

## Actas de los Talleres de JISBD

- [JISBD](#)
- [Talleres](#)

Volumen 4. Número 2

DSDM 2010. Desarrollo de Software Dirigido por Modelos [\[TOC\]](#)

Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos.



Valencia, España

7 de septiembre de 2010

Editor(es):

Orlando Avila-García

Open Canarias S.L. (España)

Jordi Cabot

INRIA-École des Mines de Nantes (Francia)

Javier Muñoz

Prodevelop S.L. (España)

Jose Raúl Romero

Universidad de Córdoba (España)

Antonio Vallecillo

Universidad de Málaga (España)

Tabla de Contenidos

Sesión 1: Transformaciones

Adaptation of transformations to metamodel changes

Páginas 1-9

Jokin García y Oscar Diaz

[PDF](#)

Automatización de la Selección de Transformaciones Alternativas Basada en Atributos de Calidad

Páginas 10-18

Javier Gonzalez-Huerta, Emilio Insfran y Silvia Abrahão

[PDF](#)

De flujos de navegación a Spring Web Flow. Un primer acercamiento a las transformaciones verticales en MWACSL

Páginas 19-28

Antonia M. Reina-Quintero, Jesús Torres-Valderrama y Miguel Toro-Bonilla

[PDF](#)

Mejorando el nivel de automatización en el desarrollo dirigido por modelos de editores gráficos

Páginas 29-37

Alvaro Jimenez, Juan Manuel Vara, Veronica Andrea Bollati y Esperanza Marcos

[PDF](#)

Ingeniería inversa de eventos GUI en aplicaciones RAD mediante MDD

Páginas 38-46

Óscar Sánchez Ramón, Jesús Sánchez Cuadrado y Jesús García Molina

[PDF](#)

Sesión 2: Métodos

A Technological Framework to support Model Driven Method Engineering

Páginas 47-56

Mario Cervera, Manoli Albert, Victoria Torres, Vicente Pelechano, Javier Cano y Begoña Bonet

[PDF](#)

SOFIA: Smart Objects for Intelligent Applications – ADK

Páginas 57-59

Jesús Fernández Gómez-Pimpolo y Raúl Otaolea

[PDF](#)

A MDA Approach for Deriving Functional Testing Software for Validation of Active Applications in RDB

Páginas 60-69

Harith Al-Jumaily, Dolores Cuadra y Paloma Martínez

[PDF](#)

Análisis de Comunicaciones como un enfoque de requisitos para el desarrollo dirigido por modelos

Páginas 70-77

Marcela Ruiz, Sergio España, Arturo González y Óscar Pastor

[PDF](#)

Modernizing Legacy Systems through Runtime Models

Páginas 78-87

Ricardo Perez-Castillo, Barbara Weber, Ignacio García-Rodríguez de Guzmán y Mario Piattini

[PDF](#)

Decisiones arquitectónicas y tecnológicas como líneas de producto en el desarrollo dirigido por modelos

Páginas 88-97

Jose García-alonso, Jose Javier Berrocal Olmeda y Juan Manuel Murillo

[PDF](#)

Sesión 3: Herramientas y Aplicaciones

EMF4CPP: a C++ Ecore Implementation

Páginas 98-106

Andrés Senac, Diego Sevilla y Gregorio Martínez

[PDF](#)

Generación de modelos de servicios en SoaML desde modelos de procesos de negocio en BPMN con QVT

Páginas 107-116

Andrea Delgado, Francisco Ruiz, Ignacio García-Rodríguez de Guzmán y Mario Piattini

[PDF](#)

Variability issues in MARTE for SPL Model Analysis

Páginas 117-125

Lorea Belategui, Goiuria Sagardui, Joseba Andoni Agirre y Leire Etxeberria

[PDF](#)

Un motor de generación de código dirigido por modelos de base de datos, como punto de partida para la implantación de una plataforma MDA en la administración balear

Páginas 126-135

Víctor García Pau y José A. Carsí

[PDF](#)

Definición y ejecución de métricas en el contexto de ADM

Páginas 136-145

Javier Luis Canovas, Belén Cruz y Jesus Garcia-Molina

[PDF](#)

Modelos weaving para trazabilidad de requisitos Web en A-OOH

Páginas 146-155

José Alfonso Aguilar Calderon, Irene Garrigos y Jose-Norberto Mazon  
[PDF](#)

© 2010, SISTEDES.

Última actualización 14 de Julio de 2010.

# Generación de modelos de servicios en SoaML desde modelos de procesos de negocio en BPMN con QVT

Andrea Delgado

Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República  
1300 Montevideo, Uruguay  
[adelgado@fing.edu.uy](mailto:adelgado@fing.edu.uy)

Ignacio García-Rodríguez  
de Guzmán

Grupo Alarcos  
Departamento de Tecnologías y  
Sistemas de Información  
Universidad de Castilla – La Mancha  
13071 Ciudad Real  
[ignacio.grodriguez@uclm.es](mailto:ignacio.grodriguez@uclm.es)

Francisco Ruiz,  
Mario Piattini

Grupo Alarcos  
Departamento de Tecnologías y  
Sistemas de Información  
Universidad de Castilla – La Mancha  
13071 Ciudad Real  
[{francisco.ruizg,mario.piattini}@uclm.es](mailto:{francisco.ruizg,mario.piattini}@uclm.es)

## Resumen

El modelado de procesos de negocio en las organizaciones permite explicitar el conjunto de actividades que son necesarias para lograr los objetivos del negocio definidos. Tradicionalmente éstos han sido parte del área del negocio sin demasiada relación con su implementación tecnológica por el área de sistemas, que en general se ha realizado con una visión vertical de la organización por sus secciones. La incorporación de cambios a los sistemas y la interacción entre éstos ha requerido esfuerzos importantes, debido en parte a la visión vertical del desarrollo y al alto acoplamiento entre los procesos de negocio (muchas veces implícitos en los sistemas) y su implementación. La realización de procesos de negocio con servicios introduce una capa de intermedia, que permite mayor agilidad en los cambios en los procesos o las tecnologías. La generación automática de servicios en SoaML desde procesos de negocio en BPMN que se presenta, aporta a la agilidad organizacional proveyendo además trazabilidad entre los elementos involucrados.

## 1. Introducción

El paradigma de Computación Orientada a Servicios (Service Oriented Computing, SOC) basa el diseño de aplicaciones masivas distribuidas y evolutivas en servicios [1] elementos de software reutilizables mediante los cuales consumidores y proveedores de servicios interactúan en forma desacoplada para realizar procesos de negocio en secuencias de invocaciones a servicios. La Arquitectura Orientada a Servicios (Service Oriented

Architecture, SOA) es un estilo de arquitectura para la implementación con servicios [2][3]. Los servicios constituyen actualmente una de las implementaciones más utilizadas, junto con motores de procesos, para soportar las diversas etapas de la Gestión de Procesos de Negocio (Business Process Management, BPM) [4]. La introducción de una capa de servicios intermedia entre la capa de definición de los procesos de negocio y la capa de implementación en distintas tecnologías, aporta a reducir el acoplamiento entre las mismas, vía los servicios definidos. Los servicios realizan y por lo tanto se mapean a actividades, sub-procesos y/o procesos de negocio completos, relacionando el diseño de software con la definición de los procesos. A su vez, los servicios definidos son implementados por componentes y sistemas de software, mapeando el diseño de software con su implementación. De esta forma, se agiliza la introducción de cambios tanto en los procesos de negocio como en su implementación, minimizando el impacto de los cambios en una capa sobre la otra [2][3].

Sin embargo, las ventajas de la introducción de servicios en las organizaciones no han sido totalmente alcanzadas, en parte por la escasez o poca utilización de metodologías para guiar el desarrollo orientado a servicios. Si bien la implementación y ejecución de servicios en los últimos años ha madurado considerablemente, el modelado de servicios aún está en definición. Este modelado es fundamental entre otras cosas, para permitir la automatización de distintas etapas del desarrollo de software utilizando el paradigma de Desarrollo Dirigido por Modelos (Model Driven Development, MDD)[5], incluyendo la generación de código desde estos modelos. El estándar Soa Modeling Language (SoaML) [6] del Object

Management Group (OMG) es un paso en este sentido. SoaML define un metamodelo y un perfil UML para modelar servicios, extendiendo UML.

En este artículo se presenta la definición de transformaciones QVT [7] desde modelos de procesos de negocio en Business Process Modeling Notation (BPMN) [8] a modelos de servicios en SoaML, para automatizar la obtención de servicios directamente desde los procesos de negocio. El estándar BPMN constituye actualmente el estándar más importante para modelado de procesos de negocio, y el estándar SoaML constituye la base para la homogeneización del modelado de servicios, que hasta su definición se realizaba con distintas notaciones y por lo tanto distintos elementos y distinta semántica asociada a los mismos [1]. Estas transformaciones se integran en la definición del marco MINERVA (Model drIveN and sERvice oRIented framework for the continuous business processes improVement & relAted tools) [9] que integra los paradigmas BPM, SOC y MDD para soportar la mejora continua de procesos de negocio. La utilización de estándares es la base del marco, por lo que además de BPMN y SoaML se utiliza QVT para las transformaciones.

El resto del documento se organiza de la siguiente forma: en la sección 2 se presentan los principales elementos de SoaML y BPMN en base al modelo de proceso de negocio “Otorgar Préstamo”, en la sección 3 se describen las transformaciones QVT definidas, en la sección 4 se muestran las transformaciones sobre el ejemplo presentado, y finalmente en la sección 5 se presentan conclusiones y trabajo futuro.

## 2. Principales elementos de los estándares BPMN y SoaML

Para ilustrar los elementos de BPMN se introduce en la Figura 1 el modelo de proceso de negocio “Otorgar Préstamo” que se utilizará como ejemplo a lo largo de este artículo. BPMN especifica un único tipo de diagrama, el Business Process Diagram (BPD), un conjunto de elementos núcleo con el cual modelar la mayoría de los procesos de negocio y un conjunto completo. Los elementos definidos se agrupan en: Swimlanes, Objetos de Flujo, Objetos de Conexión y Artefactos. En el grupo de Swimlanes se definen: Pool que representa una entidad o rol participante (ej. Banco), y Lane que representa una sub-partición

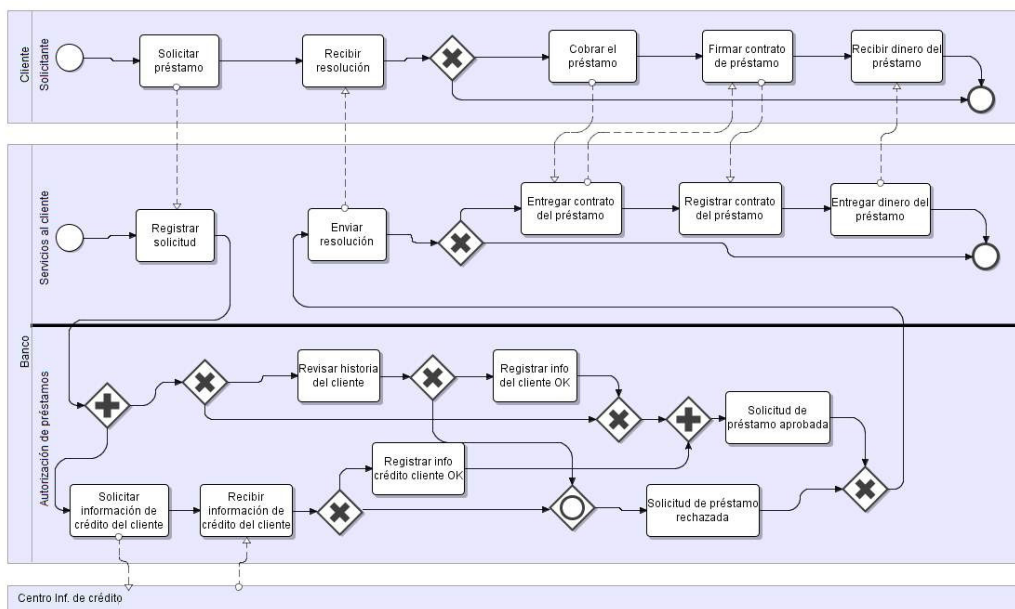


Figura 1. Proceso de negocio “Otorgar Préstamo” en BPMN

dentro de un Pool para organizar actividades (ej. Servicios al Cliente).

El grupo de objetos de Flujo contiene los distintos tipos de Actividad que representa el trabajo que la organización realiza, y puede ser atómica o un sub-proceso compuesto de otras actividades (ej. de Actividad atómica Solicitar Préstamo). También están entre los Objetos de Flujo, los puntos de decisión utilizados para controlar la divergencia y convergencia del flujo del proceso, que puede ser paralelo (AND, nodos con cruz horizontal en Autorización de Préstamo), exclusivo (XOR, nodos con cruz inclinada), inclusivo (OR, nodo con el círculo en blanco) o complejo. Incluye también los eventos que son algo que ocurre en el curso del proceso afectando su flujo, con una causa (trigger) y un impacto (result) y pueden ser de inicio, intermedios o fin, y pueden tener un tipo como tiempo o error (ej. eventos de inicio y fin simples).

El grupo de objetos de conexión define los objetos de flujo de secuencia que muestran el orden en que las actividades del proceso se realizan, desde el inicio hasta el fin, y el flujo de mensajes entre dos participantes (las flechas enteras son de secuencia y las punteadas de mensajes). Los objetos de flujo de secuencia solo pueden utilizarse dentro de un mismo pool, entre distintos pools se deben utilizar los objetos de flujo de mensajes. En el grupo de Artefactos se

definen objetos de texto, de datos y grupos. El estándar se encuentra actualmente bajo modificación ya que la última versión 2.0 beta1 introduce nuevos elementos y cambia otros existentes en los estándares previos.

Por otro lado, el estándar SoaML define servicio (Service) como un recurso que permite el acceso a una o más capacidades provistas, mediante la interface definida, que es ejercitado consistentemente con restricciones y políticas especificadas en la descripción del servicio. La arquitectura general de servicios se especifica en la colaboración ServicesArchitecture, donde se muestran los roles participantes y los contratos de servicios definidos. En la Figura 2 se presenta una parte del modelo en SoaML correspondiente a la ServicesArchitecture para el proceso de negocio "Otorgar Préstamo" presentado antes, con tres contratos de servicio a modo de ejemplo. Un servicio se provee por una entidad (provider, ej. :Banco) para su uso por otros (consumers, ej. :Cliente), donde los consumidores del servicio pueden no ser conocidos para el proveedor. Un servicio puede tener una o más interfaces y un contrato asociado. Un contrato de servicio (ServiceContract, en el ejemplo uno es :Registrar solicitud) es una colaboración que define los términos, condiciones, interfaces y coreografía en que los participantes acuerdan para usar el servicio. Las interfaces pueden ser de tipo simple

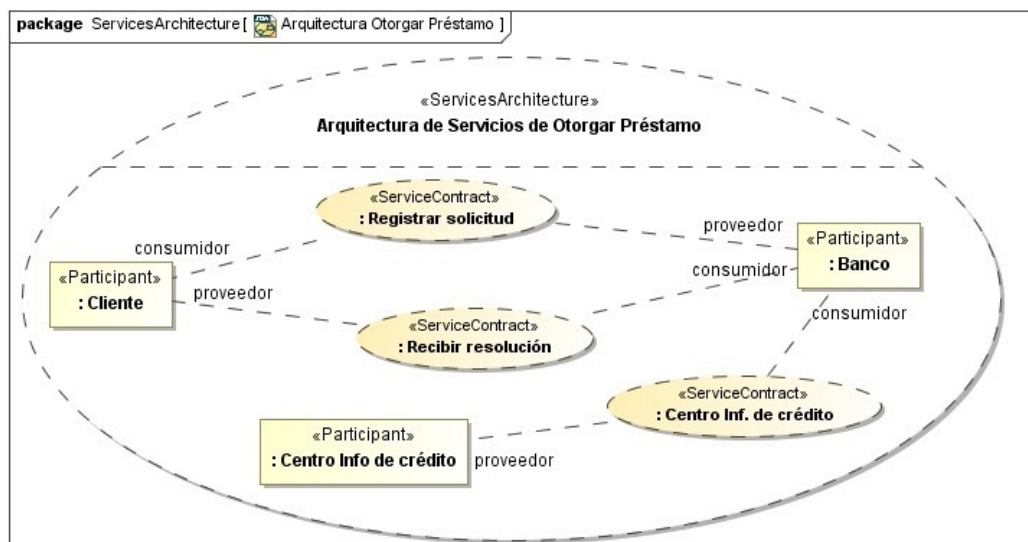


Figura 2. Parte de la Arquitectura de servicios para el Proceso de negocio "Otorgar Préstamo" en SoaML

UML Interface, o de tipo interface de servicio (ServiceInterface) que es una clase definida con tres secciones principales: las interfaces provistas y requeridas, la clase interface y el protocolo de comportamiento (choreography). Los participantes (Participant) son componentes de software, organizaciones o sistemas que proveen y usan servicios, ofreciendo capacidades en un ServicePoint y requiriendo otras en un RequestPoint, ambos especializando Port UML. Un canal de servicios (ServiceChannel) permite la comunicación entre estos, y los tipos de mensajes (MessageType) especifican el tipo de información a intercambiar en las operaciones. La arquitectura interna de cada participante se presenta en una ParticipantArchitecture que se realiza por cada participante, definiendo servicios internos que permiten proveer los servicios externos ofrecidos.

El estándar se encuentra actualmente en su versión beta2, en la cual se introdujeron cambios como el nombre de ServicePoint y RequestPoint a Service y Request respectivamente, y la sustitución del uso de la ParticipantArchitecture por el mismo tipo de colaboración ServicesArchitecture para modelar la arquitectura interna de cada participante.

### 3. Transformaciones QVT de BPMN a SoaML

Si bien la realización de procesos de negocio con servicios tiene asociadas las ventajas mencionadas previamente, la definición de los servicios “correctos” para soportar estos procesos es todavía un desafío. Existen varias iniciativas para proveer soporte metodológico para derivar servicios desde procesos de negocio y automatizar esta derivación cuando sea posible, pero el problema está aún en estudio. Para guiar el

desarrollo orientado a servicios en MINERVA incluimos la metodología Business Process Service Oriented Methodology (BPSOM) [10] a la cual estamos agregando la generación automática de servicios desde procesos de negocio que se presenta, en las actividades de BPSOM definidas.

La importancia de generar servicios automáticamente directamente desde los procesos de negocio es la de la generación automática: permitir el reuso de conocimiento, reducir los errores de diseño, registrar la trazabilidad de las relaciones entre elementos de los distintos modelos, y mejorar la productividad. Al diseñar o generar servicios otro aspecto importante a tener en cuenta es la existencia de servicios ya implementados en la organización que puedan ser reutilizados. En BPSOM se considera con una actividad que explícitamente requiere la revisión de un catálogo de servicios existentes, que serán incluidos en el modelo SoaML obtenido. En este sentido la re-ejecución de transformaciones deberá preservar estas definiciones.

Un elemento clave para definir las transformaciones son las relaciones entre elementos de un modelo de procesos de negocio y un modelo de servicios. Para conceptualizar estas relaciones y los conceptos involucrados en cada modelo, definimos una ontología de modelado [11] que se incluye en MINERVA para su utilización como referencia conceptual. Teniendo en cuenta la conceptualización realizada, hemos identificado varias correspondencias entre los conceptos involucrados en el modelado de procesos de negocio y servicios, que luego trasladamos a los metamodelos de BPMN y SoaML. En la Tabla 1 se muestran las principales correspondencias definidas entre los metamodelos involucrados, y las transformaciones QVT que los relacionan.

**Tabla 1.** Principales relaciones y conceptos en las transformaciones QVT

Modelo Procesos de Negocio	Transformación QVT	Modelo Servicios
BusinessProcessDiagram	ProcessToModel	ServicesModel
Pool	PoolToParticipant	Participant
Lane	LaneToParticipant	Participant
Activity+IncomingMessages	ActivityMessageToServicePoint	ServicePoint
Activity+OutgoingMessages	ActivityMessageToRequestPoint	RequestPoint
MessagingEdge+Target	PoolMessageToServicePoint	ServicePoint
MessagingEdge+Source	PoolMessageToRequestPoint	RequestPoint

Las relaciones identificadas nos permiten obtener los primeros elementos en el modelo SoaML para definir la arquitectura de alto nivel. En la Tabla 2 se presenta un sub-conjunto de las transformaciones QVT definidas. De cada Pool se obtiene un participante en la arquitectura que serán quienes ofrezcan y requieran servicios, de cada Lane se obtiene un participante interno a éstos, que se utilizará en la definición de la arquitectura interna asociada al participante. Las actividades que se tendrán en cuenta para las transformaciones son las marcadas con el tipo “Service” que indica que esa actividad será soportada por un servicio. El marcado del modelo BPMN para generar servicios, lo realiza el Arquitecto de software como parte del diseño de servicios. Otras actividades podrán ser manuales o de cualquier otro tipo de BPMN y no serán

transformadas. De los mensajes que entran a las actividades podemos identificar los servicios a proveer (para su ejecución) en los ServicePoint asociados al participante que corresponde al pool que contiene a la actividad. De los mensajes que salen de las actividades podemos identificar los servicios requeridos de otros participantes a los que corresponde el Pool que contiene a la actividad. En SoaML los ServicePoint y RequestPoint son conjugados, esto es si un participante brinda un servicio en un ServicePoint habrá otro que lo requiera en un RequestPoint. Debido a que en BPMN los mensajes pueden asociarse también a Pools sin necesidad de que estén específicamente asociados a una actividad en el Pool, deben ser identificados para generar también los servicios asociados. Esto ocurre generalmente cuando de un Pool no se conoce su

**Tabla 2.** Sub-conjunto de las transformaciones QVT definidas de BPMN a SoaML.

<b>Reglas de relaciones definidas en QVT</b>
<b>top relation ProcessToModel</b> { checkonly domain bpmn bp : bpmn::BpmnDiagram{name = pn}; enforce domain soaml sm : SoaML::Model{name = pn };}
<b>top relation PoolToParticipant</b> { checkonly domain bpmn p : bpmn::Pool{name = pn}; enforce domain soaml s : SoaML::Participant{name = pn};}
<b>top relation LaneToParticipant</b> { checkonly domain bpmn p : bpmn::Lane{name = pn }; enforce domain soaml s : SoaML::Participant{ name = pn };}
<b>top relation ActivityMessageToServicePoint</b> { checkonly domain bpmn c : bpmn::Activity{lanes = p : bpmn::Lane{ }, activityType = bpmn::ActivityType::Task, incomingMessages = im : bpmn::MessagingEdge{ }, name = cn}; enforce domain soaml t : SoaML::ServicePoint { participant = s : SoaML::Participant { }, isService = true, name = cn}; when { p.pool.bpmnDiagram.pools.lanes.activities -> exists (x:bpmn::MessageVertex   (x.outgoingMessages.target = c.incomingMessages.target) and (x.oclAsType(bpmn::Activity). activityType = c.activityType)) or (p.pool.bpmnDiagram.pools -> exists (x:bpmn::MessageVertex   (x.outgoingMessages.target = c.incomingMessages.target))); PoolToParticipant (p.pool, s); } }
<b>top relation PoolMessageToServicePoint</b> { checkonly domain bpmn c : bpmn::MessagingEdge{ target = d : bpmn::MessageVertex {name = cn} enforce domain soaml t : SoaML::ServicePoint { participant = s : SoaML::Participant { }, isService = true, name = cn }; when {PoolToParticipant (d, s);} }



estructura interna (o no se quiere mostrar) y solo se definen los mensajes intercambiados con otros Pools. La granularidad de los servicios es un tema complejo y ampliamente tratado en la literatura [2][3]. Si bien en general es recomendable que la granularidad sea de tipo gruesa, servicios básicos generados podrán ser combinados en un servicio de mayor granularidad, si se considera adecuado. Como se puede observar en la Tabla 2, las primeras tres reglas permiten obtener el Model SoaML de servicios desde el diagrama de procesos con la regla "ProcessToModel", y los participantes desde cada Pool y desde cada Lane dentro de los Pool, con las reglas "PoolToParticipant" y "LaneToParticipant". Los primeros serán utilizados para definir la arquitectura de servicios para el proceso de negocio completo, los segundos serán utilizados para describir la arquitectura interna de cada uno de los anteriores.

En la cuarta regla nombrada como "ActivityMessageToServicePoint" se generan los ServicePoint que serán los servicios provistos por los participantes con que esté asociada la actividad que los determina. En esta regla se recorren las actividades en el modelo BPMN para determinar las conexiones con mensajes existentes. Solo estamos transformando actividades conectadas con mensajes, esto es, actividades que se encuentran en distintos Pools, o Pools que se encuentran conectados con actividades. La expresión OCL en la cláusula when chequea que los mensajes entrantes a la actividad evaluada provenga de un Pool o de otra actividad. Esto es una restricción del metamodelo de la herramienta que estamos utilizando ya que los eventos y los gateways también están definidos como actividades, por lo que tuvimos que cambiar la restricción original de que el tipo de actividad fuera Servicio.

La última regla denominada "PoolMessageToServicePoint" genera los ServicePoint que no están asociados con actividades sino directamente con Pools, como permite BPMN cuando un Pool no se encuentra expandido. Estas transformaciones no son las únicas que se pueden definir para obtener los elementos que estamos transformando, por lo cual estamos investigando también otras formas de obtenerlos, y de obtener el resto de los elementos para generar modelos SoaML completos.

#### 4. Ejemplo de las transformaciones

El ejemplo corresponde al proceso de negocio de un banco genérico "Otorgar Préstamo" en el cual hay tres participantes involucrados: el Cliente que solicita el préstamo, el Banco que recepciona la solicitud, la evalúa y autoriza o no el préstamo, y el Centro de Información de Créditos al cual el Banco le solicita información sobre la situación crediticia del Cliente, cuando la solicitud se encuentra en evaluación. En la Figura 1 en la sección 2 se puede ver el modelo del proceso de negocio "Otorgar Préstamo" en BPMN.

Para realizar este ejemplo integramos varias herramientas al entorno Eclipse, ya que hemos definido en la dimensión de las herramientas de MINERVA el soporte [12] basado en Eclipse. Por lo que utilizamos para el modelado de procesos de negocio el plug-in de Eclipse BPMN Modeler [13], y para el modelado SoaML el plug-in Magic Draw Cameo SOA+ [14] debido a que no hay aún demasiadas implementaciones del estándar y queríamos una integración en Eclipse para tener todas las herramientas en el mismo entorno. Estamos definiendo nuestra implementación de un plug-in Eclipse para SoaML que integraremos. Desde el modelo SoaML es posible generar el código asociado con el motor MDA de ModelPro [15]. Para la definición y ejecución de las transformaciones QVT usamos el plug-in de Eclipse MediniQVT [16].

Como se observa en la Figura 1, además de los tres Pool mencionados el Banco tiene dos Lanes una para "Servicios al Cliente" y otra para "Autorización de Préstamos". Existen tres mensajes enviados del Cliente al Banco, tres mensajes enviados del Banco al Cliente, y dos mensajes entre el Banco y el Centro de Autorización de Créditos. Los metamodelos del BPMN Modeler y de SoaML se cargan en el entorno Eclipse como metamodelos Ecore [17] mediante las facilidades que provee Eclipse, y el modelo del proceso de negocio "Otorgar Préstamo" en formato XMI, que cumple con el metamodelo de BPMN Modeler puede entonces ser transformada en un modelo SoaML en formato XMI que cumple con el metamodelo SoaML. Luego de ejecutar las transformaciones QVT se obtiene el archivo XMI que corresponde con el modelo SoaML generado.

```

<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SoaML="http://www.soaml.org/SoaML/1.0.0" xsi:schemaLocation="http://www.soaml.org/SoaML/1.0.0 ../metamodel/SoaML.ecore">
<SoaML:Participant xmi:id="_VIVgY3LEd-QcLXVKWdxxv" name="Autorización de préstamos"/>
<SoaML:Participant xmi:id="_VIVHc33LEd-QcLXVKWdxxv" name="Servicios al cliente"/>
<SoaML:Participant xmi:id="_VIVHc23LEd-QcLXVKWdxxv" name="Solicitante"/>
<SoaML:Participant xmi:id="_VIVHd33LEd-QcLXVKWdxxv" name="Centro Inf. de crédito"/>
<SoaML:ServicePoint xmi:id="_VIVug33LEd-QcLXVKWdxxv" name="Centro Inf. de crédito" isService="true"
participant="_VIVHd33LEd-QcLXVKWdxxv"/>
<SoaML:Participant xmi:id="_VIVugm33LEd-QcLXVKWdxxv" name="Banco"/>
<SoaML:ServicePoint xmi:id="_VIVuh33LEd-QcLXVKWdxxv" name="Registrar solicitud" isService="true"
participant="_VIVugm33LEd-QcLXVKWdxxv"/>
<SoaML:ServicePoint xmi:id="_VIXVkg33LEd-QcLXVKWdxxv" name="Entregar contrato del préstamo" isService="true"
participant="_VIVugm33LEd-QcLXVKWdxxv"/>
<SoaML:ServicePoint xmi:id="_VIXVkm33LEd-QcLXVKWdxxv" name="Registrar contrato del préstamo" isService="true"
participant="_VIVugm33LEd-QcLXVKWdxxv"/>
<SoaML:ServicePoint xmi:id="_VIX8o33LEd-QcLXVKWdxxv" name="Recibir información de crédito del cliente" isService="true"
participant="_VIVugm33LEd-QcLXVKWdxxv"/>
<SoaML:Participant xmi:id="_VIX8om33LEd-QcLXVKWdxxv" name="Cliente"/>
<SoaML:ServicePoint xmi:id="_VIX8p33LEd-QcLXVKWdxxv" name="Recibir resolución" isService="true"
participant="_VIX8om33LEd-QcLXVKWdxxv"/>
<SoaML:ServicePoint xmi:id="_VIX8om33LEd-QcLXVKWdxxv" name="Firmar contrato de préstamo" isService="true"
participant="_VIX8om33LEd-QcLXVKWdxxv"/>
<SoaML:ServicePoint xmi:id="_VIX8om33LEd-QcLXVKWdxxv" name="Recibir dinero del préstamo" isService="true"
participant="_VIX8om33LEd-QcLXVKWdxxv"/>
<SoaML:Model xmi:id="_VIX8v33LEd-QcLXVKWdxxv"/>
</xmi:XMI>

```

Figura 3. Archivo XMI generado correspondiente al modelo SoaML

La Figura 3 presenta el archivo XMI generado, y muestra los participantes generados correspondientes al Cliente, Banco y Centro de Información de Crédito, más los participantes internos al Banco “Servicios al Cliente” y “Autorización de Préstamo”. Para el Centro de Información de Crédito se obtiene el ServicePoint del mismo nombre, ya que al no estar expandido no se conocen sus actividades. Para el Banco se obtienen los ServicePoint: “Registrar solicitud”, “Entregar contrato del préstamo”, “Registrar contrato del préstamo” de la interacción de mensajes con el Cliente, y el ServicePoint

“Recibir información de crédito del cliente” de la interacción con el Centro de Información. Para el Cliente se generan los ServicePoint: “Recibir resolución”, “Firmar contrato del préstamo” y “Recibir dinero del préstamo” de la interacción con el Banco. Todos los ServicePoint generados tienen el nombre de la actividad asociada, el atributo isService en True, y tienen asociado el identificador del participante (interno) que los provee. Luego el archivo XMI se debe cargar en el modelador para visualizar los diagramas y desde estos generar el código. En la Figura 4 se muestra en forma gráfica la parte del diagrama

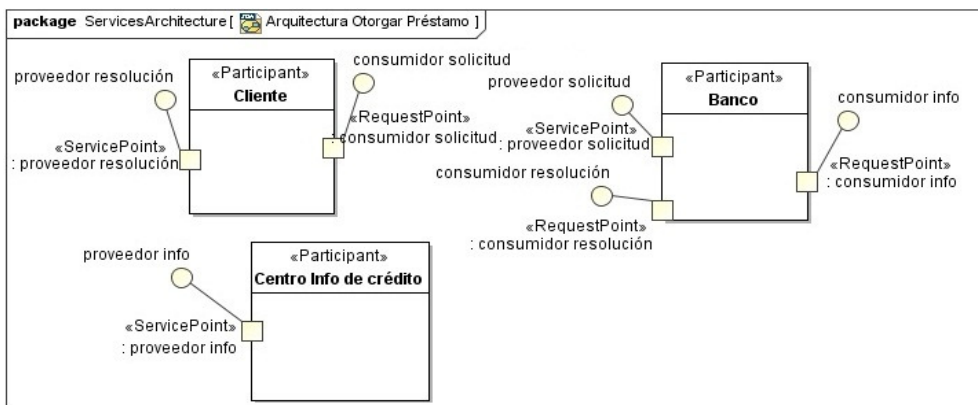


Figura 4. Parte del Diagrama ServicesArchitecture en SoaML

ServicesArchitecture asociada con la generación de estos elementos básicos. El diagrama generado es parte del tipo de diagrama de ServicesArchitecture para definir la arquitectura de alto nivel para el proceso de negocio. Estamos trabajando en primera instancia con la vista estructural de servicios, para luego incorporar la generación de la parte dinámica asociada por ejemplo, a los contratos. La generación de los participantes involucrados y de los ServicePoint y RequestPoint asociados para proveer y requerir servicios entre ellos, corresponde a los elementos básicos con que se debe contar para presentar los diagramas SoaML completos. Como puede observarse en la Figura 4, se muestran los tres participantes del proceso de negocio generados, con ejemplos de puertos ServicePoint y RequestPoint asociados a cada uno, mostrando el tipo de las interfaces. Si bien aún no se generan los contratos de servicio, ni las interfaces asociadas con sus operaciones, parámetros y tipos de mensajes, la obtención de los elementos básicos de la ServicesArchitecture provee la base para el resto de las transformaciones.

## 5. Trabajos relacionados

Existen varias propuestas para transformar y generar modelos de software desde procesos de negocio utilizando lenguajes existentes como QVT o ATL[18] o utilizando nuevos enfoques o lenguajes definidos. BPMN es la notación más utilizada para modelar procesos de negocio, y UML para el modelado de software y servicios. En [19] se proponen transformaciones de BPMN a UML pero con foco en elementos de seguridad agregados al modelo BPMN, para obtener primero un DA de UML y de ahí derivar casos de uso y clases de análisis. [20] también transforma BPMN en UML anotando el modelo BPMN con información a ser procesada por las transformaciones, obteniendo varios artefactos UML como DA, casos de uso y diagramas de colaboración y deployment. Ambos trabajos difieren del nuestro en que no usamos artefactos intermedios para pasar del modelo BPMN al modelo SoaML sino que lo hacemos directamente.

En [21] se modela el comportamiento de sistemas Web con cuatro PIMs: servicios de usuario, casos de uso extendidos, proceso de servicios y composición de servicios, definiendo

reglas de mapeo entre los modelos que se automatizan parcial o totalmente. En [22] se agrega un modelo de valor desde el cual se definen transformaciones a modelos de casos de uso, automatizadas con ATL, que también se usa en [23] definiendo dos tipos de reglas de transformación: de generación básica para crear elementos del modelo destino y de ligamiento para generar links entre éstos. A diferencia usamos metamodelos estándar existentes, derivando servicios directamente desde los procesos de negocio, con QVT.

En [24] se definen transformaciones conceptuales basadas en la sucesiva aplicación de patrones desde la capa superior a la capa inferior, utilizando grafos para hacer pattern matching. [25] define dos pasos para transformar procesos de negocio en servicios: identificar tareas en los procesos que son invocaciones a servicios, luego integrando el modelado de procesos de negocio y objetos en un modelo de servicios del negocio (Business Service Model, BSM) mediador entre los requerimientos y su implementación. En contraste utilizamos modelo de diseño de servicios sin mediadores, y no proponemos nuevos patrones sino la utilización de patrones de procesos existentes, los patrones de procesos de negocio [26] principalmente para la validación de los modelos. Estamos explorando también su uso como guía en transformaciones más complejas.

## 6. Conclusiones y trabajo futuro

La propuesta presentada para obtener modelos de servicios en SoaML desde modelos de procesos de negocio en BPMN mediante transformaciones QVT, pretende automatizar lo más posible la generación de servicios desde procesos de negocio, en base a la utilización de notaciones y lenguajes estándares. Si bien las transformaciones corresponden a un conjunto reducido de reglas para un conjunto reducido de elementos seleccionados de BPMN y SoaML, creemos que proveen la base para la definición de transformaciones para el resto de los elementos, que es en lo que estamos trabajando actualmente.

Estas transformaciones se integran en la metodología BPSOM para desarrollo orientado a servicios desde procesos de negocio definido en el marco MINERVA que constituye el eje del trabajo de investigación. El marco está basado en

la integración y utilización de diversos estándares existentes para procesos de negocio, servicios y desarrollo dirigido por modelos. En ese sentido BPMN, SoaML y QVT, alineado con MDA, son estándares de OMG integrados para la generación automática de servicios presentada en este artículo. De la misma forma el soporte de herramientas definido está basado en el entorno Eclipse, integrando diversos plug-ins que soporten las distintas etapas del desarrollo. Adicionalmente estamos trabajando en la definición de un plug-in de Eclipse propio para implementar el estándar SoaML que se integre en el entorno.

El conjunto completo de transformaciones QVT con el soporte metodológico y de herramientas integrado en el marco MINERVA servirá para guiar todo el proceso de desarrollo de aplicaciones orientadas a servicios desde procesos de negocio. Creemos que MINERVA puede resultar una guía útil y soporte de herramientas basado en Eclipse, para ser utilizado en organizaciones que requieran una integración rápida y fácil de metodologías, herramientas y conceptos para adoptar los paradigmas BPM, SOC y MDD.

**Agradecimientos.** Este trabajo ha sido parcialmente financiado por la Agencia Nacional de Investigación e Innovación (ANII), Uruguay, proyecto ALTAMIRA (Junta de Comunidades de Castilla-La Mancha, España, Fondo Social Europeo, PII2I09-0106-2463), proyecto PEGASO /MAGO (Ministerio de Ciencia e Innovación MICINN, España, y Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01).

## Referencias

- [1] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenge, IEEE Computer Society, (2007).
- [2] Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design, PrenticeHall, (2005).
- [3] Krafzig, D., et.al, Enterprise SOA, SOA: Best Practices, Prentice Hall, (2005).
- [4] Smith, H., Fingar, P., Business Process Management: The third wave, Meghan-Kieffer, (2003).
- [5] Mellor, S., Clark, A., Futagami, T., Model Driven Development - Guest editors introduction, IEEE Computer Society, September/October, (2003).
- [6] Soa Modeling Language (SoaML), v.1.0 Beta1, OMG, (2009).
- [7] Query/Views/Transformations (QVT), v.1.0, OMG, (2008).
- [8] Business Process Modeling Notation (BPMN), OMG, (2008).
- [9] Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M.: MINERVA: Model driven and service oriented framework for the continuous business processes improvement & related tools, In: 5th IW on Engineering Service-Oriented Applications (WESOA'09), in ICSOC, Stockholm, (2009).
- [10] Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M., Towards a Service-Oriented and Model-Driven framework with business processes as first-class citizens, In: 2nd IC on Business Process and Services Computing (BPSC'09), Leipzig, (2009).
- [11] Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M., Towards an ontology for service oriented modeling supporting business processes, 4th. IC on Research Challenges in Information Science (RCIS'10), Nice, (2010).
- [12] Delgado, A., García - Rodríguez de Guzmán, I., Ruiz, F., Piattini, M.: Tool support for Service Oriented development from Business Processes, 2nd Int. Work. on Model-Driven Service Engineering (MOSE'10), in TOOLS, Málaga, (2010)
- [13] SOA Tools Platform (STP) BPMN Modeler, <http://www.eclipse.org/bpmn/>
- [14] Magic Draw Cameo SOA+, <http://www.nomagic.com/>
- [15] ModelPro, <http://modeldriven.org/>
- [16] Medini QVT, ikv++ technologies ag, <http://projects.ikv.de/qvt/>
- [17] Eclipse Process Framework Composer (EPF Composer), <http://www.eclipse.org/epf/>
- [18] Jouault, F., Kurtev, I., Transforming Models with ATL (ATLAS Transformation Language), Satellite Events at MoDELS Conference (2006)
- [19] Rodríguez, A.; Fernández-Medina, E.; Piattini, M.: Towards CIM to PIM Transformation: From Secure Business Processes Defined in

- BPMN to Use-Cases. 5th International Conference on Business Process Management (BPM'07), (2007).
- [20] Liew,P., Kontogiannis,K. Tong,T., A Framework for Business Model Driven Development,12th Int. Workshop on Sw. Tech. and Eng. Practice (STEP'04), (2004).
- [21] de Castro, V., Marcos, E., López Sanz, M., A model driven method for service composition modelling: a case study, Int. J. Web Engineering and Tech., Vol. 2, No. 4, (2006)
- [22] de Castro V., Vara Mesa J. M., Herrmann E., Marcos E., A Model Driven Approach for the Alignment of Business and Information Systems Models, (2008)
- [23] Touzi J., Benaben F., Pingaud H., Lorré J.P., A model-driven approach for collaborative service-oriented architecture design, Int. J. of Prod. Economics,Vol.121 Is. 1, (2009)
- [24] Gacitua-Decar V., Pahl C., Pattern-based business-driven analysis and design of service architectures, 3rd Int. Conf. on Software and Data Technologies SE (ICSOFT'08), (2008)
- [25] Rychly M., Weiss P., Modeling of SOA: from business process to service realization, 3rd Int. Conf. on Evaluation of Novel Approaches to SE, (ENASE 08), (2008)
- [26] van der Aalst, W.; ter Hofstede, A.; Kiepuszewski, B.; Barros, A. 2003. Workflow Patterns, In Distributed and Parallel Databases, 14(3), pages 5-51.