Services Conceptualization within SOA/BPM methodology

Patricia Bazán
LINTI Facultad de Informática UNLP
La Plata, Buenos Aires, Argentina
pbaz@info.unlp.edu.ar

Roxana Giandini
LIFIA Facultad de Informática UNLP
La Plata, Buenos Aires, Argentina
giandini@lifia.info.unlp.edu.ar

Gabriela Perez
LIFIA Facultad de Informática UNLP
La Plata, Buenos Aires, Argentina
gperez@lifia.info.unlp.edu.ar

Elsa Estevez
United Nations University – EGOV Center at IIST
Macao SAR, China
elsa@iist.unu.edu

Javier Diaz
LINTI Facultad de Informática UNLP
La Plata, Buenos Aires, Argentina
jdiaz@info.unlp.edu.ar

***Abstract*— The SOA/BMP methodology proposes a model for aligning business processes with the services providing the functionality required by them. Within such methodology, the conceptualization of services and their mapping with software components enable to reduce the gap between business processes life cycle and their executable versions. Although the methodology proposes a set of steps, the interactions between such steps still is to be formalized. The use of meta-models is an alternative for such formalization, since they enable to unambiguously define the syntax for the languages used in each step and the transformation rules between them, serving as a previous step towards automatic transformations. This work presents a proposal for integrating a meta-model for services – proposed in a previous work and integrated with BPM; and a meta-model for components – defined by the SCA (Service Component Architecture) standard. The contribution of this paper enables to formalize the interactions between two steps of the SOA/BPM methodology.**

Keywords - SOA, BPM, MDD, Meta-model, Software Components, SCA

I. INTRODUCCIÓN

La visión integrada de la orientación a procesos de negocio y la orientación a servicios, conduce a definir un marco metodológico que ordena los conceptos y pauta el ciclo de vida de los procesos de negocios con el objetivo de obtener procesos ejecutables. Al respecto, el trabajo desarrollado en [2] define un modelo de integración entre SOA (*Service Oriented Architecture*) [1] y BPM (*Business Process Management*) [3]. Este modelo propone una metodología general y abarcadora para desarrollar proyectos con enfoque SOA y BPM, que consta de ocho etapas: 1) Etapa de Organización y Plan Estratégico, 2) Etapa de Identificación y Especificación de Requisitos con enfoque de procesos, 3) Etapa de Modelado del Negocio, 4) Etapa de Modelado de Procesos, 5) Etapa de Modelado de Servicios, 6) Etapa de Definición de Componentes, 7) Etapa de Implementación de Componentes y 8) Etapa de Administración y Seguimiento (ver Figura 1). La metodología propuesta delinea una nueva visión global que identifica cada etapa y su interacción con la siguiente, cubriendo tanto el ciclo de vida de los procesos de negocio como el del software de una manera unificada. Sin embargo, algunas de las interacciones entre estas etapas necesitan aún profundizarse y definirse con precisión. El objetivo de nuestros trabajos de investigación es mejorar la formalización de cada etapa y generar productos más robustos que permitan retroalimentar el ciclo.

Una de las interacciones más evidentes y centrales en la metodología propuesta [2] es la de los procesos con los servicios. Por esta razón este trabajo se focaliza en revisar y mejorar la interacción entre: 1) procesos - modelados como paso de refinamiento desde el modelo del negocio y 2) servicios - conceptualizados para dar respuesta a las actividades de los procesos. Los servicios conceptualizados pueden ser considerados como nuevos servicios de negocios construidos para realizar las actividades de los procesos, o bien como funciones de negocios de aplicaciones existentes, que serán reusadas como servicios.

Asimismo, además de la identificación de los puntos de interacción entre etapas más relevantes de la metodología, este trabajo busca lograr la formalización de los productos de cada etapa a través de la aplicación del mecanismo de metamodelado. Definir un metamodelo aporta precisión al lenguaje y permite construir herramientas para la edición de elementos del modelo, facilitando la realización de pruebas reales de la metodología planteada. El metamodelado es un mecanismo para definir formalmente la sintaxis de lenguajes de modelado sin ambigüedades y permite que distintas herramientas puedan interpretar los modelos [4].

En este marco, se ha definido en [6] un metamodelo de conceptualización de servicios a partir de los procesos y sus

978-1-4673-0793-2/12/$31.00 ©2012 IEEE

actividades, sustentado en una adaptación del metamodelo de SOAF (*Service Oriented Architecture Framework*) [5] y combinado con el metamodelo de BPMN [8] [24]. El metamodelo propuesto se instanció con un ejemplo concreto y se desarrolló un prototipo de un editor gráfico que provee la sintaxis concreta del mismo, permitiendo contar con una herramienta para editar servicios a partir de las actividades de los procesos.

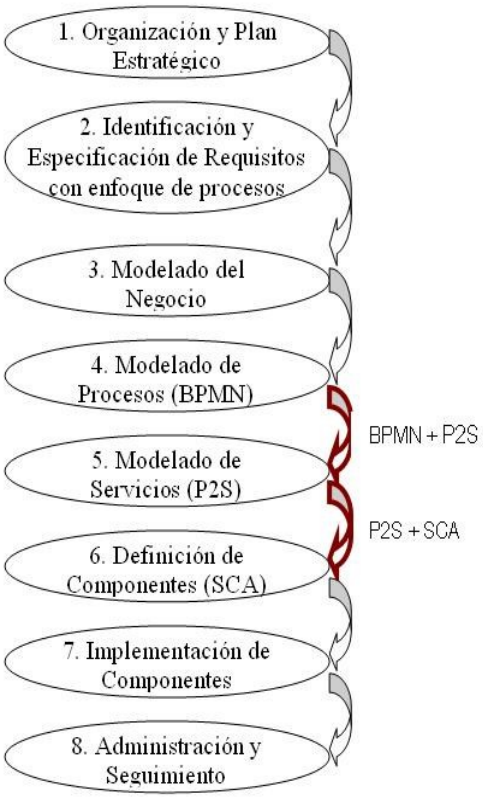


FIGURA 1. Etapas de la metodología integradora SOA/BPM

A partir del metamodelo de conceptualización de servicios planteado en [6], se obtiene una herramienta para modelar servicios. El paso de interacción siguiente (según [2]) es la definición de componentes de software que deben combinarse para absorber el flujo de ejecución que marca el proceso.

La arquitectura SCA (*Service Component Architecture*) define un enfoque general para crear componentes y describir cómo éstos trabajan juntos [1]. En particular, las especificaciones SCA definen cómo crear componentes y cómo combinarlas en aplicaciones completas, más allá de la tecnología de componentes usada.

Este trabajo realiza una integración entre el metamodelo de conceptualización de servicios P2S (*ProcessToService*) planteado en [6] y el metamodelo de SCA, como paso de interacción entre las etapas que modelan servicios – etapa 5 y etapa 6 - de la metodología. Esta integración define también un conjunto de reglas de transformación entre elementos de cada lenguaje que se presenta como una tabla de asociación de conceptos y que representará en futuros trabajos las condiciones para escribir funciones de transformación.

Esta integración de metamodelos contribuye a un objetivo más global que es producir una nueva versión de la metodología SOA/BPM. El beneficio es maximizar la automatización de tareas en base a transformaciones de modelos, logrando una formalización de los pasos de interacción entre cada una de las etapas de la metodología.

La propuesta de integración de metamodelos de este articulo tiene por objetivo formalizar otro de los pasos de interacción entre el modelado de servicios y el modelado de componentes en el marco de metodología integradora SOA/BPM [2], con el propósito de automatizar la producción de un modelo de componentes de software alineado con la producción de modelos de procesos del negocio, a través del modelado de servicios.

El resto del trabajo se organiza de la siguiente manera. En la Sección 2, se clasifican y analizan algunos trabajos relacionados y se discute la influencia de los mismos en nuestra propuesta. En la Sección 3, se introduce sucintamente la metodología integradora SOA/BPM que sustenta este trabajo y se ilustra a través de un caso de estudio. En la Sección 4, se muestra el metamodelo de conceptualización de servicios P2S como paso previo a la etapa de definición de componentes que se resolverá a través del estándar SCA; la definición de componentes se detalla en la Sección 5. En la Sección 6, se propone una forma de integración entre el metamodelo de SCA y el metamodelo de conceptualización de servicios P2S. Por último, se plantean las conclusiones.

II. TRABAJOS RELACIONADOS

La investigación realizada se centró en verificar el estado del arte en materia de propuestas metodológicas en torno a reducir la brecha entre procesos y servicios que se fundamenten en metamodelos, para poder aplicar transformaciones entre modelos.

En este sentido, hemos clasificado el estudio realizado en tres grandes grupos. Por un lado, hemos encontrado diversos trabajos que aplican la técnica del metamodelado, con el objetivo de lograr versiones ejecutables de procesos de negocio a través de un workflow (Sección A).

Por otro lado, encontramos un grupo de trabajos que definen diversas reglas de transformación entre modelos (Sección B) y demuestran la ventaja de utilizar estas reglas de transformación tanto para automatizar las transformaciones como para utilizarlas para verificar los procesos modelados.

Finalmente, en la Sección C, agrupamos los trabajos que plantean el proceso de desarrollo de software con MDA (*Model-Driven Architecture*), tanto en enfoques puramente orientados a servicios como en otros donde el objetivo es disminuir la brecha entre la especificación y el modelado de procesos y servicios. Si bien los trabajos en esta categoría podrían incluirse en la Sección A, se opta por incluirlos en una categoría separada porque su enfoque está más vinculado a la arquitectura que al uso de lenguajes de modelado.

En las secciones siguientes, se explica cómo la metodología propuesta se nutre de múltiples contribuciones de trabajos relacionados aquí descriptos. Sin embargo, la principal diferencia es que la misma aborda una visión integradora de todo el ciclo de vida de los procesos y su gestión.

A. *Metamodelos para ejecutar procesos*

En [11] se presenta un metamodelo MOF (Meta-Object Facility) para crear y ejecutar modelos de workflow, a través de la herramienta USE (UML-*based Specification Environment*) [14]. Este metamodelo permite validar propiedades estáticas de los modelos de workflow durante el modelado del proceso, observando invariantes OCL. Originado en [13], este trabajo plantea usar diagramas de clase de UML para definir un metamodelo de procesos, y diagramas de actividad (DA) para metamodelar el ciclo de vida de las actividades de los procesos. En [11], los autores permiten validar la ventaja de utilizar un enfoque dirigido por modelos como paso de interacción entre las etapas de nuestra metodología.

Por su parte en [12], se presenta el enfoque MDD en el área de definición de workflow y se muestra que el uso de un perfil UML permite ofrecer a los DA de UML como notación viable para la definición de workflows integrando a su vez datos y recursos. Una vez más, este trabajo permite confirmar que los metamodelos - propuestos como productos de cada etapa de nuestra metodología para así transformarla en un enfoque dirigido por modelos, son un componente esencial en el estado del arte actual.

Estos trabajos resultaron interesantes a la hora de validar la ventaja de utilizar un enfoque dirigido por modelos, sin embargo no pudieron aplicarse a nuestra propuesta, en forma completa dado que no consideran la visión integradora de la misma, siendo sí un gran aporte conceptual.

B. *Metamodelos y reglas de transformación*

En [15], los autores definen una transformación desde los DA de UML hacia YAWL (*Yet Another Workflow Language*) - un lenguaje de workflow formal que captura cerca de 20 patrones de workflow. Si bien en este trabajo se presenta a YAWL como contrapunto de BPEL, la diferencia encontrada a la luz de nuestro trabajo da cuenta de la validez de pensar en la transformación de modelos como mecanismo útil a la hora de tender un puente entre lenguajes de diferentes niveles de abstracción y formalidad.

En [16], se presenta un prototipo de una herramienta de modelado que aplica reglas basadas en grafos para identificar problemas en modelos de procesos de negocios. El lenguaje gráfico utilizado para modelar los procesos es EPC (*Event-Driven Process Chains*) y sobre el mismo se elabora una técnica que puede aplicarse incluso a modelos de procesos de negocio incompletos. Si bien la propuesta utiliza un lenguaje particular de modelado, los principios subyacentes pueden ser aplicados también a BPMN. Este trabajo aporta a nuestra propuesta la visión de contar con una validación de modelos en etapas tempranas del desarrollo del proyecto.

En [17] se especifica e implementa un conjunto de reglas de mapeo conceptuales entre los metamodelos de BPMN [24] y SCA [1]. Este trabajo constituye un valioso aporte a nuestra propuesta, dado que no solamente utiliza metamodelos y define reglas de transformación, sino que también utiliza los lenguajes sobre los que nos basamos. La diferencia sustancial radica en que nuestra metodología integradora propone una etapa intermedia entre el modelado de procesos en BPMN y el modelado de componentes en SCA, definiendo así una conceptualización de servicios.

C. *MDA como método de desarrollo de software*

En [18] y [22] se presenta el método de desarrollo de software orientado a servicios SOD-M, que constituye un método dirigido por modelos para el desarrollo de servicios; en particular, Web Services. Los procesos de negocios representan el CIM (*Computacional-Independent Model*); la visión del sistema de información se representa primero por el PIM (*Plataform-Independent Models*) y luego se especifica en el PSM (*Platform-Specific Models*) con SoaML [21]. Los autores comparan a su vez su propia propuesta con otras orientadas a la ingeniería de software como SoaML [21] y SOMA [20]. Esto convalida nuestra idea de que el trabajo constituye un aporte desde el enfoque de desarrollo de software mientras que nuestra propuesta presenta una visión integradora de todo el ciclo de vida de un proyecto gestionado por procesos.

El trabajo presentado en [19], desarrolla técnicas y herramientas para dar soporte al ciclo de vida de los servicios. Se presenta un modelo para alinear modelos de procesos con especificaciones de workflow. La principal ventaja de este enfoque es la creación de un modelo de mapeo entre la realidad del negocio y la de Tecnologías de la Información (*Business-IT-Mapping Model -BIMM*), para evitar la definición de reglas de mapeo complejas. Este trabajo convalida nuestra visión.

Finalmente en [23], se presenta un enfoque dirigido por modelos para tender un puente entre los requerimientos del negocio y los sistemas multi-agentes, considerando el uso de agentes en contraste a enfoques basados en motores BPEL. El trabajo se basa en los servicios de modelado a nivel PIM con SoaML y su conexión a los modelos de agentes a nivel PSM. La investigación se fundamenta en el proyecto SHAPE (*Semantically-enabled Heterogeneous Service Architecture and Platforms Engineering*). Este trabajo representa un aporte al enfoque dirigido por modelos e influye en nuestra propuesta ya que reduce la brecha entre los procesos y los servicios, pero considera a estos últimos desde el modelo de agentes.

III. METODOLOGÍA INTEGRADORA SOA/BPM APLICADA A UN CASO DE ESTUDIO

Recientemente, la propuesta BPM ha adquirido una atención considerable tanto en la academia como en la industria del software. BPM es una estrategia para gestionar y mejorar el rendimiento de un negocio optimizando sus procesos a través del modelado, ejecución y medida de rendimiento dentro de un ciclo de mejora continua.

SOA es un enfoque diferente para el diseño y construcción de sistemas flexibles y adaptables que asisten al desarrollo de sistemas en un entorno empresarial dinámico. Si bien no es un

concepto nuevo, está siendo ampliamente estudiado en la academia y utilizado en la práctica.

En un entorno SOA, los servicios pueden compartirse y reutilizarse en varios procesos de negocio. El resultado es un entorno altamente adaptable, con mayor retorno de inversión en el desarrollo de aplicaciones, mejoras en la integración y mayor rapidez en el despliegue de sistemas.

En [2] se propone una metodología que provee un modelo de integración de aplicaciones dentro de una organización, de modo de alinear los procesos que definen su funcionamiento con los servicios que dan soporte a la funcionalidad. Esta metodología, presentada en la Figura 1, pone un fuerte acento en la identificación de requisitos con enfoque de procesos. La misma, cuenta con una etapa de modelado del negocio como paso previo al modelado de procesos y servicios. A partir de los servicios modelados, se suceden dos etapas de diseño e implementación de componentes, respectivamente. Finalmente, existen dos etapas globales: una de organización y plan estratégico al principio, y otra de administración y seguimiento posterior, que envuelven y atraviesan las fases internas.

Según se plantea en [2], la etapa de modelado del negocio produce un mapa de procesos que junto a los requisitos identificados y especificados y los casos de uso del negocio, sientan las bases para el modelado de los procesos de negocio.

A partir del modelo del negocio, el modelado de los procesos de negocios constituye un paso de refinamiento, donde los procesos se enriquecen con aspectos funcionales mediante la definición de casos de uso del sistema y la descripción de los objetos de información. Este paso de refinamiento permite descomponer el proceso en tareas y definir roles y flujos que aportan documentación textual a la descripción de procesos de negocios (BPD).

Una vez obtenido el modelo de procesos se encuentra el terreno propicio para identificar los elementos (servicios) que realizarán los procesos de negocios, transformándose estos últimos en consumidores de dichos servicios.

En las próximas secciones se introduce un caso de estudio para mostrar la aplicación de algunas de las etapas.

A. Caso de estudio

En esta sección presentamos un caso de estudio que corresponde modelado del proceso que se desarrolla dentro de un centro de reparaciones de una empresa de transportes, entre los conductores que solicitan la reparación de sus vehículos y el taller que los lleva a cabo. El proceso genera una orden de trabajo (OT) cuando el conductor llega con el transporte a reparar. Por cada actividad de reparación, si puede planificarse, se programa la asignación del box y de un mecánico por cada ítem de la OT. Se realiza la reparación calculando costo real y material utilizado y luego se cierra la OT.

En [6] se presenta la aplicación de P2S a un caso de estudio donde se utiliza el metamodelo de conceptualización, mapeando las actividades de un proceso como servicios del modelo de servicios. Se parte de un modelo en BPMN donde se identifican las actividades del proceso. A partir de las mismas se obtiene un conjunto de servicios que se documentan con los círculos con cordones, según lo determina la metodología propuesta. La manera de relacionar servicios a partir de actividades, se encuentra sustentada por nuestro metamodelo, utilizando la referencia "realiza" entre actividades y servicios.

En este trabajo, expandimos este caso de estudio formalizando el paso de interacción del metamodelo de conceptualización de servicios al modelado de componentes, utilizando la integración con el metamodelo SCA.

B. Modelado y clasificación de procesos

El corazón del modelado de procesos de negocio lo constituyen los procesos mismos. Aplicando distintos niveles de abstracción, existen tres conceptos relacionados: 1) Procesos, 2) Subprocesos y 3) Tareas. Un Proceso es una red de "cosas para hacer". Un Subproceso es un proceso en sí mismo, cuya funcionalidad es parte de un proceso más grande. El proceso de menor nivel y que no puede ser descompuesto, se considera una Tarea.

En la etapa de modelado de procesos de negocio (ver Figura 1, etapa 4), se utiliza BPMN, y los elementos notacionales que se identifican en su metamodelo, según [6].

En la Figura 2 se observa el modelo en BPMN del proceso enunciado como caso de estudio, construido con el editor BPMN de Eclipse.

C. Modelado y clasificación de servicios

Los servicios pueden verse, dentro del paradigma de orientación a objetos, como objetos que aportan más información y menos acoplamiento dado que generalmente no se asocian, sino que su comportamiento es desencadenado por la meta-información que poseen, en cuanto a contratos de servicio y contexto de ejecución. Debido a su similitud con los objetos, admiten que sean modelados en términos de interfaces.

Inspirado en las clases de UML, Erl [7] define el símbolo de servicios como un círculo que incluye dos áreas: una para especificar el nombre del servicio y la otra para enumerar las capacidades del mismo.

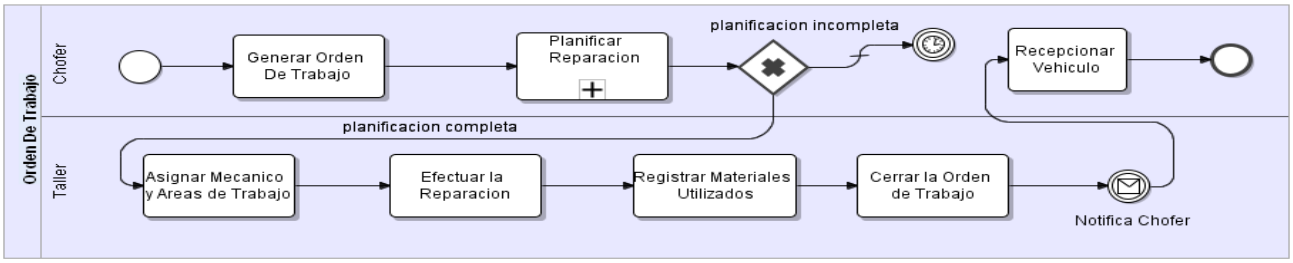


FIGURA 2. Caso de estudio - Proceso de una orden de trabajo

Un servicio simple provee una colección de capacidades que se agrupan según el contexto funcional establecido por el servicio. La capacidad de un servicio representa la función específica del mismo a través de la cual es invocado. Se expresan dentro del contrato de servicio. Los contratos de servicio indican las interfaces que proveen los servicios - sus operaciones y sus parámetros; favoreciendo la catalogación de los servicios y su consecuente composición para ampliar las funcionalidades.

La granularidad de un servicio comunica el nivel de detalle del mismo asociado a su alcance funcional. No existe un único criterio que indique el nivel de granularidad adecuado para un servicio, pero los principios de diseño y las clasificaciones de servicios que se adopten impactan en forma directa en el nivel de granularidad.

La clasificación propuesta en nuestro metamodelo tiene en cuenta servicios de: negocio, entidad y de infraestructura:

- S*ervicios de negocio* - tienen como alcance funcional las entidades del negocio, son de alta reusabilidad y son agnósticos a muchos procesos de negocios.
- *Servicios de entidad* - tienen límites funcionales asociados a las clases dentro de un modelo de clases. Tienen menos reuso potencial y se componen con otros servicios.
- *Servicios de infraestructura* - son servicios que resuelven servicios de tecnología o aspectos transversales. A modo de ejemplo, un servicio que establece la conexión con una base de datos relacional en forma genérica o un servicio centralizado que informa condiciones de excepción, ambos son ejemplos de servicios transversales.

Los servicios son agrupados en función de un comportamiento común, en vez de encapsular dicho comportamiento junto con los datos, como sucede en la orientación a objetos. La Figura 3 muestra dos servicios relacionados con el caso de estudio.

IV. METAMODELO DE CONCEPTUALIZACIÓN DE SERVICIOS

El metamodelo de conceptualización de servicios SOAF puede adaptarse y enfocar el aspecto de los procesos específicamente tomados como un conjunto de actividades, siendo cada una de ellas la realización de un servicio visto como una abstracción de funcionalidad.

Así definimos un metamodelo para conceptualizar estos servicios basado en el metamodelo de conceptualización de SOAF [5] y en el metamodelo de BPMN [8] que fue inspirado en el que propone Eclipse [16] para modelar procesos de negocio. El metamodelo propuesto, llamado P2S, conceptualiza a los servicios como una generalización de los componentes internos y los servicios externos. Está construido como instancia del lenguaje estándar MOF (*Meta Object Facility*) [10], que constituye la capa más abstracta en la Arquitectura 4 capas de modelado de la OMG [9] [4].

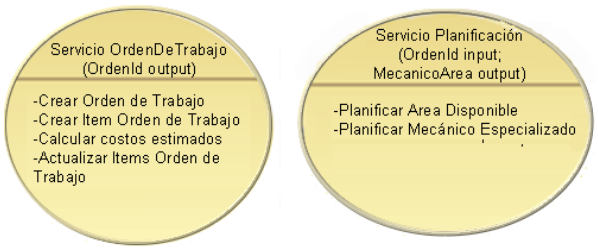


FIGURA 3. Caso de estudio – Servicios del proceso de una orden de trabajo

El proceso de obtener servicios a partir de las actividades de un proceso de negocio es uno de los puntos clave en la interacción entre procesos y servicios. El metamodelo P2S determina que la metaclase Servicio "realiza" una Actividad. Este aspecto se pone en práctica aplicando una combinación top-down y bottom-up. Es decir, una vez aplicado el análisis top-down para identificar las actividades, se aplica un análisis bottom-up para obtener los componentes funcionales que dan respuesta a las actividades del negocio. La premisa es que los componentes contengan la información a ser gestionada: 1) las operaciones para manipular la información, y 2) las reglas de negocio que gobiernan dicha manipulación.

Luego de este análisis bottom-up, se determinan cuáles actividades son realizadas por cada servicio identificado, el cual, a su vez, estará clasificado según la generalización de metaclases del metamodelo.

Este metamodelo fue instanciado con el caso de estudio y se construyó un editor gráfico [6] utilizando plugins de Eclipse que está incluido en el proyecto Eclipse Modeling [25, 26, 27].

En la próxima sección presentamos la etapa de definición de componentes de la metodología SOA/BPM [2], cuyo modelado se sustenta en el metamodelo de SCA y proponemos una integración entre dicho metamodelo y nuestro metamodelo de conceptualización de servicios P2S.

V. ETAPA DE DEFINICIÓN DE COMPONENTES EN LA METODOLOGÍA

En el marco de la metodología SOA/BPM [2], a fin de mantener su independencia, los servicios deben encapsular la lógica dentro de un contexto que puede ser una tarea, una entidad de negocio o algún otro agrupamiento. Para que los servicios puedan orquestarse y desplegar su lógica de funcionamiento, deben intervenir en la ejecución de las actividades del negocio, para lo cual deben poder establecerse relaciones con aquellos que quieren usarlos.

Concluida la etapa de identificación y modelado de servicios, continúa la etapa de definición de componentes. En esta etapa, se empaquetan los servicios como componentes de granularidad gruesa, para dar respuesta a los requisitos funcionales y no funcionales identificados en etapas iniciales y que son utilizados por procesos de negocios que los requieren.

A. SCA y su metamodelo

SCA (*Service Component Architecture*) define un enfoque general para crear componentes y describir cómo éstos trabajan juntos [1]. Siendo actualmente un estándar de OASIS [28], fue creado originalmente por un grupo de representantes de la industria, incluyendo, entre otros, BEA, IBM, Oracle y SAP.

Las especificaciones SCA definen cómo crear componentes y cómo combinarlas en aplicaciones completas. Los componentes pueden ser construidos con JAVA y otros lenguajes usando los modelos de programación basados en SCA, o pueden construirse con otras tecnologías, como BPEL (*Business Process Execution Language*) o el framework Spring. Más allá de la tecnología de componentes usadas, SCA define un mecanismo común para especificar cómo se combinan componentes en aplicaciones de software.

En la Figura 4 se presenta un metamodelo simplificado del estándar SCA definido en [1].

Allí se observa que: un *Composite* contiene *Reference*, *Component*, *Wire, Service y Property*. A su vez un *Component* contiene *ComponentReference* y *ComponentService*. *Reference* y *ComponentReference* son subclases de *BaseReference* y *ComponentService* y *Service* son subclases de *BaseService*, que contiene *Operation*. *Reference* y *Wire* se relacionan con *ComponentReference*. A su vez, *Service* y *Wire* se relacionan con *ComponentService*. *Component* contiene *PropertyValue*, mientras que *ComponentType* contiene *Property*.

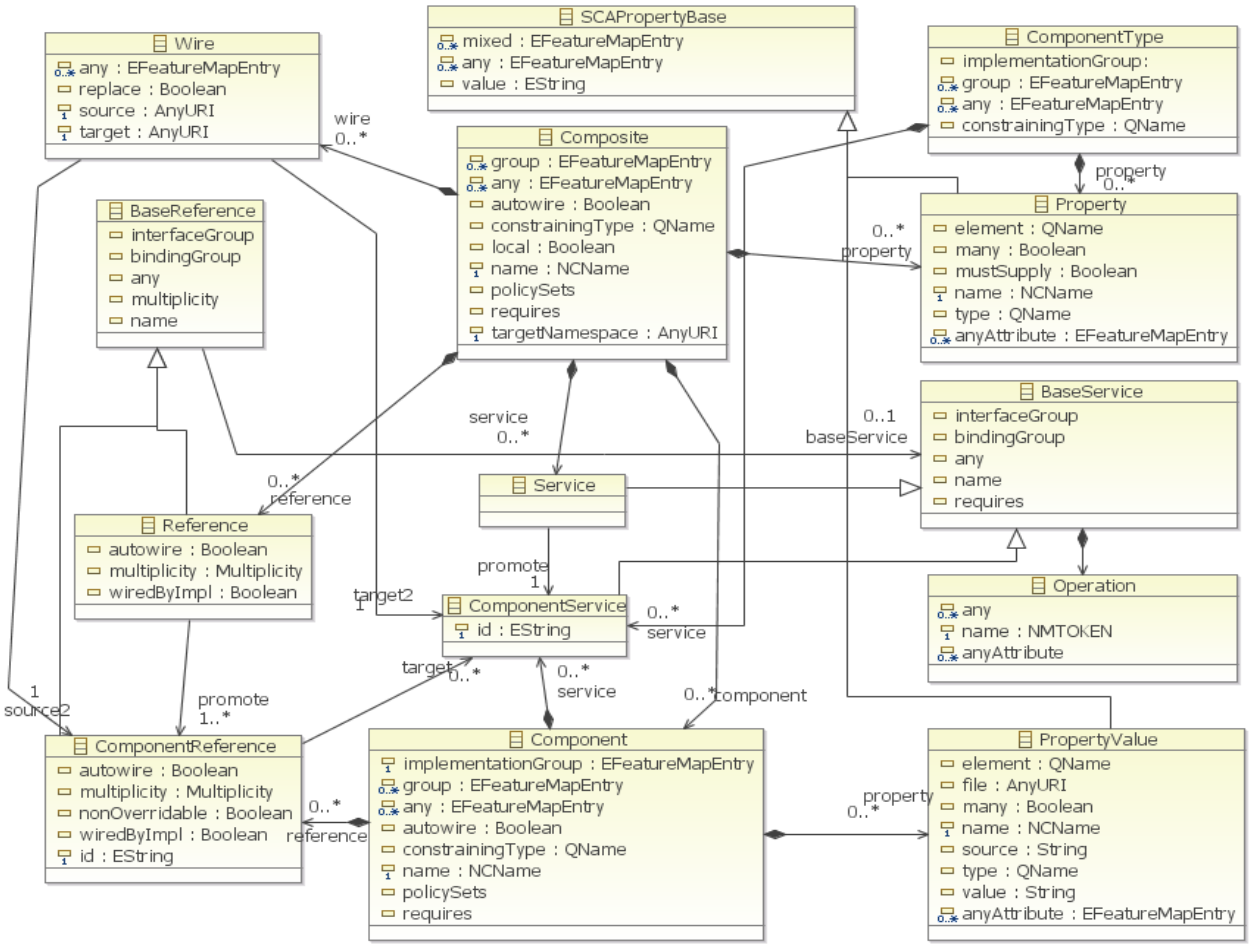


FIGURA 4. Metamodelo simplificado de SCA

B. Descripción y representación gráfica de los elementos.

Dentro del estándar SCA, los elementos metamodelados poseen una representación gráfica y una descripción que se detalla a continuación.

1) Compuestos

Un compuesto es una estructura lógica, donde sus componentes pueden ejecutarse dentro de un único proceso y en un único computador, o en procesos y computadores diferentes.

Una aplicación puede definirse como un único compuesto o combinando varios compuestos. Además los componentes que se usan en cada compuesto pueden compartir tecnologías o ser de tecnologías diferentes.

Un compuesto SCA se describe con un archivo XML de configuración, con extensión *.composite* y que se denomina

Service Component Definition Language (SCDL, pronunciado "skiddle").

2) Componentes

Los componentes son las unidades atómicas desde las cuales se construye una aplicación SCA.

Un componente es una instancia de una implementación que ha sido configurada adecuadamente. La implementación es el código que realmente provee de funcionalidad al componente - como un clase Java o un proceso BPEL. La configuración, expresada en SCDL, define cómo el componente interactúa con el mundo externo.

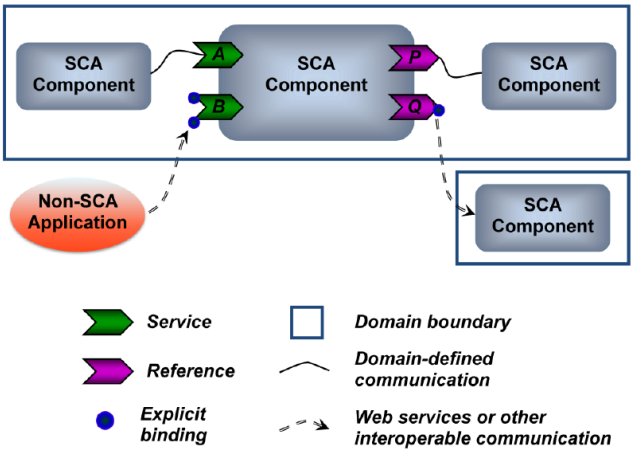
Cada componente se sustenta sobre un conjunto de abstracciones que incluyen servicios, referencias, propiedades y enlaces para especificar su interacción con el exterior (como se observa en la Figura 5). A su vez, cada componente implementa una lógica de negocio expuesta como servicio (representado por el símbolo verde). Cada servicio provee una cantidad de operaciones que pueden ser accedidas por el cliente del componente. La manera de describir el servicio depende de la tecnología para implementar el componente. Por ejemplo, si el componente es una clase Java, los servicios se describen usando interfaces, mientras que si se trata de un componente implementado en BPEL, se describe usando WSDL.

Además de proveer servicios a sus clientes, los componentes pueden consumir servicios de otros componentes. Esto se describe indicando los servicios mediante referencias (símbolos violetas). Cada referencia define una interface que contiene operaciones que el componente necesita invocar.

3) Enlaces o bindings

Los servicios y referencias permiten que el componente se comunique con otro. Sin embargo, no se especifica cómo se produce ese enlace.

En la Figura 6, se observan los componentes SCA identificados en nuestro caso de estudio. Si bien varios de los componentes se vinculan directamente a componentes SCA, cabe destacar que los enlaces SCA permiten representar también el flujo de trabajo subyacente. Esta figura fue construida con una herramienta para SCA para Eclipse [29].



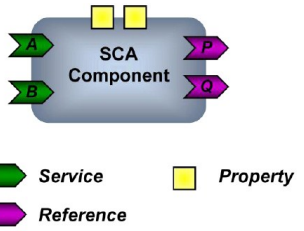


FIGURA 5. Enlaces y bindings SCA [1]

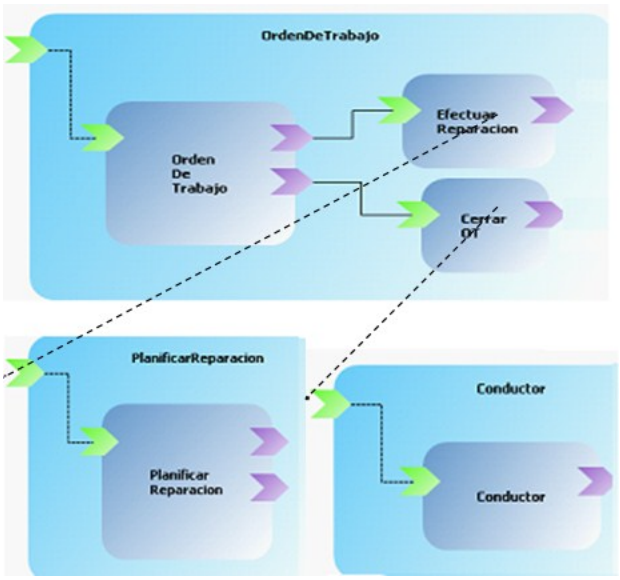


FIGURA 6. Caso de estudio – Modelo de Componentes SCA

En la próxima sección proponemos una integración entre el metamodelo de conceptualización de servicios y el metamodelo de SCA que nos permite obtener un nuevo paso de interacción entre dos etapas de la metodología propuesta [2].

VI. INTEGRACIÓN DE METAMODELOS DE SERVICIOS (P2S) Y COMPONENTES (SCA)

El metamodelo de SCA descripto en la sección anterior, define un lenguaje para especificar un modelo de componentes que cubre parcialmente los aspectos relevantes de cada elemento gráfico. En este sentido, debería interpretarse que la interface que define *Reference* en el gráfico corresponde a *BaseService* en el metamodelo. Además, en el gráfico sólo se distingue por color un *Service* de un *Reference*, mientras que en el metamodelo son metaclases diferentes con atributos distintos.

Dado que se trata de un metamodelo simplificado, es posible que algunos elementos no estén completos, pero esta simplificación resulta suficiente para analizar la integración con el metamodelo de conceptualización de servicios (P2S).

Uno de los aspectos más evidentes al analizar una integración entre P2S y SCA proviene del hecho que *Component* en P2S es una especialización de *Service*, mientras

que en SCA, tanto *Component* como *Service* son parte de *Composite*. Es decir, *Service* no comparte el mismo nivel de abstracción en los dos metamodelos.

Más allá de este punto, se observa que el estándar SCA permite, a través de su metamodelo, definir un lenguaje de especificación para la etapa 6 de la metodología (Definición de componentes). La interacción entre esta etapa y la anterior (etapa 5 – Modelado de Servicios) se podrá resolver instanciando ambos metamodelos, y poniendo cuidado para resolver la ambigüedad del concepto *Service*.

La solución propuesta para resolver la ambigüedad de este elemento es definir que *Service* en P2S representa un modelo de servicio y no el servicio en sí mismo. Se trata de un servicio conceptual que puede ser un componente o un servicio externo (tal como se describe en P2S). En este sentido, la metaclase *Service* de P2S pasa a nombrarse como *ConceptualService* y así se puede vincular ambos metamodelos a través de la metaclase *Component*.

A su vez, la metaclase *Component* que se utiliza como medio de vinculación entre los dos metamodelos, puede también renombrarse como *ConceptualComponent* en P2S y así se puede relacionar con *Component* en SCA. De la misma manera, se define *ConceptualService* en P2S para relacionarlo con *ComponentService* en SCA. La figura 7 presenta una versión simplificada de esta propuesta de integración de metamodelos.

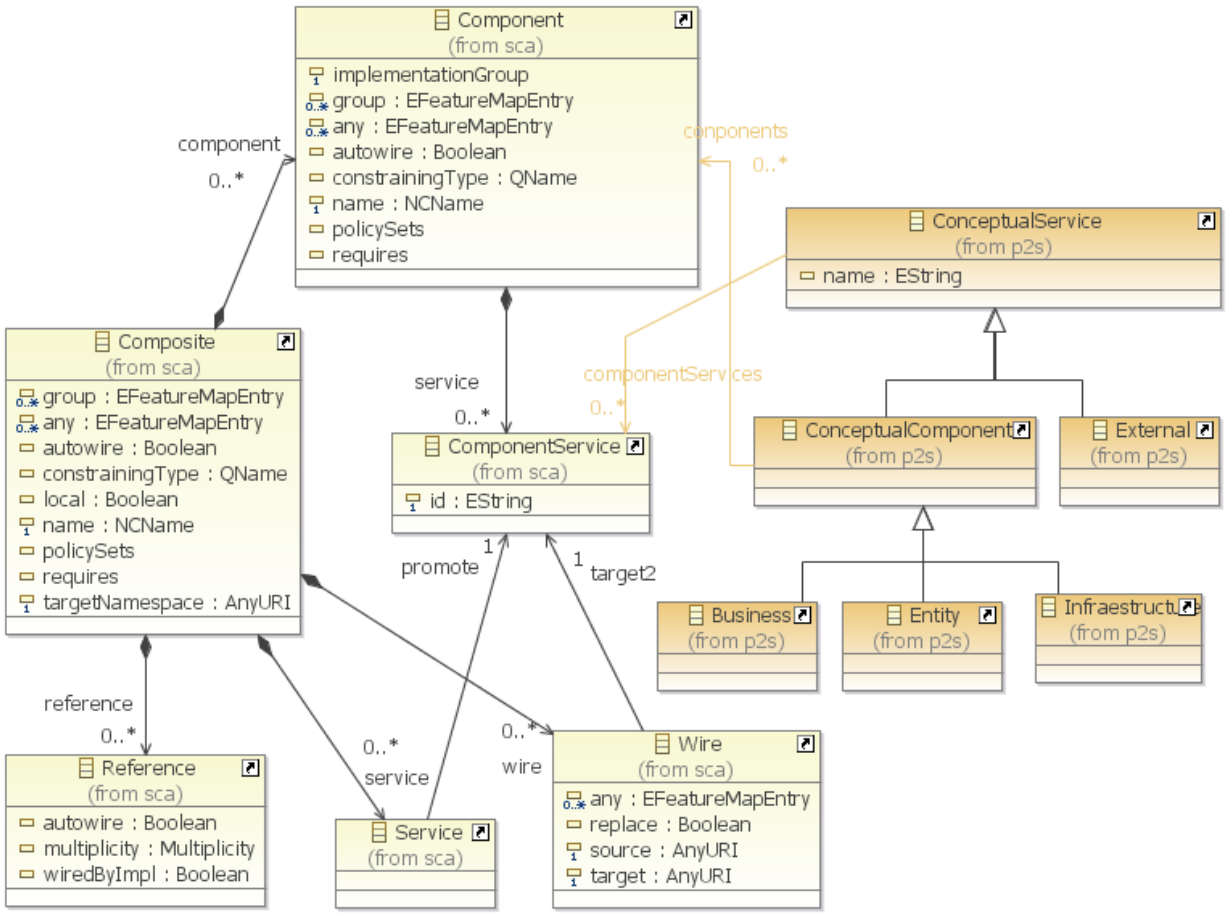


FIGURA 7. Vinculación del metamodelo P2S con SCA

A la luz de lo expuesto, la Tabla 1 presenta un conjunto de reglas de vinculación entre las metaclases de cada metamodelo. Estos mapeos constituyen un primer paso en la definición de transformaciones entre metamodelos. Si bien el desarrollo de este punto será objeto de investigaciones futuras, presentamos a continuación una breve explicación de cada mapeo.

Como mencionamos en la Sección V, un modelo SCA consiste de una serie de compuestos que definen la configuración de un dominio. Este dominio representa las capacidades o funcionalidades del negocio que son controladas por una organización y son el insumo para ejecutar las actividades de un proceso de negocio.

En este sentido, un *Compuesto* SCA equivale a un *Proceso* en el metamodelo P2S y es a su vez el *ProcessModel* de BPMN. Este mapeo conceptual da cuenta que en realidad cada proceso de negocio tiene su representación de implementación como un *Compuesto* SCA.

El mapeo de *Pool* a *Component* se sustenta en la idea de que *Component* es un metaclase contenedora de *ComponentService* y por lo tanto, puede desagregarse en más

partes. Además, *Pool* puede interpretarse como un participante de un proceso de negocio que produce una función de negocio esperada; siendo ésta última un *Component*.

Las actividades de los procesos (*Activity*) se mapean a *ComponentService* que son la implementación de un servicio como instancia para proveer o consumir funciones básicas de negocios.

Por último, *Connection* y *Artifact*, se mapean a *Wire* y *Reference* como representación conceptual de los flujos que permiten orquestar el proceso.

TABLA I. REGLAS DE INTEGRACION ENTRE METAMODELOS

METACLASES BPM	*METACLASES P2S*	*METACLASES SCA*
Process Model	Process	Composite
Pool	ProcessService	Component
Connection	Contrato	Wire
Artifact	TypedElement	Reference
Activity	ConcpetualService	Component Service

VII. CONCLUSIONES

El presente trabajo constituye un paso más en la mejora de la metodología integradora SOA/BPM planteada en [2] y enriquecida en [6], mediante la conceptualización de servicios modelados en una de las etapas de dicha metodología.

Esta visión contribuye a un objetivo más global que es producir una nueva versión -dirigida por modelos- de la metodología SOA/BPM, maximizando la automatización de tareas en base a la transformación de modelos. El objetivo es lograr una formalización de los pasos de transformación e interacción entre cada una de las etapas de la metodología.

Asimismo, se espera poder contribuir con un nuevo enfoque metodológico que logre una combinación balanceada del uso alternado de técnicas de desarrollo top-down para el desarrollo de proceso de negocios y bottom-up para el desarrollo de servicios.

En este trabajo presentamos una integración entre el metamodelo de conceptualización de servicios presentado e implementado en [6], con el metamodelo de SCA a los efectos de contar con una sintaxis de lenguaje de modelado para la siguiente etapa de la metodología que consiste en obtener un modelo de componentes a partir de un modelo de servicios. Esta integración define un conjunto de reglas que serán utilizadas en trabajos futuros para obtener en forma automática un modelo de componentes, a partir de servicios modelados.

AGRADECIMIENTOS – Los autores agradecen los valiosos comentarios de los revisores anónimos que contribuyeron a mejorar la calidad de este trabajo.

REFERENCES

[1] SCA Assembly Model Specification 1.1, Open SOA Collaboration, (2009).

[2] Bazán P. “Un modelo de integrabilidad con SOA y BPM”. Tesis de Maestría en Redes de Datos. Facultad de Informática. *Universidad Nacional de La Plata*. (2010).

[3] Weske M., “Business Process Management: Concepts, Languages, Architectures”, *Springer,* pp 3-67, ISBN 978-3-540-73521-2, (2008).

[4] Pons, C., Giandini, R., Pérez, G., “Desarrollo de Software Dirigido por Modelos. Conceptos teóricos y su aplicación práctica”. *EDULP & McGraw-Hill Educación*. (2010). ISBN: 978-950-34-0630-4. Capitulo 4.

[5] Erradi, A., Anand, S., Kulkarni, N. N.: SOAF: An Architectural Framework for Service Definition and Realization, IEEE International Conference on Services Computing, *IEEE*, (2006), pp. 151-158.

[6] Bazan P., Perez G, Giandini R., Diaz J. “Process - service interaction using an SOA/BPM methodology”. Proceedings de XXX Conferencia Internacional de la Sociedad Chilena de Ciencia de la Computación (SCCC'2011). Curico, Chile, Noviembre 2011.

[7] Erl, T., “SOA Principles of Service Design”, *Prentice Hall*, ISBN-13: 9780132344821. 2007, pp.25-119.

[8] Giandini, R., Pérez, G., Pons, C., “Un lenguaje de Transformación específico para Modelos de Proceso del Negocio”. Proceedings de. XXXVI Conferencia Latinoamericana de Informática (CLEI 2010), octubre de 2010, Asunción, Paraguay.

[9] Object Management Group (OMG), http://www.omg.org

[10] OMG/MOF Meta Object Facility (MOF) 2.0. OMG Adopted Specification, october 2003, http://www.omg.org, http://www.eclipse.org/stp/sca/.

[11] Bruning, J.; Gogolla, M.; , "UML Metamodel-based Workflow Modeling and Execution," Enterprise Distributed Object Computing Conference (EDOC), *IEEE,* vol., no., pp.97-106, (2011) URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6037564&isnumber=60375532006.

[12] Kalnins, A., Vitolins, V., “Use of UML and model transformations for workflow process definitions” Databases and Information Systems, Baltic DB&IS'2006, edited by Olegas Vasilecas, Johann Eder, Albertas Caplinskas, Vilnius, Technika, (2006), pp. 3-15.

[13] Brüning, J., Gogolla, M., Forbrig, P., "Modeling and formally checking workflow properties using UML and OCL". Perspectives in Business Informatics Research, *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, (2010), ISBN 978-3-642-16101-8, vol 64, pp. 130 a 145, URL: http://dx.doi.org/10.1007/978-3-642-16101-8_112009.

[14] Gogolla, M., Büttner, F., Richters, M., “USE: A UML-Based Specification Environment for Validating”, *Science of Computer Programming,* vol 69, (2007), pp 27–34.

[15] Zhaogang, H. , Li, Z., “From UML 2.0 Activity Diagram to YAWL : The Controlflow Aspect”, ICEES 2011, 14–16 October 2011, Singapore.

[16] Kühne, S., Kern, H., Gruhn, V., Laue, R., “Business process modeling with continuous validation”, *Journal of Software Maintenance and Evolution: Research and Practice*”, (2010), vol 22, issue 6-7, pp 547-566, DOI: 10.1002/smr.517.

[17] Dahman, K., Charoy, F., Godart, C., "Generation of Component Based Architecture from Business Processes: Model Driven Engineering for SOA", *IEEE 8th European Conference on* Web Services (ECOWS), (2010), pp.155-162, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5693257&isnumber=5693237.

[18] De Castro, V., Marcos, E., Vara, J.M., “Applying CIM-to-PIM model transformations for the service-oriented development of information systems”, *Information and Software Technology*, vol 53, issue 1, January 2011, pp 87-105, ISSN 0950-5849, DOI: 10.1016/j.infsof.2010.09.002. (http://www.sciencedirect.com/science/article/pii/S0950584910001588).

[19] Buchwald, S., Bauer, T., Reichert, M., “Bridging the Gap Between Business Process Models and Service Composition Specifications”. In: *Service Life Cycle Tools and Technologies: Methods, Trends and Advances*, (2011), pp. 124-153. ISBN 978-1613501597.

[20] Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K., “SOMA: a method for developing service-oriented solutions”, *IBM System Journal*, 47 3 (2008).

[21] OMG, Service Oriented Architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS). OMG document: ad/2008-08-04, (2009), URL: http://www.omg.org/docs/ad/08-08-04.pdf.

[22] De Castro, M., Wieringa, "Towards a service-oriented MDA-Based approach to the alignment of business process with it systems: from the business model to a Web Services composition model", *International Journal of Cooperative Information Systems* (IJCIS), vol 18, issue 2, (2009), pp. 225-260, DOI: 10.1142/S0218843009002038.

[23] Hahn, C., Dmytro, P., Fischer, K., "A Model-Driven Approach to Close the Gap between Business Requirements and Agent-Based Execution". *Proceedings of the 4th Workshop on Agent-based Technologies and applications for enterprise interoperability*, Toronto, Canada, (2010).

[24] Business Process Model and Notation (BPMN) 2.0, Beta 1, OMG, May 2009.

[25] Eclipse Modeling Framework EMF. http://www.eclipse.org/modeling/emf/

[26] The Eclipse Project. Home Page, *IBM Corp*, (2000).

[27] Eclipse EuGENia, URL: www.eclipse.org/gmt/epsilon/doc/eugenia/.

[28] Advancing open standars for the information society, URL:http://www.oasis-open.org/.

[29] Herramientas para SCA (Service Component Arquitecture) para la plataforma Eclipse, URL: http://www.eclipse.org/soa/sca/.