

Desarrollo de servicios con SoaML desde procesos de negocio en BPMN: metodología y automatización

Andrea Delgado¹, Ignacio García-Rodríguez de Guzmán², Francisco Ruiz²

¹ Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, 1300 Montevideo, Uruguay
adelgado@fing.edu.uy

² Grupo Alarcos, Depto. de Tecnologías y Sistemas de Información, Universidad de Castilla – La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, España
{ignacio.grodriguez, francisco.ruiz}@uclm.es

Resumen. La sinergia entre los paradigmas de Gestión de Procesos de Negocio (Business Process Management, BPM) y Computación Orientada a Servicios (Service Oriented Computing, SOC) establecida en los últimos años, aporta al cierre de la brecha negocio-sistemas mediante la realización de procesos de negocio (PN) como servicios. Si bien la implementación y ejecución de servicios es un área que en los últimos años ha madurado considerablemente, el diseño y modelado de servicios aún está en definición. El modelado de servicios es fundamental entre otras cosas, para la automatización de distintas etapas del desarrollo en base al paradigma de Desarrollo Dirigido por Modelos (Model Driven Development, MDD). El estándar SoaML es un paso para el avance en dicha área, definiendo un metamodelo y un perfil UML para el modelado de servicios, con estereotipos específicos. En este artículo presentamos la extensión de nuestra metodología BPSOM para desarrollo de servicios desde PN, integrando dos aspectos principales: el nuevo estándar SoaML para diseño y modelado de servicios, y transformaciones QVT para generación automática de servicios en SoaML desde modelos de PN en BPMN.

Palabras clave: Business Process Management (BPM), Service Oriented Computing (SOC), Model Driven Development (MDD), metodologías de diseño y desarrollo de servicios, generación automática, QVT, BPMN, SoaML.

1 Introducción

La realización de procesos de negocio (PN) con servicios se ha convertido en una de las formas preferidas para implementar el software para informatizar los procesos de negocio en las organizaciones. La aplicación conjunta de los paradigmas de Gestión de Procesos de Negocio (Business Process Management, BPM) [1][2][3] y de Orientación a Servicios (Service Oriented Computing, SOC) [4] provee la base conceptual y técnica, para acercar la visión de las áreas del negocio y software en pos del objetivo común que es poder realizar el negocio que define a la organización de la forma más adecuada, permitiendo la introducción de cambios tanto en los PN como

en el software asociado, minimizando el impacto de dichos cambios y permitiendo su incorporación con la mayor agilidad posible. La Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) [5][6] es un estilo de arquitectura que soporta el paradigma de desarrollo SOC, usualmente implementado con Servicios Web (Web Services, WS). Si bien la implementación y ejecución de servicios es un área que en los últimos años ha madurado considerablemente, el diseño de modelos de servicios aún está en definición. El modelado de servicios es fundamental entre otras cosas, para la automatización de distintas etapas del desarrollo de software utilizando el paradigma de Desarrollo Dirigido por Modelos (Model Driven Development, MDD) [7] que agrega a dicha visión el enfoque centrado en modelos, tanto de PN como de servicios, para relacionar y transformar elementos de un modelo origen con elementos de otro modelo destino, mediante la definición de correspondencias entre los mismos. El enfoque de Arquitectura Dirigida por Modelos (Model Driven Architecture, MDA) [8] es la propuesta de OMG para MDD. El estándar de Modelado de Arquitecturas Orientadas a Servicios (Service Oriented Architecture Modeling Language, SoaML) [9] de OMG es un paso para el avance del modelado de servicios, que define un metamodelo y un perfil UML, extendiendo la notación UML.

La metodología de desarrollo orientado a servicios desde PN (Business Process Service Oriented Methodology, BPSOM) que guía el desarrollo de servicios desde PN, integra la dimensión metodológica del marco MINERVA [10][11] definido para soportar la mejora continua de PN realizados por servicios y basada en modelos, integrando los paradigmas mencionados de BPM, SOC y MDD. MINERVA comprende tres dimensiones en las que se integran conceptos [12], metodologías [13] [14], y herramientas [15], para soportar las definiciones y el trabajo en cada fase definida. En este artículo se presenta la extensión de las definiciones de BPSOM [13], integrando dos aspectos principales para la realización de PN con servicios: el uso del estándar SoaML para modelado de servicios, y la definición de transformaciones con el lenguaje Query/View/Transformations (QVT) [16] entre modelos de PN especificados en la notación de Modelado de Procesos de Negocio (Business Process Modeling Notation, BPMN)[17] y modelos de servicios en SoaML. Esto permite generar automáticamente los servicios necesarios desde los modelos de PN en base a las relaciones identificadas entre elementos de ambos metamodelos.

El resto del artículo está organizado como sigue: en la sección 2 se presenta el modelado de PN y servicios con los estándares BPMN y SoaML asociados mediante el ejemplo que se introduce, y el enfoque que tomamos para su generación automática; en la sección 3 se describe brevemente la metodología BPSOM detallando la utilización del estándar SoaML en cada actividad definida, los diagramas a realizar, los roles responsables, los estereotipos a utilizar, siguiendo el ejemplo introducido en la sección anterior; en la sección 4 se describen los trabajos relacionados y finalmente en la sección 5 se presentan conclusiones y trabajo futuro.

2 Modelado de PN y servicios

Hemos detectado siete principios básicos a tener en cuenta al realizar la integración de los paradigmas SOC y MDD a los procesos de negocio [18]: el modelado de PN y de servicios, transformación de modelos, enfoque metodológico, uso de patrones

(procesos y diseño), procesos colaborativos y herramientas de soporte, que fueron tenidos en cuenta en la definición de MINERVA. Para el modelado de PN y servicios se utiliza una amplia variedad de notaciones existentes, siendo los principales BPMN para PN y UML para servicios. Aunque SoaML es un estándar de reciente definición, es esperable que sea rápidamente incorporado por la comunidad de software para el modelado de servicios, ya que extiende UML que es ampliamente utilizado.

2.1 Elementos clave en BPMN2

La notación para Modelado de Procesos de Negocio (BPMN) [17] de OMG, es una notación estándar para modelar visualmente flujos de procesos que tiene como objetivo proveer notación común para analistas del negocio que crean los flujos iniciales de los procesos y desarrolladores de software responsables de la tecnología e implementación de los procesos. La versión actual (BPMN2) introduce varios cambios a las anteriores, organizando el modelado de PN en base a cuatro construcciones principales: *Process*, *Collaboration* y *Choreography* que integran el núcleo de elementos definidos, y *Conversation* que es un uso particular y una descripción informal de una *Collaboration*. De esta forma es posible especificar una orquestación (*Process*) que define un proceso en los límites de una organización que lo controla, una colaboración (*Collaboration*) que permite especificar la interacción entre procesos de dos o más organizaciones que interactúan en base a mensajes y flujo de mensajes, donde ninguna tiene el control absoluto de la interacción; una coreografía (*Choreography*) que describe en detalle la interacción entre procesos de una colaboración con foco en los mensajes intercambiados, y una conversación que muestra una visión global de una colaboración. Estos elementos se conectan al elemento de definiciones (*Definitions*) que es el elemento raíz en la jerarquía. Se modelan explícitamente los servicios definidos integrando varios elementos como interfaces (*Interfaces*), operaciones (*Operations*), y mensajes (*Messages*) de entrada y salida y sus relaciones. El resto de elementos de las versiones anteriores de BPMN se mantienen: objetos de flujo, de conexión, swimlanes y artefactos.

En la Fig. 1 se presenta el modelo BPMN del proceso “Admisión y Registro de Paciente para Cirugía Mayor Ambulatoria (CMA)”, uno de los PN que estamos trabajando con el Hospital General de Ciudad Real. Muestra los participantes “Hospital Público Local”, “Paciente” y “Registro Central de Salud”, donde el paciente solicita programar la CMA (por ej. página Web), la secretaría del Hospital reserva día y hora tentativos que le son enviados al paciente (por ej. e-mail o sms) a la vez que se solicita el registro médico del paciente al Registro Central de Salud. El paciente se presenta el día asignado para la CMA entregando la orden para la cirugía, se chequean las precondiciones para la CMA (análisis de sangre, medicamentos, ayuno, etc.) y si hay algún problema se informa al paciente y se cancela la cirugía, de lo contrario se realizan los pasos definidos para preparar al paciente para la CMA.

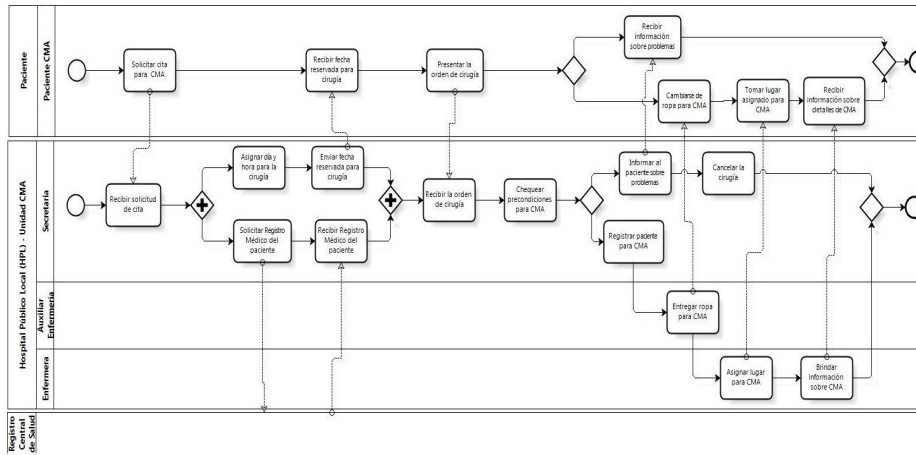


Fig. 1. Proceso “Admisión y Registro de paciente para CMA” en BPMN

2.2 Elementos clave en SoAML

El Lenguaje de Modelado para Arquitecturas Orientadas a Servicios (Service Oriented Architecture Modeling Language, SoAML) [9] de OMG provee un perfil UML y un metamodelo que extiende el metamodelo UML para diseñar servicios en SOA, la versión actual del estándar es la beta 2. En SoAML se define un servicio como una oferta de valor según una o más capacidades (abstracción de la habilidad de actuar y producir una salida y resultado) que tiene interface/s y un contrato asociados. Las interfaces pueden ser de tipo Interface de Servicio (*ServiceInterface*) o Interface simple (*Interface*) UML. Un Contrato de Servicios (*ServiceContract*) define los términos, condiciones, interfaces y coreografía en que los participantes acuerdan para utilizarlo, esta última puede expresar con cualquier diagrama de comportamiento (*Behavior*) UML, generalmente uno de secuencia. Una Arquitectura de Servicios (*ServiceArchitecture*) es una colaboración UML que presenta los participantes, contratos de servicios y roles en los mismos, brindando una visión global de los servicios provistos y requeridos. Los Participantes (*Participants*) pueden ser componentes de software, organizaciones, o sistemas que proveen y usan éstos servicios, ofreciendo capacidades en puntos de servicio (*Service*) y requiriendo servicios en puntos de solicitud (*Request*), siendo ambos (puntos de servicios y solicitud) especializaciones de *Port* UML (en la versión anterior beta 1: *ServicePoint* y *RequestPoint*). Un canal de servicios (*ServiceChannel*) modela la comunicación entre proveedores y consumidores de servicios, el tipo de mensaje (*MessagesType*) especifica la información intercambiada. La Fig. 2 muestra uno de los diagramas de servicios de SoAML, el diagrama de Arquitectura de Servicios (*ServicesArchitecture*, *SA*) correspondiente al PN de la Fig. 1.

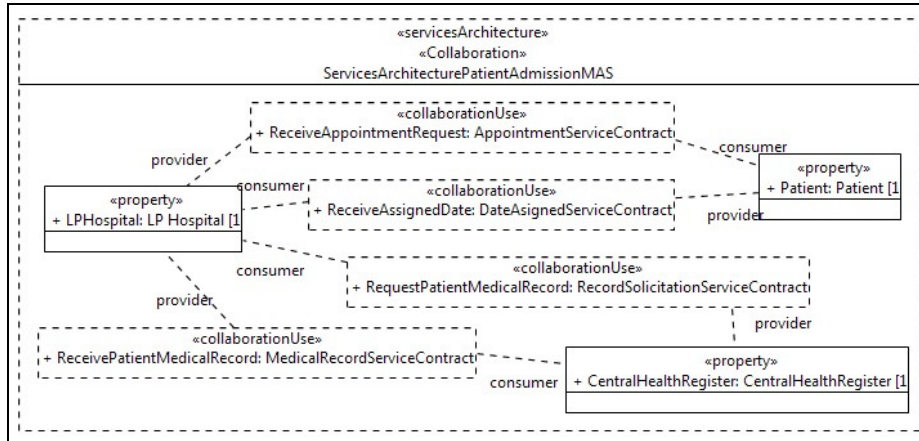


Fig. 2. Arquitectura de Servicios SoAML para el proceso de la Fig. 1

El diagrama de Arquitectura del modelo de servicios en la Fig. 2 muestra los participantes “Hospital PL”, “Paciente” y “RegistroCentralSalud” que fueron definidos como pools en el PN en la Fig. 1. La Fig. 2 muestra también cuatro contratos de servicios (*ServiceContract*) correspondientes a los servicios provistos y requeridos por los participantes involucrados. Cada participante tiene un rol en la interacción con el servicio que puede ser proveedor y consumidor que se muestran en el diagrama. Esta conexión define también los puertos (*Service o Request*) donde los participantes proveen o consumen cada servicio. El diagrama de la Fig. 2, así como el resto de los diagramas de SoAML que se presentan en el artículo, están realizados con el plug-in SoAML para Eclipse que estamos desarrollando integrado en MINERVA.

2.3 Derivación y generación de servicios desde PN

La metodología BPSOM define como derivar servicios desde modelos conceptuales de PN, identificando los participantes involucrados y, para éstos, los servicios provistos y consumidos, contratos asociados con definición de interfaces, parámetros de entrada y salida y mensajes intercambiados. Con la aparición del estándar SoAML hemos incluido en la metodología su utilización en base a las definiciones previas realizadas, incluyendo en las actividades todos los elementos y diagramas de SoAML, y adecuándolas cuando se ha visto necesario. Adicionalmente al uso de BPMN y SoAML, definimos transformaciones QVT entre los metamodelos de dichos estándares, que nos permiten automatizar (parcialmente) la generación de servicios desde PN, obteniendo modelos de servicios desde modelos de PN. Nuestro enfoque para la generación de servicios en SoAML desde modelos de PN en BPMN sigue los principios de MDA y se basa completamente en el uso de estándares OMG. En la Fig. 3 se muestra la visión MDA propuesta por el marco MINERVA.

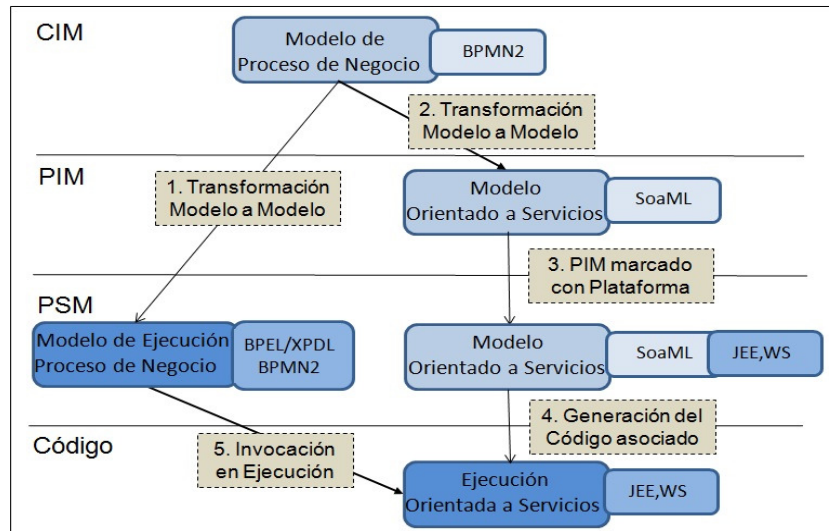


Fig. 3. Aplicación de MDA a MINERVA para generación de servicios desde PN

El modelo de PN en BPMN constituye el Modelo Independiente de la Computación (Computation Independent Model, CIM) en la transformación y el modelo de servicios en SoaML constituye el Modelo Independiente de la Plataforma (Platform Independent Model, PIM). Desde el modelo de servicios en SoaML se obtiene el Modelo Específico de la Plataforma (Platform Specific Model, PSM) en la plataforma seleccionada y el código asociado con motores MDA (ej. ModelPro¹), lo que permite completar la trazabilidad desde el PN a su implementación. Los servicios serán luego invocados desde los motores de procesos ejecutando el PN en BPEL/XPDL/BPMN2 siendo el PSM y (código) ejecutable obtenido desde el modelo BPMN2 que con la transformación identidad es también el PIM (no incluido en la Fig.3). Las transformaciones QVT entre los metamodelos de los estándares (paso 2 Fig.3) se basan en una ontología definida previamente que relaciona los elementos de modelado de PN y servicios [10], obteniendo luego las correspondencias de la Fig.4.

BPMN 2.0	SoaML beta2	BPMN 2.0	SoaML beta2
Definitions (complete model)	Model <Model>	Interface <Interface>	Interface <Interface>
Process	Participant <Class>	Operation <Operation>	Operation <Operation>
Message	MessageType <Class>	MessageRef In Out	Parameter In Out <Parameter>
ServiceTask (provider)	Service (of Participant) <Port>	Collaboration	Services Architecture <Collaboration>
Task (consumer)	Request (of Participant) <Port>	Collaboration Participant	Participant Part <Property>
MessageFlow sourceRef targetRef	ServiceContract <Collaboration> Provider <Property> provider Consumer <Property> consumer	MessageFlow sourceRef targetRef	CollaborationUse <Collaboration Use> Dependency consumer provider <Dependency>

Fig. 4. Correspondencias clave definidas entre elementos de BPMN2 y SoaML beta 2

¹ <http://modeldriven.org/>

3 Definición de BPSOM

BPSOM está definida para ser integrada en el proceso de desarrollo de software existente en una organización, ya que nuestro objetivo es reutilizar el conocimiento existente y solo agregar elementos específicos para guiar el desarrollo orientado a servicios (OS) desde PN. En [13] se presenta el proceso de definición de BPSOM, sus disciplinas, actividades, roles y artefactos, así como un ejemplo de su utilización. Desde esa definición hemos implementado la metodología como un *Method plug-in* en Eclipse Process Framework (EPF) Composer [15] y publicado como sitio Web [20], para proveer interoperabilidad con procesos definidos de la misma forma. A continuación extendemos la definición inicial de BPSOM introduciendo la de este artículo, dos elementos que consideramos clave para el desarrollo con servicios: SoaML para modelado de servicios, y la generación automática de modelos de servicios con transformaciones QVT desde BPMN2 a SoaML.

Las disciplinas que BPSOM define como prioritarias para el desarrollo orientado a servicios desde PN son: Modelado del Negocio, Diseño e Implementación. Cada disciplina incluye actividades clave para realizar la definición, especificación, diseño e implementación de servicios, así como roles involucrados, artefactos a producir (y requeridos) por las actividades y plantillas para su realización. Otras disciplinas de Ingeniería necesarias, como Requisitos, Verificación y Despliegue, son las propias del proceso de desarrollo existente, para poder reutilizar el proceso con el cual las personas están acostumbradas a trabajar. Lo mismo se define para las disciplinas de soporte como Aseguramiento de la Calidad, Gestión de la Configuración y Gestión del Proyecto. En la Fig. 5 se muestra la definición general de BPSOM y su uso en el contexto del proceso de desarrollo de software existente en una organización.

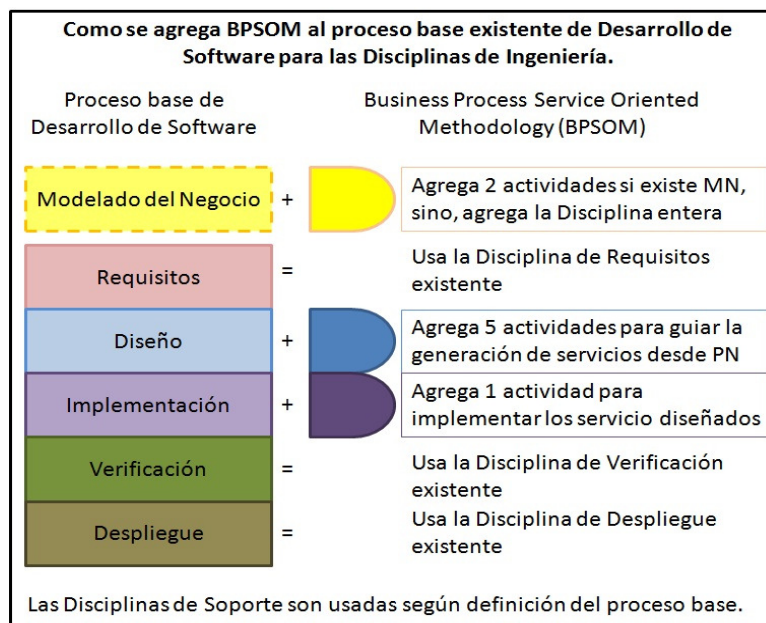


Fig. 5. Definición y uso de BPSOM en el contexto del proceso base de software existente

3.1 Disciplina Modelado del Negocio

La disciplina de Modelado del Negocio tiene como objetivo principal comprender el entorno organizacional e identificar los PN que definen su negocio, constituyendo uno de los pilares de BPSOM. En esta disciplina participan los Analistas del Negocio por el área del Negocio, y los Analistas de Sistemas y Arquitecto por el área de software.

BM1 – Evaluar la Organización objetivo. Tiene como objetivo involucrar al equipo de proyecto con la organización para la cual se realiza el desarrollo, si el área de software es de la misma organización entonces se podrán reutilizar documentos del negocio existentes donde se describa el área del negocio, los objetivos, su operación, empleados, tecnologías, entre otros. Adicionalmente el Modelo de Motivación del Negocio (Business Motivation Model, BMM) [21] de OMG, se utiliza para expresar los objetivos y estrategia del negocio y otra información del negocio relevante, que luego pueda ser relacionada directamente con la definición de servicios en SoaML.

BM2 – Identificar los Procesos de Negocio. Esta actividad es una de las actividades clave para el desarrollo de servicios desde PN, ya que constituye la entrada principal para comprender y describir los PN de la organización, principalmente los relacionados con la aplicación en desarrollo. Se utiliza BPMN como notación para especificar los modelos de PN, Es posible especificar los PN en otras notaciones pero las transformaciones QVT desarrolladas no podrían ser usadas. Para el modelado de PN recomendamos el uso de los patrones de procesos (*Workflow patterns*) [22] que proveen análisis y soluciones a problemas comunes de modelado. Un aspecto importante a especificar son los límites del negocio, indicando quién y qué interactúa con la organización, así como los puntos de interacción con el resto de las partes involucradas. En general toda la información puede ser especificada en los modelos de PN utilizando alguna herramienta de modelado BPMN (como BizAgi). Imágenes de los modelos se pueden incluir también en el documento de PN para su comunicación, donde se describen si es necesario, los PN en lenguaje natural para proveer una guía del modelado e incluir detalles de partes específicas del modelo que se quieran aclarar. Un ejemplo de modelo de PN en BPMN es el mostrado en la Fig.1

3.2 Disciplina de Diseño

La disciplina de Diseño es el otro pilar de la metodología BPSOM. Su objetivo principal es la definición y especificación de servicios desde PN, así como el reuso de los servicios existentes en la organización. En esta disciplina participan como responsable principal el Arquitecto, y los Analistas de Sistemas y Desarrolladores como soporte al diseño y especificación de los servicios.

D1 – Identificar y categorizar servicios. En esta actividad se identifican los servicios necesarios para realizar los PN en desarrollo, teniendo como principales entradas los PN especificados previamente y el Documento de la Arquitectura de Software (SAD) definido según el proceso de desarrollo de software existente. Los servicios que la organización debe proveer y los servicios que la organización

requiere consumir de otras partes involucradas, se identifican en base a los mensajes que se intercambian, cada parte definida por una calle (*Pool*) en el PN. Los servicios internos a la organización se identifican dentro de los carriles (*Lane*) que presenta cada calle, que serán luego modelados dentro de cada organización. En esta actividad se modela con SoaML la Arquitectura de Servicios (ServicesArchitecture, SA) especificando los participantes, contratos de servicios y los roles que cada participante cumple como proveedor o consumidor del servicio. La definición de servicios se realiza en alto nivel mostrando la visión general de servicios necesarios y los participantes que conectan, para todas las organizaciones involucradas. La Fig. 2 muestra el ejemplo del diagrama de SA en SoaML para el PN presentado en la Fig. 1.

Las calles del modelo de PN (definición de *Pool* en BPMN) se corresponden con Participantes en SoaML, luego para cada carril (*Lane*) dentro de la calle se generan participantes internos a éstos, en cada organización al refinar los modelos. Los servicios se identifican con las actividades identificadas para soportar servicios, a las que se les indica su tipo como *Service*, y con los mensajes intercambiados entre las calles. Teniendo en cuenta la dirección de los mensajes definimos que la actividad, en la cual el mensaje es entrante será soportada por el servicio asociado, y la actividad para la cual el mensaje es saliente será la que lo consuma. La calle que contiene cada actividad asociada con los servicios identificados define el participante que lo provee o lo requiere, dependiendo de si la actividad es soportada por un servicio provisto o requiere invocar uno. Esta información se incluye en el contrato del servicio (*ServiceContract*) para indicar los roles de proveedor y consumidor definidos. Para generar el diagrama de Arquitectura desde el modelo de PN, el arquitecto tiene que especificar en el modelo de PN que actividades relacionadas mediante mensajes de entrada y salida son de tipo *Service*, marcándolas con el estereotipo de BPMN, ya que otras actividades podrán ser de tipo manual o no soportadas por servicios, por lo que no serán tenidas en cuenta para la generación de los servicios requeridos. Junto con el diagrama de Arquitectura es se genera también el diagrama de participantes del PN incluyendo sus puertos y los servicios provistos y requeridos (*Service*, *Request*) según la identificación realizada. En la Fig. 6 se muestran los participantes del PN de ejemplo, así como sus puertos con servicios provistos y requeridos.

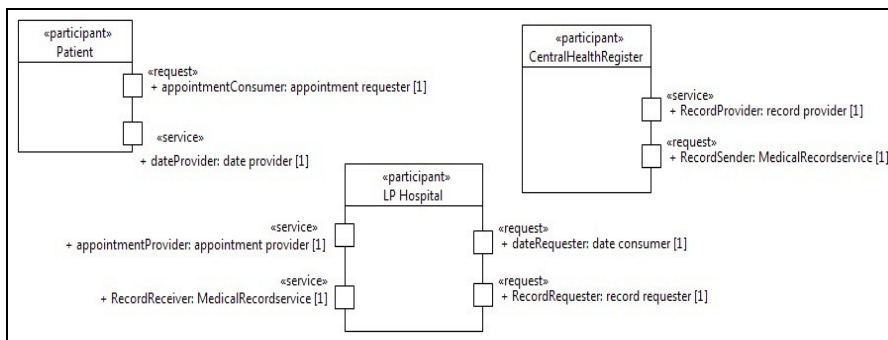


Fig. 6. Participantes con sus puertos y servicios provistos y requeridos

D2 – Especificar servicios. La especificación de servicios refiere a la definición de toda la información necesaria para su posterior implementación. Esto incluye definir toda la información del contrato de servicio (*ServiceContract*): interfaces, operaciones, parámetros de entrada y salida, pre y post condiciones si existieran, y la coreografía asociada al intercambio de mensajes por las partes involucradas. Esta información puede modelarse una vez generados los diagramas anteriores en base a las definiciones del servicio, o puede ser generada parcialmente desde el modelo de PN. Para esto último el arquitecto debe incorporarla al modelo de PN para aquellas actividades que fueron marcadas de tipo *Service*. La información relacionada con los mensajes de entrada y salida tiene que ser especificada (por ej. en los mensajes intercambiados), indicando los parámetros y tipos a ser intercambiados entre las partes. Es posible entonces definir los elementos *MessageType* a ser usados como parámetros en las operaciones, como se muestra en la Fig. 7.

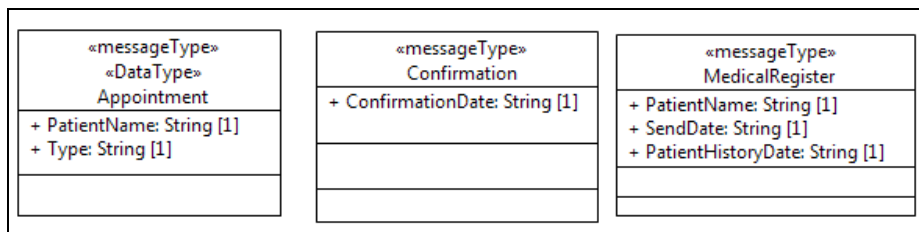


Fig. 7. Definiciones de MessageType a ser usados en las operaciones de los servicios

El contrato del servicio (*ServiceContract*) puede tener asociada también una coreografía para mostrar la interacción entre los participantes, interfaces, operaciones y parámetros utilizados. Teniendo en cuenta esta información para cada servicio, se generan las partes más importantes del contrato de servicio asociado, que son completadas posteriormente por el arquitecto o por los desarrolladores, que también tendrán que completar los detalles de implementación del servicio generado. Los principales elementos correspondientes al contrato de servicio para el servicio “RecibirSolicitudCita” se muestran en la Fig.8: la definición de los roles proveedor y consumidor y la asociación entre estos, así como su coreografía asociada. Se definen también las interfaces, operaciones y parámetros para el servicio, que no se muestran en detalle por razones de espacio.

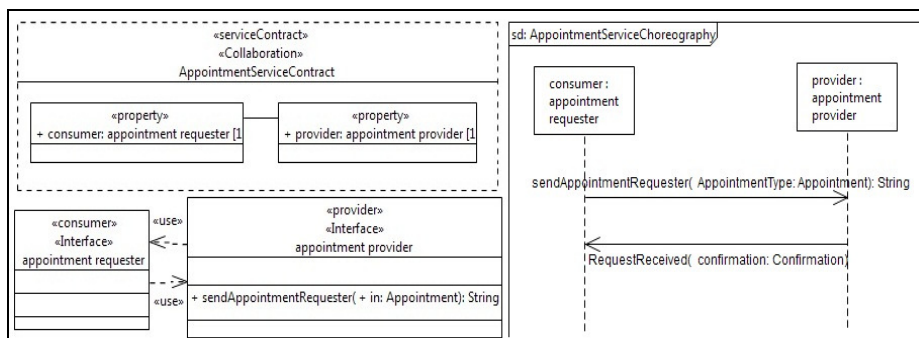


Fig. 8. Especificación del servicio “ReceiveAppointmentRequest”

D3 – Investigar servicios existentes. Esta actividad tiene como principal objetivo reusar lo más posible los servicios existentes en la organización, para lo cual se define un Catálogo de Servicios centralizado donde cada nuevo servicio que se desarrolla es registrado, y en el cual para cada proyecto se buscan servicios que provean la funcionalidad requerida o similar. En ambos casos, en un diagrama de componentes de SoaML se define un componente de tipo *adapter* o *wrapper* para relacionar el servicio generado en el modelo de servicios con su implementación existente.

D4 – Asignar servicios a componentes. Los componentes que implementarán los servicios generados serán definidos y presentados en el diagrama de componentes. SoaML provee el componente participante para definir la implementación de participantes y servicios, definiendo nuevos componentes a generar, y si los componentes existen, la definición de *adapters* o *wrappers* para usarlos, proveyendo las modificaciones necesarias para su invocación.

D5 – Definir la interacción de servicios. La interacción de servicios puede definirse como orquestación o coreografía de servicios de la misma forma que fue definida para los PN. Un diagrama de secuencia mostrando todos los servicios, o varios diagramas de secuencia mostrando distintos subconjuntos de servicios correspondientes a distintos subprocesos en el PN, proveen esta información. Esta actividad no tiene diagrama correspondiente en SoaML y se realiza mediante el diagrama de secuencia UML, en forma manual por el arquitecto.

3.3 Transformaciones QVT

El algoritmo general se compone de varias relaciones QVT que se van invocando a medida que los elementos necesarios van siendo generados. A modo de ejemplo en la Fig. 9 se muestra la generación de la especificación de servicios en notación gráfica de QVT, donde la conexión entre los estereotipos SoaML aplicados a los elementos base UML no se indica ya que está implícita en las referencias de los atributos.

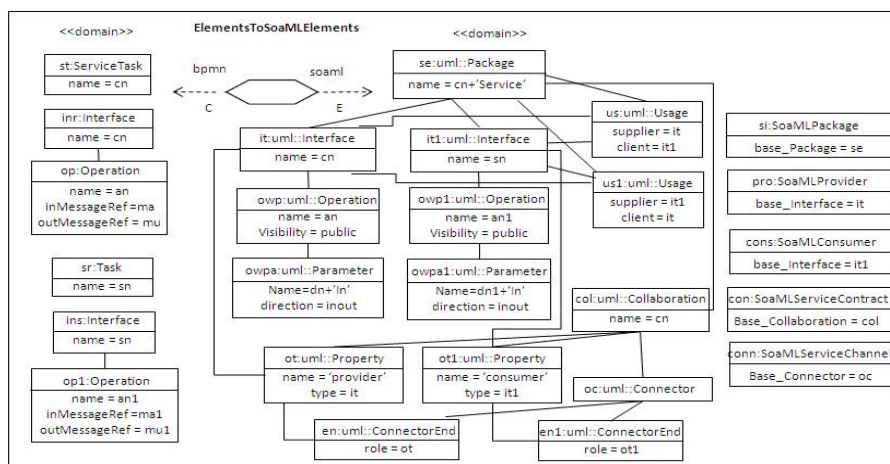


Fig. 9. Una de las relaciones QVT para la generación de servicios en notación gráfica QVT

4 Trabajos relacionados

En los últimos años se han definido varias metodologías para desarrollo de servicios, contemporáneamente con BPSOM, tanto desde el área académica como desde el área industrial, las que se mencionan en [13], y se discuten también en [18]. Entre ellas, se pueden mencionar la propuesta de Papazoglou et. al [23] que define una metodología para diseño y desarrollo con servicios desde PN, para desarrollo específico con WS, con seis fases realizadas iterativamente: planificación, análisis y diseño, construcción y testing, provisionamiento, despliegue, ejecución y monitoreo. El plug-in SOMA [24] de RUP para definir servicios de negocio y software en una metodología consolidada, como en nuestra propuesta, agrega elementos específicos pero al proceso RUP, y la metodología del proyecto SHAPE [25] que también integra SoaML pero con distintas guías de desarrollo que nuestra propuesta y sin generación automática. Kohlborn et al. [26] propone un enfoque consolidado que combina definiciones de los trabajos de desarrollo con servicios revisados, agregando también nuevos ítems. Define dos partes principales para el proceso: la derivación de servicios del negocio y la derivación de servicios de software para soportarlos.

En [27] Herold et. al propone un enfoque dirigido por modelos con cuatro fases: desarrollo del negocio, análisis de requisitos, diseño arquitectónico y modelado de implementación, con guías y transformaciones para moverse de uno modelo al otro. De Castro et al. [28] define un método para composición de servicios con un proceso que establece varios pasos para la generación de modelos, definiendo metamodelos, modelos y artefactos a obtener. En [29] de Castro et. al integran un modelo de valor del negocio para derivar artefactos desde el mismo usando ATL [30]. Este lenguaje para transformaciones también es utilizado en Touzi et. al [31] donde se definen modelos, metamodelos y transformaciones para ir desde un PN colaborativo (CIM) a un modelo SOA (PIM) generando el código en BPEL en base a PIM4SOA. Nuestra propuesta difiere de las anteriores en varios sentidos. En primer lugar la definición de la metodología BPSOM para guiar el desarrollo de servicios desde PN se agrega al proceso base existente en la organización, facilitando su adopción. En segundo lugar las transformaciones se definen y ejecutan con QVT en el mismo entorno de desarrollo, obteniendo los modelos de servicios en SoaML directamente desde modelos de PN en BPMN2, sin generar ni utilizar ningún artefacto intermedio. Desde los modelos SoaML a su vez, es posible generar el código correspondiente integrando motores MDA en el mismo entorno. En tercer lugar, la guía conceptual y automática está integrada y detallada claramente en la metodología BPSOM, haciéndola fácilmente accesible y utilizable en el entorno de desarrollo de software.

5 Conclusiones y trabajo futuro

La metodología BPSOM está definida para guiar desarrollos orientados a servicios desde procesos de negocio, siendo integrada en el marco MINERVA para mejora continua de PN, en el que estamos trabajando desde 2009. BPSOM se ha pensado para ser agregada al proceso de desarrollo existente en la organización, agregando únicamente los elementos clave para desarrollo con servicios. En trabajos previos mencionados se ha presentado su implementación como *Method plug-in* de EPF

Composer y publicada como sitio web para su interoperabilidad con otros procesos definidos en la misma forma. En este artículo hemos presentado la integración del estándar SoaML para modelado de servicios indicando en cada actividad de BPSOM que diagramas de SoaML deben ser realizados, brindando guías para su realización, así como los roles involucrados y sus responsabilidades. La definición de servicios se va completando en forma iterativa e incremental desde la definición de la Arquitectura de Servicios hasta la especificación completa de los mismos, incluyendo interfaces, operaciones, parámetros asociados y mensajes intercambiados. Asimismo, hemos incluido en BPSOM la generación automática de modelos de servicios en SoaML desde modelos de PN en BPMN2 mediante transformaciones QVT desde los metamodelos de BPMN a SoaML, que se ejecutan en el entorno Eclipse definido por MINERVA. Recientemente hemos completado las transformaciones QVT que permiten obtener modelos SoaML completos y que además pueden ser visualizados en el plug-in SoaML² de Eclipse que hemos desarrollado, en el cual están modelados todos los diagramas SoaML presentados en este artículo. A partir de los modelos SoaML es posible generar el código asociado mediante motores MDA existentes, cuya integración en el entorno definido es uno de los trabajos que estamos realizando actualmente. Esto permite cerrar el ciclo de modelado–generación–diseño–implementación propuesto. Asimismo estamos trabajando en un caso real de PNs del Hospital General de Ciudad Real para la validación de la propuesta, desde el modelado de PN en BPMN, la generación de modelos de servicios en SoaML, hasta la ejecución de los PN en motores de procesos invocando los servicios generados.

Agradecimientos. Este trabajo ha sido financiado parcialmente 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, Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01), proyecto INGENIOSO (Junta de Comunidades de Castilla-La Mancha, España, PEII11-0025-9533) y proyecto MOTERO (Junta de Comunidades de Castilla-La Mancha, España, PEII11-0366-9449)

Referencias

1. Weske, M. 2007. BPM Concepts, Languages, Architectures, Springer.
2. Smith, H., Fingar, P. 2003. Business Process Management: The third wave, Meghan-Kieffer.
3. van der Aalst, W.M.P., ter Hofstede, A., Weske, M., Business Process Management: A Survey, In: International Conference on Business Process Management, (2003)
4. Papazoglou, M., Traverso, P., Dustdar, S. and Leymann, F. Service-Oriented Computing: State of the Art and Research Challenge, IEEE Computer Society, (2007)
5. Krafzig, D., Banke, K., Slama, D., Enterprise SOA, SOA: Best Practices, Prentice Hall, (2005)
6. Erl, T., SOA: Concepts, Technology, and Design, Prentice Hall, (2005)
7. Mellor, S., Clark, A., Futagami, T., MDD, Guest editors intro, IEEE CS, (2003)
8. Object Management Group (OMG), Model Driven Architecture (MDA), (2003)
9. Object Management Group (OMG), SOA Modeling Language (SoaML), (2009)

² <http://alarcos.esi.uclm.es/MINERVA/TOOLS/>

10. 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, 5th Int. Work. on Engineering Service-Oriented Applications (WESOA'09), with ICSSOC 2009, Estocolmo, Noviembre (2009).
11. Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M.: A Model-driven and Service-oriented framework for the BP improvement, Journal of Systems Integration (JSI), Vol.1, No.3, ISSN: 1804-2724, Julio (2010).
12. Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M., Towards an ontology for service oriented modeling supporting business processes, 4th. International Conference on Research Challenges in Information Science (RCIS'10), Nice, Mayo (2010).
13. 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, Marzo (2009)
14. Delgado A., Ruiz F., García-Rodríguez de Guzmán I.: Mejora continua de procesos de negocio basada en PmCOMPETISOFT integrando BPM. III Taller sobre Procesos de Negocio e Ingeniería de Servicios (PNIS'10) en XV JISBD'10, Valencia, Setiembre (2010)
15. Delgado, A., García - Rodríguez de Guzmán, I., Ruiz, F., Piattini, M.: Tool support for Service Oriented development from Business Processes, 2nd International Workshop on Model-Driven Service Engineering (MOSE'10), Málaga, Junio (2010)
16. Object Management Group (OMG), Query/Views/Transformations(QVT), (2008)
17. Object Management Group (OMG), Business Process Modeling Notation (BPMN2), (2011)
18. Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M., Application of service-oriented computing and model-driven development paradigms to business processes: a systematic review, 5th Int. Conf. on Software and Data Techs. (ICSOFT'10), Atenas, (2010)
19. Delgado, A., García - Rodríguez de Guzmán, I., Ruiz, F., Piattini, M.: From BPMN business process models to SoaML service models: a transformation-driven approach. 2nd Int.Conf. on Software Tech. and Engineering (ICSTE 2010), San Juan de Puerto Rico, Octubre (2010)
20. Delgado, A., BPSOM Methodology <http://alarcos.esi.uclm.es/MINERVA/BPSOM/> (2010)
21. Object Management Group (OMG), Business Motivation Model (BMM), (2010)
22. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14, 5–51, (2003)
23. Papazoglou, M., van den Heuvel, W., Service-oriented design and development methodology, Int. J. Web Engineering and Technology, Vol. 2, No. 4, pp.412-462, (2006)
24. IBM-SOMA, < http://www.ibm.com/developerworks/rational/downloads/06/rmc_soma/>
25. Stollberg, M., et al, A Customizable Methodology for the MDE of Service-based System Landscapes. 4th Workshop on Modeling, Design, and Analysis for the Service Cloud (MDA4ServiceCloud'10), with ECMFA'10, Paris, June, (2010)
26. Kohlborn T., Korthaus A., Chan T., Rosemann M., Identification and Analysis of Business and SE Services- A Consolidated Approach, IEEE Transactions on Services Comp., (2009)
27. Herold S., Rausch A., Bosl A., Ebell J., Linsmeier C., Peters D., A Seamless Modeling Approach for Service-Oriented Information Systems, In ITNG'08, 5th International Conference on Information Technology:New Generations, (2008)
28. de Castro, V., Marcos, E., López Sanz, M., A model driven method for service composition modelling: a case study, Int. J. Web Engineering and Technology, Vol. 2, No. 4, (2006)
29. 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)
30. Jouault, F., Kurtev, I., Transforming Models with ATL (ATLAS Transformation Language), Satellite Events at the MoDELS Conference, (2005)
31. Touzi J., Benaben F., Pingaud H., Lorré J.P., A model-driven approach for collaborative service-oriented architecture design, Int. Journal of Prod. Economics, Vol.121 Is.1, (2009)