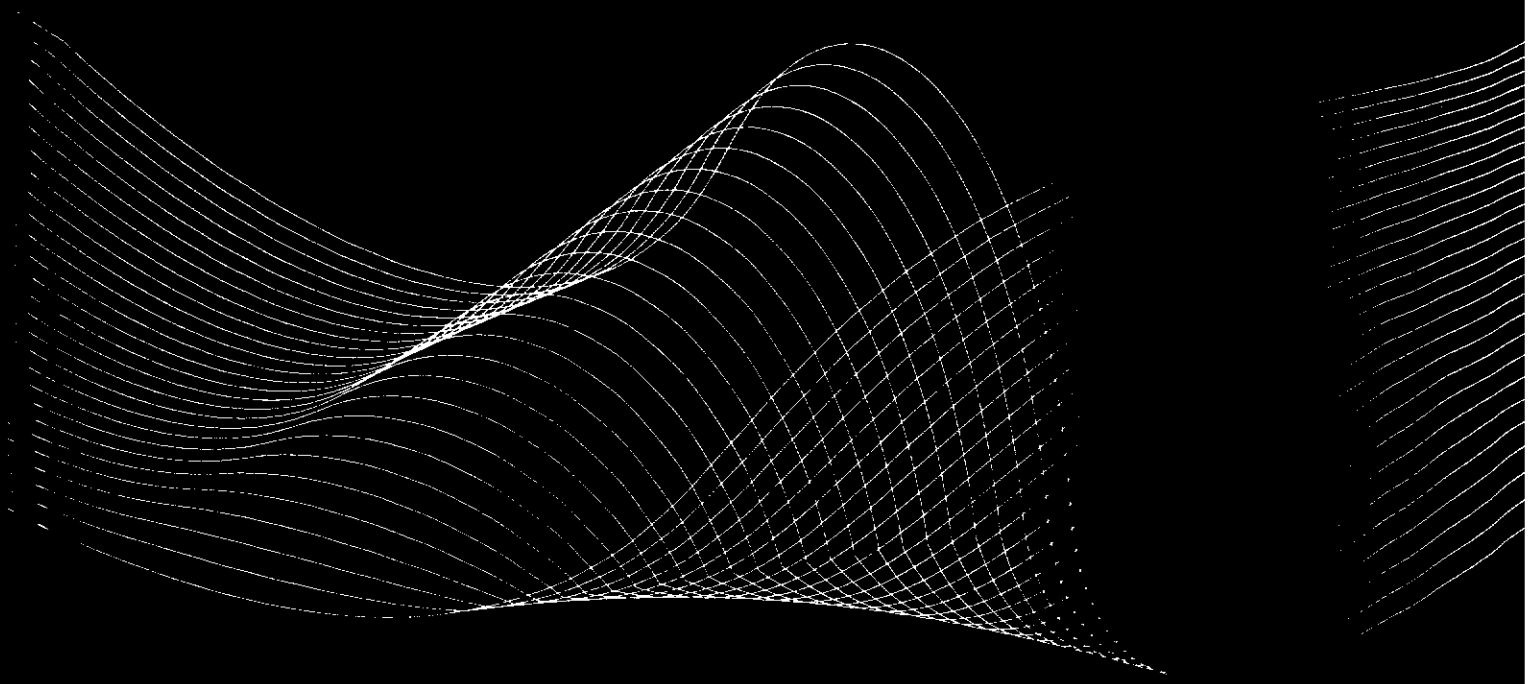
▶**Del 24 al 28 de abril de 2006**
La Plata | Buenos Aires | Argentina

ideas.06

9° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software

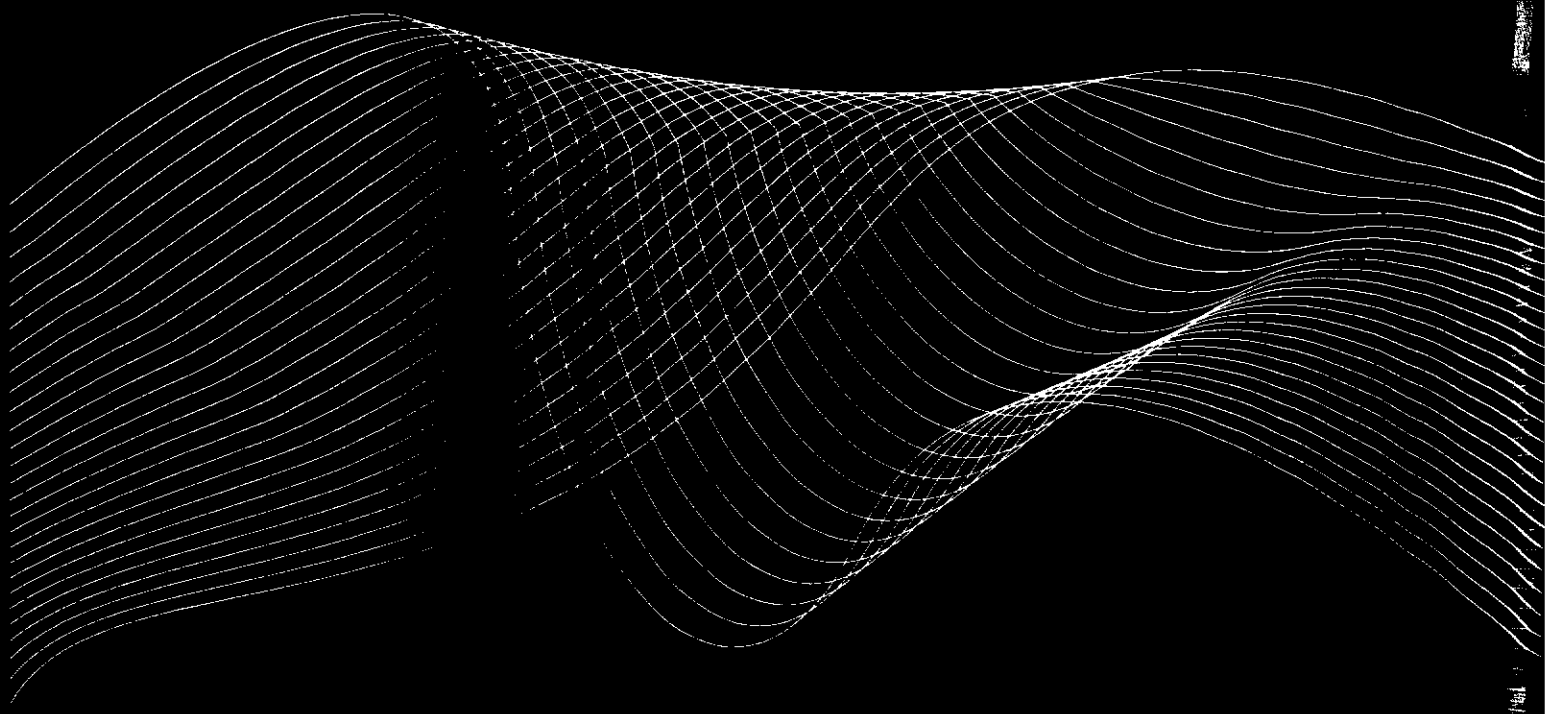


Auspician:

Centro Latinoamericano de Estudios en Informática

Microsoft

SADIO



ISBN-10:950-34-0360-X ISBN-13 :978-950-34-0360-0

Actas

IDEAS'06

9º Workshop Iberoamericano de
Ingeniería de Requisitos y
Ambientes de Software

24 al 28 de Abril de 2006
La Plata, Argentina

Editores
Jaelson Castro
Luca Cernuzzi
Silvia Gordillo

Actas del 9º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software IDEAS'06

ISBN-10: 950-34-0360-X
ISBN-13: 978-950-34-0360-0

PRÓLOGO

El presente volumen contiene los trabajos aceptados y presentados en el IX Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software – IDEAS 06 celebrado en la ciudad de La Plata, Argentina, del 24 al 28 de Abril de 2006.

Muy brevemente quisiéramos mencionar los números y recordar los pasos que dimos juntos para el proceso de evaluación.
Originalmente se han postulado 107 resúmenes que finalmente han resultado en la postulación de 100 artículos. Cada articulo ha sido asignado a 3 revisores y se han discutido las eventuales discrepancias con respecto a un mismo articulo para alcanzar, donde fuera posible, un consenso. Finalmente, en la difícil decisión de aceptación/rechazo hemos adoptado el criterio de aceptar como artículos todos y solo aquellos que obtuvieron un promedio y un promedio ponderado (considerando el nivel de experiencia del evaluador) igual o superior a 4.5 sobre 7; para una sesión especial se han aceptado adicionalmente, en calidad de poster, aquellos artículos que obtuvieron un promedio y un promedio ponderado igual o superior a 4 sobre 7.
Así, el resultado final ha sido que de los 100 artículos originalmente postulados, 33 han sido aceptados para su presentación integral y otros 8 para su presentación como poster.

Consideramos que el nivel de postulación es síntoma de un creciente interés por el workshop y nos hace pensar que él mismo se está afianzando cada vez más como un foro importante de Ingeniería de Software (en términos amplios) para Ibero América. Al mismo tiempo, el porcentaje de aceptación indica un nivel de exigencia interesante para la realidad Iberoamericana que puede ayudar a posicionar mayormente el workshop para obtener un mayor reconocimiento en las calificaciones de los autores en sus respectivos países.

Evidentemente, todo esto no hubiera sido posible sin la valiosa colaboración de distintos actores a quien van nuestros más sinceros agradecimientos. Entre ellos, cabe mencionar a los autores por su esfuerzo en investigación, los revisores, ya sean miembros del comité de programa o adicionales, que han tenido que cumplir un esfuerzo particularmente intenso por la cantidad de postulaciones y el proceso adoptado, los miembros del comité organizador, los disertantes de los tutoriales, el comité de conducción de

IDEAS y todas las demás personas que de distintas formas han colaborado a que este evento pueda ser un ocasión importante para la comunidad científica en nuestra región.

Esperamos puedan disfrutar de IDEAS 06 así como de la calida acogida de La Plata.

Silvia Gordillo	**Jaelson Castro**	**LucaCernuzzi**
Presidente Conferencia	Co-Pte. Comité de Programa	Co-Pte. Comité de Programa

Comité de Programa

Presidencia de la Conferencia

Silvia Gordillo, Universidad Nacional de La Plata, Argentina

Co-Presidentes del Comité de Programa

Jaelson Castro , Universidad Federal do Pernabuco, Brasil
Luca Cernuzzi, Universidad Católica "Nuestra Señora de la Asunción", Paraguay

Miembros del Comité de Programa

Amador Duran, U. Servilla, España
Alexandre Vasconcelos, U.F. Pernambuco,Brasil
Antonio Brogi, U. de Pisa, Italia
Antonio Vallecillo, U. de Málaga, España
Carme Quer, U. P. de Catalunya, España
Cecilia Bastarrica, U. de Chile, Chile
Daniel Riesco, U. de San Luis, Argentina
Ernesto Cuadros, UNAS, Perú
Ernesto Pimentel, U. de Málaga, España
Francisco Ruiz, U. Castilla de la Mancha, España
Guilherme H. Travassos, UFRJ, Brasil
Hernán Astudillo, UTFSM, Chile
Joao Falcão e Cunha, U.do Porto, Portugal
João Araújo, U. Nova de Lisboa, Portugal
José Carlos Maldonado, U. de São Paulo, Brasil
Júlio Leite, PUC-Rio, Brasil
Luis Olsina, U. Nacional de La Pampa, Argentina
Marcelo Campo, UNICEN, Argentina
Marcelo Frias, UBA, Argentina
Mario Piattini, U. Castilla-La Mancha, España
Miguel Katrib, U. de La Habana, Cuba
Óscar Pastor, U. Politécnica de Valencia, España
Pere Botella, U. Politécnica de Catalunya, España
Raúl Monje, UTFSM, Chile
Ricardo de A. Falbo, UFES, Brasil
Tereza Kirker, UNIMEP, Brazil
Xavier Franch, U. P. de Catalunya, España

Revisores adicionales

Gabriel Infante-López - UBA, Argentina
Diego Garbervetsky – UBA, Argentina
Carlos Lopez Pombo - UBA, Argentina
Nazareno Aguirre – UBA, Argentina
Juan Pablo Galeotti – UBA, Argentina
Alejandro Vaisman – UBA, Argentina
Silvia Gordillo – U. Nacional de La Plata, Argentina
Gustavo Rossi - U. Nacional de La Plata, Argentina
Claudia Pons - U. Nacional de La Plata, Argentina
Maria de los Angeles Martín - U. Nacional de La Pampa, Argentina
Hernán Molina - U. Nacional de La Pampa, Argentina
Guillermo Covella - U. Nacional de La Pampa, Argentina
Rodrigo de Oliveira Spinola - UFRJ, Brazil
Marco Pereira Araujo - UFRJ, Brazil
Wladmir Araujo Chapetta – UFRJ, Brazil
Tayana Uchoa Conte – UFRJ, Brazil
Ana Candida Cruz Natali - UFRJ, Brazil
Jobson Massolar da Silva - UFRJ, Brazil
Paula Gomes Mian - UFRJ, Brazil
Carla Silva - UFPE, Brazil
Alex Sandro Gomes – UFPE, Brazil
Sandro Ronaldo Bezerra – UFPE, Brazil
Fernanda Ma. R. Alencar – UFPE, Brazil
Regiane Andrade Brito – UFPE, Brazil
Carlos Menezes de Albuquerque – UFPE, Brazil
Carina Alves – UFPE, Brazil
Thaizel Fuentes - U. de La Habana, Cuba
Abel Marrero - U. de La Habana, Cuba
Marcello Visconti - Universidad Técnico Federico Santa María, Chile
Horst H. von Brand - UTFSM, Chile
Benjamin Piwowarski - U. de Chile, Chile
Sara Corfini – U. de Pisa, Italy
Razvan Popescu - U. de Pisa, Italy
Massimo Cossentino - ICAR – CNR, Italy
Enrique Vargas – U. Católica "Nuestra Señora de la Asunción", Paraguay
Vicente González – U. Católica "Nuestra Señora de la Asunción", Paraguay
Magalí González – U. Católica "Nuestra Señora de la Asunción", Paraguay
Alfredo Paz-Valderrama - UNAS, Perú
Percy Pari Salas – UNAS, Perú

Alfredo Paz-Valderrama - UNAS, Perú
Raul Romero - U. de Málaga ,Spain
Nathalie Moreno - U. de Málaga ,Spain
Manuel F. Bertoa - U. de Málaga ,Spain
José Luis Pastrana - U. de Málaga, Spain
Juan Pablo Carvallo - U. Politécnica de Catalunya, Spain
Xavier Burgués - U. Politécnica de Catalunya, Spain
Enric Mayol - U. Politécnica de Catalunya, Spain
Nuria Rodríguez - U. Politécnica de Catalunya, Spain
Tomás Aluja - U. Politécnica de Catalunya, Spain
Emilio Insfrán – U. Politécnica de Valencia, Spain
Vicente Pelechano – U. Politécnica de Valencia, Spain
Hugo Estrada Esquivel - U. Politécnica de Valencia, Spain
Pedro Valderas - U. Politécnica de Valencia, Spain
Javier Muñoz - U. Politécnica de Valencia, Spain
Marta Ruiz - U. Politécnica de Valencia, Spain
Juan Sánchez Díaz - U. Politécnica de Valencia, Spain
Nelly Condori-Fernandez – U. Politécnica de Valencia, Spain
Alicia Martínez Rebollar - U. Politécnica de Valencia, Spain
Silvia Abrahão - U. Politécnica de Valencia, Spain
Victoria Torres - U. Politécnica de Valencia, Spain
Brian Matthews - CCLRC Rutherford Appleton Laboratory
Simon Lambert - CCLRC Rutherford Appleton Laboratory

Comité Organizador

Co-Presidentes del Comité Organizador

Claudia Pons, Universidad Nacional de La Plata, Argentina
Roxana Giandini, Universidad Nacional de La Plata, Argentina

Miembros

Claudia Banchoff, Universidad Nacional de La Plata, Argentina
Gabriela Pérez, Universidad Nacional de La Plata, Argentina
Ileana Carrizo, Universidad Nacional de La Plata, Argentina

Auspicios

Universidad Nacional de La Plata (UNLP)
Sociedad Argentina de Informática (SADIO)
Centro Latinoamericano de Estudios en Informática (CLEI)
Microsoft Argentina

Diseño Gráfico y Comunicación Visual

Axel Hochegger

INDICE

Una Aproximación Dirigida por Modelos para el Diseño de Bases de Datos XML Seguras

Belén Vela[1], Eduardo Fernández-Medina[2], Esperanza Marcos[1] y Mario Piattini[2]

(1) Grupo Kybele. Departamento de Lenguajes y Sistemas Informáticos
Universidad Rey Juan Carlos
C/ Tulipán, s/n - 28933 Móstoles, Madrid, España
{Belen.Vela, Esperanza.Marcos}@urjc.es
(2) Grupo Alarcos. Departamento de Tecnologías y Sistemas de Información
Centro Mixto de Investigación y Desarrollo de Software UCLM-Soluziona
Universidad de Castilla-La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real, España
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Resumen. En este artículo se propone una aproximación metodológica para el desarrollo dirigido por modelos de bases de datos (BD) XML seguras. Esta propuesta se enmarca dentro de MIDAS, una metodología dirigida por modelos para el desarrollo de Sistemas de Información Web (SIW) basada en la arquitectura MDA (*Model Driven Architecture*) propuesta por el OMG (*Object Management Group*). El proceso de desarrollo de una BD XML en MIDAS propone usar como modelo independiente de plataforma (PIM, *Platform Independent Model*) el modelo conceptual de datos y como modelo específico de plataforma (PSM – *Platform Specific Model*) el modelo de esquemas XML, ambos representados en UML. En este trabajo se modifican dichos modelos para poder añadir aspectos de seguridad para el caso de que la información almacenada se considere crítica. Se propone, por un lado, el uso de la extensión de UML para incorporar los aspectos de seguridad a nivel de PIM y, por otro lado, se modificará el perfil para esquemas XML previamente definido, con el fin de incorporar los aspectos de la seguridad en todas las etapas del desarrollo de una BD XML. También se definirán los *mappings* para pasar del PIM Seguro al PSM para BD XML Seguras. El proceso de desarrollo de la BD XML Segura se mostrará mediante un caso de estudio: un SIW para la gestión de información hospitalaria en una BD XML.

Palabras clave: Desarrollo de bases de datos XML Seguras, Seguridad en bases de datos, XML, MDA, UML, OCL.

1 Introducción

XML es el estándar actual para el intercambio de información y el transporte de datos entre aplicaciones heterogéneas. Tradicionalmente la información de los documentos XML se almacenaba en ficheros o en Sistemas de Gestión de Bases de Datos (SGBD) convencionales, o bien haciendo la correspondiente transformación a una base de datos (BD), generalmente relacional, o utilizando campos de tipo fichero, por ejemplo, de tipo CLOB (*Character Large OBject*). Sin embargo, la aparición de las BD XML proporciona una alternativa mejor y más directa para almacenar y gestionar los documentos XML. En la actualidad, existen diferentes soluciones para el almacenamiento de documentos XML que según [22], se pueden clasificar en dos

grupos: las BD XML nativas como Tamino [20], eXcelon XIS [6], eXist [3] o ToX [1]; y las extensiones de BD XML, que permiten el almacenamiento de documentos XML en SGBD convencionales, habitualmente relacionales u objeto-relaciones (OR). Por ejemplo, Oracle [19] incluye desde la versión 9*i* release 2 nuevos aspectos para el almacenamiento de XML (XML DB de Oracle). También otros productos como IBM DB2 XML Extender [13] o Microsoft SQLXML [17] incluyen extensiones para el almacenamiento de XML. En [22] se realiza un estudio de las principales soluciones de BD XML.

Para la mayoría de las organizaciones la gestión, seguridad y confidencialidad de la información resulta ser un tema crítico [5]. Además, algunos autores remarcan que la seguridad en BD debe ser tenida muy en cuenta, no como un aspecto relegado al final de su desarrollo, sino incluido en todas las etapas de su ciclo de vida [4,10,12]. Incluso la Fundación para el Control y Auditoría de Sistemas de Información insiste en que los gestores han de asegurar que la seguridad se considera de manera integral y explícita en todas las etapas del desarrollo de los sistemas de información [14]. También en el caso de las BD XML es la seguridad un punto clave que debe ser considerado explícitamente y que tiene que tenerse en cuenta de forma ortogonal a todo el proceso de desarrollo de este tipo de BD [11].

Aunque existen diferentes ideas para integrar la seguridad en el proceso de desarrollo de sistemas de información, en el ámbito de las BD la seguridad se suele considerar sólo a nivel de criptografía. Recientemente, hemos propuesto una metodología para diseñar BD relacionales teniendo en cuenta los aspectos de seguridad de la información en todas las etapas del proceso de desarrollo [8]. Pero, sin embargo, no existen trabajos que traten el tema de la seguridad a la hora de desarrollar una BD XML.

En este artículo vamos a integrar el aspecto de seguridad en la aproximación metodológica para el desarrollo de BD XML [21] que se enmarca en MIDAS [15], una metodología dirigida por modelos para el desarrollo de Sistemas de Información Web (SIW). MIDAS propone el uso de estándares a lo largo de todo el proceso de desarrollo, así como el uso de UML para el modelado del SIW independientemente del nivel de abstracción o del aspecto del sistema a modelar. Dado que UML no permite representar directamente todos los modelos necesarios, MIDAS incorpora algunas extensiones de UML existentes y define, o adapta, otras nuevas, siempre que es necesario [7,16].

MIDAS propone una arquitectura dirigida por modelos, basada en MDA (*Model Driven Architecture*) propuesta por el *Object Management Group* (OMG) [18] y considera a la hora de modelar el sistema los aspectos de contenido, hipertexto y comportamiento. Todos estos aspectos se consideran a nivel de Modelos Independientes de Computación (CIM - *Computation Independent Model*), Modelos Independientes de Plataforma (PIM - *Platform Independent Model*) y Modelos Específicos de Plataforma (PSM - *Platform Specific Model*) [2]. La Fig. 1 muestra de forma resumida la arquitectura dirigida por modelos de MIDAS simplificada, donde se proponen los CIM, comunes a todo el sistema, así como los diferentes PIM y PSM para representar los aspectos de contenido, hipertexto y comportamiento.

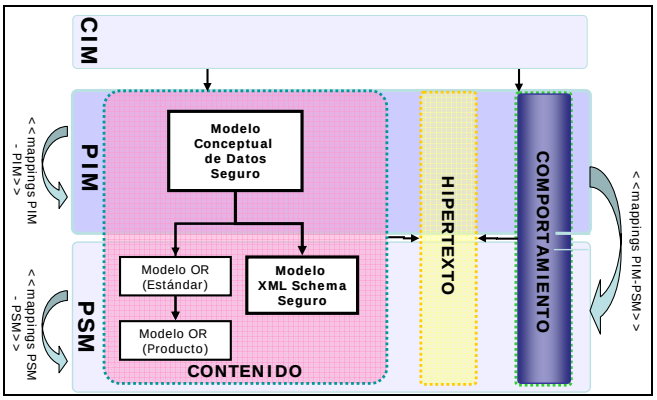


Fig. 1. Arquitectura de MIDAS simplificada

Se contempla una tercera dimensión que incluye otros aspectos a tener en cuenta en el desarrollo de un SIW, como son la arquitectura del sistema o la seguridad, que son ortogonales a los presentados en la Fig. 1. La Fig. 2 muestra la arquitectura de MIDAS con las tres dimensiones mencionadas.

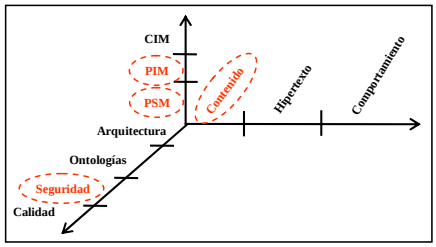


Fig. 2. Dimensiones a considerar en un desarrollo de un SIW

Este trabajo se centra en el aspecto del ***contenido,*** que se corresponde con el concepto tradicional de una BD, y el aspecto ortogonal de la ***seguridad*** para los niveles de ***PIM*** y ***PSM*** (ver Fig. 2). En la siguiente sección 2 veremos el desarrollo de BD XML Seguras en el marco de MIDAS. Como PIM de datos se usa el modelo conceptual de datos que se representará mediante un diagrama de clases extendido, usando la extensión a UML propuesta en [7,9] para incorporar los aspectos de seguridad a este nivel. El perfil para PIM de datos seguros se resumirá en la sección 2.1. Como PSM en MIDAS se propone usar el modelo OR o el modelo de esquemas XML, dependiendo de la tecnología que se necesite utilizar. En este artículo se muestra la parte relativa al diseño de BD XML Seguras y por lo tanto, como PSM se usará el modelo de esquemas XML, usando el perfil para BD XML previamente definido en [21]. En el apartado 2.2 de este artículo presentaremos una adaptación del mismo para incorporar los aspectos de la seguridad específicos para este tipo de BD XML. También se mostrarán en el apartado 2.3 las reglas de transformación para pasar del PIM de datos seguro al PSM de datos, que será el esquema de la BD XML Segura. Estas reglas están basadas en las definidas en [21], donde se describen unos *mappings* para obtener el PSM de datos, pero sin considerar los aspectos de seguridad. En este artículo adaptaremos dichas reglas para obtener el esquema de la BD XML Segura incluyendo las restricciones necesarias para la seguridad.

En la sección 3 de este trabajo se presenta también un caso de estudio de un SIW para la gestión y el análisis de información de hospitales, mediante el que se ha definido y validado la propuesta para el desarrollo de una BD XML Segura.

Finalmente, en la sección 4 se muestran las principales conclusiones obtenidas y se plantean futuros trabajos.

2 Desarrollo de Bases de Datos XML Seguras en MIDAS

Como ya se ha dicho en el apartado anterior, en este trabajo nos centramos en el aspecto de **contenido** de MIDAS, que se corresponde con el concepto tradicional de una BD. El desarrollo de una BD depende de varios aspectos: en primer lugar, de si ya existe una BD en la organización y, por otro lado, de la tecnología que se desea utilizar, es decir, si se desea utilizar una BD OR [16] o una BD XML [21]. Además es necesario tener en cuenta si la BD que se desea desarrollar incluye información que sea necesario proteger y que, por lo tanto, sea necesario recoger aspectos de seguridad desde las etapas más tempranas del desarrollo de la BD.

Vamos a pasar a describir en detalle el proceso de desarrollo de una BD XML Segura partiendo de cero, así como las técnicas y los modelos necesarios:

- A nivel de **PIM** se realiza el diseño conceptual de datos. Para ello se utiliza el modelo conceptual de datos, sin tener en cuenta la tecnología seleccionada, ya que se trata de un modelo independiente de la plataforma. Este PIM de datos se representa mediante un diagrama de clases UML. En el caso de nuestra propuesta se usará, como ya se ha dicho anteriormente, el diagrama de clases UML extendido, para poder representar los aspectos de seguridad junto con un conjunto de restricciones de seguridad que han sido expresadas a través del lenguaje OSCL [9], como veremos en el subapartado siguiente (2.1).
- A nivel de **PSM** se lleva a cabo el diseño lógico de datos. En este punto sí que es necesario tener en cuenta la tecnología seleccionada, que en nuestro caso será una BD XML. Se parte del PIM de datos seguro obtenido en el nivel anterior y se aplican las reglas de transformación que se resumen en el subapartado 2.3. El PSM de datos seguro se representará mediante un esquema XML en UML extendido (ver subapartado 2.2). En este caso, el esquema de la BD XML será el esquema XML obtenido, que considerará los aspectos de seguridad necesarios.

La tabla 1 resume las tareas, técnicas y notaciones que se deben llevar a cabo a la hora de desarrollar una BD XML Segura

Tabla 1. Proceso de desarrollo de una BD XML Segura

	MIDAS: Desarrollo de BD XML Seguras		
Nivel	Tareas	Modelos	Notación
PIM	Diseño Conceptual de Datos con Seguridad	Modelo Conceptual de Datos Seguro	Diagrama de Clases (UML Extendido)
PSM	Diseño Lógico de Datos con Seguridad	Modelo Lógico de Datos Seguro	Esquemas XML (MIDAS-UML)

Ahora vamos a pasar a ver en primer lugar el PIM de datos seguro, resumiendo el perfil de UML que se ha desarrollado para modelar BD seguras a nivel de PIM (subapartado 2.1), después se verá el PSM de datos seguro, resumiendo el perfil de UML para esquemas XML adaptado para BD XML Seguras (subapartado 2.2), y

finalmente, se resumen las reglas para transformar el PIM de datos seguro en el PSM de datos para incluir en el esquema XML los aspectos de seguridad que se han definido en el PIM de datos (subapartado 2.3).

2.1 PIM de Datos Seguro

Para desarrollar un PIM de datos seguro se ha desarrollado un perfil de UML (más detalles pueden ser encontrados en [9]). El perfil de UML definido permite clasificar tanto a datos como a usuarios bajo varios criterios de clasificación, con el objeto de poder llevar a cabo un control de acceso obligatorio. Los criterios son los siguientes:

- **Niveles de seguridad:** Permiten definir una jerarquía de niveles, como los tradicionales en entornos militares: sin clasificar, confidencial, secreto y alto secreto.
- **Roles de usuarios:** Permiten definir un conjunto jerárquico de roles de usuarios que representan las funciones jerárquicas dentro de una empresa.
- **Categorías de usuarios:** Permiten definir una organización o clasificación horizontal (no jerárquica) de grupos de usuarios.

Además de esta información de clasificación, este perfil permite definir tres tipos de restricciones:

- **Reglas de clasificación dinámica de datos:** Permiten definir los datos de clasificación de distintas instancias dependiendo del valor de uno o varios atributos de las instancias.
- **Reglas de autorización:** Permiten definir qué usuarios podrán acceder a qué datos y realizar qué acciones, dependiendo de una condición expresada en OCL.
- **Reglas de auditoría:** Permiten especificar situaciones en las que nos interesa registrar un rastro de auditoría para analizar qué usuarios acceden a la información. Para ello se definen condiciones expresadas con OCL.

Para la definición de todos estos elementos, consideramos el perfil UML, llamado *Conceptual Secure DB* (extensión de UML y OCL para diseñar BD seguras), que está compuesto de un conjunto de tipos de datos, valores etiquetados y estereotipos, además de la definición de un conjunto de reglas bien formadas.

El paquete que contiene todos los estereotipos que han sido definidos en este perfil de UML puede ser analizado en la Fig. 3. Estos estereotipos se pueden clasificar en las siguientes tres categorías:

- Los necesarios para representar la información de seguridad en los *elementos del modelo* (el propio modelo, las clases, atributos, asociaciones e instancias).
- Los necesarios para modelar las *restricciones de seguridad* para a) definir la clasificación dinámica de cualquier elemento del modelo, b) definir reglas de autorización, y c) definir reglas de auditoría dependiendo de tipos de acceso quizás de alguna condición expresada en OCL.
- El estereotipo *UserProfile*, que es necesario para especificar restricciones dependiendo de alguna propiedad de un usuario o de un grupo de usuarios, por ejemplo, dependiendo de la ciudadanía, edad, etc.

Una descripción en detalle de todos estos estereotipos, así como de todos los valores etiquetados que han sido definidos para estos estereotipos puede ser analizada en [9].

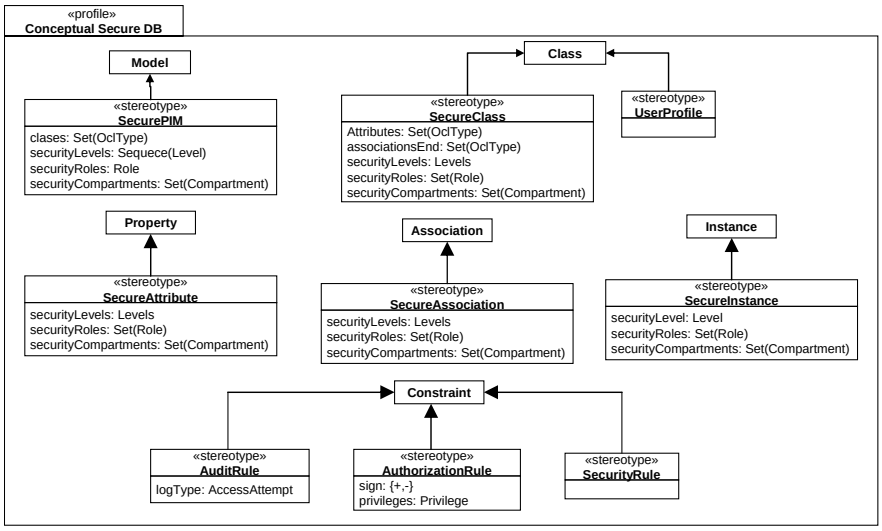


Fig. 3. Perfil para BD Seguras: *Conceptual Secure DB*

2.2 PSM de Datos Seguro

En MIDAS se propone usar como PSM de datos el modelo de esquemas XML, representado en UML extendido, utilizando el perfil definido en [21]. Para incluir los aspectos de seguridad en este modelo, en este artículo, hemos adaptado dicho perfil añadiendo los elementos necesarios para poder recoger la seguridad.

En la Fig. 4 se muestra los elementos que se han añadido con el fin de adaptar el perfil para poder representar esquemas XML seguros mediante un diagrama de clases UML. La extensión define un conjunto de estereotipos nuevos para permitir recoger en notación gráfica en UML todos los componentes de un esquema XML seguro, manteniendo las asociaciones, el orden y el anidamiento entre los distintos elementos.

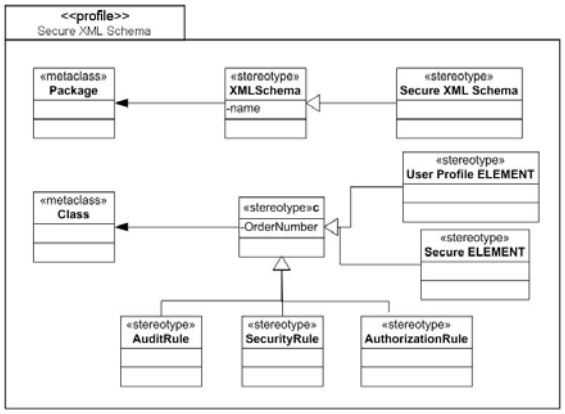


Fig. 4. Perfil para esquemas XML seguros: *Secure XML Schema*

2.3 Reglas de Transformación para pasar del PIM al PSM de Datos Seguro

De forma similar a como las metodologías para BD relacionales u objeto-relacionales proponen algunas reglas para la transformación de un esquema conceptual en un esquema lógico estándar, en MIDAS se proponen reglas de transformación para pasar del nivel de PIM al de PSM de datos. En este trabajo, se han definidos las reglas necesarias para obtener un PSM de datos seguro partiendo del PIM de datos seguro. A continuación, mostraremos estas reglas para recoger las características propias de la seguridad, basándonos en el trabajo de [21], donde se definieron los *mappings* para obtener un esquema de una BD XML.

- **Transformación del PIM de datos seguro**
  El modelo conceptual de datos, es decir, el *PIM de datos seguro,* se transforma, a nivel de PSM, en un esquema XML denominado '*Secure Data PSM*'. Este se representará con un paquete UML estereotipado con <<Secure XML SCHEMA>> que incluye todos los componentes del PSM de datos seguro. Además también contendrá los atributos de seguridad (*securityLevel*, *securityRoles* y *securityCompartments)* del PIM de datos seguro. Estos se definirán dentro del esquema XML como elementos globales. Estos atributos de seguridad se podrían haber incluido como atributos del esquema, sin embargo, si se representasen de este modo, no se considerarían elementos de primer orden, ni se podría recoger que pueden tener una cardinalidad máxima múltiple.
- **Transformación de la clase *User Profile***
  Esta clase incluye la información que se desee restringir sobre uno o varios usuarios. Se transformará incluyendo un elemento global estereotipado con <<User Profile ELEMENT>>, que contendrá un *complexType* de tipo secuencia con todos los atributos de la clase representados como subelementos.
- **Transformación de clases seguras**
  De forma genérica una clase UML se transforma en un elemento del esquema XML con el mismo nombre que la clase de la que procede [21]. Para transformar las clases UML *seguras*, estereotipadas con <<SecureClass>> hay que incluir además las características seguras que tengan las mismas. Las clases seguras pueden tener tres atributos específicos: *securityLevel*, *securityRoles* y *securityCompartments*. Las clases seguras se transformarán en elementos seguros estereotipados con <<Secure ELEMENT>>. Cada elemento seguro contendrá un tipo complejo de tipo secuencia, que contendrá como subelementos, entre otros, los atributos seguros, indicando con el atributo de los subelementos *maxOccurs* el número de posibles instancias de los atributos de seguridad.
- **Transformación de atributos seguros**
  Dado que los atributos de una clase, según la propuesta de [21], se transforman como subelementos del elemento que representa la clase UML a la que pertenecen los atributos, si un atributo tiene sus propios atributos de seguridad asociados, estos se representarán como subelementos del elemento que representa el atributo correspondiente. Así, los atributos de seguridad definidos sobre un atributo se transformarán en subelementos <<Secure ELEMENT>>.
- **Transformación de asociaciones seguras**
  En el caso de la transformación de asociaciones, en [21] se hizo un estudio detallado de la forma más apropiada para mapear las mismas al nivel de PSM. Las asociaciones entre dos clases se transforman, de forma genérica, incluyendo

un subelemento en uno de los elementos, correspondiente a una de las clases implicada en la relación, con una o varias referencias al otro elemento implicado en la asociación. En el caso de ser una asociación con seguridad, este subelemento tendría a su vez subelementos para representar a los atributos de seguridad correspondientes (*securityLevel*, *securityRoles*, *securityCompartment*) estereotipados como <<Secure ELEMENT>>.

- **Transformación de restricciones de seguridad**
  A la hora de realizar la transformación de las restricciones de seguridad que se hayan definido a nivel de PIM de datos seguro, estas se pueden definir para cualquier elemento (modelo o clase), aunque lo normal suele ser definirlo a nivel de clase. Si se definen a nivel de modelo, se crearán elementos globales dentro del esquema para recogerlo, en los demás casos se crearán subelementos del elemento del que dependan. Existen diferentes restricciones de seguridad:
  a) las reglas de auditoría (***AuditRule***): Se transformarán creando un subelemento estereotipado con <<AuditRule>> con nombre "AuditRule_" más el número de la regla. Este elemento será de tipo *complexType* y contendrá una secuencia de dos elementos: un elemento *AuditRuleType* de tipo simpleType de tipo base *string* con una restricción de tipo *enumeration* con los valores: *all*, *frustatedAttempt*, *successfullAccess*; y otro elemento *AuditRuleCondition* que será un elemento de tipo *string*, que contendrá la expresión XPATH asociada a la expresión en OCL.

```
<complexType>
     <sequence>
        <element name= "AuditRuleType">
            <simpleType>
                   <restriction base= "string">
                      <enumeration value= "all"/>
                      <enumeration value= "frustatedAttempt"/>
                      <enumeration value= "successfullAccess"/>
                   </restriction>
             </simpleType>
        </element>
        <element name= "AuditRuleCondition" type="string"/>
     </sequence>
</complexType>
```

  b) las reglas de autorización (***AuthorizationRule***): Se transformarán creando un subelemento estereotipado con <<AuthorizationRule>> con nombre "AuthorizationRule_" más el número de la regla. Este elemento será de tipo complexType y contendrá una secuencia con tres elementos: un elemento *AuthorizationRuleSign* de tipo simpleType de tipo base *string* con una restricción de tipo *enumeration* con los valores: + o - ; otro elemento *AuthorizationRulePrivileges* de tipo simpleType de tipo base *string* con una restricción de tipo *enumeration* con los valores: *read*, *insert*, *delete*, *update* y *all*; y otro elemento *AuthorizationRuleCondition* que será de tipo *string* y contendrá la expresión XPATH asociada a la expresión en OCL.

```
<complexType>
     <sequence>
        <element name= "AuthorizationRuleSign">
               <simpleType>
                      <restriction base="string">
                         <enumeration value="+/> <enumeration value="-"/>
                      </restriction>
               </simpleType>
        </element>
```

```
        <element name= “AuthorizationRulePrivileges”>
               <simpleType>
                      <restriction base=”string”>
                         <enumeration value=”read”/>
                         <enumeration value=”insert”/>
                         <enumeration value=”delete”/>
                         <enumeration value=”update”/>
                         <enumeration value=”all”/>
                      </restriction>
               </simpleType>
    </element>
    <element name= “AuthorizationRuleCondition” type=”string”/>
  </sequence>
</complexType>
```

c) la clasificación dinámica de cualquier elemento del PIM (***SecurityRule***) se transforma creando un subelemento estereotipado con <<SecurityRule>>, con nombre “SecurityRule_” más el número de la regla. Este elemento será de tipo complexType y contendrá una secuencia con un elemento de tipo *string* con la expresión XPATH asociada a la expresión en OCL.

```
<complexType>
     <sequence>
        <element name= “SecurityRuleCondition” type=”string”/>
     </sequence>
</complexType>
```

En la Tabla 2 se resumen las reglas de transformación para pasar del PIM de datos al correspondiente PSM utilizando tecnología de BD XML. En [21] aparecen las reglas de forma detallada, sin incluir el aspecto de la seguridad.

Tabla 2. Reglas de transformación para pasar del PIM al PSM de datos seguro

PIM de datos	**PSM de datos**
PIM de datos seguro	**Esquema XML con seguridad <<Secure Data PSM>>**
Atrib. securityLevels	Elemento global del esquema (maxOccurs=unbounded)
Atrib. securityRoles	Elemento global del esquema
Atrib. securityCompartments	Elemento global del esquema (maxOccurs=unbounded)
Clase Segura	**Elemento XML con seguridad <<Secure Element>>**
Atrib. securityLevels	Subelemento (maxOccurs=unbounded)
Atrib. securityRoles	Sublemento
Atrib. securityCompartments	Sublemento (maxOccurs=unbounded)
Clase Segura User Profile	**Elemento XML global <<User Profile Element>>**
Atributo	**Subelemento**
Atrib.securityLevels	Subelemento (maxOccurs=unbounded)
Atrib. securityRoles	Sublemento
Atrib. securityCompartments	Sublemento (maxOccurs=unbounded)
Asociación	
Atrib. securityLevels	Subelemento del elemento de la asociación (maxOccurs=unbounded)
Atrib. securityRoles	Subelemento del elemento de la asociación
Atrib. securityCompartments	Sublemento de elemento de la asociación (maxOccurs=unbounded)
Restricción	Subelemento con complexType de tipo secuencia con subelementos
AuditRule	- AuditRuleType de simpleType con restricción enumeration - AuditRuleCondition de tipo string, que contendrá la expresión XPATH asociada a la expresión en OCL
AuthorizationRule	- AuthorizationRuleSign de simpleType con restricción enumeration - Privileges de simpleType con restricción enumeration - AuthorizationRuleCondition de tipo string, que contendrá la expresión XPATH asociada a la expresión en OCL
SecurityRule	- Subelemento de tipo string con expresión XPATH

3 Caso de Estudio

En esta sección aplicamos nuestra extensión de UML para desarrollar el PIM de datos seguro de un caso de estudio en el contexto de sistemas de información hospitalarios. A continuación, a través de las reglas de transformación que hemos definido en la sección 2.3, construimos una BD XML Segura utilizando la otra extensión de UML para PSM de datos seguros.

Para este ejemplo se ha definido una jerarquía de usuarios simplificada (ver Fig. 5 izquierda), compuesta de un rol genérico "Empleado de Hospital", que se separa en los roles "Personal Sanitario" y "Personal no Sanitario". El primero de estos roles se divide en "Doctor" y en "Enfermero", mientras que el segundo se especializa en "Mantenimiento" y en "Administrativo". Adicionalmente, para este caso de estudio se ha considerado tres niveles de seguridad, "Sin Clasificar", "Secreto" y "Alto Secreto" (ver Fig. 5 derecha). Para este ejemplo, por simplicidad no hemos definido categorías de usuarios, pero podrían haberse considerado categorías regionales (Región España, Región Argentina, etc.), criterios profesionales (Pediatría, Cirugía, etc.).

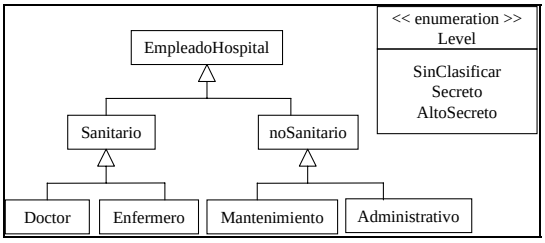


Fig. 5. Jerarquía de Roles de Usuarios y Niveles de Seguridad

La Fig. 6 muestra el PIM de datos seguro, representado con un diagrama de clases que contiene muchos detalles, pero nosotros explicaremos tan sólo algunas clases para poder enfocar nuestra atención en los aspectos de seguridad. Las clases del diagrama que mencionaremos son (el resto son obvias y no profundizaremos en ellas) *UserProfile, Paciente* y *Admisión*. La clase *UserProfile* contiene los distintos campos de información que son registrados para todos los usuarios que tendrán acceso a la BD. La clase *Paciente* contiene información de todos los pacientes del hospital, y puede ser accedida por los usuarios que tengan al menos el nivel de seguridad secreto y realicen roles administrativos o sanitarios. La clase *Admisión* contiene la información de todas las admisiones del hospital, y puede ser accedido por usuarios que realizan tareas sanitarias o de administración, y que además tengan un nivel de seguridad secreto. Ponemos en esta clase, que los atributos *diagnóstico*, *resultado* y *medicinas* sólo podrán ser accedidos por personal sanitario (y no administrativo), y que el atributo *coste* puede ser visto por administrativos, pero no por personal sanitario. Existe una asociación entre las clases *Paciente* y *Admisión*, y además, podemos apreciar que existen numerosas restricciones de seguridad asociadas a estas clases.

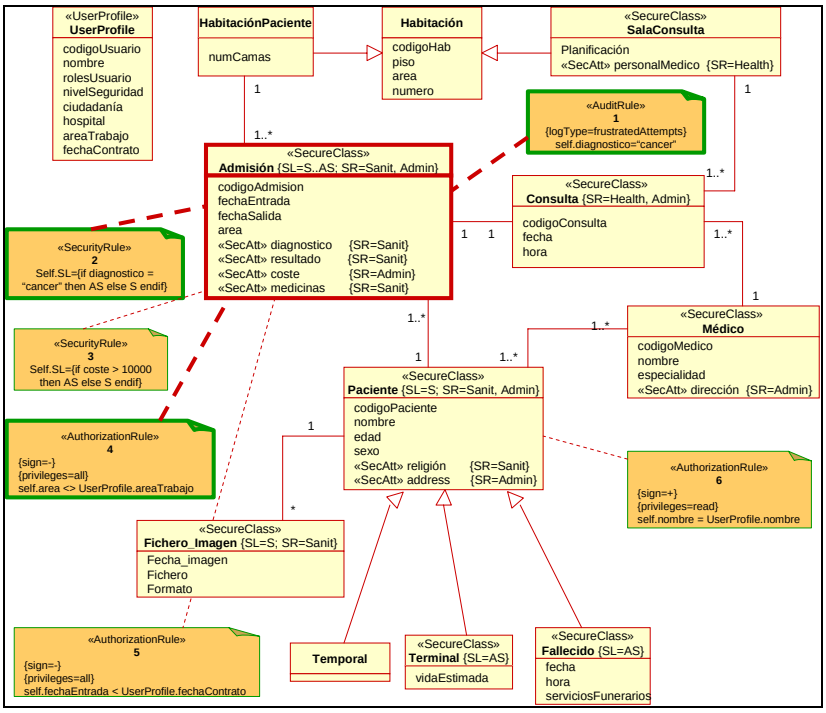


Fig. 6. PIM de datos con Restricciones de Seguridad

Hemos identificado cada una de las restricciones de seguridad con un número. El detalle de cada una es como sigue:

1. Se trata de una restricción estereotipada que representa una regla de auditoría. Esta regla especifica que todos los accesos que sean denegados por el mecanismo de control de acceso (valor etiquetado logType igual a frustratedAttempts), correspondientes a instancias de la clase *Admisión* cuyos diagnóstico sea cáncer (expresión OCL self.diagnostico='cancer'), deberían ser registrados para futura auditoría. Esta regla de auditoría nos ayudará a identificar posibles atacantes que traten de acceder a información confidencial sin tener los permisos necesarios.
2. Esta restricción define una regla de seguridad dinámica que especifica el nivel de seguridad de cada instancia de la clase *Admisión*. Si el valor del atributo diagnóstico es 'cancer', entonces el nivel de seguridad de la admisión será alto secreto, y de lo contrario sólo será secreto.
3. Esta es otra regla de seguridad dinámica para la clase *Admisión*. En este caso, el nivel de seguridad dependerá del valor del atributo *coste*, que indica el precio del servicio hospitalario.
4. El concepto que modela la cuarta restricción es una regla de autorización. Podríamos denegar el acceso (signo = -) a la información de admisiones a usuarios cuya área de trabajo sea diferente que el área sanitaria de una admisión particular (self.area <> UserProfile.areaTrabajo).
5. Por motivos de confidencialidad, podríamos denegar el acceso (signo = -) a la información de admisiones a todos los doctores cuya fecha de contrato en el hospital sea posterior a la fecha de entrada en el hospital de los pacientes

(self.fechaEntrada < UserProfile.fechaContrato). Esta restricción especifica esa regla de autorización.

6. Finalmente, podemos considerar a los pacientes como usuarios especiales del sistema en el sentido de que podrían tener acceso solamente a sus propios datos personales. En ese caso, es necesario especificar una regla de autorización positiva (signo = +) indicando como condición que el nombre del usuario sea igual al nombre del paciente (self.name = UserProfile.nombre).

Podemos observar que utilizando la extensión de UML, es posible especificar un amplio abanico de restricciones de confidencialidad en el PIM de datos seguro.

A la hora de transformar el PIM de datos seguro de la Fig. 6 aplicaremos las reglas definidas en el apartado 2.3 y obtendremos el PSM de datos. En la Fig. 7 podemos ver parte del PSM de datos seguro que hemos obtenido.

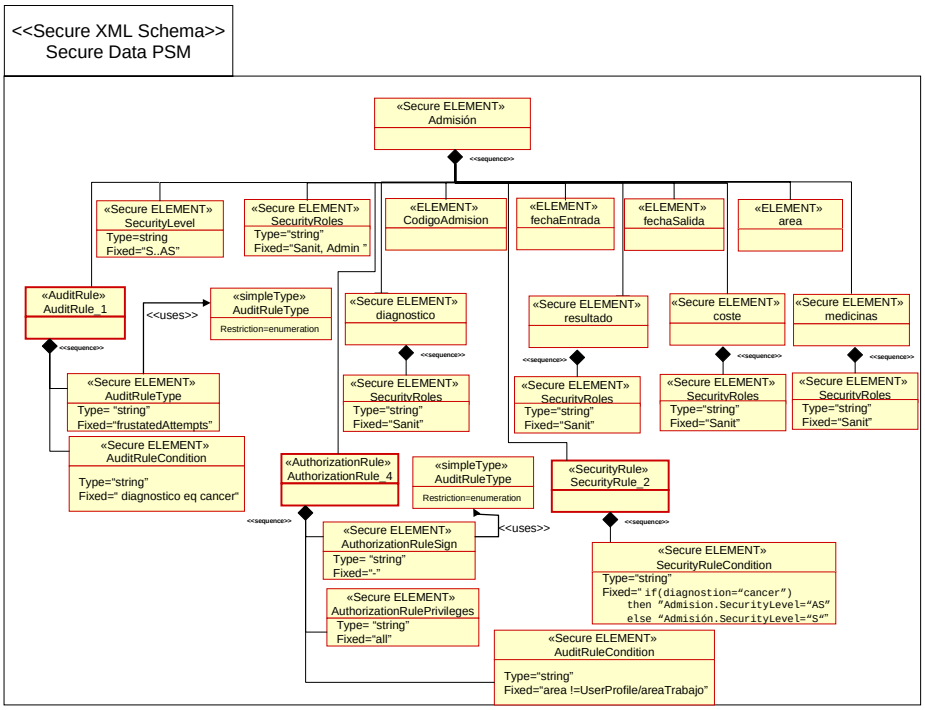


Fig. 7. Parte de PSM de datos con Seguridad

En primer lugar, hemos creado un paquete UML estereotipado con <<Secure XML Schema>> denominado *Secure Data PSM*, que incluirá todos los elementos obtenidos al transformar el PIM de datos seguro. En nuestro caso, por motivos de espacio, nos hemos centrado en la transformación de la clase ***Admisión*** y en la transformación de algunas de las restricciones, concretamente, las reglas ***AuditRule1***, ***AuthorizationRule4*** y ***SecurityRule2***, siguiendo los pasos indicados en la sección 2.3.

A la hora de transformar la clase segura *Admisión* se creará en primer lugar un elemento estereotipado con <<Secure ELEMENT>> que contendrá un complexType con los subelementos que representan a los atributos propios de la clase (codigoAdmisión, fechaEntrada, fechaSalida, area). Además contendrá los

subelementos correspondientes a los atributos de seguridad (*Security Level* y *Security Roles*) y estereotipados con <<Secure ELEMENT>>. El valor de estos atributos se recogerá con el atributo *fixed* de estos elementos.

Los atributos *diagnostico*, *resultado*, *coste* y *medicinas* de dicha clase son atributos seguros y por lo tanto, se representarán como elementos seguros <<Secure ELEMENT>> y por lo tanto, tienen sus propios subelementos que representan los atributos seguros.

4 Conclusiones y Trabajos Futuros

Hoy en día existen diversas soluciones para el almacenamiento de datos XML, pero aún no existe una metodología para el diseño sistemático de BD XML que incorpore la seguridad en todo el proceso de desarrollo, si la información a almacenar se considera crítica.

En este trabajo se ha integrado el aspecto de seguridad en la aproximación metodológica para el desarrollo de BD XML en el marco de MIDAS que es una metodología dirigida por modelos para el desarrollo de SIW basada en MDA. Según el proceso especificado para el desarrollo de BD XML Seguras, para el PIM de datos seguro se usa la extensión a UML para incorporar los aspectos de seguridad a nivel conceptual y para el PSM de datos seguro se ha modificado el perfil para BD XML previamente definido, con el fin de incorporar los aspectos de la seguridad. Además, se han definido las reglas de transformación para pasar del PIM de datos seguro al PSM de datos seguro, que será el esquema de la BD XML Segura. Se ha desarrollado el caso de estudio para la gestión de información hospitalaria para validar nuestra propuesta; en este artículo se muestra una parte del mismo, en el que se define el esquema de la BD XML Segura para una parte del PIM de datos seguro desarrollado.

Existen diferentes direcciones en las que vamos a seguir trabajando para extender la propuesta de este artículo. Una de las líneas, en la que hemos empezado a trabajar es en la automatización de las transformaciones de las restricciones expresadas en OCL a nivel de PIM para convertirlas al lenguaje XPATH. Además se pretende realizar la implementación de diferentes casos de estudios, para detectar nuevas necesidades, así como analizar las facilidades para incorporar los aspectos de seguridad que proporcionan los diferentes gestores de BD XML, tanto los nativos como las extensiones para XML que tienen los SGBD. Por otro lado, también vamos a incluir el aspecto de la seguridad en el módulo para el desarrollo semi-automático de BD XML de la herramienta CASE (MIDAS-CASE) que estamos desarrollando.

Agradecimientos

Esta investigación se ha llevado a cabo en el marco de los siguientes proyectos: GOLD financiado por el Ministerio de Educación y Ciencia (TIN2005-00010/), DIMENSIONS (PBC-05-012-2) financiado por FEDER y por la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha, RETISTIC (TIC2002-12487-E) y CALIPO (TIC2003-07804-CO5-03), de la Dirección General de Investigación del Ministerio de Ciencia y Tecnología.

Referencias

1. Barbosa, D., Barta, A., Mendelzon, A., Mihaila, G., Rizzolo, F. y Rodriguez-Gianolli, P. *ToX - The Toronto XML Engine*, International Workshop on Information Integration on the Web, Rio de Janeiro, 2001.

2. Cáceres, P., Marcos, E. y Vela, B. *A MDA-Based Approach for Web Information System Development.* Workshop in Software Model Engineering in UML Conference. San Francisco, USA, octubre, 2003.
3. Chaudhri, A.B., Rashid, A. y Zicari, R. (Eds.). *XML Data Management. Native XML and XML-Enabled Database Systems.* Addison Wesley, 2003.
4. Devanbu, P. y Stubblebine, S. *Software engineering for security: a roadmap*, in: A. Finkelstein (Ed.), The Future of Software Engineering, ACM Press, 2000, pp. 227-239.
5. Dhillon, G. y Backhouse, J. *Information System Security Management in the new Millennium.* Communications of the ACM. 43, 7. pp.: 125-128, 2000.
6. eXcelon Corporation. *Managing DXE. System Documentation Release 3.5.* eXcelon Corporation. Burlington. Recuperado de: www.excelon.corp.com, 2003.
7. Fernández-Medina, E. y Piattini, M. *UML for the Design of Secure Databases.* Proceedings to the The 1st International Workshop on Security in Information Systems (SIS 2002) (Into the 4th International Conference on Enterprise Information Systems), pp. 25-38. Abril, 2002. Ciudad Real (España), 2002.
8. Fernández-Medina, E. y Piattini M. *Designing secure databases.* Information & Software Technology 47(7): 463-477. 2005
9. Fernández-Medina, E. y Piattini, M. Extending OCL for Secure Database Design. in International Conference on the Unified Modeling Language (UML 2004). 2004. Lisboa, Portugal. Springer-Verlag, LNCS 3273. pp. 380-394. 2004.
10. Ferrari E. y Thuraisingham B., *Secure Database Systems*, in: M. Piattini, O. Díaz (Ed.), Advanced Databases: Technology Design. Artech House, 2000.
11. Ferrari, E. *Secure DataBase Systems.* Segunda Reunión RETISBD. Murcia (España), junio 2001.
12. Ghosh, A., Howell C., Whittaker J., *Building software securely from the ground up*, IEEE Software 19 (1) (2002) 14-17.
13. IBM Corportation. *IBM DB2 Universal Database -XML Extender Administration and Programming, Product Documentation Version 7.* IBM Corporation, 2000.
14. ISACF, Information Security Governance. Guidance for Boards of Directors and Executive Management, Information Systems Audit and Control Foundation, USA, 2001.
15. Marcos, E., Vela, B., Cáceres, P. y Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design.* DASWIS 2001. Yokohama (Japan), noviembre, 2001. Springer-Verlag, LNCS 2465, pp. 227-238, septiembre, 2002.
16. Marcos, E., Vela, B. y Cavero J.M. *Methodological Approach for Object-Relational Database Design using UML.* Journal on Software and Systems Modeling (SoSyM). Springer-Verlag. Ed.: R. France y B. Rumpe. Vol. SoSyM 2, pp.59-72, 2003.
17. Microsoft Corporation. *Microsoft SQL Server - SQLXML 2.0*, System Documentation. Microsoft Corporation, 2000.
18. OMG. *MDA Guide Version 1.0.* Document number omg/2003-05-01. Ed.: Miller, J. y Mukerji, J. Recuperado de: http://www.omg.com/mda, 2003.
19. Oracle Corporation. *Oracle XML DB. Technical White Paper.* Recuperado de: www.otn.com, enero, 2003.
20. Software AG. *Tamino X-Query. System Documentation Version 3.1.1.* Software AG, Darmstadt, Alemania. Recuperado de: www.softwareag.com, 2001.
21. Vela, B., Acuña, C. y Marcos, E. *A Model Driven Approach for XML Database Development*, 23rd. International Conference on Conceptual Modelling (ER2004). Springer Verlag, LNCS 3288, pp. 780-794. 2004.
22. Westermann, U. y Klas W. *An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions.* ACM Computing Surveys, Vol. 35 (4), pp. 331-373, diciembre, 2003.