JISBD2007-07: An extension of the Relational Metamodel of CWM to represent Secure Data Warehouses at the Logical Level

E. Soler, J. Trujillo, E. Fernández-Medina y M. Piattini

Abstract— **Generally, security and audit measures for Data Warehouses (DWs) are defined in the final implementation on top of commercial systems because there is not a standard for the exchange and the operability of metadata. The Common Warehouse Metamodel (CWM) proposal is broadly accepted as the standard for the interchange and the interoperability of metadata. Nevertheless, it does not allow us to specify security measures. In this paper, we make use of the extension mechanisms provided by the CWM to extend the relational package to specify, at the logical level, the security and audit rules captured during the conceptual modeling phase of the DWs design. Moreover, in order to show the benefits of our extension, we apply it to a case study related to the management of the pharmacies consortium businesses.**

Keywords— **data warehouses, common warehouse metamodel, security.**

I. Introducción

Según el actual desarrollo de la tecnología digital, las organizaciones adoptan más sistemas informatizados, que depende tanto de bases de datos (BD) como de Almacenes de Datos (ADs). En consecuencia, la supervivencia de las organizaciones depende de una apropiada manipulación de la seguridad y confidencialidad de la información [1]. Normalmente en los proyectos de ADs los aspectos de seguridad se implementan en fases finales de diseño. Sin embargo, la seguridad de la información es un serio requisito que debe ser cuidadosamente considerado, no como un aspecto aislado, sino como un elemento que esté presente en todas las etapas del ciclo de vida de desarrollo, desde el análisis de requisitos hasta la implementación y mantenimiento [2]. Lo anterior justifica que es crucial especificar medidas de confidencialidad en el diseño de ADs y hacerlas cumplir.

Por otro lado, es ampliamente aceptado que el diseño de ADs se basa en el modelado multidimensional (MD), el cual estructura la información en hechos y dimensiones. Para el diseño de ADs nos basamos en la Arquitectura dirigida por Modelos (MDA) [3]. MDA propone modelos a diferentes niveles: a nivel conceptual el modelo independiente de la plataforma (PIM) y a nivel lógico el modelo dependiente de la plataforma (PSM). En nuestro contexto el PIM se corresponde con el metamodelo presentado en los trabajos [4]. [5] y [6], donde los autores extienden la propuesta basada en UML [7] para incorporar aspectos de seguridad en el diseño conceptual de ADs. El PSM se corresponde con nuestra extensión de CWM a nivel lógico.

En [8] ha sido empleado MDA para el desarrollo de ADs, eligiendo al metamodelo relacional de CWM [9] como PSM. El metamodelo relacional de CWM permite el intercambio entre BDs relacionales para la mayoría de los sistemas comerciales [10]. Sin embargo, las medidas de seguridad y auditoría no pueden ser modeladas en CWM porque este no proporciona constructores de modelado para representar la seguridad de datos tales como, los derechos de acceso para usuarios o roles [11]. La mayoría de los enfoques de control de acceso de datos están basados en las estructuras de metadatos de productos de software específicos [12], de manera que integrar la seguridad relacionada con los metadatos en CWM beneficia el soporte de la seguridad y facilita el establecimiento de un mecanismo de control de acceso estandarizado para ADs [11]. Según MDA no necesitamos los metadatos de un SGBD específico, sino un metamodelo que nos permita representar medidas de seguridad y auditoría para ADs a nivel lógico. Por lo tanto, en este artículo presentamos una extensión del metamodelo relacional de CWM usando sus propios mecanismos de extensión. Está fuera del alcance de este artículo la implementación de este metamodelo mediante una transformación MDA (vea el trabajo futuro en la sección VII).

El resto de este artículo se estructura como sigue. El trabajo relacionado es discutido en la sección II. El modelado MD seguro es introducido en la sección III. La sección IV muestra una visión general de CWM. La sección V presenta nuestra extensión del metamodelo relacional de CWM, en la siguiente sección, es decir sección VI proponemos un caso de estudio para mostrar los beneficios de nuestra extensión. Finalmente, la sección VII presenta las principales conclusiones y delínea el trabajo futuro inmediato.

Este proyecto ha sido parcialmente financiado por los proyectos METASIGN (TIN2004-00779) del Ministerio Español de Educación y las Ciencias, del gobierno regional de Valencia, DIMENSIONS (PBC-05-012-1) y DADS (PBC-05-012-2) FEDER y por la consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha (España).

E. Soler es profesor del departamento de informática de la universidad de Matanzas. Cuba (correo.: emilio.soler@umcc.cu).

J. Trujillo es profesor del departamento de sistemas y lenguajes informáticos de la universidad de Alicante. España (correo.: jtrujillo@dlsi.ua.es).

E. Fernández-Medina y M. Piattini imparten docencia y pertenecen al grupo ALARCOS de la universidad de Castilla-La Mancha. España (correo.: eduardo.fdzmedina@uclm.es, mario.piattini@uclm.es).

II. TRABAJO RELACIONADO

La literatura más relevante sobre este tema comprende varias iniciativas para incluir seguridad en el diseño de ADs. En [13] los autores describen un modelo prototipo para la seguridad en ADs basado en los metadatos que permite definir vistas de datos para cada grupo de usuarios, sin embargo, no permite especificar restricciones complejas de confidencialidad. Rosenthal y Sciore [14], extienden los permisos de SQL y crean un mecanismo de inferencias para establecer la seguridad en ADs. Otro intento es la arquitectura para Sistemas de Información Federados (SIF) y ADs que preserva la integración multinivel entre los SIF y los ADs [15]. Estas aproximaciones ([13], [14] y [15]) son atractivas pero solo se refieren a temas prácticos tales como la adquisición, almacenamiento y el control de acceso en el lado *On-Line Analytical Processing* (OLAP). Ninguna de ellas examina la representación de la seguridad en los niveles conceptual y lógico para el diseño de ADs.

Por otro lado, existen iniciativas más elaboradas que proponen modelos de autorización para el diseño de ADs. Por ejemplo, en [16] los autores proponen un concepto de seguridad para OLAP, que constituye un modelo de seguridad para ADs basado en roles. Priebe y Pernul [12] proponen una metodología de diseño similar a la clásica metodología de diseño para bases de datos (análisis de requisitos, diseño conceptual, lógico y físico) cubriendo los requisitos y la implementación en sistemas comerciales. En [17] los mismos autores extienden el modelo ADAPTed UML para la fase del diseño conceptual, especificando una metodología y un lenguaje de restricciones multidimensional para el modelado conceptual de la seguridad OLAP. En [18] los autores muestran que los privilegios de acceso para ADs y OLAP pueden ser expresados más intuitivamente que mediante las sentencias de los permisos de SQL, su modelo de control de acceso se centra específicamente en expresividad y usabilidad. Estas propuestas ([16], [12] y [17]) ofrecen modelos de seguridad a nivel conceptual por medio de restricciones de seguridad, pero básicamente tratan con operaciones OLAP, aunque las propuestas [12] y [17] constituyen una de las mejores referencias en esta área. Como resumen, estos trabajos implementan las reglas de seguridad consideradas en su aproximación conceptual en sistemas comerciales de bases de datos. Por otro lado, nosotros basamos nuestra propuesta en los trabajos [4]. [5] y [6], en los cuales los autores abogan por el diseño de medidas de seguridad en todas las fases del diseño de los ADs, desde el nivel conceptual hasta la implementación. Por consiguiente, en este artículo, extendemos formalmente a CWM de modo que nos permita transformar automáticamente todas las reglas de seguridad consideradas a nivel conceptual en una representación lógica del ADs.

Existen numerosas propuestas que extienden a CWM con diferentes propósitos: para el modelado lógico objeto-relacional orientado al almacenamiento de datos y el correspondiente proceso ETL [19], para la biblioteca universal de minería de datos que implementa algoritmos y métodos de minería de datos [20], para el registro de la traza de la evolución de la información de los metadatos para poder mantener consistencia durante la evolución de la metaclase [21], para representar e integrar los metadatos generados por datos y los metadatos de líneas de implementación [22] y para construir modelos conceptuales de limpieza y calidad de datos aplicables al contexto operacional y de *data warehousing* [23]. Sin embargo ninguna de las propuestas anteriores extiende el metamodelo relacional de CWM con aspectos de seguridad. Solo el trabajo presentado en [24] muestra cómo CWM puede ser adecuado para representar medidas de seguridad a nivel lógico, aunque no es formalmente extendido a través de los mecanismos de extensión que ofrece CWM.

III. MODELADO MULTIDIMENSIONAL

Las principales propiedades del modelado multidimensional son representadas por un perfil UML [7] que está basado en el modelado conceptual OO. En [5] el anterior perfil es reutilizado para el diseño de un modelo conceptual MD que permite clasificar tanto información como usuarios, para representar los principales aspectos de seguridad en el modelado conceptual de ADs. Por lo tanto, este perfil nos permite clasificar la información de seguridad que será usada en nuestro modelado conceptual de ADs. Para cada elemento del modelo (la clase *Fact*, la clase *Dimension*, *FactAttribute*, etc.), es definida su información de seguridad especificando una secuencia de niveles de seguridad, un conjunto de categorías de usuarios y un conjunto de roles de usuario. Son consideradas además, restricciones de seguridad para especificar seguridad en atributos. Estas restricciones y la información de seguridad indican las propiedades de seguridad que tienen los usuarios para permitir el acceso a la información. Este perfil es llamado *Secure Data Warehouses* (SECDW), su descripción es representada como un paquete de UML. Todas las restricciones anteriores *AuditRule*, *AuthorizationRule* y *SecurityRule* se modelan como notas de UML.

En el modelado seguro multidimensional (MD) considerado (*Secure Multidimensional Modeling*) las propiedades estructurales del modelado MD se representan mediante diagramas de clases UML que organizan la información en hechos y dimensiones de manera muy clara. Estos hechos y dimensiones se representan mediante las clases *SFact* y *SDimension* respectivamente, donde S significa la abreviación de la palabra *secure*. Con respecto a las *SDimensions*, cada nivel en la jerarquía de clasificación es especificado por una clase *SBase*. Una asociación de clases *SBases* especifica la relación entre dos niveles de una jerarquía de clasificación. Cada clase *SBase* también debe contener un atributo seguro OID (SOID) y un atributo *SDescriptor* (SD). La clase llamada *UserProfile* contendrá la información de todos los usuarios con acceso al modelo multidimensional. En la Fig. 4 de la sección 6 mostramos un ejemplo de modelado MD.

En la siguiente sección presentamos una descripción general de CWM, destacando los diferentes mecanismos para

su extensión.

IV. Una visión general de CWM

El principal propósito de CWM [9] es permitir el almacenamiento de los metadatos de inteligencia del negocio y el fácil intercambio entre las herramientas de almacenaje, las plataformas de almacenaje y los repositorios de metadatos en ambientes heterogéneos y distribuidos. CWM está basado en los tres estándares de la industria: i) UML - *Unified Modeling Language*, un estándar de OMG para el modelado, ii) MOF - *Meta Object Facility*, un estándar de OMG para el metamodelado y repositorio de metadatos, y iii) XMI – XML *Metadata Interchange*, un estándar de OMG para el intercambio.

El estándar UML define un rico lenguaje de modelado que es soportado por un amplio rango de herramientas de diseño gráfico. El estándar MOF (*Model Object Facility*) define un marco extensible para definir modelos para metadatos y ofrecer herramientas con interfases programables para almacenar y poder acceder a los metadatos en un repositorio. El estándar XMI (XML *Metadata Interchange*) permite que los metadatos sean intercambiados mediante un flujo o mediante ficheros con un formato estándar basado en XML. CWM ha sido diseñado para ajustarse al "modelo MOF" perteneciendo al metamodelo de la capa M2. Para más detalles sobre las diferentes capas de metamodelos de CWM referimos al lector a [9] y [10].

A. Organización de CWM

CWM está organizado en 21 paquetes separados, agrupados en cinco capas escalables por medio de roles similares (más detalles en [9]). Para nuestros propósitos nos centramos en la capa *Resource*, más precisamente en el paquete *Relational*. La capa *Resource* describe la estructura de las fuentes de datos que actúan como origen o destino en un intercambio. El paquete relacional es un metamodelo relacional que describe los metadatos correspondientes a las fuentes de datos relacionales. El acceso a estas fuentes de datos se realiza a través de una interfase relacional, como por ejemplo un SGBD relacional nativo, *Object* DB *Conectivity* o *Java* DB *Conectivity*.

B. Mecanismos de extensión de CWM

CWM proporciona mecanismos de extensión para construir metamodelos específicos. Según [9], hay dos técnicas generales para extender a CWM: i) uso del mecanismo general de extensión que ofrece UML *Object Model* mediante valores etiquetados y estereotipos. Esta aproximación es normalmente usada para extensiones menores (por ejemplo, para adicionarle atributos al modelo de objetos) que no son lo suficientemente significativas para requerir la creación de un modelo específico. ii) Las extensiones de modelos no normativa o extensiones modeladas (para más detalles consulte [10]) documentados como paquetes de metamodelos adicionales para extender el metamodelo de CWM. Esta propuesta es utilizada para extensiones más complejas, el propio CWM está construido siguiendo este tipo de extensión. Para representar requisitos de seguridad a nivel lógico necesitamos introducir nuevas clases y asociaciones, por ello, la extensión no normativa es el mecanismo preferido, pues no se trata de una simple extensión [10].

En la siguiente sección vamos a utilizar el mecanismo de extensión no-normativa para extender el metamodelo *Relational* de CWM.

V. La extensión relacional

Esta extensión el paquete relacional de CWM define nuevas clases para poder representar a nivel lógico todos los requisitos de seguridad y auditoría capturados durante la fase del modelado conceptual de ADs. Esta extensión será llamada metamodelo seguro relacional de ADs (SECRDW), la cual depende de los paquetes de CWM: *Relational*, *Core* y *DataTypes*.

En la Fig. 1 mostramos en color gris las nuevas clases que conforman el paquete SECRDW. La metaclase *SSchema* (*SCatalog*) especializa a *Schema* (*'Catalog*) para permitir un esquema (catálogo) seguro. *STable* y *UserProfile* especializan a la metaclase Table. *SColumn* es especializada en la metaclase *Column*. La tabla *UserProfile* es una tabla que almacena información de usuarios con acceso al sistema, estos derechos son especificados por *SecurityProperty* (*securityLevel*, *securityCompartment* y *securityRole*). *STable* y *SColumn* tienen asociada información de seguridad mediante *SecurityProperty* (*securityLevel*, *securityCompartment* y *securityRole*). *SecurityProperty* especializa a la metaclase *Class* del *Core*, con ella establecemos mediante los valores de *securityLevel*, *securityCompartment* y *securityRole* propiedades de acceso sobre tablas y columnas que el usuario debe cumplir para poder acceder.

AuditConstraint es útil para analizar y registrar los accesos a tablas y columnas realizados por los usuarios durante el uso del sistema. *ARConstraint* permite definir reglas para especificar las políticas de seguridad multinivel n tablas y columnas. *AURConstraint*, puede coexistir con *ARConstraint* para especificar el acceso a tablas y columnas, de esta manera permite definir modelos de seguridad mucho más elaborados. La metaclase *SecurityConstraint* lógicamente hereda propidades de la clase *Constraint* del *Core*. Los tipos de datos se estudian más a fondo en la siguiente sección.

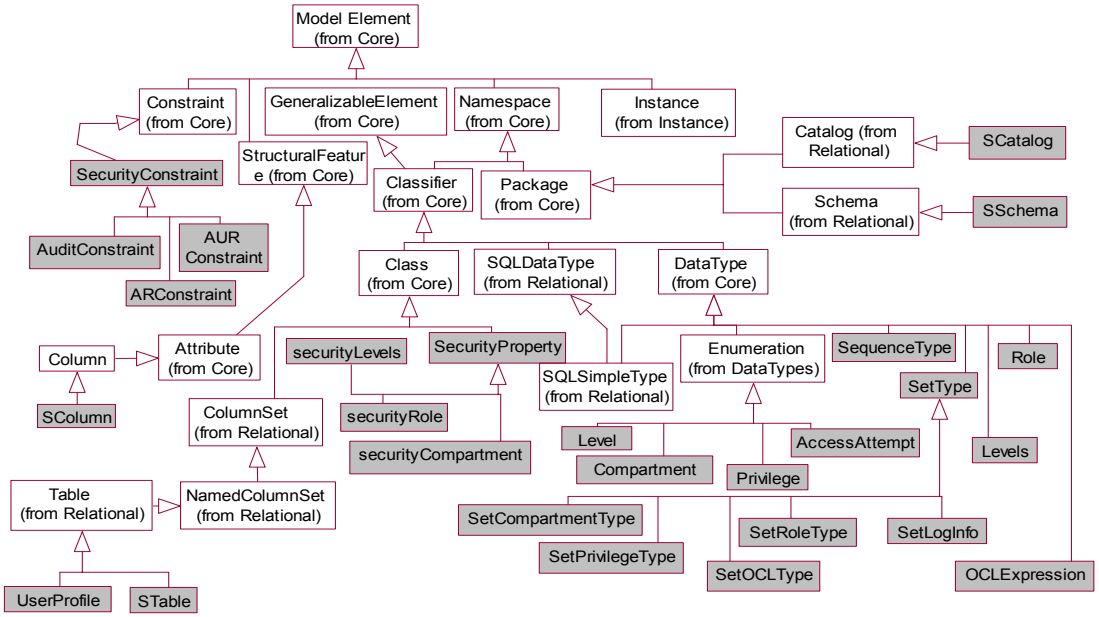

Fig. 1 Herencia en el metamodelo SECRDW

A. *Nuevos tipos de datos*

En general, los paquetes de CWM solo soportan atributos cuyos tipos de datos son considerados necesarios para el intercambio de información entre sistemas [9]. Para representar a nivel lógico información de seguridad y auditoría necesitamos nuevos tipos de datos. En la Fig. 2 aparecen en color gris las nuevas clases, que en un caso heredan de *DataType* y en otro de *Enumeration*. Estos nuevos tipos de datos son necesarios para modelar tanto propiedades de acceso (mediante *securityProperty*) como requisitos y restricciones (mediante *SConstraint*) a *STable*, *UserProfile* y *SColumn*.

La clase *SequenceType* representa un tipo CWM de dato que permite especificar todos los niveles de seguridad que pueden ser usados por los elementos del modelo (ordenados del menos al más restrictivo). *Level* es una enumeración ordenada compuesta por todos los niveles de seguridad que han sido considerados (*unclassified*, *confidential*, *secret* y *top Secret*). *Compartment* es una enumeración compuesta por todas las categorías de usuario que han sido consideradas. *Privilege* es una enumeración compuesta por todos los privilegios que han sido considerados (*read*, *insert*, *delete*, *update*, *all*). *Attempt* es una enumeración compuesta por los diferentes tipos de accesos que han sido considerados (*all*, *frustratedAttempt*, *successfullAccess*, *none*). *Levels* es un intervalo de niveles compuesto por *lowerlevel* y *upperlevel*. Si los niveles de seguridad coinciden *lowerlevel* y *upperlevel* coinciden, todas las instancias tendrán el mismo nivel de

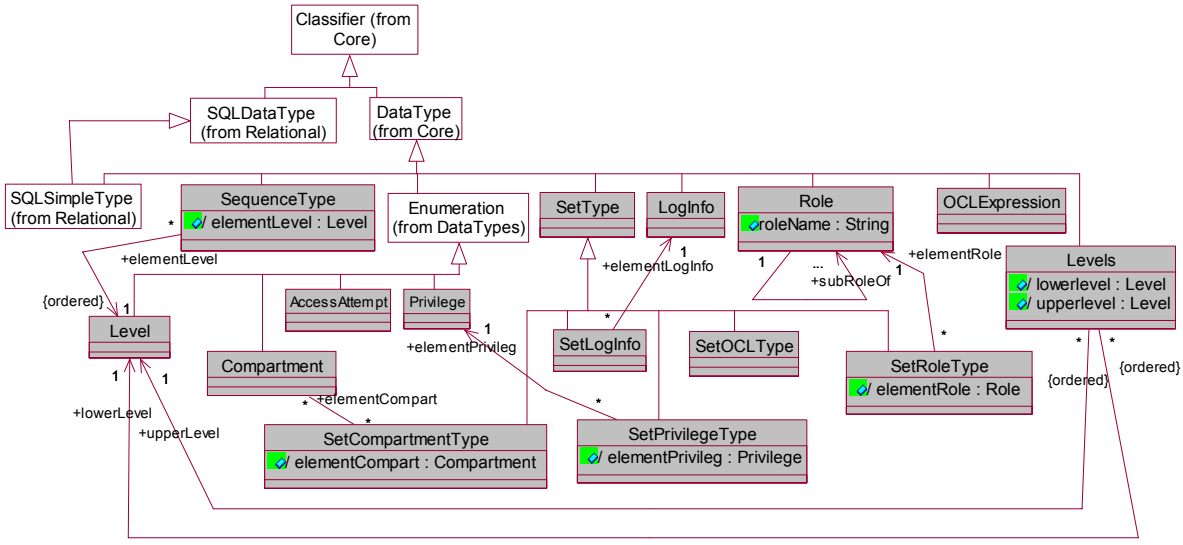

Fig. 2 Nuevos tipos de datos

seguridad, en caso contrario, el nivel específico será definido de acuerdo a una *securityConstraint*. *OCLExpression* define una expresión *Object Constraint Language* (OCL) que tienen que cumplir los usuarios del sistema en alguna condición. *Role* representa la estructura de roles jerárquica definida. *SetRoleType* especifica un conjunto de roles de usuarios, cada rol es la raíz de un subárbol en la jerarquía de roles considerada. *SetCompartmentType* representa al conjunto de *compartments* definidos por la organización. *SetPrivilegeType* especifica los privilegios que un usuario puede recibir o perder. *SetOCLType* especifica las tablas involucradas en una consulta realizada por un usuario del sistema. *SetLogInfo* especifica los elementos que se quieren registrar para una futura auditoría, normalmente se refiere al sujeto que solicita el acceso (*subjectID*), las tablas o columnas accedidas (*objectID*), la operación solicitada (*action*), el tiempo de la solicitud (*time*) y el responsable del control de acceso (*response*).

B. *Nuevas clases y asociaciones*

El paquete SECRDW define los contenedores *SCatalog* y *SSchema* respectivamente. *SCatalog* es un repositorio local de metadatos que describe todas las bases de datos mantenidas por el motor del SGBD relacional. *SSchema* es una colección de *STables* y *securityProperties* que garantiza la seguridad a nivel de modelo. *ColumnSet* representa cualquier forma de datos relacional. *STable* y *UserProfile* heredan de *Table*, ambos contienen *Columns*. Obsérvese en la Fig. 3 que la tabla *UserProfile* contiene columnas para especificar las propiedades de acceso (*securityProperty*) que tiene el usuario. *UserProfile* a diferencia de *STable* es única y no tiene asociación con el resto de las tablas del sistema. *ForeignKey* asocia columnas de una tabla con columnas de otra tabla. La clase *PrimaryKey* hereda de *UniqueConstraint*. Las metaclases *PrimaryKey* y *ForeignKey* son dominadas por la metaclase *STable* (vea la Fig. 3). Para representar medidas de seguridad y auditoría en el nuevo metamodelo, adicionamos algunas metaclases. La metaclase *SecurityProperty* hereda de la metaclase *Class* (del *Core*) y se especializa como las metaclases *SecurityLevels*, *SecurityCompartments* y *SecurityRoles*. Además, para representar restricciones de seguridad, reglas de autorización en el nuevo metamodelo adicionamos las clases *ARConstraint* y *AURConstraint* que heredan de *SecurityConstraint*. Para especificar restricciones dependiendo de una información particular de un usuario o un grupo de usuarios, introducimos la metaclase *UserProfile*. Obsérvese en la Fig. 3 las nuevas clases que hemos adicionado al paquete relacional de CWM, así como las nuevas asociaciones entre ellas. Las nuevas clases contienen atributos cuyos tipos han sido especificados en la Fig. 2. Estos atributos permiten representar toda la información de seguridad capturada durante la etapa del modelado conceptual de los ADs.

En la siguiente sección vamos a mostrar cómo usamos la extensión realizada en la representación a nivel lógico de un modelo MD seguro.

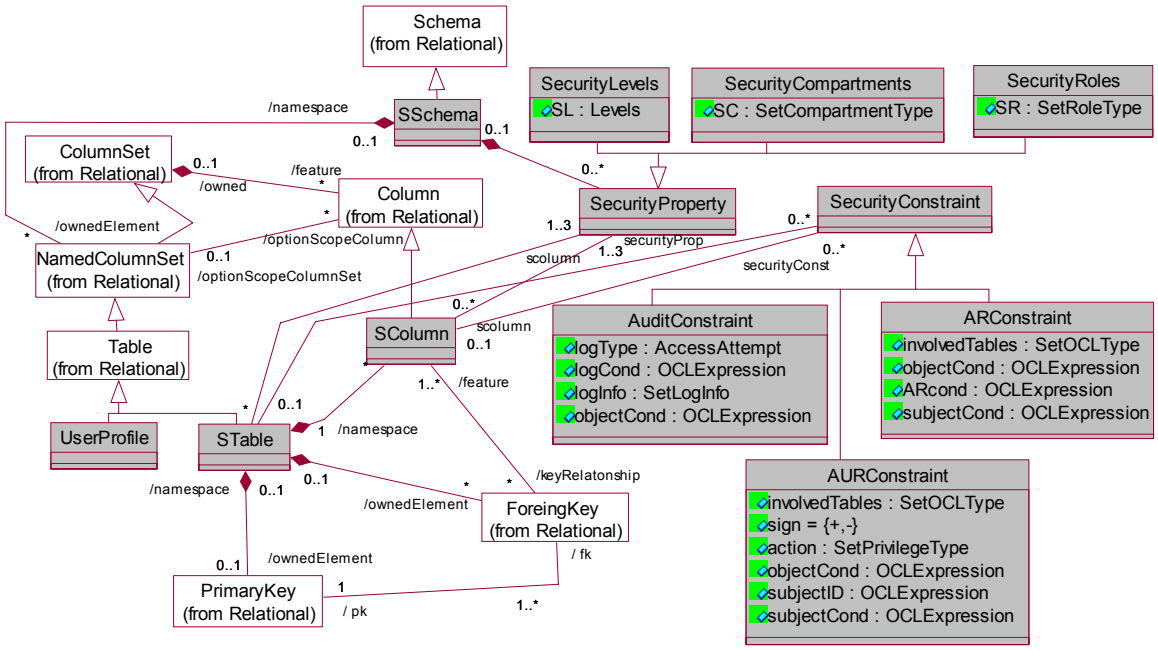

Fig. 3 Nuevas clases y asociaciones

VI. CASO DE ESTUDIO

En esta sección, aplicamos nuestra extensión del metamodelo relacional de CWM en el contexto de un consorcio farmacéutico. El consorcio administra varias farmacias que brindan diferentes tipos de servicios a la comunidad y quiere controlar todo lo referente a las ventas de medicamentos mediante las recetas médicas. Para definir una clasificación de datos y usuarios que es típico para este tipo de negocio (la más general es *Pharmacy Employee*, el cual es especializado en los

roles *Pharmacist* y *nonPharmacist*, los cuales a su vez son especializados en los roles *assistant* y *technicians* en el primer caso, y como *maintenance* y *administrative* en el último). Hemos considerado los siguientes niveles de seguridad: *confidential*, *secret* y *topSecret*. Dentro de la empresa existe un grupo de *farmacovigilancia*, que vela por la seguridad del uso de ciertos fármacos y un comité que vela por la salud de sus clientes, por ello hemos definido cuatro *securityCompartments*: *pharmacovigilanceCenter*, *generalCenter*, *healthOversightCenter* y *comercialManagerCenter*.

A. *Definiendo el PIM*

En la Fig. 4 mostramos una instancia del metamodelo SECDW, es decir, una instancia de nuestro SMD PIM, que ilustra una parte del ADs que se requiere para el problema anterior. La *SFact Sales_Prescription* (estereotipo *SFact*) contiene toda la información de las ventas en una o más farmacias, y puede ser accedida por usuarios que tienen el nivel de seguridad *secret* o *topSecret*, desempeñan el rol *Administrative* o *Pharmacist* y pertenecen a uno de los *compartments pharmacovigilance Center*, *healthOversightCenter* o *comercialManagerCenter*.

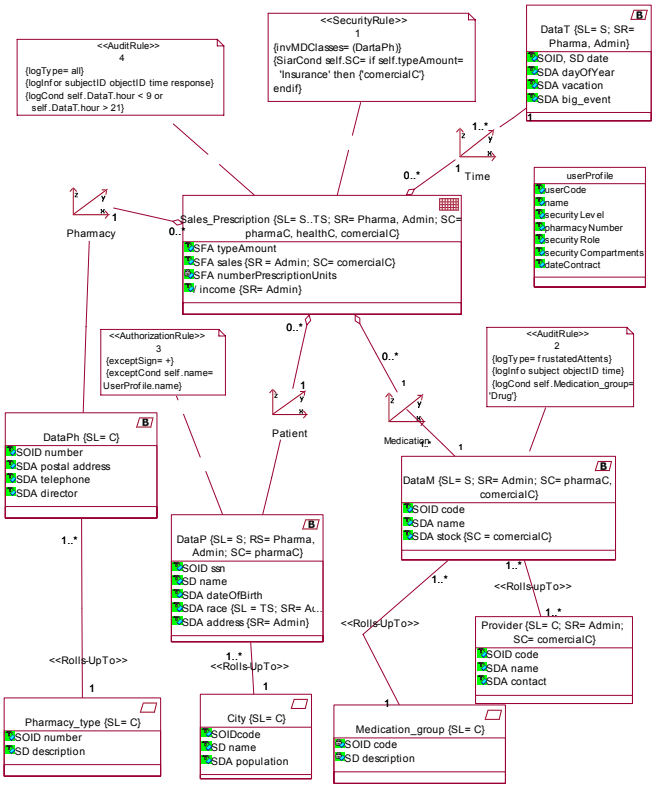

Fig. 4 Modelo MD seguro

El atributo sales solo puede accedido por los usuarios que desempeñan el rol *administrative* (valor etiquetado SR del atributo *sales*) y pertenece al *compartment comercialManagerCenter*, por consiguiente el acceso a este atributo está prohibido para otros usuarios. El atributo *income* puede ser accedido solo por usuarios que desempeñan el rol *administrative* (valor etiquetado SR del atributo *income*). Otras clasificaciones estáticas para los usuarios del modelo conceptual representado en la Fig. 4 son: La *SFact Sales_Prescription* contiene cuatro dimensiones (*Pharmacy*, *Patient*, *Medication* y Time) que contienen jerarquías de *SBases*. El acceso a es tas jerarquías de *SBases* se establece de manera análoga a como se hizo con la *SFact*. La clase *UserProfile* contiene la información de todos los usuarios que tendrán acceso a este modelo multidimensional. Cada usuario tiene asociado una *securityLevels* (SL), un *securityRoles* (SR) y un *securityCompartments* (SC).

Varias restricciones de seguridad han sido definidas usando los estereotipos, los valores etiquetados y las restricciones especificadas. Los siguientes párrafos se corresponden con las notas 1 y 2 de la Fig. 4:

1. Para cada instancia de la case de hecho *Sales_Prescription*, si el tipo de payment es a través de un seguro el *securityCompartment* será *comercialManagerCenter* (valor etiquetado SC). Esta restricción es solo aplicada si el usuario realiza una consulta cuya información viene de *DataPh*.
2. Se quiere registrar el sujeto, objeto y tiempo para cada intento de acceso frustrado a la descripción de las drogas en *DataM* (*Data Medication*).

B. *Definiendo el PSM*

A partir del PIM representado en la Fig. 4, aplicamos relaciones QVT [25] para obtener una instancia del metamodelo SECRDW, es decir, una instancia de nuestro SMD PSM (vea la Fig. 5) que representa un esquema *snowflake* para ADs a nivel lógico. Esta instancia del PSM se corresponde con una instancia del metamodelo extendido en la sección V-B. Con la extensión de CWM formalizamos los conceptos para una plataforma relacional, aunque ellos son cercanos al modelado multidimensional cuando el paradigma lógico utilizado no fuese una representación del ADs tan parecida a la del modelo relacional, entonces la transformación del modelo conceptual al relacional resulta muy interesante en lo que respecta a la seguridad.

El hecho *Sales_Prescription* es representado en la Fig. 5 mediante la *STable Sales_Prescription*. En esta tabla representamos todas sus columnas, así como también toda la información de seguridad asociada que restringe el acceso a la propia tabla y a sus columnas. Cada *SBase* es transformada en una *STable*. La clase *UserProfile* es transformada en la tabla *UserProfile*. Para representar la relación *many-to-many* entre las tablas *DataM* y *Provider* hemos creado una tabla puente. Las informaciones de seguridad (SL, SR y SC) representadas en la tabla *Sales_Prescription* de la Fig. 5 constituyen instancias de la clase *SecurityProperty* que aparece en la Fig. 3. Esta información de seguridad es modelada a nivel lógico en el encabezamiento de la propia tabla (vea la Fig. 5).

Las restricciones seguridad *SecurityRule* 1, *AuditRule* 2, *AuthorizationRule* 3 y *AuditRule* 4 que aparecen en la Fig. 4 son transformadas en instancias son representadas en la Fig. 5 como notas UML con los nombres *ARConst* 1, *AudConst* 2,

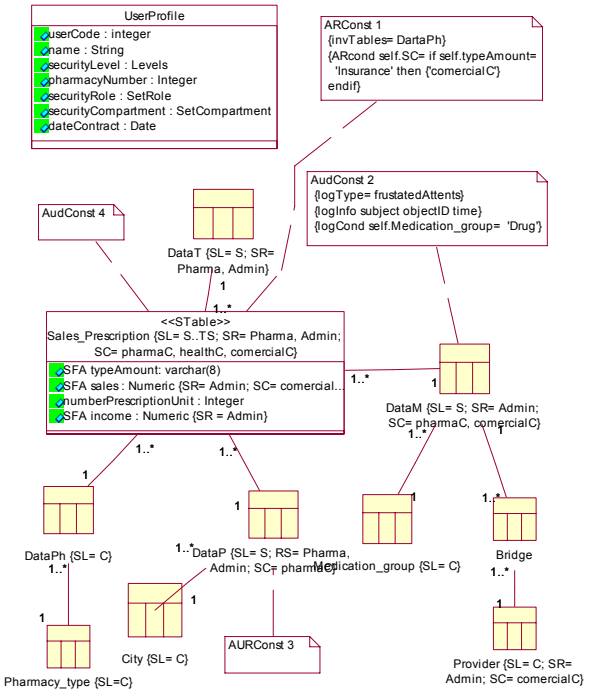

Fig. 5 Esquema snowflake a nivel lógico

AURConst 3 y *AudConst* 4, sin embargo, para hacer la Fig. 5 más comprensible, solo mostramos los atributos de las clases *ARConst* 1 y *AudConst* 2, los que constituyen instancias de las clases representadas en la Fig. 2.

```
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. 6 Implementando restricciones en Oracle

C. *Ejemplo de código en Oracle*

Para finalizar nuestro caso de estudio mostramos algunas implementaciones de los aspectos de seguridad modelados en la instancia del metamodelo SECRDW que aparecen en la Fig. 5. Hemos elegido la versión 10 del SGBD *Oracle* ya que brinda facilidades para la seguridad y la auditoría mediante sus componentes llamados *Oracle Label Security* (OLS10g), *Virtual Private Databases* (VPD) y *Oracle Fine Grained Auditing* (FGA). Para explicar los aspectos de seguridad que contempla nuestra extensión primeramente creamos una política de seguridad llamada *'MyPolicy'* y *Levels*, *compartments* y grupos de jerarquías válidos.

En la Fig. 6 a) mostramos cómo el usuario *User1* satisface las propiedades de seguridad para la tabla *Sales_Prescription*. Fig. 6 b) muestra como definimos y establecemos la información de seguridad para la tabla *Sales_Prescription* a través de las funciones etiquetadas de OLS, aunque no podemos considerar seguridad a nivel de columna. *ARConst* 1 es implementada mediante la función etiquetada que aparece en la Fig. 6 c). FGA nos permite definir e implementar la *AudConst* 2 (vea la Figura 6d)), aunque no podemos implementar *logType* ni *logCond* pues FGA no nos permite elegir el tipo de auditoría *logType* ni la condición referente a columnas de tablas diferentes (*logCond*).

VII. Conclusiones

En este trabajo hemos presentado una extensión del paquete relacional de CWM para representar a nivel lógico todos los requisitos de seguridad y auditoría capturados durante la fase del modelado conceptual de ADs. Esta propuesta está alineada con MDA, la que nos permite contemplar aspectos de seguridad en todas las fases de diseño de los ADs, desde el PIM con la propuesta del modelado conceptual basado en UML, así como su correspondiente representación a nivel lógico basada en el presente trabajo. Para mostrar la validez de nuestra extensión hemos desarrollado un caso de estudio que ilustra cómo modelamos a nivel lógico todas las medidas de seguridad y auditoría representadas durante la fase del modelado conceptual de los ADs. Nuestro trabajo futuro inmediato consiste en rehusar la implementación desarrollada en *Borland Together Architect* que aparece en [26] para conseguir una transformación MDA automática entre el PIM seguro y el PSM seguro desarrollado en este artículo.

Referencias

[1] Dhillon, G., Backhouse, J. Information Systems Security Management in the New Millenium, Communications of the ACM, vol. 43, no. 7, pp. 125-128, 2000.

[2] Devanbu, P., Stubblebine, S. Software Engineering for Security: a Roadmap, The Future of Software Engineering, ACM Press, New York, pp. 227-239, 2000.

[3] OMG, MDA Guide Version 1.0.1, Disponible: http://www.omg.org/cgi-bin/doc?omg/03-06-01, 2003.

[4] Fernandez-Medina, E., Trujillo, J., Villarroel, R., Piattini, M., Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses, Decision Support Systems, vol. 42, no. 3, pp. 1270-1289, 2006.

[5] Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M., Developing Secure Data Warehouses with a UML Extension, Information Systems, vol. 32, no. 6, pp. 826-856, 2007.

[6] Villarroel, R., Fernández-Medina, E., Piattini, M. A UML 2.0/OCL Extension for Designing Secure Data Warehouses, Journal of Research and Pract. in Inf. Tech., vol. 38, no. 1, pp. 31-43, 2006.

[7] Luján, S., Trujillo, J., A UML Profile for Multidimensional Modeling in Data Warehouses, Data Knowledge and Engineering, vol. 59, no. 3, pp. 725-769, 2006.

[8] Mazón, J. N., Trujillo, J., An MDA Approach for the Development of Data Warehouses," Decision Support Systems, vol. 45, no.1, pp. 41-58, 2008.

[9] OMG, Common Warehouse Metamodel Specification 1.1, Disponible: http://www.omg.org/cgi-bin/doc?formal/03-03-02.
[10] Poole, J., Chang, D., Tolbert, D., Mellor, D., Common Warehouse Metamodel Developer´s Guide. Indianapolis, Indiana: Wiley Publishing, Inc, 2003.
[11] Melchert, F., Schwinn, A., Herrmann, C., Winter, R., Using Reference Models for Data Warehouse Metadata Management, Eleventh Americas Conference on Information Systems, pp. 1316-1326, Omaha, USA, august 11-14, 2005.
[12] Priebe T., Pernul, G., Towards OLAP Security Design - Survey and Research Issues, DOLAP'00, pp. 33-40, McLean, VA. USA, november 10, 2000.
[13] Katic, N., Quirchmayr, G., Schiefer, J., Stolba, M., Tjoa, A. M., A Prototype Model for Data Warehouse Security Based on Metadata, DEXA'98, pp. 300-308, Vienna, Austria, august 26-28, 1998.
[14] Rosenthal A., Sciore, E., View Security as the Basic for Data Warehouse Security, DMDW'00, pp. 8.1-8.8, Stockholm, Sweden, june 5-6, 2000.
[15] Saltor, F., Oliva, M., Abelló, A., Samos, J., Building Secure Data Warehouse Schemas from Federated Information Systems, in Heterogeneous Information Exchange and Organizational Hubs., D. T. Bestougeff, Ed.: Kluwer Academic, 2002, pp. 123-134.
[16] Kirkgöze, R., Katic, N., Stolba, N., Tjoa, A. M., A Security Concept for OLAP, DEXA'97, pp. 619-626 ,Toulouse, France, 1997.
[17] Priebe T., Pernul, G., A Pragmatic Approach to Conceptual Modeling of OLAP Security, ER'01, pp. 311-324, Yokohama, Japan, november 27-30, 2001.
[18] Essmayr, W., Weippl, E., Lichtenberger, F., Winiwarter, W., Mangisengi, O., An Authorization Model for Data Warehouses and OLAP, Workshop on Security in Distributed Data Warehousing, in conjunction with SRDS'01, pp. 9-13, New Orleans, Louisiana, USA, october 2001.
[19] Maier, T., A Formal Model of the ETL process for OLAP-Based Web Usage Analysis, KDD Workshop on Web Mining and Web Usage Analysis (WebKDD'04), pp. 23-34, Seatle, Washington, USA, 2004.
[20] Thess M., Bolotnicov, M., XELOPES Library Documentation Version 1.2.3, Prudsys AG, 2004.
[21] Zhao X., Huang, Z., A Formal Framework for Reasoning on Metadata Based on CWM, ER'06, pp. 371-384 Tucson, AZ, USA, november 6-9, 2006.
[22] Santana A. S., Moura, A. M. d. C., Metadata to Support Transformations and Data & Metadata Lineage in a Warehousing Environment, Data Warehousing and Knowledge Discovery (DAWAK'04), pp. 249-258, Zaragoza, Spain, 2004.
[23] Amaral G. C. M., Campos, M. L. M., AQUAWARE: A Data Quality Support Environment for Data Warehousing, XIX Simpósio Brasileiro de Banco de Dados (SBBD'04), pp. 121-133, Brasília, DF, Brasil, 2004.
[24] Soler, E., Villarroel, R., Trujillo, J., Fernández-Medina, E., Piattini, M., Representing Security and Audit Rules for Data Warehouses at the Logical Level by using the Common Warehouse Metamodel, ARES'06, pp. 914-921 , Vienna, Austria, april 20-22, 2006.
[25] OMG, MOF 2.0 QVT Final Adopted Specification, Disponible en: http://www.omg.org/cgi-bin/doc?ptc/2005-11-01, 2005.
[26] Mazón, J. N., Pardillo, J., Trujillo, J., Applying transformations to Model Driven Data Warehouses, DOLAP'06, pp. 57–66, Krakow, Poland, september 4-8, 2006.

Emilio Soler es graduado de matemática de la Universidad Pedagógica de Matanzas (Cuba) y profesor auxiliar del departamento de Informática de la Universidad de Matanzas (Cuba). Soler actualmente es estudiante de doctorado del departamento de sistemas informáticos de la universidad de Alicante (España). Sus actividades de investigación incluyen seguridad en almacenes de datos, MDA y sistemas de información. Ha publicado y presentado artículos en conferencias nacionales e internacionales de ciencias de la computación tales como ICCSA, ARES, JISBD, WOSIS e IDEAS.

Juan Trujillo es profesor en la Escuela de Informática de la Universidad de Alicante, España. Trujillo obtuvo su Doctorado en Informática en la Universidad de Alicante (España) el año 2001. Sus intereses de investigación incluyen modelado de bases de datos, diseño conceptual de almacenes de datos, bases de datos multidimensionales, OLAP, y análisis y diseño orientado a objetos con UML. Ha publicado artículos en conferencias internacionales y revistas tales como ER, UML, ADBIS, CaiSE, WAIM, *Journal* de Gestión de Bases de Datos (JDM) e IEEE *Computer*. Participa como miembro de Comité de Programa de varios talleres y conferencias tales como ER, DOLAP, DSS, y SCI. También ha participado como revisor de varias revistas tales como JDM, KAIS, ISOFT y JODS.

Eduardo Fernández-Medina es Doctor y Master en Informática. Es profesor asistente en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha (España). Su actividad de investigación es seguridad en bases de datos, almacenes de datos, servicios Web y sistemas de información. Y también en métricas de seguridad. Es coeditor de varios libros y capítulos de libros en estos temas, y tiene varias docenas de artículos en conferencias nacionales e internacionales. Participa en el grupo de investigación ALARCOS del Departamento de Informática en la Universidad de Castilla-La Mancha (España). Pertenece a varias asociaciones de investigación y profesionales (ATI, AEC, AENOR, IFIP WG11.3 etc.).

Mario Piattini es Master y Doctor en Informática por la Universidad Politécnica de Madrid. Auditor de Sistemas de (*Information System Audit and Control Association*). Actualmente es Catedrático de Universidad en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha (España). Autor de varios libros y artículos sobre bases de datos, ingeniería de software y sistemas de información. Pertenece al grupo de investigación ALARCOS del Departamento de Informática en la Universidad de Castilla-La Mancha (España). Sus intereses de investigación son: diseño de bases de datos avanzadas, calidad de bases de datos, métricas de software, métricas orientadas a objeto, mantenimiento de software.