



IV Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM'07)



DSDM'07

: Evento a celebrar en el marco de las XII Jornadas de Ingeniería del Software y Bases de Datos 2007 :

11 de Septiembre, 2007
<http://www.taro.ull.es/dsdm07>

DSDM'07

- | | | | | | | |
|--------------|----------------|----------|-----------------|--------------------|---------|-----------------|
| Presentación | Contribuciones | Programa | Call For Papers | Fechas Importantes | Comités | Otras Ediciones |
|--------------|----------------|----------|-----------------|--------------------|---------|-----------------|

PROGRAMA DEL TALLER DSDM 2007

- 08:45 - 09:00 REGISTRO Y RECEPCIÓN
- 09:00 - 10:30 SESIÓN: GESTIÓN DE MODELOS
 Un acercamiento a la representación e implementación de la diferencia entre modelos. Jose E. Rivera, Antonio Vallecillo
- Representación de las Relaciones en los Metamodelos con el lenguaje Ecore. Iván García-Magariño, Jorge Jesús Gomez-Sanz, Juan Pavón
- El metamodelado de un framework: Spring Web Flow. Antonia M. Reina, Jesús Torres, Miguel Toro
- 10:30 - 11:00 PAUSA CAFÉ
- 11:00 - 12:30 SESIÓN: TRANSFORMACIÓN DE MODELOS
 Generating Operation Contracts from UML Class Diagrams: A Template-based Approach. Jordi Cabot, Cristina Gómez
- Desarrollo de modelos multidimensionales de almacenes de datos basado en MDA: del análisis de requisitos al modelo lógico. Jose-Norberto Mazón, Juan Trujillo
- Implementación de Heurísticas en QVT para la obtención de Clases de Análisis a partir de Modelos de Proceso de Negocio Seguros. Ignacio García-Rodríguez de Guzmán, Alfonso Rodríguez, Eduardo Fernández-Medina, Mario Piattini
- From Requirements to Web System Design. An Automated Approach using Graph Transformations. Sergio Segura, David Benavides, Antonio Ruiz-Cortés, María José Escalona
- 12:30 - 12:45 DESCANSO
- 12:45 - 14:30 SESIÓN: HERRAMIENTAS MDA Y EVALUACIÓN
 RubyTL a ATC: un caso real de transformación de transformaciones. Jesús Sánchez Cuadrado, Eduardo Victor Sánchez Rebull, Jesús García Molina, Antonio Estévez García
- Hacia un Marco Práctico de Evaluación de Tecnologías de Transformación de Modelos en la Plataforma Eclipse. Emilio A Sánchez, Gabriel Merino, Javier Muñoz
- Una revisión de herramientas MDA. Verónica Andrea Bollati, Juan Manuel Vara, Belén Vela, Esperanza Marcos
- 14:30 - 16:30 COMIDA
- 16:30 - 18:00 SESIÓN: MEDICIÓN Y DISEÑO ARQUITECTÓNICO
 Herramienta para la Definición de Modelos de Medición Software. Laura Sánchez, Francisco Ruiz, Mario Piattini
- Asignación de Tratamientos a Responsabilidades en el contexto del Diseño Arquitectónico Dirigido por Modelos. David Ameller, Xavier Franch
- Medición del Software mediante Transformaciones QVT. Beatriz Mora, Francisco Ruiz, Félix García, Mario Piattini
- 18:00 - 18:30 PAUSA CAFÉ
- 18:30 - 19:15 MESA REDONDA

: Zaragoza - España - 11 de Septiembre de 2007 :

Implementación de Heurísticas en QVT para la obtención de Clases de Análisis a partir de Modelos de Proceso de Negocio Seguros

Ignacio García-
Rodríguez de Guzmán
Grupo de investigación
ALARCOS
Instituto de I+D UCLM-
INDRA Universidad de
Castilla-La Mancha
Ignacio.Grodriguez@uclm.es

Alfonso Rodríguez
Departamento de Auditoría e
Informática
Universidad del Bio Bio
Chillán
Chile
alfonso@ubiobio.cl

Eduardo Fernández-Medina y
Mario Piattini
Grupo de investigación ALARCOS
Instituto de I+D UCLM-INDRA Universidad de
Castilla-La Mancha
{Eduardo.FdezMedina,Mario.Piattini}@uclm.es

Resumen

Dada la importancia actual de los procesos de negocio para mantener la competitividad de las empresas, cada vez se presta más y más atención a la forma en que estos se representan. Estándares como UML 2.0 y BPMN tratan de abordar su representación, sin embargo no contemplan la inclusión de ciertos aspectos relacionados con la seguridad, que son cruciales para el desempeño de los procesos. Habiendo solventado la problemática de la representación de Procesos de Negocio Seguros en trabajos anteriores, se proponen a continuación una serie de reglas definidas con QVT, que implementan un conjunto de heurísticas para la derivación de clases de análisis a partir de modelos de procesos de negocio seguros. Con esto, tanto los procesos de negocio (con BPMN) y los modelos de análisis (con UML) se representan a nivel de modelo. La definición de las reglas de transformación con QVT nos permite de este modo especificar en el modelo de análisis, las restricciones y necesidades de seguridad en los procesos de negocio desde el comienzo del modelado del sistema.

1. Introducción

Tal y como se ha venido observando en estos últimos años, la competitividad de las empresas está fuertemente vinculada con su capacidad para adaptarse a los cambios que surgen en el mercado.

En este contexto los procesos de negocio han resultado ser un valioso recurso que esta siendo utilizado para describir, estandarizar y adaptar, la forma en que las organizaciones reaccionan ante los estímulos externos y la manera en que interactúan con proveedores, clientes, socios comerciales y competidores [23].

Por su parte, la globalización de la economía y el uso intensivo de tecnologías de información y comunicaciones, ha implicado que las empresas no sólo amplíen sus negocios sino que también aumenten su vulnerabilidad. Como consecuencia, el número de ataques sobre los sistemas ha crecido, de manera que tarde o temprano algún intruso tendrá éxito.

Aunque la importancia de la seguridad en procesos de negocio es ampliamente aceptada, la perspectiva de los expertos del negocio en relación con la seguridad, hasta ahora, había sido escasamente tratada. En [20, 21] hemos introducido la representación de requisitos de seguridad en procesos de negocio. Para ello hemos extendido el Diagrama de Actividad de UML 2.0 [17] (UML 2.0-AD) y el Diagrama de Procesos de Negocio de BPMN [5] (BPMN-BPD), las dos principales notaciones para BP del mercado [14]. La extensión, que hemos llamado BPSec, permite capturar un conjunto de requisitos de seguridad que pueden ser especificados por el experto del negocio dando como resultado un SBP (Secure Business Process).

Por otra parte, ingeniería del software está siendo influenciada por la transformación de modelos, ya que con ello se apunta a resolver los problemas de

tiempo, costes y calidad asociados a la creación de software. La propuesta de OMG (Object Management Group) en este ámbito se denomina MDA (Model-Driven Architecture) [16]. MDA es un marco de trabajo en que debe ser posible (i) hacer una especificación de un sistema independiente de la plataforma que lo va a soportar, (ii) especificar plataformas, (iii) seleccionar una determinada plataforma para el sistema y (iv) transformar la especificación del sistema en una especificación para una plataforma en particular. La transformación entre modelos debe ser expresada de manera clara y precisa usando un lenguaje definido para ese propósito, como por ejemplo, VIATRA [6], ATL [4] o QVT (Query-View-Transformation) [18].

En el ámbito de MDA un SBP es un modelo independiente de computación (CIM, Computation Independent Model). En este nivel un SBP, descrito con BPMN o UML, es el punto de partida para la construcción de software.

En este trabajo se describen transformaciones entre modelos usando el lenguaje QVT. El objetivo principal es pasar desde un SBP hacia un conjunto de Clases de Análisis que forman parte de un modelo independiente de plataforma (PIM, Platform Independent Model). Hemos usado UML 2.0-AD como modelo de referencia para describir las transformaciones desde CIM hacia PIM (C2P). No obstante, como un SBP también puede ser descrito con BPMN-BPD hemos descrito reglas para transformar conceptos del modelo de BPMN-BPD hacia UML 2.0-AD. Esta transformación, desde CIM hacia CIM (C2C) es previa a la obtención de Clases de Análisis.

Tanto el modelo de SBP como el modelo de Clases de Análisis pueden ser utilizados como complemento en un proceso de desarrollo de software. Para ello hemos optado por el Proceso Unificado (UP) [13, 19], ya que es un método de construcción de software consolidado y exitoso.

Para presentar nuestra propuesta hemos organizado este artículo de la siguiente forma: en la Sección 2 presentaremos los principales aspectos relacionados con el modelado de procesos de negocio, en la Sección 3 se mostrará, de manera resumida, la forma en que se puede especificar requisitos de seguridad en procesos de negocio. En la Sección 4 describiremos las transformaciones que permiten obtener clases de análisis a partir de la especificación de un proceso de negocio seguro, en la Sección 5 mostraremos

un ejemplo ilustrativo y, finalmente, en la Sección 6 presentaremos nuestras conclusiones.

2. Modelado de Procesos de Negocio

Un modelo es una vista simplificada de una realidad compleja. Es una manera de crear una abstracción, que permita, al mismo tiempo, eliminar los detalles irrelevantes y centrarse en los aspectos importantes. Junto con ello, los modelos efectivos también facilitan el intercambio de puntos de vista entre diversos interesados en el negocio, permitiéndoles ponerse de acuerdo sobre aspectos fundamentales y orientar el trabajo hacia objetivos comunes [7]. En el modelado de proceso de negocio el objetivo principal es producir una descripción de la realidad, por ejemplo la forma en que se lleva a cabo una transacción comercial, que permita entenderla y eventualmente modificarla con el propósito de incorporar mejoras. Por lo tanto, es importante contar con una notación que permita modelar con la mayor claridad posible la esencia del negocio.

Para el modelado de procesos de negocio se han usado diversas técnicas. Entre las más importantes podemos encontrar: diagramas de flujo, diagramas de flujo de datos, diagramas entidad-relación, diagramas de transición de estado, gráficas Gantt, la familia IDEF (Integration DEfinition for Function modeling), redes de Petri, simulación, técnicas basadas en el conocimiento (inteligencia artificial) y diagramas de actividad de roles (Rol Activity Diagrams) [1, 9].

Sin embargo, en la actualidad, y de acuerdo con el estado de la industria del modelado de BP [14], es posible identificar a Unified Modeling Language (UML) y Business Process Modeling Notation (BPMN) entre los principales estándares.

Desde la perspectiva de la ingeniería de software, la descripción de un proceso de negocio es una fuente de requisitos que son necesarios, concisos, libres de perturbaciones propias de la implementación y no ambiguos. De manera que estos requisitos pueden ser transformados en conjunto de diagramas útiles para la construcción del software [10].

3. Seguridad en Procesos de Negocio

Los trabajos relacionados con la especificación de requisitos de seguridad en procesos de negocio [2,

11, 12, 15, 22, 25] coinciden en que es necesario capturar tempranamente requisitos de seguridad y que se deben incluir en un proceso de desarrollo de software. No obstante, no hemos encontrado trabajos que usen UML2.0-AD o BPMN-BPD para especificar procesos de negocios que incluyan requisitos de seguridad considerando la perspectiva del experto del negocio.

En trabajos previos hemos propuesto BPSec, una extensión para UML2.0-AD y BPMN-BPD, mediante la cual es posible representar requisitos de seguridad en procesos de negocio. Esta extensión está compuesta por un conjunto de estereotipos, restricciones y valores etiquetados que permiten adaptar el metamodelo de UML y BPMN, para incorporar estos nuevos elementos. En la Figura 1 se muestra la relación entre los estereotipos definidos en BPSec (en color gris) y los elementos de UML 2.0-AD con los cuales se relacionan.

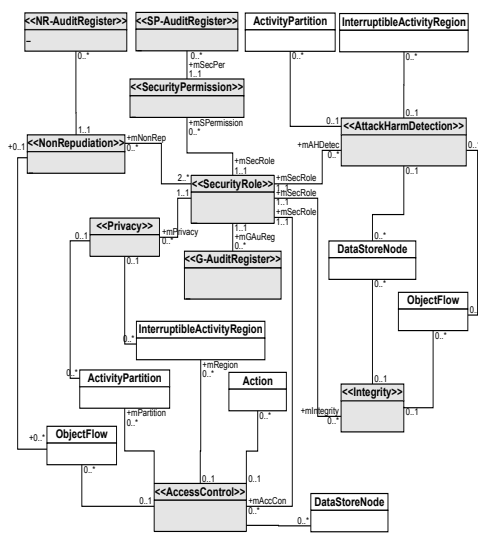


Figura 1. Estereotipos de BPSec y elementos de UML 2.0-AD

Los requisitos de seguridad que hemos considerado en BPSec son comprensibles para los analistas de negocios y no ambiguos para los expertos en seguridad. El subconjunto (no limitado) de requisitos de seguridad está compuesto por: Detección de Ataques y Amenazas, Control de Acceso, Integridad

Privacidad y No repudio. En la Figura 2a se muestra el icono genérico empleado para representar requisitos de seguridad. Si se desea indicar que un requisito de seguridad además debe tener registro de auditoría se debe usar un candado con una esquina doblada (Figura 2b).

El símbolo genérico usado para representar un requisito de seguridad con o sin registro de auditoría debe ser especializado con una abreviatura que permita identificar un requisito específico. Las abreviaturas válidas son: *AD* (AttackHarmDetection) (Figura 2c), *AC* (AccessControl) (Figura 2d), *I* (Integrity) (Figura 2e), *P* (Privacy) (Figura 2f) y *NR* (NonRepudiation) (Figura 2g).

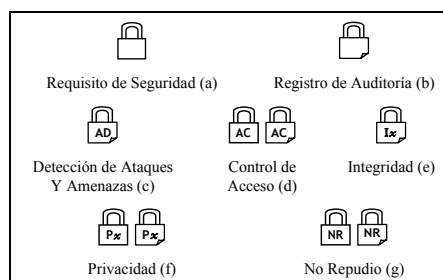


Figura 2. Iconos para representar seguridad en Procesos de Negocio

4. Descripción de transformaciones entre modelos

Consideramos que un proceso de negocio, construido por un analista de negocios, además de ser útil en el ámbito del negocio propiamente dicho, puede ser usado en un proceso de construcción de software. Desde allí se pueden obtener importantes requisitos del sistema, punto de partida en todos los procesos de desarrollo de software modernos.

Un esquema general de los elementos que están presentes en nuestra propuesta se muestra en la Figura 3. En la primera columna (izquierda) se muestra los tres tipos de modelos que conforman MDA. En la última columna se muestran los flujos de trabajo del Proceso Unificado. En la parte central está nuestra propuesta y los artefactos que son derivados de su aplicación. Las

transformaciones que proponemos en este trabajo están destacadas en gris oscuro (C2C y C2P-1).

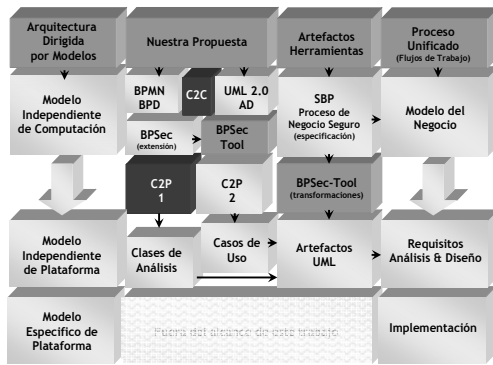


Figura 3. Esquema general de nuestra propuesta

Si se observa la Figura 3 en forma horizontal se puede ver que hay una correspondencia entre los elementos que presentamos en nuestra propuesta con los modelos de MDA y los flujos de trabajo de UP. De manera que un SBP es un modelo CIM que puede ser usado como complemento en el flujo de trabajo “Modelo del Negocio”. Y las Clases de Análisis que forman parte de un modelo PIM pueden ser usadas en los flujos de trabajo “Requisitos” y “Análisis & Diseño”.

En la transformación C2C (CIM→CIM) se especifica el mapeo entre conceptos de BPMN-BPD y UML 2.0-AD. En [26] se hace una comparación de ambas notaciones en que se considera la capacidad técnica que tiene cada una para representar patrones y su legibilidad. No obstante, en este trabajo consideramos la equivalencia de elementos de una y otra notación en función del objeto del proceso de negocio que es representado. Por ejemplo un *Pool* o *Lane* en BPMN-BPD representa un área en que se pueden agrupar un conjunto de actividades que lleva a cabo un determinado actor en un proceso de negocio. El elemento equivalente es *ActivityPartition* de UML 2.0-AD. En nuestro caso hemos puesto especial atención en los elementos de BPMN-BPD y UML 2.0-AD que están relacionados con BPSec (ver clases de UML 2.0-AD en la Figura 1). Las transformaciones están especificadas en lenguaje QVT gráfico y se muestran en la Figura 4.

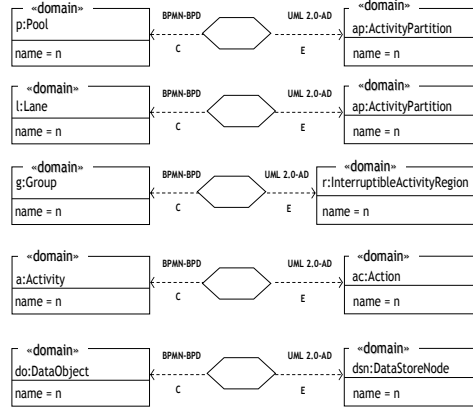


Figura 4. Reglas QVT para transformaciones desde BPMN-BPD hacia UML 2.0-AD

En la transformación C2P (CIM→PIM), se parte desde una especificación de un SBP para obtener un modelo de Clases de Análisis. En la revisión de la literatura hemos encontrado dos trabajos que abordan de manera directa este tipo de transformaciones. En el primero de ellos [3] se transforman diagramas de actividad en clases de análisis. La transformación no se hace en forma automática y se utiliza una versión previa a UML 2.0. Y en el segundo de trabajo [24], el diseñador de software examina el modelo del BP, descrito con BPMN, extrayendo las clases UML que son refinadas posteriormente. Nuestro trabajo se diferencia de los anteriores porque: (i) el modelo SBP no es intervenido manualmente ni antes ni después de hacer las transformaciones, (ii) las transformaciones son especificadas con el lenguaje QVT, (iii) tanto el SBP como las clases de análisis están vinculadas con UP y (iv) hemos diseñado una herramienta, BPSec-Tool, que permite especificar un SBP y automatiza la obtención de las Clases de Análisis. Dada la falta de motores QVT, esta herramienta implementa de manera algorítmica las transformaciones, aunque no obstante será mejorada en el futuro cuando aparezcan este tipo de entornos específicos para QVT. Aunque en principio los modelos no se manipulan, tal y como se a dicho en (i), con BPSec-Tool podría hacerse de manera sencilla, al basarse en un entorno gráfico.

Para mostrar las reglas de transformación (C2P) de manera más clara las hemos separado en dos tablas.

La Figura 5 muestra de manera esquemática cómo

se obtiene el diagrama de clases de análisis. De acuerdo a la Figura 5, mediante una transformación T1 (ver Tabla 1) se crea la parte del diagrama de clases directamente inferible a partir del diagrama de actividad que representa el proceso. Por otro lado, mediante otra transformación T2 (ver Tabla 2), se modifica este diagrama de clases con los aspectos de seguridad especificados mediante el modelo BPSec.

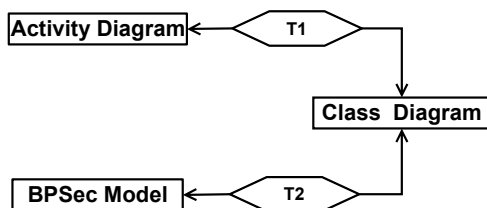


Figura 5. Modelos origen y transformaciones involucradas para la generación del diagrama de clases

En la Tabla 1 se muestran, en forma de reglas QVT, las heurísticas que se siguen para poder transformar aspectos generales de un proceso de negocio. Los elementos de UML 2.0-AD que pueden ser transformados directamente en clases de análisis son: particiones (*Activity Partition*), regiones (*Interruptible Activity Region*) y almacenes de datos (*Data Store Node*). Las acciones (*Action*), contenidas en regiones y/o particiones, son transformadas en operaciones de las clases derivadas desde las regiones o particiones en que se encuentran contenidas (ver *relation R4*). Además es posible obtener las relaciones de composición entre las clases derivadas de particiones que están divididas en sub-particiones (ver *relation R5*).

Tabla 1. Reglas QVT: UML 2.0-AD hacia Clases de Análisis

```

transformation ActivityDiagram2ClassDiagram
top relation R1
{
  checkonly domain uml_ActivityDiagram ap:ActivityPartition { name = n }
  enforce domain uml_ClassDiagram c:Class { name = n }
  where {
    ap.containedNode -> forAll(cn:Action | R4(cn));
    ap.containedNode-> for(ap:ActivityPartition | R5(c,ap));
  }
}
top relation R2
{
  fullName:String;
  checkonlydomain uml_ActivityDiagram r:InterruptibleActivityRegion{
    name = fullName
  }
}
  
```

```

enforce domain uml_ClassDiagram c:Class{
  name = fullName
}
}
where{
  r.containedNode->iterate(a:Action;
  activityRegionsName:Set(String) |
  activityRegionsName->including(a.name));
  activityRegionsName->iterate(s:String; fullName |
  fullName = fullName.concat(s));
  r.containedNodeforall(cn:Action | R4(cn));
}
}
top relation R3
{
  checkonly domain uml_ActivityDiagram dsn:DataStoreNode { name = n }
  enforce domain uml_ClassDiagram c:Class { name = n }
}
}
relation R4
{
  checkonly domain uml_ActivityDiagram ac:Action { name = n, inPartition=ap }
  enforce domain uml_ClassDiagram c:Class{
    name = ap.name,
    ownedOperation = OWNEDOPER:Set(Operation)
  }
  enforce domain uml_ClassDiagram op:Operation { name = n, ownerClass=c }
  where{
    OWNEDOPER->including(op);
  }
}
}
relation R5
{
  checkonly domain uml_ClassDiagram composingClass:Class {
    name = cName
  }
  checkonly domain uml_ActivityDiagram ap:ActivityPartition {
    name = apName
  }
  enforce domain uml_ClassDiagram composedClass:Class{
    name = apName
  }
  enforce domain uml_ClassDiagram assoc:Association{
    name = assocName,
    memberEnd = MEND: OrderedSet (Property),
    ownedEnd = OEND: OrderedSet (Property),
    navigableOwnedEnd = NOEND: OrderedSet (Property)
  }
  where{
    composingClassProp:Property {
      type = composingClass.name,
      name = 'm'+ composingClass.name
    };
    composedClassProp:Property {
      type = composedClass.name,
      name = 'm'+composedClass.name
      aggregation = AggregationKind.composite
    };
    MEND = MEND->including (composingClassProp);
    MEND = MEND->including (composedClassProp);
    OEND->including(composingClassProp);
    maininClass.ownedAttribute->including(secClassProp);
  }
}
}
  
```

En la Tabla 2 se muestran, también en QVT, la implementación de las heurísticas que permiten transformar los aspectos de seguridad que han sido definidos en un proceso de negocio a través de la extensión BPSec. Por ejemplo una especificación de un requisito de seguridad (*AttackHarmDetection*, *AccessControl*, *Integrity*, *Privacy* o *NonRepudiation*) da origen a una clase de análisis que toma el nombre del requisito de

seguridad (ver *top relation R6*). Del mismo modo, las especificaciones de registro de auditoría dan origen a otras clases de análisis. Por ejemplo, en *top relation R12*, a partir de una especificación de control de acceso con registro de auditoría, se crean las clases *SecurityPermission*, *SecurityRole* y la clase *SP-AuditRegister* que se crea invocando a *relation R14*.

Tabla 2. Reglas QVT: BPSec hacia Clases de Análisis

```

transformation BPSec2ClassDiagram
top relation R6
{
  checkonly domain uml_BPSec sr:SecurityRequirement (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = n)
}
top relation R7
{
  checkonly domain uml_BPSec sr:SecurityRequirement (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = "SecurityRole")
}
top relation R8
{
  checkonly domain uml_BPSec sr:SecurityRequirement (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = nc)
  where { nc= if (n="I" or n="AD") then "G_AuditRegister" endif;
  R13(c,tmp:Class(name="SecurityRole"));
}
top relation R9
{
  checkonly domain uml_BPSec ar:AuditRegister (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = nc)
  where { nc= if (n="AC") then "G_AuditRegister" endif;
  R13(c,tmp:Class(name="SecurityRole"));
}
top relation R10
{
  checkonly domain uml_BPSec ar:AuditRegister (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = nc)
  where { nc= if (n="P") then "G_AuditRegister" endif;
}
top relation R11
{
  checkonly domain uml_BPSec ar:AuditRegister (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = nc)
  where { nc= if (n="NR") then "NR_AuditRegister" endif;
}
top relation R12
{
  checkonly domain uml_BPSec ar:AuditRegister (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name = nc)
  where { nc= if (n="AC") then "SecurityPermission" endif;
  R13(c,tmp:Class(name="SecurityRole"));
  R14();
}
}
relation R13
{
  checkonly domain uml_ClassDiagram leftClass:Class (name = lcName)
  checkonly domain uml_ClassDiagram rightClass:Class (name = rcName)
  enforce domain uml_ClassDiagram assoc:Association{
    name = lcName+'_'+rc,
    memberEnd = MEND: OrderedSet (Property),
    ownedEnd = OEND: OrderedSet (Property),
    navigableOwnedEnd = NOEND: OrderedSet (Property)
  }
  where{
    leftClassProp:Property { name = 'm'+lcName, type = lcName
  };
    rightClassProp:Property { name = 'm'+rcName, type =rcName
  };
    MEND = MEND->including (leftClassProp);
    MEND = MEND->including (rightClassProp);
  }
}

```

```

leftClass.ownedAttribute->including(rightClassProp);
rightClass.ownedAttribute->including(leftClassProp);
}
}
relation R14
{
  checkonly domain uml_BPSec ar:AuditRegister (requirementtype = n)
  enforce domain uml_ClassDiagram c:Class (name =nc)
  where { nc= if (n="AC") then "SP_AuditRegister" endif;
  R13(c,tmp:Class(name="SecurityPermission"));
}
}

```

En síntesis, las transformaciones C2C deben ser aplicadas cuando el SBP haya sido descrito con BPMN-BPD. De esta forma, se tendrá sólo un modelo de entrada (UML 2.0-AD) para llevar a cabo las transformaciones C2P.

5. Ejemplo

En esta sección mostramos un SBP, el resultado de la aplicación de las reglas de transformación y el modelo de clases de análisis.

El proceso de negocio, relacionado con la admisión de pacientes en una institución médica, se inicia con una Solicitud de Admisión que es rellenada por un Paciente. Este documento, es enviado al Área de Administración para capturar la información relacionada con los seguros médicos y verificar la existencia de una Ficha Clínica asociada al paciente. Una vez que se verifica que la documentación del paciente sea válida y esté completa es enviada al Área Médica. El área de Evaluación Médica determina, a través de un conjunto de pruebas de pre-admisión, la condición medica del paciente. Si fuera necesario se harán exámenes adicionales que deberán ser registrados desde el punto de vista clínico y económico. Finalmente se completa el documento Evaluación Médica con información acerca del paciente, el cual se le envía. El proceso de negocio termina cuando el paciente ha recibido la Evaluación Médica. Desde el punto de vista de la seguridad se ha identificado Privacidad (con anonimato) para la partición Paciente, con el propósito de prevenir la obtención y almacenaje de información sensible acerca del paciente. Se ha especificado No Repudio sobre el flujo de datos "Solicitud Admisión" con la intención de evitar la negación de la recepción del documento. Los requisitos de seguridad Control de Acceso con Registro de Auditoría y Privacidad con confidencialidad han sido especificados sobre la región que se encuentra entre las particiones

Admisión y Contabilidad. También se ha especificado Integridad (alta) sobre el almacén de datos “Ficha Clínica” con la intención de proteger de manera especial ese documento. Finalmente, se ha especificado Detección de Ataques y Amenazas sobre el almacén de datos “Evaluación Médica” con la idea de registrar todos los eventos relacionados con intentos, fallidos y exitosos, de ataques o daños que se hagan sobre el documento. El proceso de negocio ha sido descrito con UML 2.0-AD y BPsec, para los cual hemos usado la herramienta BPsec-Tool. La pantalla en que se ha descrito el proceso de negocio “Admisión de Pacientes” se muestra en la Figura 5.

El resultado de la aplicación de las reglas QVT se describe en la Tabla 3.

Las clases de análisis derivadas de la aplicación de las reglas se muestran en la Figura 6. Se ha coloreado en gris las clases que están relacionadas con los aspectos de seguridad que han sido especificados en el proceso de negocio.

Tabla 3. Reglas aplicadas en “Admisión de Pacientes”

top relation R1:	Paciente, Área Administración, Admisión, Contabilidad, Área Médica, Evaluación Médica y Exámenes
top relation R2:	AdmisiónContabilidad
top relation R3:	Solicitud Admisión, Datos Contables, Ficha Clínica, Evaluación médica
relation R4:	En <i>Paciente</i> : Rellenar solicitud y Recibir Evaluación Médica. En <i>Área de Admisión</i> : Capturar Información de Seguros, Verificar Ficha Clínica, Crear Ficha Clínica, Rellenar información de Costos y Almacenar Datos. En <i>Administración</i> : Capturar Información de Seguros, Verificar Ficha Clínica y Crear Ficha Clínica. En <i>Contabilidad</i> : Rellenar información de Costos y Almacenar Datos. En <i>Región O1</i> : Capturar Información de Seguros, Verificar Ficha Clínica, Crear Ficha Clínica, Rellenar información de Costos y Almacenar Datos. En <i>Área Médica</i> : Hacer pruebas de pre-admisión, Evaluación exámenes Paciente, Rellenar Ficha Clínica, Rellenar Informe Paciente, Rellenar Datos Contables, Realizar Exámenes y Rellenar Ficha Clínica. En <i>Evaluación Médica</i> : Hacer pruebas de pre-admisión, Evaluación exámenes Paciente, Rellenar Ficha Clínica y Rellenar Informe Paciente. En <i>Área Médica</i> : Rellenar Datos Contables, Realizar Exámenes y Rellenar Ficha Clínica.
relation R5:	Área Administración compuesta por Admisión y Contabilidad y Área Médica compuesta por Evaluación médica Exámenes.
top relation R6:	Privacy, Nonrepudiation, AccessControl y Privacy, Integrity y AttackHarmDetection
top relation R7:	SecurityRole
top relation R8:	Security, Role, G-AuditRegister
top relation R9:	SecurityRole, G-AuditRegister
top relation R10:	G-AuditRegister
top relation R11:	NR-AuditRegister
top relation R12:	SecurityRole, G-AuditRegister, SP-AuditRegister
relation R13:	Se crean las relaciones entre las clases derivadas: <i>Privacy</i> con <i>Paciente</i> , <i>AdmisiónContabilidad</i> y <i>SecurityRole</i> ; <i>NonRepudiation</i> con <i>Solicitud Atención</i> ; <i>AccessControl</i> con <i>Admisión Contabilidad</i> y <i>SecurityRole</i> ; <i>Integrity</i> con <i>Ficha Clínica</i> y <i>SecurityRole</i> ; <i>AttackHarmDetection</i> con <i>Evaluación médica</i> y <i>SecurityRole</i> ; <i>SecurityRole</i> con <i>G-AuditRegister</i> y <i>SecurityPermission</i> ; <i>SecurityPermission</i> con <i>SP-AuditRegister</i>
relation R14:	SP-AuditRegister

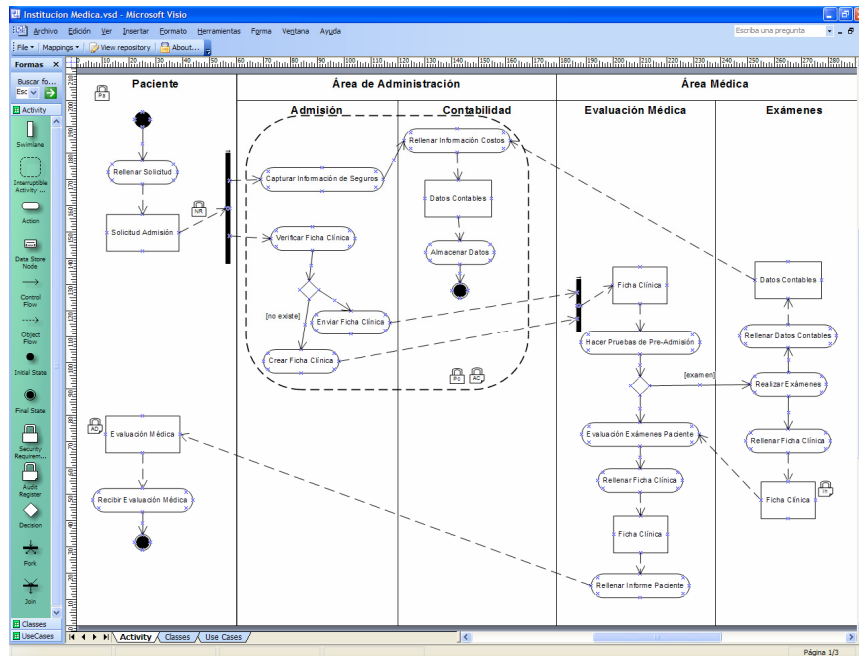


Figura 6. Admisión de Pacientes

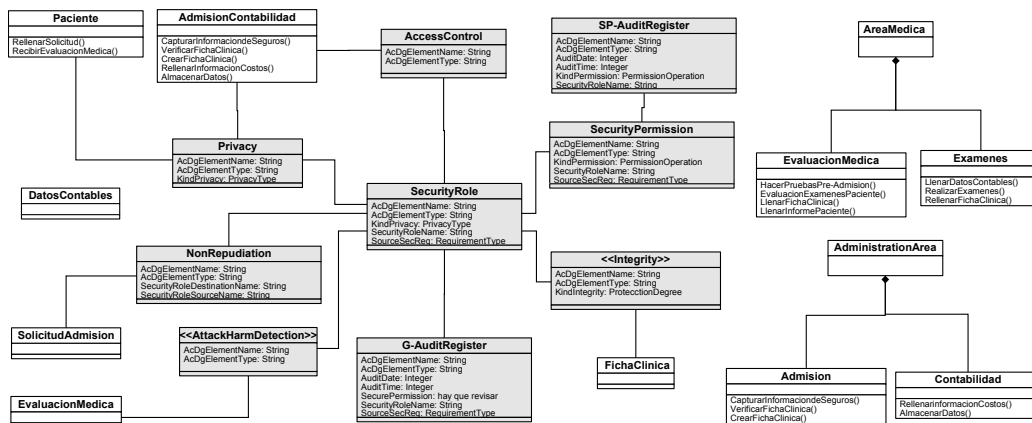


Figura 7. Clases de Análisis obtenidas desde “Admisión de Pacientes”

6. Conclusiones y trabajo futuro

La identificación temprana de los requisitos de un sistema favorece la calidad del mismo y reduce los costes y tiempo de desarrollo.

En este trabajo hemos presentado un conjunto de reglas que permiten pasar desde modelos especificados de manera independiente de computación hacia modelos independientes de plataforma en el ámbito de la arquitectura dirigida por modelos. Concretamente, se ha obtenido un modelo de Clases de Análisis a partir del modelo de un proceso de negocio seguro. Este último ha sido especificado con el Diagrama de Actividad de UML 2.0 extendido con BPSec, que permite incorporar requisitos de seguridad en el modelado de procesos de negocio. Para la especificar las transformaciones hemos descrito un conjunto de reglas usando el lenguaje QVT.

Tanto el proceso de negocio seguro y las clases obtenidas pueden ser utilizadas en los flujos de trabajo del las etapas de “Modelo del Negocio”, “Requisitos” y “Análisis & Diseño” del Proceso Unificado.

El trabajo futuro está orientado a enriquecer la especificación de las transformaciones orientado a obtener modelos de clases de análisis más completos, contemplando elementos y relaciones hasta ahora no considerados. Junto con ello mejoraremos el prototipo de la herramienta que

permite especificar un proceso de negocio seguro y realizar en forma automática las transformaciones.

Agradecimientos

Esta investigación es parte de los proyectos DIMENSIONS (PBC-05-012-1) y MISTICO (PBC06-0082), ambos parcialmente financiado por el FEDER y por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha, España; COMPETISOFT (506AC287) concedido por CYTED y ESFINGE (TIN2006-15175-C05-05/) otorgado por la Dirección General de Investigación del Ministerio de Ciencia y Tecnología, España.

Referencias

- [1] Aguilar-Savén, R. S. Business process modelling: Review and framework, International Journal of Production Economics, Vol. 90 (2), (2004), pp. 129-149.
- [2] Backes, M., Pfitzmann, B., y Waider, M. Security in Business Process Engineering, International Conference on Business

- Process Management (BPM), Eindhoven, Netherlands., (2003), pp. 168-183.
- [3] Barros, J. P. y Gomes, L. From Activity Diagrams to Class Diagrams, Workshop Dynamic Behaviour in UML Models: Semantic Questions In conjunction with Third International Conference on UML, York, UK, (2000).
- [4] Bézivin, J., Breton, E., Dupé, G., y Valduriez, P. The ATL Transformation-based Model Management Framework, IRIN-Université de Nantes, N° 03.08, (2003) 17 p.
- [5] BPMN. Business Process Modeling Notation Specification: OMG Final Adopted Specification, dtc/06-02-01, 2006.
- [6] Csertan, G., Huszerl, G., Majzik, I., Pap, Z., Pataricza, A., y Varro, D. VIATRA-Visual Automated Transformations for Formal Verification and Validation of UML Models, 17th IEEE International Conference on Automated Software Engineering (ASE), (2002), pp. 267-270.
- [7] Eriksson, H.-E. y Penker, M. Business Modeling with UML, OMG Press, (2001) p.
- [8] Firesmith, D. Specifying Reusable Security Requirements, Journal of Object Technology, Vol. 3 (1), January-February., (2004), pp. 61-75.
- [9] Giaglis, G. M. A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques, International Journal of Flexible Manufacturing Systems, Vol. 13 (2). (2001), pp. 209-228.
- [10] Harmon, P. The OMG's Model Driven Architecture and BPM, 2005: .Business Process Trends, Vol. 2 (5), 2004.
- [11] Herrmann, G. y Pernul, G. Viewing Business Process Security from Different Perspectives, 11th International Bled Electronic Commerce Conference, Slovenia., (1998), pp. 89-103.
- [12] Herrmann, P. y Herrmann, G. Security requirement analysis of business processes, Electronic Commerce Research, Vol. 6 (3-4), (2006), pp. 305-335.
- [13] Jacobson, I., Booch, G., y Rumbaugh, J. The Unified Software Development Process, (1999) 463 p.
- [14] Lonjon, A. Business Process Modeling and Standardization, BPTrends, In <http://www.bptrends.com/>, (2004).
- [15] Maña, A., Montenegro, J. A., Rudolph, C., y Vivas, J. L. A business process-driven approach to security engineering, 14th. International Workshop on Database and Expert Systems Applications (DEXA). Prague, Czech Republic., (2003), pp. 477-481.
- [16] Object Management Group. MDA Guide Version 1.0.1, 2003.
- [17] Object Management Group. Unified Modeling Language: Superstructure: version 2.0, formal/05-07-04, 2005.
- [18] QVT. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, (2005) 204 p.
- [19] Rational Software. Rational Unified Process, Best Practices for Software Development Teams, 2001.
- [20] Rodríguez, A., Fernández-Medina, E., y Piattini, M. Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes, 3rd International Conference on Trust, Privacy and Security in Digital Business (TrustBus), Krakow-Poland, (2006), pp. 51-61.
- [21] Rodríguez, A., Fernández-Medina, E., y Piattini, M. A BPMN Extension for the Modeling of Security Requirements in Business Processes, IEICE Transactions on Information and Systems, Vol. E90-D (4), (2007), pp. 745-752.
- [22] Röhm, A. W., Pernul, G., y Herrmann, G. Modelling Secure and Fair Electronic Commerce, 14th. Annual Computer Security Applications Conference, Scottsdale, Arizona, (1998), pp. 155-164.
- [23] Roser, S. y Bauer, B. A Categorization of Collaborative Business Process Modeling Techniques, 7th IEEE International Conference on E-Commerce Technology Workshops (CEC 2005), Munchen, Germany, (2005), pp. 43-54.
- [24] Rungworawut, W. y Senivongse, T. Using Ontology Search in the Design of Class Diagram from Business Process Model, Enformatika, Transactions on Engineering, Computing and Technology, Vol. 12, (2006), pp. 165-170.

- [25] Vivas, J. L., Montenegro, J. A., y Lopez, J. Towards a Business Process-Driven Framework for security Engineering with the UML, Information Security: 6th International Conference, ISC, Bristol, U.K., (2003), pp. 381-395.
- [26] White, S. A. Process Modeling Notations and Workflow Patterns, in BPTrends March, 2004: BPTrends March, 2004, 2004.