (i.e, functional requirements)
Icon	Fig. 6 a)
Constraints	Context UML::Kernel::Package inv: self.isStereotyped("*GOModel*") implies - A *GOModel* can only contain *Strategic*, *Decision*, *Information*, *Requirement*, *BusinessProcess*, *Measure* and *Context* classes: self.contents-> select (co| co.oclIsKindOf(class))-> forAll (g| g.oclIsTypeOf(Strategic) or g.oclIsTypeOf(*Decision*) or g.oclIsKindOf(*Information*) or g.oclIsTypeOf(*Requirement*) or g.oclIsTypeOf(*BusinessProcess*) or g.oclIsTypeOf(Measure) or g.oclIsTypeOf(Context)) - It is not possible to create an *IDependency* between *GOModels* (only to *SOModel*): self.clientDependency-> forAll (d| d.supplier -> forAll (me| me.oclIsTypeOf(*SOModel*)))
Tagged values	None
Name	*SOModel*
BaseClass	Package (from SR)
Description	This Package class contains the secure i* model for DWs
Icon	Fig. 6 a)
Constraints	Context UML::Kernel::Package inv: self.isStereotyped("SOModel") implies - A *SModel* can only contain *SSoftgoals*, *SRoles*, *SLevels*, *SCompartment*, *SConstraintRule*, *SConstraintAudit* and *SConstraintAuthorization* classes: self.contents-> select (co| co.oclIsKindOf (class))->forAll (m| m.oclIsTypeOf(*SSoftgoal*) or m.oclIsTypeOf(*SRole*) or m.oclIsTypeOf (*SLevel*) or m.oclIsTypeOf(*SCompartment*) or m.oclIsTypeOf(*SConstraintRule*) or m.oclIsTypeOf(*SConstraintAudit*) or m.oclIsTypeOf(*SConstraintAuthorization*) - A *SOModel* can only be associated by means of *IDependency* with *GOModel*: self.allOppositeAssociationEnds-> forAll (ao| ao.participant.oclIsKindOf(*GOModel*) and ao.oclIsKindOf (*IDependency*) - The actor of a *SOModel* can only be a *SecurityManager*: self.actor.isStereotyped(*SecurityManager*)
Tagged values	None
Name	*GSAModel*
BaseClass	Package (from SR)
Description	This Package contains the *SOModel*, *GOModel* packages and their relationship
Icon	None
Constraints	None
Constraints	UML::Kernel::Package inv: self.isStereotyped("*GSAModel*") implies A *GSAModel* can only contain *GOModels* and *SOModels* packages: self.contents-> select (po| oclIsKindOf(Package)) -> forAll (me| me.oclIsTypeOf(*GOModel*) or me.oclIsTypeOf(SOModel)) A *GSAModel* can only contain one *SOModel* (and only one): self.ownedElement-> select(me| me.oclIsTypeOf(*SOModel*))->size()= 1
Tagged values	None

Table 2
Class stereotypes

Stereotypes	
Name	*SecurityManager*
BaseClass	Class (from *IActor*)
Description	This class represents the person in charge of the security organization
Icon	Fig. 6 b)
Constraints	UML::InfrastructureLibrary::Core::Consruct::Class inv: self.isStereotyped("SecurityManager") implies A *SecurityManager* can only belong to *SOModel*: self.owner.oclIsTypeOf(*SOModel*)
Tagged values	None
Name	*SConstraintRule* (Analogously are treated the *SConstraintAudit* and *SConstraintAuthorization* rules)
BaseClass	Class (from *Task*)
Description	This class represents a rule (to define multilevel security policies) that contributes to fulfilling *SSoftgoals* by the Contribution link
Icon	Icon from previous element, labeled as <<SConstraintRule>>
Constraints	Context UML::InfrastructureLibrary::Core::Consruct::Class inv: self.isStereotyped("*SConstraintRule*") implies - A *SConstraintRule* must be satisfied by means of a Contribution Association, at least one *SSoftgoal*: self.allOppositeAssociationEnds-> forAll (s|s.participant.oclIsTypeOf(*SSoftgoal*)->size >=1 and s.oclIsTypeOf(*Contribution*)
Tagged values	None
Name	*SSoftgoal*
BaseClass	Class (from Softgoal)
Description	This class represents the organization's security policy
Icon	Icon from previous element, labeled as <<SSoftgoal>>
Constraints	Context UML::InfrastructureLibrary::Core::Consruct::Class inv: self.isStereotyped("*SSoftgoal*") implies - All resources associated by Decomposition link with a *SSoftgoal* must be *SCompartment*, *SRole* or *SLevel*: self.allOppositeAssociationEnd-> forAll(so| so.participant.oclIsTypeOf(SCompartment) or so.participant.oclIsTypeOf(*SLevel*) or so.participant.oclIsTypeOf(*SRole*)) and so.oclIsTypeOf(*Decomposition* - Each *SSoftgoal* must be associated by means of *IDependency* association, either to an another *SSoftgoal* or to the following resources *BusinessProcess*, *Context* or *Measure*: self.allOppositeAssociationEnd-> forAll (si| si.participant.oclIsTypeOf(*SSoftgoal*) or si.participant.oclIsTypeOf(*BusinessProcess*) or si.participant.oclIsTypeOf(*Context*) or si.participant.oclIsTypeOf(*Measure*)) and si.oclIsTypeOf(*IDependency*) - If a SSoftgoal is associated by *Decomposition* to any resource *SLevel*, *SCompartment* or *SRole*, then it must be associated by *IDependency* to at most one of the resources *BusinessProcess*, *Context* or *Measure*: self.associationEnd-> forAll(as| as.participant.oclIsType(*SLevel*) or as.participant.oclIsType(*SCompartment*) or as.participant.oclIsType(*SRole*)) and as.oclIsTypeOf(*Decomposition*) implies the existence of (ar| ar.participant.oclIsTypeOf(*BusinessProcess*) or ar.participant.oclIsTypeOf(*Context*) or ar.participant.oclIsTypeOf(*Measure*)) and ar.oclIsTypeOf(*IDependency*)
Tagged values	None

values, Well-formedness rules and Comments.

We have defined eleven stereotypes: three specialize in the SR package, one specializes in the softgoal class, one specializes in the *IActor* class, three specialize in the Resource class and three specialize in the Task class. In Fig. 4, we have represented a portion of the UML metamodel to show where our stereotypes fit. In this figure, new stereotypes are colored in dark grey, stereotypes belonging to the previous extension of the *i** framework are filled with diagonal lines and the UML classes are white.

4.3.1. *Description.*

This extension defines a profile with which to adapt the *i** framework in order to elicit security requirements for DWs. The profile is defined by eleven stereotypes (*SOModel*, *GOModel*, *GSAModel*, *SecurityManager*, *SConstraintRule*, *SConstraintAudit* *SConstraintAuthorization*, *SSoftgoal*, *SLevel*, *SCompartment* and, *SRole*) and its constraints are written in Object Constraint Language (OCL). The set of stereotypes allows us to develop a refinement process of softgoals in order to define an SR within the context of the *i** framework. The correct use of this extension is assured thanks to the definition of constraints both in natural language and in Object Constraint Language (OCL).

4.3.2. *Prerequisite extensions*

This extension reuses stereotypes defined in the *i** profile for DWs defined in [19]. The *i** profile for DWs (see right hand side of Fig. 2, Section 3), defines the stereotypes (i.e. *Strategic*, *Decision*, *Information Requirements*, *BusinessProcess*, *Measure* and *Context*) used to capture information requirements (i.e. functional requirements) for DWs. Fig. 5 depicts how our new profile reuses stereotypes from the *i** profile for DWs.

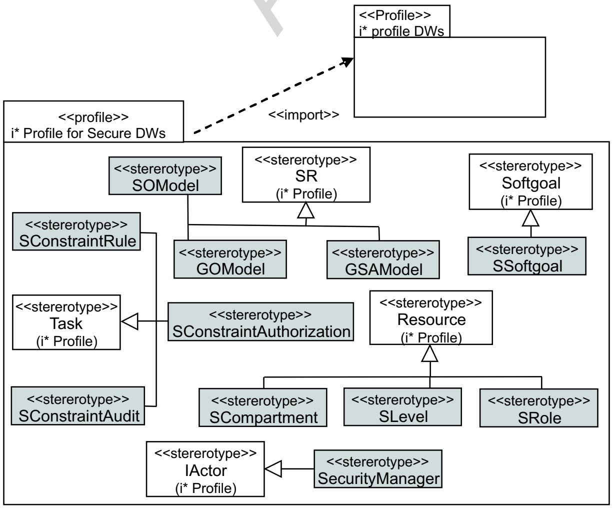


Fig. 5. Profile stereotypes with which to define security requirements

4.3.3. *Stereotypes*

We explain the stereotypes by following the structure suggested by the UML Superstructure specification [24], i.e, name, BaseClass, Description, icon, Constraint and Tagged values. We clarify that our stereotypes reuse the same icons from the *i** framework. We define the OCL operation isStereotyped(stereotypeName) which indicates whether an element is stereotyped by a string stereotypeName as follows:

```
isStereotyped (stereotypeName:String) : Boolean;
self.extension->exists (x| x.ownedEnd.type=
stereotypeName)
```

Table 3
Class stereotypes (continued)

Stereotypes	
`Name`	*SCompartment*
`BaseClass`	Class (from *Resource*)
`Description`	This class represents horizontal compartments or groups of users from the organization, such as geographical localization.
`Icon`	Icon from previous element, labeled as `<<SCompartment>>`
`Constraints`	None
`Tagged values`	None
`Name`	*SLevel*
`BaseClass`	Class (from *Resource*)
`Description`	This class represents a defined clearance level. Usually named *TopSecret* (TS), *Secret* (S), *Confidential* (C), or *Unclassified* (U).
`Icon`	Icon from previous element, labeled as `<<SLevel>>`
`Constraints`	None
`Tagged values`	None
`Name`	*SRole*
`BaseClass`	Class (from *Resource*)
`Description`	This class represents a role from the hierarchical roles defined in the organization
`Constraints`	None
`Icon`	Icon from previous element, labeled as `<<SRole>>`
`Tagged values`	None

We shall now describe each stereotype with its corresponding elements as mentioned above. Table 1 shows the stereotypes defined for the *SOModel*, *GOModel* and *GSAModel* packages (from SR) whereas, Tables 2 and 3 depict the class of stereotypes, i.e, *SecurityManager*, *SConstraintRule*, *SConstraintAudit*, *SConstraintAuthorization*, *SSoftgoal*, *SCompartment*, *SLevel* and *SRole*. The *SConstraintRule*, *SConstraintAudit*, *SConstraintAuthorization* stereotypes represent the SIARs, AURs and ARs rules from the ACA model, but they are different because they will only denote what must be refined and improved in later phases of the design. We clarify that Table 2 only shows the *SConstraintRule* rule because the other rules have a similar definition.

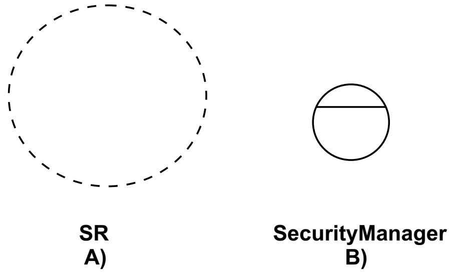


Fig. 6. Stereotype icons of packages (SR) and *SecurityManager*

4.3.4. *Tagged-values*

We do not need tagged-values in the profile.

4.3.5. *Well-Formedness rules*

Table 4 represents the well-formedness rules which are defined by means of both natural language and OCL in order to specify further constraints for the proper use of the UML profile considered.

Table 4
Well-formedness constraints

- Classes permitted in the model: All the packages in the secure CIM model must be *SOModel*, *GOModel* or *GSAModel*:
self.allContent -> forAll(oclIsKindOf(Package) implies (oclIsTypeOf(*GOModel*) or oclIsTypeOf(*SOModel*) or (oclIsTypeOf(*GSAModel*)))

4.3.6. *Comments*

In order to understand the models defined by our profile we shall now give a more detailed explanation how they are used to build a secure CIM for DWs. (See Fig. 3). First, we define the *GOModel* by applying proposal [19]: (i) we discover the intentional actors, (ii) we discover the goals (*Strategic*, *Decision* and *Information* stereotypes), (iii) we derive information requirements (*Requirement* stereotype) from information goals, and (iv) we obtain the MD concepts (*BusinessProcess*, *Measure* and *Context* stereotypes). We then define the *SOModel* by using the stereotypes defined in our profile: (i) We detect necessities according to organizations, policies, laws, rules and regulations. ii) we obtain the security requirements for the Security Manager. These requirements are modeled as *SSoftgoals*, and refined into the lower-level. During the refinement process various responsibilities and task are discovered (i.e. Roles and Compartments) along with the levels that will be used. iii) Associate *SSoftgoals* with the corresponding resources (i.e., *SCompartment*, *SRole* and *SLevel*).

Finally, as was previously stated, the *GSAModel* merges *GOModel* and *SOModel* by following these steps: i) Each *SSoftgoals* refined must be associated with the corresponding elements from the requirement model previously obtained, i.e. *BusinessProcess*, *Measure* and *Context*. ii) We analyze other security issues for the resources (i.e. *BusinessProcess*, *Measure* and *Context*), which the granularity level does not allows us to establish. Hence, constraints as tasks (*SConstraintRule*, *SConstraintAudit* and *SConstraintAuthorization*) are associated with the resources detected, which makes a positive contribution to the fulfillment of the corresponding *SSoftgoals*.

5. Pharmaceutical management: A Case Study

In this section we develop a case study through which to illustrate how the extended UML profile can be applied to the pharmaceutical consortium environment. Our proposal is aligned with MDA (see Fig. 1, Section 1). However, our main emphasis is placed on the definition of a secure CIM. Hence, we do not define QVT relations to show how the models are transformed from one level to the next level.

A pharmaceutical consortium manages several pharmacies which offer different kinds of services to the community. It wishes to control everything relating to the sales of medicines through medical prescriptions. Within the consortium there are a pharmacovigilance group which uses security to guard the use of certain medicines, a committee which guards its clients health, and a commercial group dedicated to commercialization and supply. We focus on the marketing of the prescription sales business process.

This section is divided into the following subsections: secure CIM (by using our profile), secure PIM (by using proposal [36]), secure PSM (by using the proposal [35]) and code example for Oracle DBMS which, in accordance with Fig. 1, cover the whole DWs life cycle. The MDA transformations secure CIM-secure PIM and secure PIM-secure PSM are supported by the guidelines defined in subsection 10 and by proposal [34].

5.1. *Defining a secure CIM*

Fig. 7 shows the *GOModel* defined in accordance with the problem, i.e. functional requirements elicited for DWs. The business process is related to one main actor, the marketing manager via the strategic goal "*increase prescription sales*". Two different decision goals are derived from this strategic goal: a "*decrease prescription price*" and "*give incentive to pharmacist*". The following information goals have been obtained from each of these decision goals: "*decrease prescription price*" and "*analyze prescription price*". The information requirements derived are as follows: "*analyze price by patient, prescription and pharmacy*, and *analyze sales by pharmacy, prescription and pharmacy_type*". In Fig. 7, each of these elements are defined as goals (strategic, decision and information goals) or as a task (information requirements). Furthermore, several resources are associated with the information requirements where this is

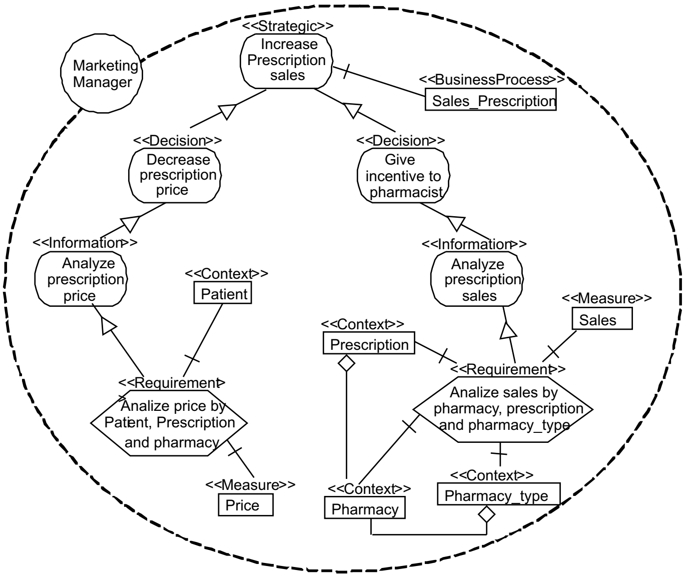


Fig. 7. Goal organization model (*GOModel*)

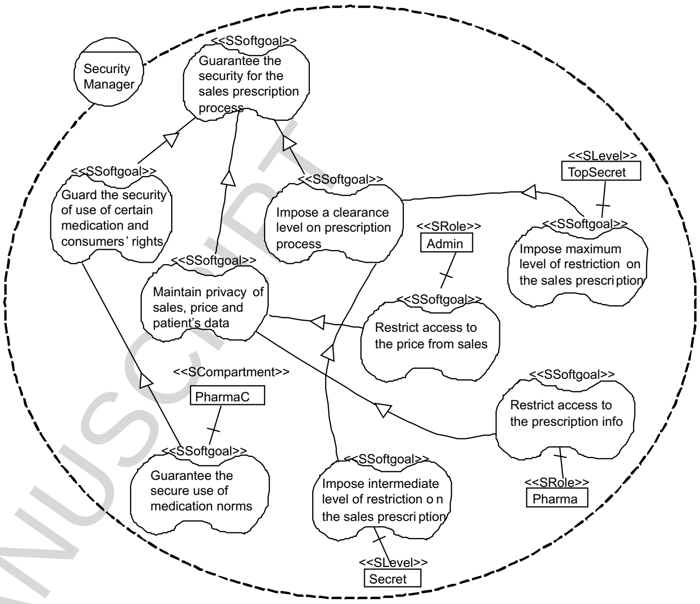

Security Manager
<<SSoftgoal>>
Guarantee the security for the sales prescription process
<<SSoftgoal>>
Guard the security of use of certain medication and consumers' rights
<<SSoftgoal>>
Impose a clearance level on prescription process
<<SLevel>>
TopSecret
<<SRole>>
Admin
<<SSoftgoal>>
Impose maximum level of restriction on the sales prescription
<<SSoftgoal>>
Maintain privacy of sales, price and patient's data
<<SSoftgoal>>
Restrict access to the price from sales
<<SCompartment>>
PharmaC
<<SSoftgoal>>
Restrict access to the prescription info
<<SSoftgoal>>
Guarantee the secure use of medication norms
<<SSoftgoal>>
Impose intermediate level of restriction o n the sales prescri ption
<<SRole>>
Pharma
<<SLevel>>
Secret

Fig. 8. Softgoal organization model (*SOModel*)

necessary such as measures and context of analysis. The measures are "*Sales* and *Price*". The elements that represent the context of analysis are "*Prescription*, *Pharmacy* and *Pharmacy_Type*", but they are related to each other, since they represent ways of aggregating the "*Pharmacy*" data. *Patient* is also the context of analysis, but does not contain relationships to other *Contexts*. According to the MDA framework, if we do not consider security measures for the DWs design, then Fig. 7 represents a CIM.

Once the functional requirements have been defined, it is necessary to describe the *SOModel* in order to elicit security requirements for DWs at the business level. By continuing with the case study, we focus on establishing security and confidentiality for the sales prescription process, which is performed by the *SecurityManager* actor via the "*guarantee the security for the sales prescription process*" *SSoftgoal*. By using a refinement process, three new softgoals: "*guard the security of use of certain medication and consumers' rights, maintain privacy of sales, price and patient's data* and *impose a clearance level on prescription process*" are obtained. See Fig. 8 for more details. Various responsibilities are discovered in this process. We therefore obtain the hierarchical relation: *PharmacyEmployee*, which is then specialized into the *Pharmacist (Pharma)* and *Administrative (Admin)* roles. Horizontal groups within the organization (compartments) are detected: *pharmacovigilanceCenter* (*pharmaC*, which is responsible for the security of the use of certain medications), and *commercialManagerCenter* (*commercialC*, which is responsible for the commercialization and supply). Restriction levels are established by means of *TopSecret* and *Secret*. Note in Fig. 8 how the security resources are associated with their corresponding *SSoftgoals*.

Finally, Fig. 9 shows the *GSAModel*, which merges *GOModel* and *SOModel* by means of the *Dependency* association (−▷−). I we are fulfill previous *SSoftgoals* we need to associate resources contained in *GOModel* (i.e. *Sales_Prescription*, *Patient*, *Price*, *Prescription*, *Pharmacy_Type* and *Sales*) with the *SSoftgoals* contained in *SOModel*. For example, "*impose maximum level of restriction on the sales prescription*" (marked in Fig. 9 with the number 1) and "*Guarantee the secure use of medication norms*" (marked in Fig. 9 with the number 5). The remaining *SSofgoals* which establish associations with resources from *GOModel* are dealt with analogously (see the *SSoftgoals* marked with the numbers 2, 3 and 4 in Fig. 9). The aforementioned *SSoftgoals* are, therefore, achieved through a *Dependency* association between the *SecurityManager* and the *MarketingManager*. *Sales_Prescription* is associated with the "*Impose maximum level of restriction on the sales prescription*" *SSoftgoals* whose *SLevel* is *TopSecret*. Moreover, other *SSoftgoals* are associated with resources (*Patient*, *Price*, *Prescription*, *Pharmacy_Type* and *Sales*).

Due to the fact that *Sales_Prescription* and *Prescription* are very valuable assets, they need additional restrictions. Fig. 9 shows how the *SOModel* has been modified with the *SRule* and *Audit* constraints, which are labeled as *SConstraintRule*, *SConstraintAudit* respectively. *SRule* contributes to the fulfillment of the *SSoftgoal* "*impose maximum level of restriction on the sales prescription*", so according to the dependency association defined, it is related to both the *BusinessProcess Sales_Prescription* and *Context Prescription* respectively. The same reasoning assures that the *Context Prescription* will be related to the *Audit* constraint. Moreover, other *SSoftgoals* are associated with resources (*Patient*, *Price*, *Prescription*, *Pharmacy_Type* and *Sales*). These are deal with in an analogous manner.

In accordance with to the classification for users of the ACA model introduced in Subsection 4.1, each of the system's user will have *securityLevel*, *securityRole* and *securi-*

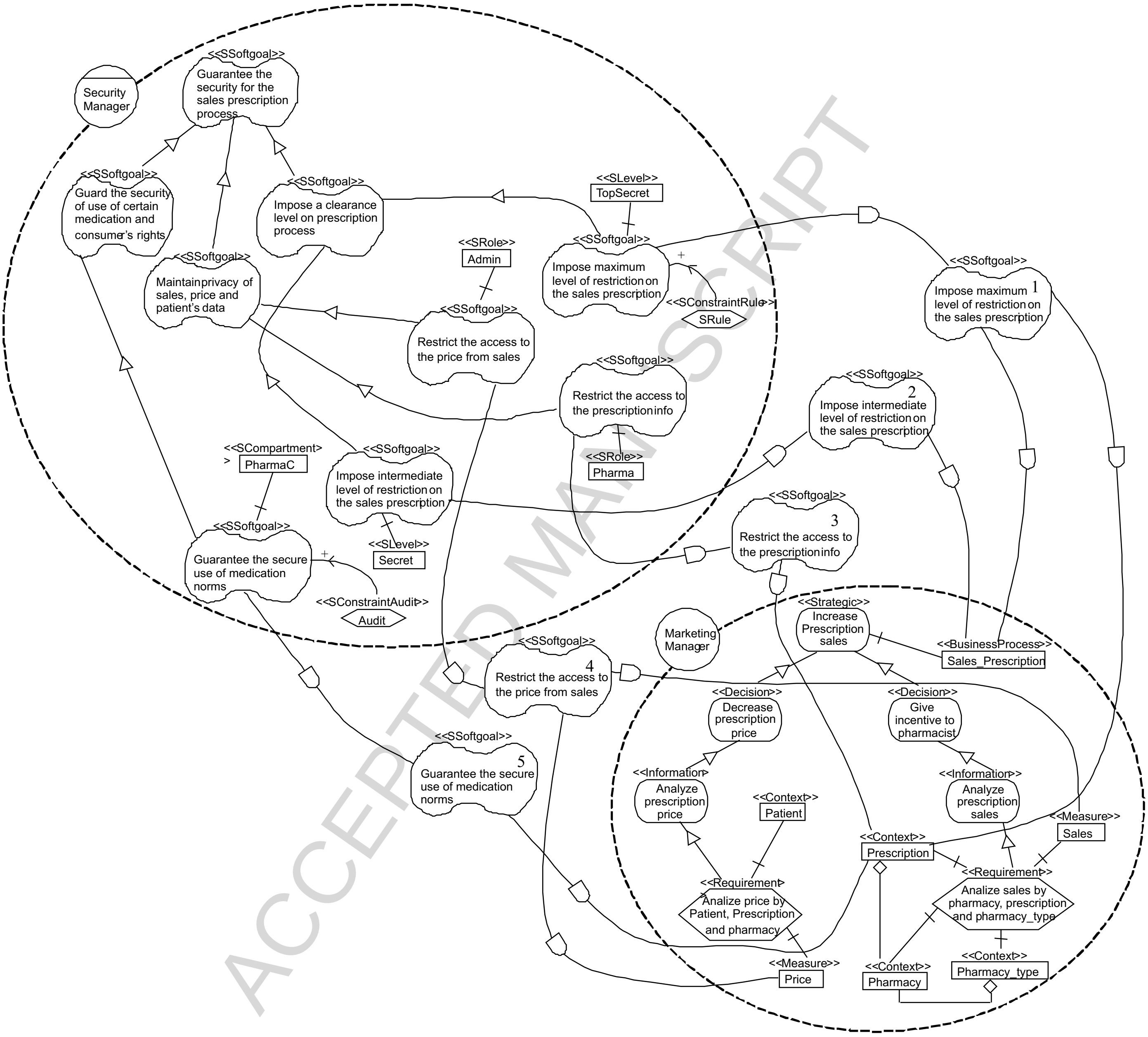

Security Manager
<<SSoftgoal>>
Guarantee the security for the sales prescription process
<<SSoftgoal>>
Guard the security of use of certain medication and consumer's rights
<<SSoftgoal>>
Impose a clearance level on prescription process
<<SSoftgoal>>
Maintain privacy of sales, price and patient's data
<<SLevel>>
TopSecret
<<SRole>>
Admin
<<SSoftgoal>>
Impose maximum level of restriction on the sales prescription
<<SConstraintRule>>
SRule
<<SSoftgoal>>
Restrict the access to the price from sales
<<SSoftgoal>>
Restrict the access to the prescription info
<<SRole>>
Pharma
<<SCompartment>>
PharmaC
<<SSoftgoal>>
Impose intermediate level of restriction on the sales prescrption
<<SLevel>>
Secret
<<SSoftgoal>>
Guarantee the secure use of medication norms
<<SConstraintAudit>>
Audit
<<SSoftgoal>>
Impose maximum level of restriction on the sales prescrption
1
<<SSoftgoal>>
Impose intermediate level of restriction on the sales prescrption
2
<<SSoftgoal>>
Restrict the access to the prescription info
3
<<SSoftgoal>>
Restrict the access to the price from sales
4
<<SSoftgoal>>
Guarantee the secure use of medication norms
5
Marketing Manager
<<Strategic>>
Increase Prescription sales
<<BusinessProcess>>
Sales_Prescription
<<Decision>>
Decrease prescription price
<<Decision>>
Give incentive to pharmacist
<<Information>>
Analyze prescription price
<<Information>>
Analyze prescription sales
<<Context>>
Patient
<<Measure>>
Sales
<<Context>>
Prescription
<<Requirement>>
Analize price by Patient, Prescription and pharmacy
<<Requirement>>
Analize sales by pharmacy, prescription and pharmacy_type
<<Measure>>
Price
<<Context>>
Pharmacy
<<Context>>
Pharmacy_type

Fig. 9. Goal/softgoal analysis model (*GSAModel* or secure CIM)

tyCompartment. Hence, we can conclude that a user has access to *Sales_Prescription* if his/her access class dominates the access class of *Sales_Prescription*, i.e. his/her security level is *TopSecret* (in this restricted case). The case study continues by defining the secure PIM, the secure PSM, and by showing a code example for Oracle DBMS.

5.2. *Defining the secure PIM*

While information and security requirements are defined at the business level, the MD elements are defined at the conceptual level in the corresponding secure PIM. As was stated in Section 1, PIM corresponds with an extension of the UML presented in [36], in which the information is clearly organized into secure facts and secure dimensions. These secure facts and dimensions are modeled by *SFact* (represented as S) and *SDimension* (↳) stereotypes, respectively. *SFact* and *SDimension* are related by shared aggregation relationships (the *Association* UML metaclass) in class diagrams. While an *SFact* is composed of measures or secure fact attributes (*SFactAttribute* stereotype, **SFA**), with respect to *SDimensions*, each aggregation level of a hierarchy is specified by classes stereotypes as *SBase* (BS). Each *SBase* class can contain several secure dimension attributes (*SDimensionAttribute*, **SDA**) and must also contain a secure descriptor attribute (*SDescriptor* attribute, **SD**). An association stereotyped as *Rolls-upTo* (`<<Rolls-UpTo>>`) between *SBase* classes

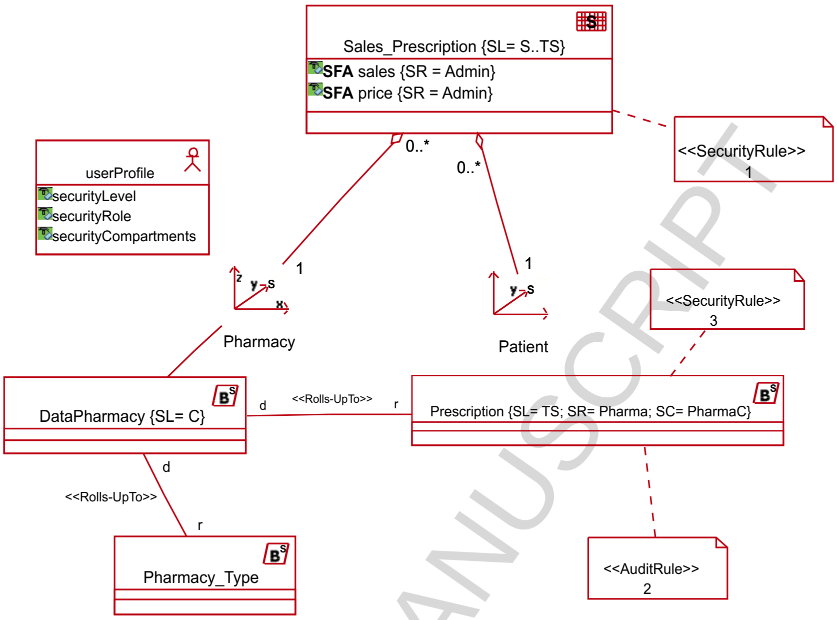


Fig. 10. An instance of the secure PIM

specifies the relationship between two levels of a classification hierarchy. Within this, role R represents the direction in which the hierarchy rolls up, whereas role D represents the direction in which the hierarchy drills down. The information about all users who are entitled to access the MD model are represented as instances of the *UserProfile* class (stereotype *UserProfile*, 웃).

Proposal [36] allows us to classify both information and users in order to represent the main security aspects in the conceptual modeling of DWs. Security information is defined for each element of the model (*SFact*, *SDimension*, *SFactAttribute*, etc) by specifying a sequence of security levels, a set of user compartments and a set of roles. Moreover, the constraints (*AuditRule*, *AuthorizationRule* and *SecurityRule*) are modeled by using UML notes. These constraints are defined by following the syntax of the ARs, AURs and SIARs rules from the ACA model (more details in [8,9,36]).

To obtain the secure PIM from the secure CIM it is necessary to apply a set of QVT [23] relations, but thid is not within the scope of this paper. We limit our effort to defining a manual transformation between the secure CIM (*GSAModel*) and secure PIM based on the guidelines presented below. Fig. 10 shows the mapping between the secure CIM (*GSAModel*) and the secure PIM.

In the sequel we suggest several guidelines for transforming the secure CIM into the secure PIM:

Guideline G1: Related to actors;
Guideline G2: Related to *BusinessProcesses*;
Guideline G3: Related to *Measures*;
Guideline G4: Related to *Context*.

Guideline G1: Actors in the *GSAModel* (secure CIM) are mapped onto the *userProfile* class of the MD model. By default the *userProfile* class will contains three attributes: *securityLevel* (SL), *securityRole* (SR) and *securityCompartment* (SC). According to the ACA model, these attributes allow us to represent the security information for each of the system's, users.

In our case study (see Fig. 10) we have only one actor, denoted as *MarketingManager*, which will be an instance of the *UserProfile* class. The values of SL, SR and SC for each actor are (for the moment) unknown due to the granularity at this level.

Guideline G2: Create a *SFact* class for each *BusinessProcess* in the *GSAModel*. The name of the *SFact* in the MD model will be the name of the *BusinessProcess* in the *GSAModel*. Several guidelines are given to obtain the security information associated with the *SFact* in the MD model.

Guideline G2.1: *SLevel*, *SRole* and *SCompartment* decomposition associated with the *BusinessProcess* resource through on *SSoftgoal* dependency in *GSAModel* are mapped as SL, SR and SC classes associated with the *SFact* that represents the corresponding *BusinessProcess*.

Guideline G2.2: Each *SConstraintRule* task that makes a positive contribution to an *SSoftgoal*, which constitutes an *SSoftgoal* dependency for the *BusinessProcess* in the *GSAModel* is mapped as a *SecurityRule* class associated with the *SFact* in the MD model. *SConstraintAudit* and *SConstraintAuthorization* tasks in the *GSAModel* are dealt with in an analogous manner.

In our case study we have only one *BusinessProcess* (see the *GSAModel* depicted in Fig. 9). According to G2 the *Sales_Prescription BusinessProcess* should be mapped onto the *Sales_Prescription SFact* (see Fig. 10). Note in Fig. 9 how the *securityManager* depends on the *MarketingManager* to achieve the *SSoftgoals* marked with the numbers 1 and 2. According to G2.1 the *Sales_Prescription SFact* is associated with the SL *Secret* (S) and *TopSecret* (TS), which are represented in its heading (see Fig. 10).

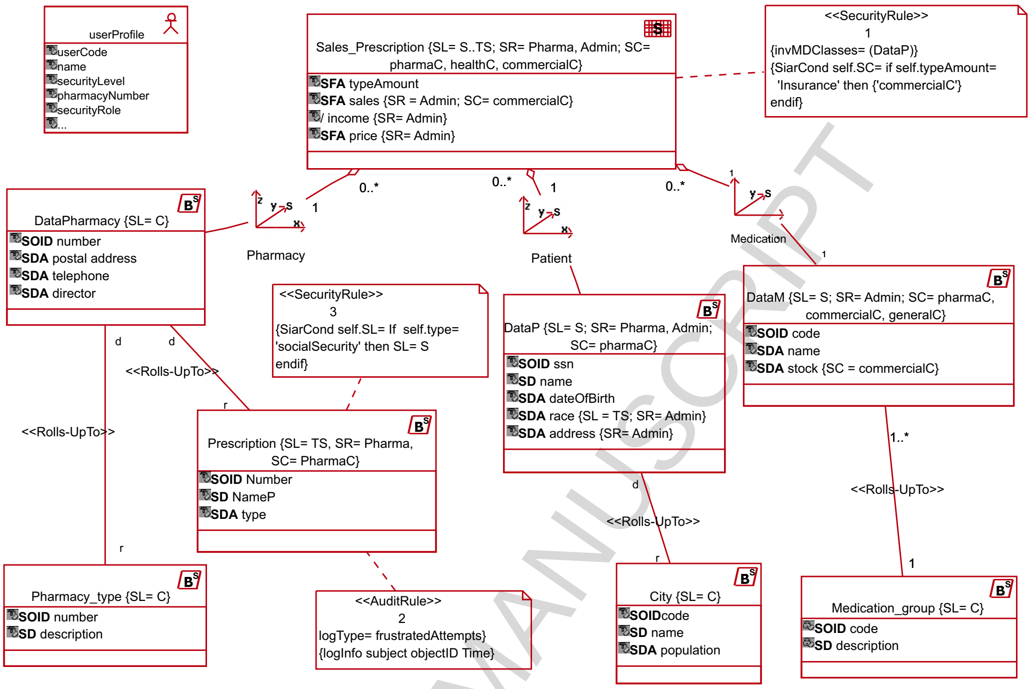


Fig. 11. An instance of the enriched secure PIM

According to G2.2 the *Sales_Prescription SFact* is associated with the *SecurityRule* 1, which is modeled in Fig. 10 by using a UML note.

Guideline G3: Each resource labeled with the stereotype `<<Measure>>` associated through the Strategic, Decision and Information goal with the *BusinessProcess* detected in guideline G2 is mapped as *SFactAttribute* for the *SFact* that corresponds with the *BusinessProcess*.

Guideline G3.1: *SLevel*, *SRole* and *SCompartment* decomposition associated with the *Measure* resource through a *SSoftgoal* dependency in *GSAModel* are mapped as SL, SR and SC classes associated with the *SFactAttribute* which represents the corresponding *Measure*.

Guideline G3.2: Each *SConstraintRule* task that makes a positive contribution to an *SSoftgoal*, which constitutes a *SSoftgoal* dependency for the *Measure* in the *GSAModel* is mapped as a *SecurityRule* class associated with the *SFact* that contains the *SFactAttribute* corresponding to the *Measure*. *SConstraintAudit* and *SConstraintAuthorization* tasks associated with *Measure* in the *GSAModel* are dealt with in an analogous manner.

In our case study we have the *Price* and *Sales Measures* (see *GSAModel* in Fig. 9) which are mapped as *SFactAttributes* in the MD model, as Fig. 10 shows. In Fig. 9 we can see that *securityManager* depends on *MarketingManager* to achieve the *SSoftgoals* marked with the number 4. Hence, according to G3.1 these attributes are associated with the SR *Admin* (see Fig. 10). These *Measures* do not have associated security constraints (*SConstraintRule*, *SConstraintAudit* and *SConstraintAuthorization*).

Guideline G4: *Context* resources which are associated through a UML aggregation represent the context of analysis, which become *SDimensions* classes in the MD model.

Guideline G4.1: Those *Contexts* that do not contain any *Context* associated through UML aggregations in the *GSAModel* are mapped as *SBases* classes which represents the root of the *SDimensions* hierarchy in the MD model.

Guideline G4.2: Each *Context* (which is not characterized by G4.1) in the *GSAModel* is mapped as an *SBase* class in the MD model. Each UML aggregation between two *Contexts* in the *GSAModel* is mapped as a *Rolls-upTo* association between the corresponding *SBases* in the MD model.

Guideline G4.3: *SLevel*, *SRole* and *SCompartment* decomposition associated with the *Context* resource through a *SSoftgoal* dependency in *GSAModel* are mapped as SL, SR and SC classes associated with the *SBase* class that represents the corresponding *Context*.

Guideline G4.4: Each *SConstraintRule* task that makes a positive contribution to an *SSoftgoal*, which constitutes an *SSoftgoal* dependency for the *Context* in the *GSAModel* is mapped as a *SecurityRule* class associated with the *SBase* which represents the corresponding *Context*. *SConstraintAudit* and *SConstraintAuthorization* tasks associated with *Context* in the *GSAModel* are deal with in an analogous manner.

In our case study we have two *Contexts* of analysis: (i) the *Patient Context* is transformed into the *Patient SDimension* in the MD model (see Fig. 10), (ii) the *Pharmacy*, *Pharmacy_Type* and *Prescription Contexts* represent the *SDimension Pharmacy*. According to G4.1 the *Pharmacy Context* represents the *SBase* root of the *SDimension Pharmacy* (see Fig. 9). If we apply G4.2, the

Prescription and *Pharmacy_Type Contexts* are mapped as *SBases* in the MD model. The UML aggregations between them and the *Pharmacy Context* are mapped as a *Rolls-upTo* association between the corresponding *SBase* classes in the MD model (see Fig. 10).

In Fig. 9 we can see that *securityManager* depends on *MarketingManager* to achieve the *SSoftgoals* marked with the numbers 1, 3 and 5. Hence, according to G4.3, the *Prescription Context* is associated with the SL *TopSecret* (TS), the SR *Pharma* and the SC *PharmaC* (see Fig. 10. According to G4.4 the *Prescription Context* is associated with the *AuditRule* 2 and the *SecurityRule* 3, which are represented in the MD model shown in Fig. 10 by using UML notes. These constraints are obtained by taking into account the *SSoftgoals* dependency marked in the *GSAModel* (see Fig. 9) with the numbers 1 and 5 between the *securityManager* and *MarketingManager*.

Observe in Fig. 10 that certain classes do not have any attributes. Also, there are incomplete *SDimensions* and the *SConstraintRule* and *SConstraintAudit* constraints. The conceptual MD model represented in Fig. 10 has a descriptive level since many elements of the organizational model which are not part of the software model totally correspond with the conceptual MD model [2]. Likewise, many elements in the software model which are made up of detailed technical software solutions and constructs are not part of the organizational model [2]. It is thus necessary to enrich the conceptual MD model presented in Fig. 10.

Fig. 11 shows an instance of our enriched secure PIM, which makes the part of the DWs that is required for the previous problem more complete. The *SFact Sales_Prescription* (stereotype *SFact*) contains all the sales information in one or more pharmacies, and can be accessed by users who have *Secret* or *topSecret security levels*, play an *Administrative* or *Pharmacist* role and belong to *pharmacovigilanceCenter*, *healthOversightCenter* (the committee which guards the health of the company's clients) and *commercialManagerCenter* compartments. The sales attribute can only be accessed by users who perform the administrative role (SR tagged values of sales attribute) and belong to the *commercialManagerCenter* compartment, and access to this attribute will therefore be forbidden to other users who are *pharmacist* and *maintenance* employees or belong to other different *commercialManagerCenter* compartments. The *income* attribute can only be accessed by users who perform the *administrative* role (SR tagged value of *income* attribute). Others static user classifications for the conceptual model classes defined in Fig. 11 are:

The *SFact Sales_Prescription*. This contains three *SDimensions* (*Pharmacy*, *Patient* and *Medication*), which contain *SBase* hierarchies. Access to these *SBase* hierarchies is established in the same way as was done with the *SFact*. The *UserProfile* has been completed in order to store information about all users who will have access to this secure MD model.

Several security constraints have been specified by using the previously defined constraints, stereotypes and tagged values. The following paragraphs correspond to notes 1, 2 and 3 in Fig. 11:

1. For each instance of the *SFact* class *Sales_Prescription*, if the type of payment is through insurance then the security compartment will be *commercialManagerCenter* (*commercialC*, tagged value SC). This constraint is only applied if the user makes a query whose information comes from the *DataPharmacy*.

2. We would like to record, for future audit, the *subject, object* and *time* of every frustrated access attempt upon *Prescription*.

3. For each instance of the *SBase* class *Prescription*, if the prescription is of the type "*socialSecurity*", then the security level will be *Secret* (*Secret*, tagged value SL).

5.3. *Defining the secure PSM*

By using the *PIM* in Fig. 10 as a starting point, we apply a set of QVT relations [34] through which to achieve an instance of the *secure PSM*. The transformation ensures that *SFact* and *SDimensions* are transformed into *STables* with their associated security information. The *UserProfile* class is transformed into a classical *Table* from *CWM*. Fig. 12 represents a star schema at the logical level, which corresponds with an instance of the relational metamodel from the CWM extended in [35].

The *SFact Sales_Prescription* is represented in Fig. 12 by means of the *STable Sales_ Prescription*. All of its columns are represented in this table along with all the associated security information, which restricts access both to the table itself and to its columns. All of the hierarchy that conforms to an *SDimension* must be represented by means of a single *STable*. Observe in Fig. 12 that the *Pharmacy STable* contains as *SColumn* the attributes from the *SBases DataPharmacy*, *Pharmacy_Type* and *Prescription* classes from Fig. 11. This occurs in an analogous manner with the *Patient* and *Medication SBases* classes. In order to build a star scheme the *Sales_Prescription* table must contain columns such as *Foreign Key*(FK) which represent *Primary Key* (PK) in the tables that correspond with *SDimensions* at the PIM level.

The security information (SL, SR and SC) represented in the classes from Fig. 11 is modeled at the logical level in the title of the table itself (See Fig. 12).

The *SecurityRule1*, *AuditRule2* and *SecurityRule3* security constraints that appear in Fig. 11 are transformed into instances of the *SecurityConstraint* from the extended relational metamodel. These instances are modeled in Fig. 12 by means of UML notes with the names *ARConst1* and *AURConst2* respectively. The *securityRule3* attempts to change the security for the *SBase Prescription* class, thus establishing new values for *securityLevel (SL)*. As we observed in Fig. 11, the security of the *SBase Pharmacy* class has been assigned to the *SColumns NumPres*, *nameP* and *type*. Hence, the constraint is transformed and applied to

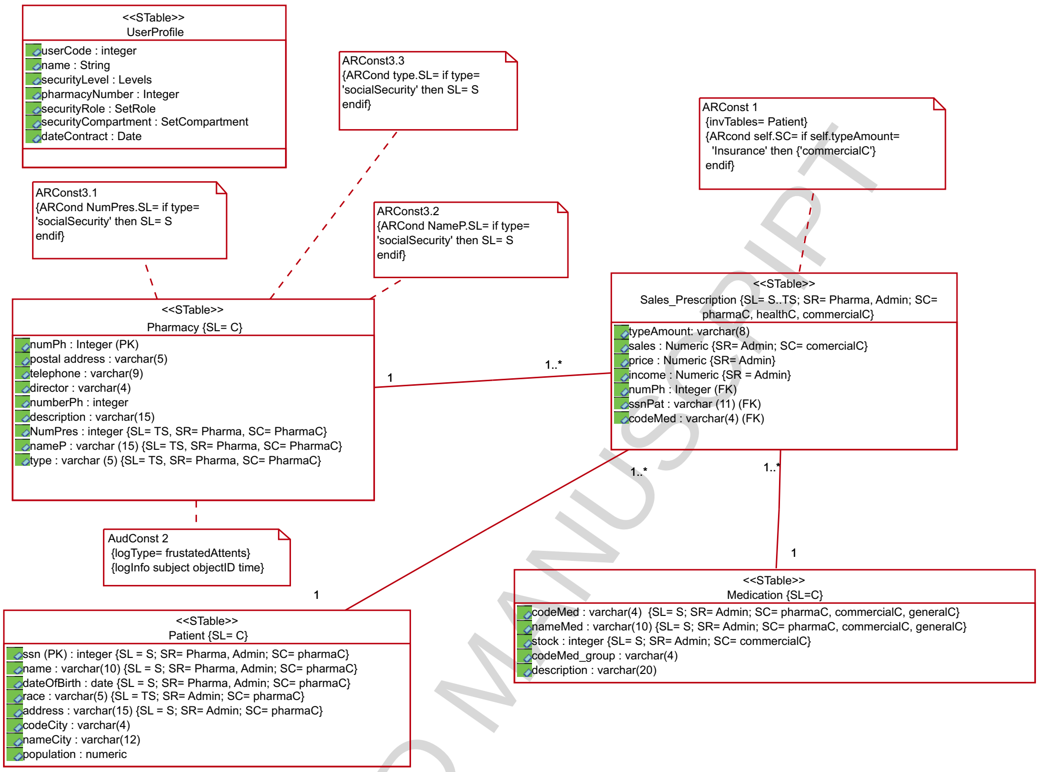

<<STable>>
UserProfile
userCode : integer
name : String
securityLevel : Levels
pharmacyNumber : Integer
securityRole : SetRole
securityCompartment : SetCompartment
dateContract : Date
ARConst3.3
{ARCond type.SL= if type=
'socialSecurity' then SL= S
endif}
ARConst 1
{invTables= Patient}
{ARcond self.SC= if self.typeAmount=
'Insurance' then {'commercialC'}
endif}
ARConst3.1
{ARCond NumPres.SL= if type=
'socialSecurity' then SL= S
endif}
ARConst3.2
{ARCond NameP.SL= if type=
'socialSecurity' then SL= S
endif}
<<STable>>
Sales_Prescription {SL= S..TS; SR= Pharma, Admin; SC=
pharmaC, healthC, commercialC}
typeAmount: varchar(8)
sales : Numeric {SR= Admin; SC= comercialC}
price : Numeric {SR= Admin}
income : Numeric {SR = Admin}
numPh : Integer (FK)
ssnPat : varchar (11) (FK)
codeMed : varchar(4) (FK)
<<STable>>
Pharmacy {SL= C}
numPh : Integer (PK)
postal address : varchar(5)
telephone : varchar(9)
director : varchar(4)
numberPh : integer
description : varchar(15)
NumPres : integer {SL= TS, SR= Pharma, SC= PharmaC}
nameP : varchar (15) {SL= TS, SR= Pharma, SC= PharmaC}
type : varchar (5) {SL= TS, SR= Pharma, SC= PharmaC}
1
1..*
1..*
1..*
1
AudConst 2
{logType= frustatedAttents}
{logInfo subject objectID time}
1
<<STable>>
Medication {SL=C}
codeMed : varchar(4) {SL= S; SR= Admin; SC= pharmaC, commercialC, generalC}
nameMed : varchar(10) {SL= S; SR= Admin; SC= pharmaC, commercialC, generalC}
stock : integer {SL= S; SR= Admin; SC= commercialC}
codeMed_group : varchar(4)
description : varchar(20)
<<STable>>
Patient {SL= C}
ssn (PK) : integer {SL = S; SR= Pharma, Admin; SC= pharmaC}
name : varchar(10) {SL = S; SR= Pharma, Admin; SC= pharmaC}
dateOfBirth : date {SL = S; SR= Pharma, Admin; SC= pharmaC}
race : varchar(5) {SL = TS; SR= Admin; SC= pharmaC}
address : varchar(15) {SL = S; SR= Admin; SC= pharmaC}
codeCity : varchar(4)
nameCity : varchar(12)
population : numeric

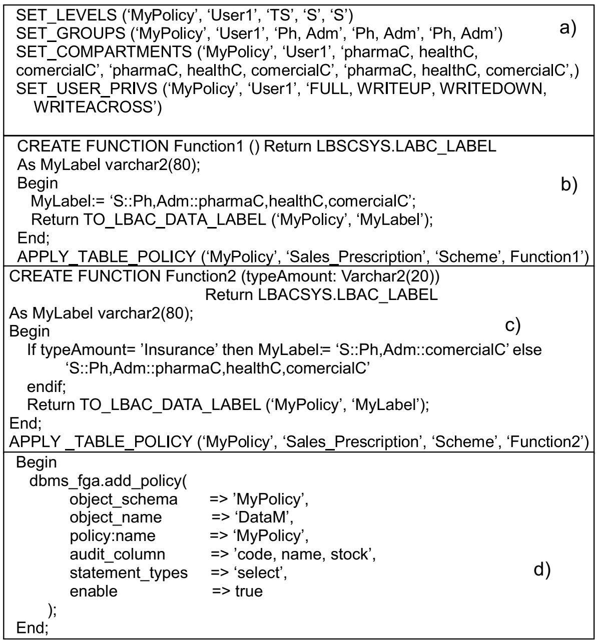
Fig. 12. An instance of the secure PSM

SColumns NumPres, *nameP* and *type*. Consequently, the *SecurityRule3* is transformed into three *ARConstraints*, which appear in Fig. 12 under the names of *ARConst3.1*, *ARConst3.2* and *ARConst3.3*, which are associated with the *SColumns NumPres*, *nameP* and *type* respectively.

5.4. *Code Example in Oracle DBMS*

Our case study will be completed by showing some implementations of the security aspects modeled in the star scheme that appears in Fig. 12. The current OMG standard used to apply the *model-code* transformations is the MOF Model to Text Transformation Language (*MOF2Text*). However, we shall briefly show the possibilities that Oracle 10g DBMS offers in order to implement security and audit measures by means of Oracle Label Security (OLS10g), Virtual Private Databases (VPD) and Oracle Fine-Grained Auditing (FGA). We shall only explain the security aspects that our extension contemplates, and to do this we have first created a security policy named "*MyPolicy*" along with valid levels, compartments and hierarchical groups.

In Fig. 13 a) we show how the *User1* satisfies the security properties for the *Sales_Prescription STable*. Fig. 13 b) shows how we define and establish the security information for the *Sales_Prescription* table by labeling functions from OLS, although it is not possible to consider security at the column level. The *ARConst 1* is implemented by means of the labeling function represented in Fig. 13 c). The FGA allows us to define and implement the *AudConst 2* (see Fig. 13d)). In *AudConst 2* we cannot implement the *logType* because FGA does not allow us to choose it.



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

 CREATE FUNCTION Function1 () Return LBSCSYS.LABC_LABEL
 As MyLabel varchar2(80);
 Begin                                                                      b)
   MyLabel:= ‘S::Ph,Adm::pharmaC,healthC,comercialC’;
   Return TO_LBAC_DATA_LABEL (‘MyPolicy’, ‘MyLabel’);
 End;
 APPLY_TABLE_POLICY (‘MyPolicy’, ‘Sales_Prescription’, ‘Scheme’, Function1’)
CREATE FUNCTION Function2 (typeAmount: Varchar2(20))
                                 Return LBACSYS.LBAC_LABEL
As MyLabel varchar2(80);
Begin                                                                   c)
   If typeAmount= ’Insurance’ then MyLabel:= ‘S::Ph,Adm::comercialC’ else
         ‘S::Ph,Adm::pharmaC,healthC,comercialC’
   endif;
   Return TO_LBAC_DATA_LABEL (‘MyPolicy’, ‘MyLabel’);
End;
APPLY _TABLE_POLICY (‘MyPolicy’, ‘Sales_Prescription’, ‘Scheme’, ‘Function2’)
 Begin
   dbms_fga.add_policy(
          object_schema      => ’MyPolicy’,
          object_name        => ‘DataM’,
          policy:name        => ‘MyPolicy’,
          audit_column       => ’code, name, stock’,
          statement_types    => ‘select’,                               d)
          enable             => true
      );
 End;
```

Fig. 13. Implementing our constraints in Oracle 10g.

6. Conclusion and Future work

In this paper we have presented a UML 2.0 profile which allows us to the define the security requirements for DWs at the business level. The extension contains the necessary stereotypes and constraints, which adapt the *i** framework in order to represent security requirements for DWs. The proposal allows us to represent certain elements of the ACA model at the business level which are virtually not considered in the requirements analysis phase of the DWs design. Our approach is used to define a formalized procedure which integrates both security and information requirements an takes the ACA model into account in the requirement analysis phase. Moreover, as our profile is MDA compliant we can define a secure CIM, which can be transformed throughout the entire DWs life cycle.

Our immediate future work consists of providing both the definition and the implementation of the QVT relations in order to establish a transformation between the CIM and the PIM levels. In addition, we plan to develop a methodology which will integrate the entire DWs life cycle within the MDA framework.

Acknowledgments.

This work has been partially supported by the METASIGN project (TIN2004-00779) from the Spanish Ministry of Education and Science, and by the QUASIMODO and MISTICO projects from the Regional Science and Technology Ministry of Castilla-La Mancha (Spain).

References

[1] A. Abelló, J. Samos, and F. Saltor, “A Framework for the classification and Description of Multidimensional Data Models”, Conference on Database and Expert Systems Applications (DEXA’01), Munich, Germany, September 2001, pp. 668-677.

[2] F. M. R. Alencar, J. Castro, G. A. C. Cysneiros, and Mylopoulos J, “From Early Requirements Analysis Modeled by the i* Technique to Later Requirements Modeled in Precise UML”, Workshop em Engenharia de Requisitos (WER’00), Rio de Janeiro, Brasil, July 2000, pp. 92-108.

[3] A. I. Antón, “Goal-Based Requirements Analysis”, IEEE International Conference on Requirements Engineering (RE’96), Colorado Springs, Colorado, USA, April 1996, pp. 136-144.

[4] J. Conallen, Building Web Applications with UML: Addison - Wesley, 2000.

[5] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, “Non-Functional Requirements in Software Engineering”, Kluwer Academic, 2000.

[6] P. Devanbu, S. Stubblebine, “Software Engineering for Security: A Roadmap”. In The Future of Software Engineering, A. Finkelstein, Ed.: ACM Press, (2000), pp. 227-239.

[7] T. R. Farkhani, and M. R. Razzazi, “Examination and Classification of Security Requirements of Software Systems” Information and Communication Technologies (ICTTA’06), (2006) pp. 2778-2783.

[8] E. Fernández-Medina, J. Trujillo, R. Villarroel, and M. Piattini, “Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses”, Decision Support Systems, 42(3), (2006), 1270-1289.

[9] E. Fernández-Medina, J. Trujillo, R. Villarroel, and M. Piattini, “Developing Secure Data Warehouses with a UML Extension”, Information Systems, 32(6), (2007), 826-856.

[10] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, “Role-Based Access Control”. Artech House, Inc. Norwood, MA, USA (2003)

[11] P. Giorgini, F. Massacci and J. Mylopoulos, “Modeling Security Requirements Through Ownership, Permission and Delegation”, International Conference on Requirements Engineering (RE’05), Paris, France, August 2005, pp. 167-176.

[12] GRL web page. http://www.cs.toronto.edu/km/GRL/

[13] N. Katic, G. Quirchmayr, J. Schiefer, M. Stolba, and A. M. Tjoa, “A Prototype Model for Data Warehouse Security Based on Metadata”, Proceedings of the 9th International Workshop on Database and Expert Systems Applications (DEXA’98), Vienna, Austria, August 1998, 300-309.

[14] R. Kimball, and M. Ross, “Data Warehouse Toolkit”. Wiley & Songs (2002)

[15] R. Kirkgöze, N. Katic, M. Stolda, and A. M. Tjoa, “A Security Concept for OLAP”, 8th International Workshop on Database and Expert System Applications (DEXA’97), Toulouse, France (1997), 619-626.

[16] A. van Lamsweerde, S. Brohez, R. De Landtsheer, and D. Janssens, “From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering”, Workshop on Requirements for High Assurance Systems (RHAS03), Monterey (CA), USA, September 2003, pp. 49-56.

[17] L. Liu, E. Yu, and J. Mylopoulos, “Security and Privacy Requirements Analysis within a Social Setting”, International Conference on Requirements Engineering (RE’03), Monterey Bay, CA, USA, September 2003, pp. 151-161.

[18] S. Luján-Mora, J. Trujillo, and I. Y. Song., “A UML profile for multidimensional modeling in data warehouses”, Data & Knowledge Engineering (DKE), 59(3), (2006), 725-769.

[19] J. N. Mazón, J. Pardillo, J. Trujillo, “A Model-Driven Goal-Oriented Engineering Approach for Data Warehouses”, International Workshop on Requirements, Intentions and Goals in Conceptual Modeling (ER Workshops 2007), Auckland, New Zealand, November 2007, pp. 255-264.

[20] J. N. Mazón, and J. Trujillo, “An MDA approach for the development of data warehouses”, Decision Support Systems, 45(1), (2008), 41-58.

[21] H. Mouratidis, P. Giorgini, and G. Manson, “Integrating Security and Systems Engineering: Towards the Modeling of Secure Information Systems”, International Conference on Advanced Information Systems Engineering (CAiSE’03), Velden, Austria, June 2003, pp. 16-20.

[22] OMG, “MDA Guide Version 1.0.1”, J. M. a. J. Mukerji, Ed.: OMG (2003)

[23] OMG, ”MOF 2.0 QVT Final Adopted Specification” (2005)

[24] OMG, Object Management Group, UML Superstructure Specification, v2.0. Available from: http://www.omg.org/cgi-bin/doc?formal/05-07-04

[25] Prakash, N., Singh, Y., Gosain, A. “Informational scenarios for data warehouse requirements elicitation”, International Conference on Conceptual Modeling (ER’04), Shanghai, China, November 2004, pp. 205216.

[26] T. Priebe and G. Pernul, “Towards OLAP Security Design - Survey and Research Issues”, Proceedings of the 3rd ACM international workshop on Data Warehousing and OLAP (DOLAP’00), Virginia, USA, 2000, pp. 33-40.

[27] T. Priebe and G. Pernul, “A Pragmatic Approach to Conceptual Modeling of OLAP Security”, 20th International Conference on Conceptual Modeling (ER’01), Yokohama, Japan, November 2001, pp. 311-324.

[28] S. Rizzi, A. Abelló, J. Lechtenbörger, and J. Trujillo, “Research in data warehouse modeling and design: dead or alive?”, (DOLAP’06), Arlington, Virginia, USA, November 2006, pp. 3-10.

[29] A. Rosenthal and E. Sciore, “View Security as the Basic for Data Warehouse Security”, Workshop on Design and Management of Data Warehouse (DMDW’00), Stockholm, Sweden, June 2000, pp. 8.1-8.8.

[30] F. Saltor, M. Oliva, A. Abelló, and J. Samos, "Building Secure Data Warehouse Schemas from Federated Information Systems", in Heterogeneous Information Exchange and Organizational Hubs., Ed.: KA, (2002), 123-134.

[31] P. Samarati, and S. De Capitani di Vimercati, Access Control: Policies, Models, and Mechanisms, International School on Foundations of Security Analysis and Design (FOSAD'00), Bertinoro, Italy, LNCS 2171, pp. 137-196.

[32] S. Jajodia, and R. Sandhu, Polyinstantiation for cover stories, Second European Symposium on Research in Computer Security, Toulouse, France, November 1992.

[33] E. Soler, J. Trujillo, E. Fernández-Medina and M. Piattini, "Application of QVT for the Development of Secure Data Warehouses: A case study", Second International Conference on Availability, Reliability and Security (ARES'07), Vienna, Austria, April 2007, pp. 829-836.

[34] E. Soler, J. Trujillo, E. Fernández-Medina and M. Piattini, "A set of QVT relations to transform PIM to PSM in the Design of Secure Data Warehouses", Second International Conference on Availability, Reliability and Security (ARES'07), Vienna, Austria, April 2007, pp. 644-654.

[35] E. Soler, J. Trujillo, E. Fernández-Medina and M. Piattini, "Building a secure star schema in data warehouses by an extension of the relational package from CWM", Computer Standards & Interfaces, 30(6), (2008), 341-350.

[36] R. Villarroel, E. Fernández-Medina, M. Piattini and J. Trujillo, "A UML 2.0/OCL Extension for Designing Secure DWs", Journal of Research and Practice in Information Technology, (38)1, (2006), 31-43.

[37] K. Weidenhaupt, K. Pohl, M. Jarke and P. Haumer, "Scenarios in System Development: Current Practice". IEEE Software, 15(2), (1998), 34-45.

[38] E. Weippl, O. Mangisengi, W. Essmayr, F. Lichtenberger, and W. Winiwarter, "An Authorization Model for Data Warehouses and OLAP", Workshop On Security In Distributed Data Warehousing, in conjunction with 20th IEEE Symposium on Reliable Distributed Systems (SRDS'2001), New Orleans, Louisiana, USA, October 2001, pp. 9-13.

[39] E. Yu, "Modelling Strategic Relationships for Process Reengineering". PhD thesis, University of Toronto, Canada (1995)

Emilio Soler is a graduate in mathematics from the Pedagogical University of Matanzas (Cuba) and is also an assistant professor at the Computer Science Department of the Matanzas University (Cuba). He is currently a PhD student at the School of Computer Science at the University of Alicante (Spain), and his research activity is in the field of security in data warehouses, MDA and information systems. He has published and presented paper at various national and international workshops and conferences in Computer Science such as ICCSA, ARES, JISBD, WOSIS and IDEAS. Contact him at emilio.soler@umcc.cu.

Juan Trujillo received a Ph.D. in Computer Science from the University of Alicante (Spain) in 2001. His research interests include database modeling, the conceptual design of data warehouses, MD 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 a Program Committee member of several workshops and conferences such as ER, DOLAP, DSS, and SCI and has also spent some time as a reviewer for several journals such as JDM, KAIS, ISOFT and JODS.

Eduardo Fernández-Medina holds a PhD. in Computer Science from the University of Castilla-La Mancha. His research activity is in the field of security in databases, data warehouses, 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 presented several dozen papers at national and international conferences (DEXA, CAISE, UML, ER, etc.). He is the author of several manuscripts in national and international journals (Information Software Technology, Computers And Security, Information Systems Security, etc. and belongs to various professional and research associations (ATI, AEC, ISO, IFIP WG11.3 etc.).

Mario Piattini has an MSc and a PhD in Computer Science from the Politechnic University of Madrid. He is a Certified Information System Auditor from the ISACA (Information System Audit and Control Association). The 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. His research interests are: advanced database design, database quality, software metrics, object- oriented metrics and software maintenance.