

Universidad de las Ciencias Informáticas

Facultad 5



Título

*Método de Modelado de Servicios basado en SOAML y
MDA en el Proceso de Desarrollo para una Arquitectura
Orientada a Servicios.*

*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Autor: Ever Bravo Barros

Tutor: Ing. Yidier Romero Zaldivar

Ciudad de la Habana, Julio de 2012

“Año 54 de la Revolución”



Un camino de mil millas comienza con un paso.

Benjamín Franklin

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ever Bravo Barros

Autor

Ing. Yidier Romero Zaldivar

Tutor

Datos del Contacto

Tutor: Ing. Yidier Romero Zaldivar.

Edad: 30.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ing. en Ciencias Informáticas.

Categoría Docente: Asistente.

E-mail: yromero@uci.cu.

Graduado de la UCI, con dos años y medio de experiencia en el tema del desarrollo de arquitecturas orientadas a servicios y profesor del departamento de Soluciones SOA del CDAE en la Universidad de las Ciencias Informáticas.

Dedicatoria

A mis padres por su dedicación y mi formación, por siempre indicarme el camino correcto, por entenderme, por apoyarme, por estar ahí, por existir, por todo...

A mi hermano por ser mi mano derecha, en quien siempre podré confiar...

A mi maravillosa mujer por ser mi estrella en las noches oscuras, por darme fuerzas para seguir luchando, por amarme y siempre estar a mi lado en los buenos y en los malos momentos, por hacerme un hombre feliz...

Agradecimientos

Agradezco a todos los que han contribuido en mi formación y en la realización de la presente investigación, especialmente:

A mis padres y mi hermano, por todo lo que se puede agradecer en una vida, por su apoyo y entrega infinita, por hacerme el hombre que soy hoy.

A mi mujer Leticia, por su amor y resistencia, por siempre estar ahí, por ser el motivo de tantas de mis alegrías, por darme tanta felicidad.

A mi tutor Yidier, por toda su preocupación y entrega.

A todos mis amigos que llevamos cinco inolvidables años juntos, por siempre tenderme la mano cuando más lo necesité, mis compañeros de aula y de edificio.

A aquellas personas que quedaron en este largo camino y que sin ellos no hubiese sido posible estar aquí.

A la UCI por tantas cosas...

A todos los que han contribuido, aún sin saberlo, a la realización de la presente investigación.

Resumen

Con el surgimiento de la Arquitectura Orientada a Servicios (SOA) como paradigma de desarrollo de software, esta fue adoptada por la gran mayoría de las industrias de desarrollo de software, pues permite la interoperabilidad de estos sistemas. La Universidad de las Ciencias Informáticas (UCI), como industria cubana de software se propone la misma meta, por lo que crea un centro para ese propósito. En los proyectos con enfoque a SOA de dicho centro, los desarrolladores de los servicios no comprenden claramente los modelos realizados por los diseñadores de servicios, constituyendo esto una debilidad en el proceso de desarrollo. En el presente trabajo se desarrolla una investigación, ofreciendo información referente a los principales conceptos tratados y definiciones asociadas al modelado de servicios, especificando en lo referente al Lenguaje de Modelado de Arquitecturas Orientadas a Servicios (SoaML) y la Arquitectura Dirigida por Modelos (MDA) y su uso para dicho modelado, realizándose así un estudio del estado del arte del tema, para definir un procedimiento de modelado de servicios utilizando el estándar de modelado de servicios SoaML y describiendo así los artefactos generados en el diseño, dándole cumplimiento al objetivo principal de la investigación.

Palabras clave: SOA, Arquitectura, Servicios, Modelado, SoaML, MDA.

Índice de Contenido

INTRODUCCIÓN	1
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 CONCEPTOS GENERALES	5
1.1.1 <i>Arquitectura Orientada a Servicios (SOA)</i>	5
1.1.2 <i>Procesos de negocio</i>	6
1.1.3 <i>Servicio Web</i>	7
1.1.4 <i>Modelado de Servicios</i>	7
1.1.4.1 <i>Elementos clave en SoaML</i>	7
1.1.4.2 <i>Estándar SoaML</i>	8
1.2 HERRAMIENTAS DE MODELADO QUE SOPORTAN SOAML	13
1.3 PARADIGMA MDE	15
1.3.1 <i>Un ejemplo de MDE: MDA</i>	16
1.3.1.1 <i>Tipos de modelos MDA</i>	17
1.3.1.2 <i>Transformaciones de modelos</i>	18
1.3.1.3 <i>El enfoque Model Driven Architecture (MDA)</i>	19
1.4 INGENIERÍA DE SISTEMAS DIRIGIDOS POR MODELOS (MDSE)	20
1.4.1 <i>Modelo de Arquitectura de Negocio (BAM)</i>	22
1.4.2 <i>Modelo de Arquitectura de Software (SAM)</i>	23
1.4.3 <i>Modelo Específico de la Plataforma (PSM)</i>	24
1.5 IMPLEMENTACIÓN DE LAS SOLUCIONES SOAML CON MDA	25
1.6 CONCLUSIONES	25
2. CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	26
2.1 SELECCIÓN DE LA HERRAMIENTA DE MODELADO	26
2.2 MODELADO DE SERVICIOS EN LA METODOLOGÍA DEL CDAE	27
2.2.1 <i>Arquitectura y diseño de las soluciones</i>	28
2.2.2 <i>Especificación y Diseño de Servicios</i>	32
2.3 CONCLUSIONES	38
3. CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	39
3.1 MÉTODO DELPHI	39
3.1.1 <i>Selección de los expertos</i>	41
3.1.2 <i>Elaboración del cuestionario para validación de la propuesta</i>	45
3.1.3 <i>Desarrollo práctico y explotación de los resultados</i>	46
3.2 CONCLUSIONES	49
CONCLUSIONES GENERALES	50
RECOMENDACIONES	51

REFERENCIAS BIBLIOGRÁFICAS	52
ANEXO 1. ENCUESTA DE AUTOVALORACIÓN	57
ANEXO 2. ENCUESTA A EXPERTOS	58
ANEXO 3. LISTADO DE EXPERTOS QUE COLABORARON	60
ANEXO 4. FACTIBILIDAD DE LAS ACTIVIDADES CONTENIDAS EN EL PROCESO GENERAL DEL MÉTODO PROPUESTO.	61
ANEXO 5. CASO DE ESTUDIO EN LA HERRAMIENTA MODELIO	63

Índice de Figuras

FIGURA 1. PROCESAMIENTO DE ARQUITECTURA DE SERVICIOS.	11
FIGURA 2. CONTRATO DE SERVICIO.	12
FIGURA 3. INTERACCIÓN.	12
FIGURA 4. MENSAJES.....	13
FIGURA 5. DIBUJO REPRESENTATIVO DE LAS ÁREAS Y TECNOLOGÍAS QUE ABARCA MDA.	16
FIGURA 6. MODELOS CIM, PIM Y PSM Y SU RELACIÓN DE TRANSFORMACIÓN.	18
FIGURA 7. EL PROCESO GENERAL BASADO EN MODELOS.	21
FIGURA 8. ACTIVIDADES DEL MODELADO DE ARQUITECTURA DE NEGOCIO.	23
FIGURA 9. ACTIVIDADES DEL MODELADO DE ARQUITECTURA DE SOFTWARE.	24
FIGURA 10. MODELO ESPECÍFICO DE LA PLATAFORMA (PSM).	24
FIGURA 11. METODOLOGÍA DE DESARROLLO SOA DEL CDAE.	28
FIGURA 12. PISCINAS EN UN DIAGRAMA BPMN.	29
FIGURA 13. INTERCAMBIO DE MENSAJES ENTRE PISCINAS EN UN DIAGRAMA BPMN.	30
FIGURA 14. MODELADO DE UNA ARQUITECTURA DE SERVICIOS.	32
FIGURA 15. INTERFAZ DEL PROVEEDOR Y TIPOS DE MENSAJE DE UN SERVICIO.	34
FIGURA 16. MODELADO DE UN CONTRATO DE SERVICIOS CON INTERFAZ DEL PROVEEDOR.	35
FIGURA 17. MODELADO DE UNA INTERFAZ DE SERVICIO.	36
FIGURA 18. WSDL GENERADO A PARTIR DE LA INTERFAZ DE SERVICIO.....	38
FIGURA 19. MODELADO DE ARQUITECTURA DE SERVICIOS (MODELIO).....	63
FIGURA 20. MODELADO DE LOS TIPOS DE MENSAJE (MODELIO).	64
FIGURA 21. MODELADO DEL CONTRATO DE SERVICIO (MODELIO).	65
FIGURA 22. MODELADO DE LA INTERFAZ DE SERVICIO (MODELIO).....	66
FIGURA 23. TRANSFORMACIÓN A LOS WSDL A PARTIR DE LAS INTERFACES DE SERVICIO (MODELIO).	67
FIGURA 24. FICHERO WSDL GENERADO DEL SERVICIO APROBACIONPERFIL (MODELIO).	68

Índice de Tablas

TABLA 1. TABLA PARA OBTENER EL COEFICIENTE DE CONOCIMIENTO.....	43
TABLA 2. TABLA PARA CALCULAR EL COEFICIENTE DE ARGUMENTACIÓN.....	43
TABLA 3. ESCALA DE PUNTOS PARA LA DETERMINACIÓN DEL (KA).....	44
TABLA 4. EJEMPLO ILUSTRATIVO DE LAS SELECCIONES DE UN EXPERTO DADO.....	44
TABLA 5. RESULTADOS OBTENIDOS EN LA ENCUESTA DE AUTOVALORACIÓN.....	45
TABLA 6. TABLA DE FRECUENCIAS ABSOLUTAS.....	46
TABLA 7. TABLA DE FRECUENCIAS ABSOLUTAS ACUMULADAS.....	47
TABLA 8. TABLA DE FRECUENCIAS RELATIVAS ACUMULADAS.....	47
TABLA 9. PUNTOS DE CORTE Y GRADO DE ADECUACIÓN PARA CADA CRITERIO.....	48
TABLA 10. GRADO DE ADECUACIÓN DE LOS INDICADORES.....	48

Introducción

En los últimos años se han experimentado grandes cambios en el área del desarrollo del software, tanto en la proliferación de nuevas tecnologías, metodologías y enfoques en dicho desarrollo que han repercutido en las organizaciones actuales, así como los cambios en los requerimientos y necesidades a nivel organizacional han repercutido en la forma de desarrollo y ejecución del software.

Lo que se hace notorio es que una necesidad que antes pudo ser medianamente satisfecha con diversidad de enfoques y tecnologías, actualmente está requiriendo respuestas más integradas.

Con el propósito de resolver estos problemas las industrias que desarrollan software se plantean la necesidad de tener sistemas de software más flexibles, por lo cual se requiere que estos sean más desacoplados e interoperables.

La Arquitectura Orientada a Servicios (SOA, del inglés Service Oriented Architecture), ha emergido en este nuevo siglo como un paradigma de desarrollo de software y se está convirtiendo en la forma moderna de concebir las mejores prácticas de ingeniería de software.

Una Arquitectura Orientada a Servicios (SOA) representa una metodología para lograr interoperabilidad entre aplicaciones y servicios web de manera tal que permita reutilizar Tecnologías de la Información (IT, del inglés Information Technology) ya existentes. (2)

La integración y coordinación de los procesos de negocio, puede verse simplificada utilizando tecnologías de integración como SOA y servicios web.

Las industrias que implementen satisfactoriamente SOA, seguramente poseerán una ventaja competitiva importante sobre las industrias que no la implementen, porque las que tienen sus servicios alineados con los negocios estratégicos de las IT, pueden reaccionar más rápido a los cambios en los requerimientos de negocio que las que no los tienen. (2)

En Cuba se desarrolla un proceso de informatización de la sociedad, el mismo dentro de sus objetivos tiene el de lograr un mayor avance socio-económico. Con la idea del Comandante en Jefe Fidel Castro Ruz de crear entidades que garantizaran el desarrollo de la informática en nuestro país surge la Universidad de las Ciencias Informáticas (UCI), con ese propósito y mediante la formación del personal capacitado, se llevaría a cabo la difícil tarea de informatizar la nación, además de contribuir a la economía

del país mediante la creación de un nuevo renglón de exportación. Se puede decir que esta idea ha dado sus primeros frutos, pues se han creado diversos proyectos para la obtención de dichos productos informáticos para empresas tanto nacionales como extranjeras.

El constante aumento del conocimiento en el campo de la informática supone la diversificación y especialización de perfiles dentro del mismo.

Atendiendo a este principio de especialización surge la idea de crear, dentro de la propia universidad, un Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) el cual contaría con el personal capacitado para dar solución a cualquier situación enmarcada en este ámbito. Dentro de este centro se le presta especial atención a los proyectos de SOA, pues definen un camino para todas aquellas organizaciones que desean alinear la estrategia de su negocio con las Tecnologías de la Información, al mismo tiempo que agilizan sus procesos, maximizan el rendimiento de sus recursos y facilitan y estandarizan la integración de los sistemas. Estos sistemas informáticos son esenciales para las empresas modernas, cada vez más global. Por su creciente aceptación y probada eficiencia, se ha estado trabajando sobre la base de obtener la experiencia necesaria que permita al centro enfrentar proyectos de este tipo.

Como la gran mayoría de las organizaciones de este tipo se esfuerzan continuamente para llegar antes al mercado y reducir el costo de desarrollo y mantenimiento de aplicaciones informáticas para apoyar sus operaciones, los desarrolladores del centro (CDAE) también se proponen esas metas aunque la realidad existente es otra. Actualmente en el CDAE existe una línea de investigación y desarrollo que se interesa por las soluciones orientadas a servicios. Para el desarrollo de proyectos con este enfoque se está elaborando una metodología de desarrollo que tiene varios procedimientos y artefactos definidos, los cuales se encuentran asociados a la documentación de un proceso de desarrollo SOA. Sin embargo, en los proyectos realizados con enfoque a SOA, no existe una buena comprensión por parte de los desarrolladores de servicios al interactuar con los modelos realizados por los diseñadores de servicios. Aunque hay presente una trazabilidad entre los modelos del negocio y los modelos de especificación de servicios, falta claridad en la representación y la especificación de la vista de implementación, cuya vista es la más cercana al implementador. Esto constituye claramente una debilidad en el proceso de desarrollo, pues los desarrolladores invierten más tiempo del estimado para la comprensión de los artefactos de entrada a la fase de implementación.

Ante el análisis de la **situación problemática** anteriormente expuesta, se plantea como **problema a resolver**: ¿Cómo mejorar la calidad de los artefactos generados en el diseño para minimizar los retrasos en la fase de implementación?

Objeto de estudio: Modelado de servicios en una arquitectura orientada a servicios.

Campo de Acción: Modelado de servicios con SoaML dentro de la metodología del CDAE.

Se plantea **como objetivo general**: Definir un procedimiento de modelado de servicios utilizando SoaML que describa los artefactos generados en el diseño para minimizar los retrasos existentes.

Objetivos específicos:

1. Realizar un estudio del estado del arte del modelado de servicios.
2. Definir un método de modelado de servicios basado en SoaML y un mecanismo basado en MDA para la generación de código fuente a partir de diagramas en SoaML.
3. Validar la propuesta con expertos.

Para dar cumplimiento a dichos objetivos específicos se proponen las siguientes **tareas de investigación**:

1. Revisión del estado del arte del modelado de servicios.
2. Caracterización del lenguaje de modelado SoaML.
3. Caracterización del enfoque de desarrollo de software MDA.
4. Elaboración del método de modelado de servicios utilizando SoaML.
5. Elaboración de un mecanismo para la generación de código fuente basado en MDA.
6. Validación de la propuesta con expertos.
7. Elaboración de un Caso Estudio.

Resultado esperado: Un método de modelado independiente de la plataforma de desarrollo, basado en UML y con facilidades para la generación de código usando MDA.

Idea a defender: Si se define un método de modelado de servicios que sea fácilmente interpretado por los desarrolladores, entonces el tiempo de implementación disminuye.

Estructura de la Tesis

Capítulo 1: Fundamentación Teórica.

En este capítulo se analizan aspectos teóricos y se ofrece información referente a los principales conceptos tratados y definiciones asociadas a SoaML y MDA. Se realiza el estudio del estado del arte del tema y se aborda sobre las principales herramientas utilizadas en el modelado de servicios en proceso de desarrollo de una SOA, para poder integrar dichos elementos y dar solución al problema en cuestión.

Capítulo 2: Propuesta de Solución

En este capítulo se realiza la propuesta de solución mediante análisis de los principales marcos tecnológicos para soluciones SoaML y su integración con MDA de tal manera que contribuya a la generación automática de artefactos derivados, teniendo en cuenta los aspectos más importantes en el uso de la herramienta escogida, lo que de conjunto con el estudio de los casos de éxito existentes en cuanto a la utilización de la misma, permitirá contar con los elementos necesarios para plantear los escenarios de negocio propicios para implementar dicha solución conjuntamente con el marco tecnológico a utilizar. Además se hace énfasis en los principios y premisas que se deben de tener en cuenta antes de aplicar el procedimiento y por último se realiza la descripción y explicación detallada del mismo.

Capítulo 3: Validación de la Solución Propuesta

Se muestra el resultado de la validación realizada conformado por un Cuadro de Expertos que exponen su criterio acerca de la propuesta presentada. Se brindan las conclusiones de la investigación.

1. Capítulo 1: Fundamentación Teórica

El presente capítulo constituye el marco teórico de la investigación a realizar. Se define una serie de conceptos necesarios para entender el objetivo fundamental del trabajo, profundizando en temas como el lenguaje de modelado de servicios en el proceso de desarrollo de una SOA, haciendo énfasis en la especificación SoaML y en su enfoque basado en MDA, sus tendencias actuales así como las herramientas más utilizadas en la actualidad. Además se llevó a cabo un estudio de la metodología utilizada para garantizar un método de modelado independiente de la plataforma de desarrollo usando MDA para la generación de código.

1.1 Conceptos Generales

1.1.1 Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) es un estilo de arquitectura que soporta el paradigma de desarrollo SOC (Service Oriented Computing), 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. (4)

SOA es una arquitectura en el cual una aplicación se constituye de servicios que se exponen y servicios que se consumen; difiere del tradicional enfoque cliente/servidor haciendo énfasis en el bajo acoplamiento entre los componentes de software. Dado que estos servicios pueden ser consumidos por diferentes sistemas y plataformas, las características de los Servicios Web son ideales para implementar esta solución, pero debe seguirse algún tipo de Modelo de Ciclo de Vida para el desarrollo de aplicaciones con esta arquitectura. (2)

Arquitectura Orientada a Servicios (SOA) es una forma de organizar y entender a las organizaciones, comunidades y sistemas para maximizar la agilidad, la escala y la interoperabilidad. (6)

SOA es un paradigma de arquitectura para la definición de cómo las personas, organizaciones y sistemas de suministro utilizan los servicios para lograr resultados. (6)

El enfoque SOA es simple, la gente, las organizaciones y los sistemas de prestación de servicios a los demás. Estos servicios permiten solucionar problemas sin hacerlo nosotros mismos o incluso sin saber cómo hacerlo, lo que ofrece eficiencia y agilidad. (6)

El paradigma SOA funciona igual de bien para la integración de las capacidades existentes, así como la creación e integración de nuevas capacidades. (6)

SOA es, esencialmente, sobre cómo las personas, organizaciones y sistemas trabajan juntos para lograr un objetivo empresarial común. La abstracción de SOA hace esto mediante la comprensión de cómo se puede ser independientes y "débilmente acoplados", pero trabajar en colaboración. En el paradigma SOA se trabaja junto por el suministro y uso de servicios de cada uno. SOA se puede aplicar a un nivel muy alto, para entender de una empresa o de la comunidad o en un nivel muy detallado para entender y especificar una operación de servicio en particular ofrecido entre los sistemas. Este amplio alcance y la capacidad de "conectar los puntos" entre las preocupaciones de negocios y la tecnología es una de las ventajas de SOA y lo que se llama Enterprise SOA. (3)

1.1.2 Procesos de negocio

La norma internacional ISO-9001 define un proceso como "una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados". (1)

Un proceso de negocio es una colección de actividades relacionadas lógicamente, que toman uno o más tipos de entradas y crean uno o más resultados que producen un valor para la organización, sus inversores o sus clientes. (1)

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) y de Orientación a Servicios (Service Oriented Computing, SOC) 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. (4)

1.1.3 Servicio Web

Un servicio web no es más que el mecanismo que posibilita el acceso a una o más capacidades del negocio, dicho acceso se provee a través del uso de una interfaz determinada, la misma es producida consistentemente con restricciones y políticas especificadas en la descripción del servicio. La implementación del servicio web se mantiene oculta al consumidor del mismo, este último debe acceder a dicho servicio a través de los modelos de información y comportamiento definidos en la interfaz.

Un servicio web es una capacidad que se ofrece a través de una interfaz bien definida y disponible a una comunidad. (6)

Los servicios web también permiten ofrecer las capacidades a los demás. (6)

1.1.4 Modelado de Servicios

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) 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) es la propuesta de OMG (Object Management Group) para MDD. El estándar de Modelado de Arquitecturas Orientadas a Servicios (Service Oriented Architecture Modeling Language, SoaML) de OMG es un paso para el avance del modelado de servicios, que define un metamodelo y un perfil UML (Lenguaje de Modelado Unificado), extendiendo la notación UML. (4)

1.1.4.1 Elementos clave en SoaML

El Lenguaje de Modelado para Arquitecturas Orientadas a Servicios (Service Oriented Architecture Modeling Language, SoaML) de OMG provee un perfil UML y un metamodelo que extiende el metamodelo UML para diseñar servicios en SOA. 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 una o varias interfaces y un contrato asociados. (4)

Las interfaces pueden ser de tipo Interfaz de Servicio (ServiceInterface) o Interfaz Simple (Interface) UML. Un Contrato de Servicios (ServiceContract) define los términos, condiciones, interfaces y coreografía que

los participantes acuerdan para utilizarlo, esta última se 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 (MessageType) especifica la información intercambiada. (4)

1.1.4.2 Estándar SoaML

Surgimiento del estándar SoaML



El Object Management Group (<http://www.omg.org>) es un consenso de la industria, normas de organización que se centran en la interoperabilidad a través de la arquitectura y el middleware. El OMG es el hogar de estándares tales como el Unified Modeling Language (UML), Business Process Modeling Notation (BPMN), el Common Object Request Broker Architecture (CORBA), Servicio de Gestión de Registros y muchos otros. El OMG con más de 20 años de edad es una organización de varios cientos de miembros y un proceso probado para la definición de estándares. (3)



Adopción del estándar SoaML

SoaML fue adoptado en 2009 por el OMG después de un proceso de tres años que involucró a varios participantes de las empresas grandes y pequeñas, así como el mundo académico. SoaML se basa en la experiencia previa, metodologías y productos de estos expertos en SOA y está diseñado para apoyar las mejores prácticas de SOA, mientras normalizan términos divergentes y anotaciones. La página web <http://www.SoaML.org> es la del estándar SoaML, la misma también lista algunos de los proveedores de herramientas que ya han lanzado herramientas o servicios en apoyo de SoaML. La adquisición de SoaML por los usuarios, herramientas y proveedores de servicios ha sido bastante rápida, lo que es una buena señal para el futuro de la norma. (3)



SoaML y UML

El Lenguaje de Modelado Unificado (UML) es el estándar de modelado de base de OMG. UML es ampliamente apoyada por herramientas, proveedores de servicios y la educación. UML se puede utilizar para una amplia variedad de propósitos y metodologías y está diseñado para ser ampliado para fines particulares el uso de perfiles. SoaML es como un perfil UML: el estándar de perfil UML para el modelado de los servicios y arquitecturas orientadas a servicios. SoaML puede conectar a la mayoría de las herramientas UML con la ayuda extra o no, y también se puede usar con otras funciones y perfiles de UML. Los usuarios de SoaML suelen tener algunos antecedentes en el modelado y UML. Algunas de las herramientas UML también se han ampliado para ofrecer capacidades específicas para modelar los servicios basados en SoaML. (3)

Alcance de SoaML

Uno de los puntos fuertes de SoaML es su alcance. Hay una gran cantidad de herramientas de tecnología específicas que permiten crear un servicio web. Sin embargo, estas herramientas de tecnología no responden a las preocupaciones de alto nivel: tienen problemas para mostrar una arquitectura orientada a servicios completa - cómo los servicios y los participantes de servicios trabajan juntos para proporcionar valor de negocio. Las herramientas de la tecnología también están mal equipadas para soportar SOA para las empresas de modelado, el gobierno o la defensa y por lo general están encerrados en una tecnología en particular y/o proveedores. Sin embargo, si la única preocupación es la creación de un par de servicios con una tecnología específica, estas herramientas especializadas pueden funcionar bien. (3)

Las organizaciones son más eficaces cuando comprenden sus servicios de tecnología en relación con su negocio de servicios, información y procesos. SoaML proporciona la capacidad de crear y aprovechar una arquitectura que ayuda a las personas, organizaciones y sistemas de colaboración a través de servicios y muestra cómo esos servicios se conectan a otras partes de la arquitectura, tales como procesos, información y reglas de negocio. (3)

Cuando se escala a partir de un servicio a los múltiples servicios que pueden tener preocupaciones relacionadas entre sí, se necesita una arquitectura. Muchas organizaciones tienen cientos de servicios

definiendo y regulando dicho inventario de servicios sin una arquitectura, eso es caótico y con pocas posibilidades de alcanzar los objetivos de negocio. (3)

Como un lenguaje arquitectónico, SoaML proporciona una forma independiente de la tecnología y estándar para crear, comunicar y aprovechar una arquitectura orientada a servicios. (3)

Descripción general de la Especificación SoaML

La especificación SoaML (Lenguaje de Modelado de Arquitectura Orientada a Servicio) se crea en respuesta a la UPMS (Perfil UML y Metamodelo de Servicios) y describe un perfil UML y metamodelo para el diseño de los servicios dentro de una arquitectura orientada a servicios. (6)

Los objetivos de SoaML son para apoyar las actividades de modelado y diseño de servicios y encajar en un enfoque global de desarrollo basado en modelos. Por supuesto, hay muchas maneras de abordar los problemas de diseño de servicios. En caso de que se ejecutará desde el punto de vista de un consumidor de servicios que solicita un servicio que se construirá. En caso de que se ejecutará desde el punto de vista de un proveedor de servicios que anuncia un servicio para aquellos que estén interesados y capacitados para usarlo. O, si se toma desde la perspectiva de un diseño de sistema que describe cómo los consumidores y proveedores interactúan para lograr los objetivos generales. En lugar de suponer ningún método en particular, el perfil y el metamodelo acomodan todas estas diferentes perspectivas de un enfoque coherente y cohesionada a la descripción de los requisitos de los consumidores, las ofertas de los proveedores y la interacción y los acuerdos entre ellos. (6)

El metamodelo SoaML se basa en el metamodelo de UML 2.0 L2 y proporciona las extensiones mínimas de UML, sólo cuando sea absolutamente necesario para lograr las metas y los requisitos de modelado de servicios. La especificación se aprovecha de la función de combinación de paquete de UML 2.0 para combinar las extensiones a UML. El perfil ofrece una versión de UML específica del metamodelo que pueden ser incorporados en las herramientas estándar de modelado UML. (6)

SoaML como se describe en esta especificación proporciona una forma estándar de arquitectura y modelos de soluciones SOA utilizando el Lenguaje de Modelado Unificado (UML). El perfil utiliza los

mecanismos de extensión integrada de UML para definir los conceptos de SOA en términos de los conceptos de UML.

SoaML se puede utilizar con las actuales herramientas UML fuera de la plataforma, pero algunas herramientas específicas de SOA pueden ofrecer un mejorado, con el apoyo del metamodelo SoaML compatible. (6)

El perfil de SoaML es compatible con la gama de requisitos de modelado de arquitecturas orientadas a servicios, incluyendo la especificación de los sistemas de servicios, la especificación de las interfaces de servicios individuales, y la especificación de las implementaciones de servicio. Todo esto se realiza con el fin de lograr un objetivo, y es enfocarlo de tal manera que contribuya a la generación automática de artefactos derivados, según un enfoque basado en MDA. (6)

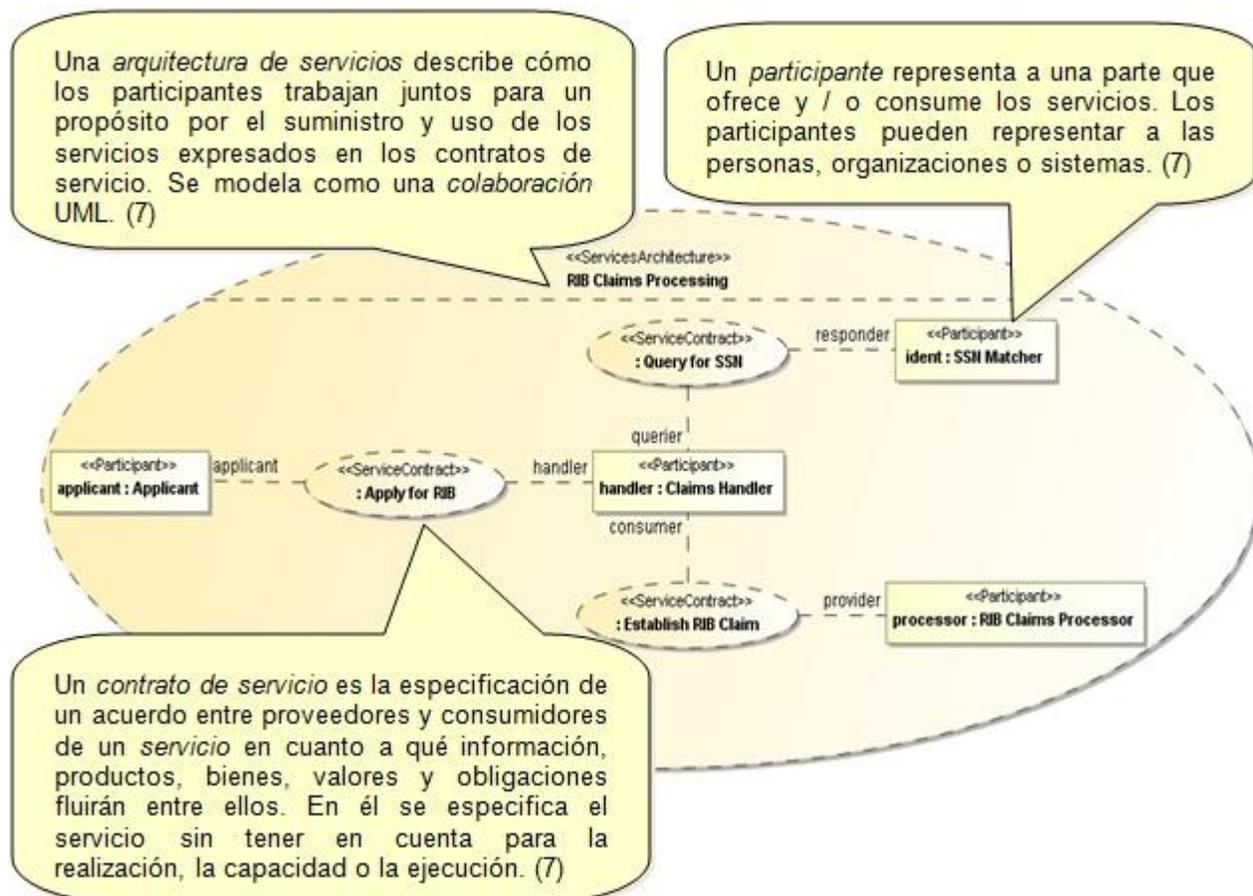


Figura 1. Procesamiento de Arquitectura de Servicios. (7)

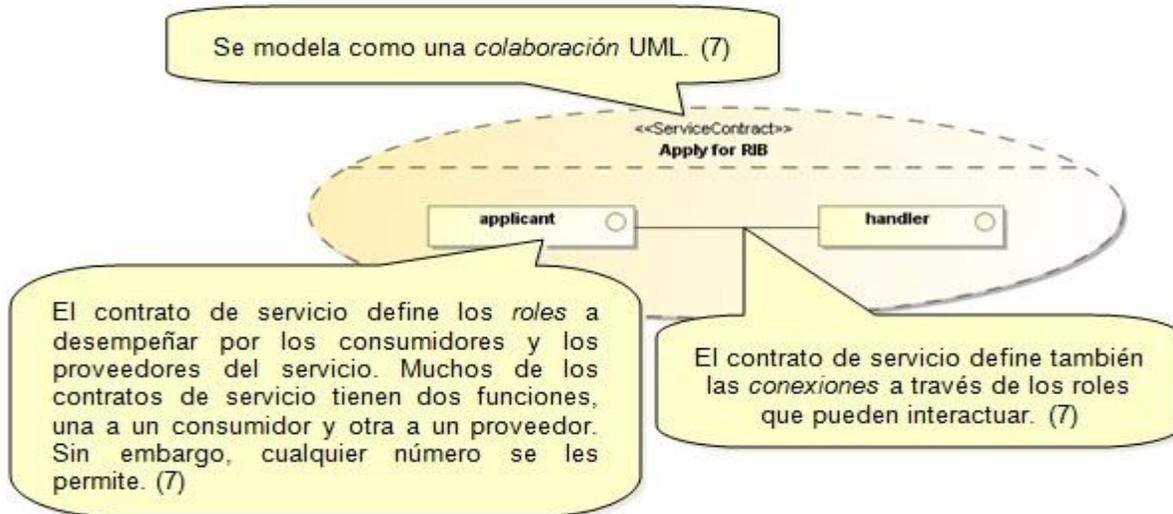


Figura 2. Contrato de Servicio. (7)

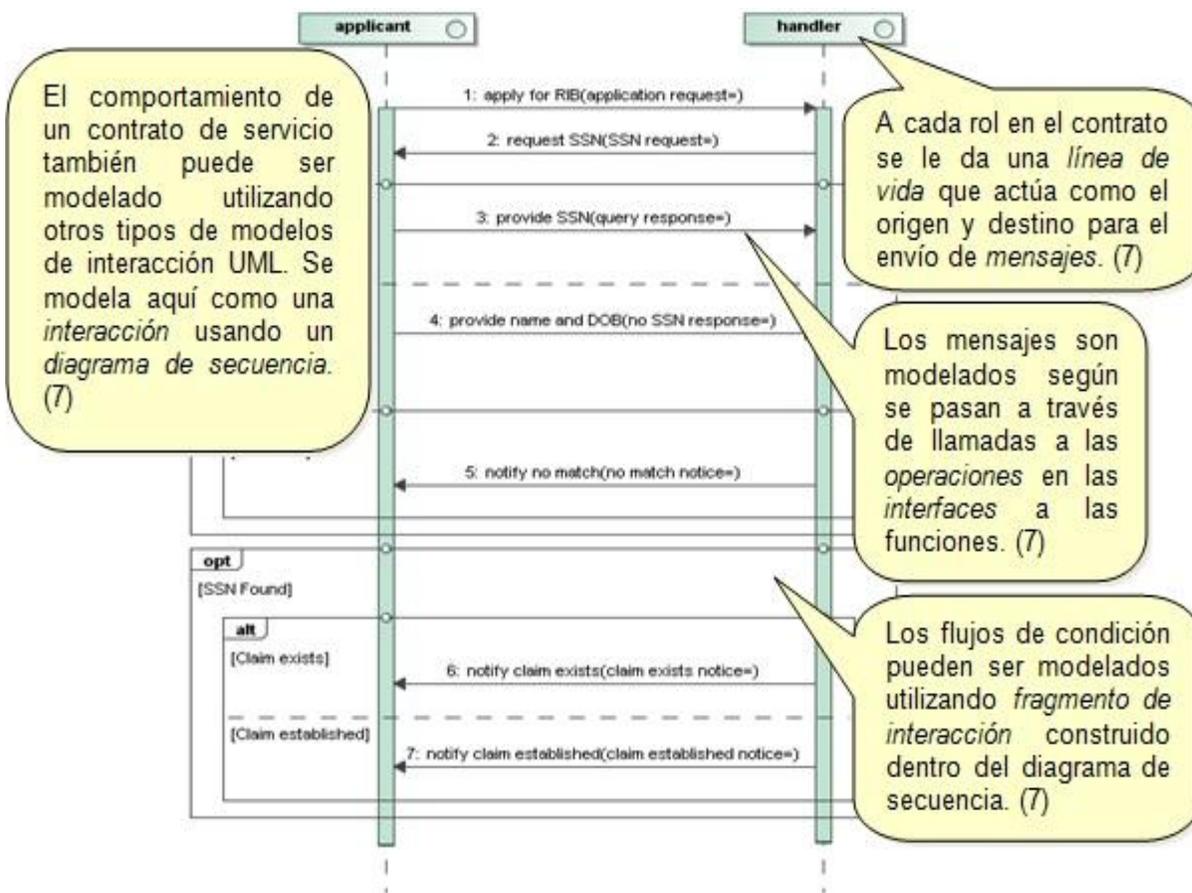


Figura 3. Interacción. (7)

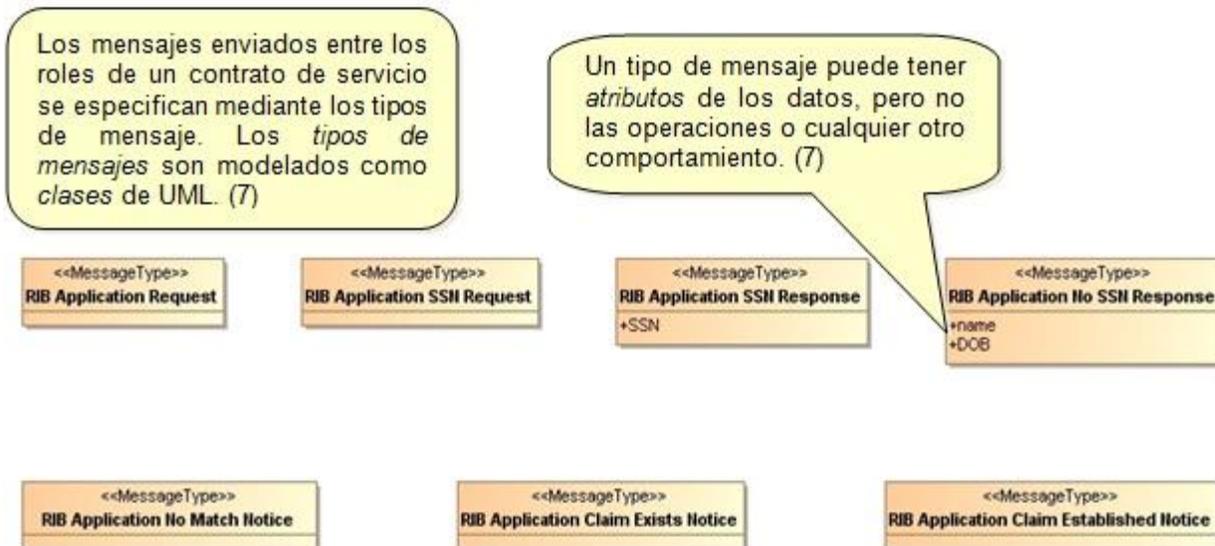


Figura 4. Mensajes. (7)

1.2 Herramientas de modelado que soportan SoAML

De la empresa **IBM**:

- IBM Rational Software Architect 7.5.4 o superior
- IBM Rational Software Modeler 7.5.4 o superior

IBM presta soporte para el lenguaje SoAML desde las versiones de las herramientas IBM Rational Software Architect 7.5.4 e IBM Rational Software Modeler 7.5.4. Además de soportar un perfil para SoAML, proporcionan un conjunto de herramientas y plantillas de modelos para estandarizar y acelerar el esfuerzo en el diseño del servicio. También tienen herramientas para generar artefactos relacionados con SOA, incluyendo código Java para las interfaces de servicio y componentes SCA, generación de ficheros de las especificaciones XSD, WSDL, Composiciones SCA y especificaciones BPEL. La empresa además recomienda el método SOMA para realizar el diseño de servicios exitosamente. (5)

La Comunidad de **ModelDriven** (The ModelDriven Community):

-ModelPro

La comunidad ModelDriven anunció en la primera versión del ModelPro el soporte de SoaML. El ModelPro es un motor de MDA disponible para producir una amplia variedad de artefactos desde los modelos. El paquete SoaML del ModelPro está habilitado para producir los ejecutables de la implementación de servicios web para las arquitecturas de servicio definidas en SoaML. Las tecnologías actuales soportadas son los servicios web, eclipse y JEE (a otras tecnologías se les están intentando dar soporte en un futuro). (5)

La empresa **SOFTEAM**:

- Modelio CASE Tool

SOFTEAM proporciona una extensión de Modelio llamada SoaMLDesigner, dicho componente es opensource. Incluye las últimas especificaciones del lenguaje SoaML e implementa una interfaz gráfica (GUI) que incluye 6 diagramas: diagrama de Capacidades, de Contrato del Servicio, de Arquitectura del Servicio, de Mensajes y de Participantes. SoaML está integrado con la metodología de SOFTEAM para arquitecturas empresariales. Esta solución integra Modelos de Negocio (Objetivos, Reglas de Negocio, Requisitos), Procesos de Negocio (BPMN), SoaML y las transformaciones a varios modelos de implementación que incluye XSD, WSDL, BPEL y Java. (5)

De la empresa **Sparx Systems**:

- Enterprise Architect 7.5 o superior

Este plugin fue proporcionado por los colaboradores de CBDI. (5)

SoaML en Enterprise Architect

Enterprise Architect permite modelar arquitecturas de servicios de forma rápida y sencilla, a través del uso de una tecnología MDG integrado con el programa de instalación de Enterprise Architect. Las instalaciones SoaML se proporcionan en forma de (5):

- ✓ Dos tipos de diagramas: diagrama de componentes SoaML y el diagrama de secuencia SoaML. Se accede a través del cuadro de diálogo Nuevo diagrama.

- ✓ Páginas SoaML en la caja de herramientas UML de Enterprise Architect.
- ✓ Elemento SoaML y las entradas de las relaciones de la caja de herramientas UML.
- ✓ Menú de acceso directo y rápido vinculador.

De la empresa **No Magic**

-MagicDraw

En el sitio de la comunidad de ModelDriven.org proporcionan un plugin para la herramienta MagicDraw. Se espera que las versiones más recientes de la herramienta ya este soportado este lenguaje de modelado. (5)

1.3 Paradigma MDE

Durante las últimas dos décadas, los avances en lenguajes y plataformas han aumentado el nivel de abstracción disponible en la tarea de desarrollo de software. Además debido a la madurez de los lenguajes de tercera generación, los desarrolladores de software están mejor equipados para afrontar y resolver los distintos problemas que se les pueden plantear. (10)

A pesar de todos estos avances, aún quedan problemas importantes que resolver. En el centro de éstos se encuentra el crecimiento de la complejidad de las plataformas, las cuales contienen miles de clases y métodos con dependencias muy complicadas que deben ser conocidas por el desarrollador. El problema se acrecienta cuando estas mismas plataformas crecen rápidamente y además aparecen otras nuevas, con el consiguiente esfuerzo de migración de unas a otras. (10)

En este último caso, cuando la evolución tecnológica de las plataformas o de los sistemas se produce, es importante conservar el mismo modelo conceptual del negocio, es decir, que la lógica del dominio del problema debería ser la misma, sea cual sea la plataforma o lenguaje que implementa dicha lógica. (10)

Para manejar el problema del crecimiento de la complejidad de los sistemas, la orientación a objetos no parece ser suficiente. Los lenguajes orientados a objetos han ido perdiendo la simplicidad con la que fueron ideados, la encapsulación no es un recurso tan útil como en principio parecía y, sobre todo, la reutilización de los objetos como componentes no ha tenido demasiado éxito en la industria del software. (10)

Parece que las propuestas centradas en el código no dan respuesta a las demandas de los sistemas actuales. Esta es la razón por la que ha aparecido una nueva propuesta centrada en modelos. A esta propuesta se le llama Ingeniería Dirigida por Modelos (MDE). (10)

1.3.1 Un ejemplo de MDE: MDA

El consorcio OMG ha desarrollado la propuesta “Arquitectura Dirigida por Modelos” o MDA como ejemplo de implementación de MDE. (10)

MDA nace con la idea establecida de separar la especificación de la lógica operacional de un sistema, de los detalles que definen cómo el sistema usa las capacidades de la plataforma tecnológica donde es implementado. (10)

Teniendo en cuenta lo anterior, los objetivos de MDA son la portabilidad, la interoperabilidad y la reusabilidad a través de la separación arquitectural. (10)

El concepto de *independencia de plataforma* aparece frecuentemente en MDA. Es la cualidad que tienen los modelos de ser independientes de las características de cualquier tipo de plataforma tecnológica. (10)

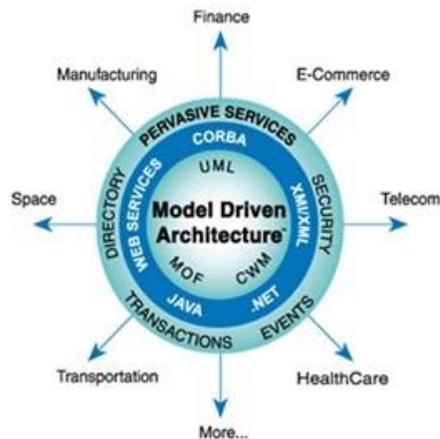


Figura 5. Dibujo representativo de las áreas y tecnologías que abarca MDA. (10)

La meta de la iniciativa MDA, promovida por el Object Management Group, es la separación de la lógica de las aplicaciones de la plataforma software en la que dicha lógica vaya a ser implementada. Esta separación permite reducir el impacto que la evolución de las tecnologías tiene en el desarrollo de aplicaciones, al permitir que una misma especificación pueda ser rectificada en diferentes plataformas

software. Por otro lado, traslada el conocimiento o la propiedad intelectual de una aplicación desde el código fuente (enfoque tradicional) a la especificación. (9)

MDA confiere a los modelos la mayor importancia en el proceso de desarrollo de aplicaciones, en detrimento del código. El modelo se convierte en el elemento más valioso (hasta el punto de afirmar que, en MDA, todo es un modelo) puesto que a partir de él, mediante una serie de transformaciones se puede obtener el código de la aplicación. (9)

El desarrollo de aplicaciones basado en MDA se basa en modelar las aplicaciones a desarrollar mediante un modelo independiente de detalles de implementación o de la plataforma software en la que la aplicación vaya a ejecutarse (PIM, Platform-Independent Model), refinar estos modelos independientes de plataforma transformándolos en otros que incorporen detalles de implementación (PSM, Platform-Specific Model), y transformar finalmente estos modelos en el código de la aplicación (Implementation Model, también llamado Code Model). (9)

Mediante la aplicación de este paradigma se cubre completamente el ciclo de vida de un sistema software, desde la captura de requisitos hasta el mantenimiento del mismo, pasando por la generación del código fuente. Para ello define tres tipos de modelos que se explican a continuación. (10)

1.3.1.1 Tipos de modelos MDA

Los tipos de modelos que MDA propone son los siguientes:

- **Modelo Independiente de la Computación (CIM):** Un CIM no muestra detalles de la estructura del sistema. A veces es llamado modelo de dominio o modelo de negocio. En el CIM se modelan los requisitos que deberá satisfacer el sistema, describiendo la situación en la cual el sistema será usado. Es muy útil tanto para ayudar a comprender el problema como para ejercer de fuente de vocabulario compartido para el uso en otros modelos. Según MDA, la especificación de requisitos de un sistema CIM debería ser transformable en un PIM y posteriormente en un PSM y viceversa. El CIM juega un papel importante como puente entre los que son unos expertos en el dominio del problema y sus requisitos y aquellos que son expertos en el diseño y construcción de artefactos software. (10)
- **Modelo Independiente de la Plataforma (PIM):** Un modelo PIM muestra el grado de independencia de plataforma necesario para poder ser usado en diferentes plataformas tecnológicas de un tipo similar. Este modelo debe tener tal nivel de abstracción que no cambie, sea

cual sea la plataforma elegida para su implementación. Con este modelo se representa la lógica del sistema y sus interacciones con el mundo exterior, sin entrar en detalle de qué tipo de tecnología implementará cada parte y cómo se adapta a una plataforma específica. (10)

- **Modelo Específico de Plataforma (PSM):** El modelo PSM es una vista del sistema para una plataforma específica. Éste combina la especificación del sistema hecha en el PIM, con los detalles que especifican la manera en que dicho sistema usa una plataforma particular. (10)

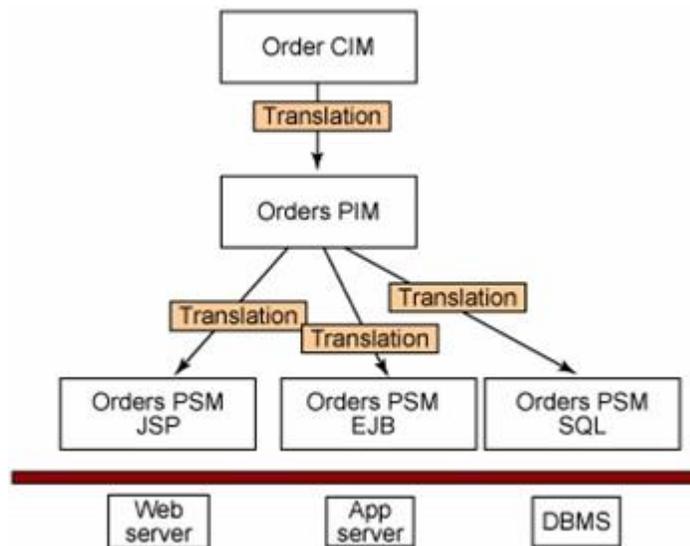


Figura 6. Modelos CIM, PIM y PSM y su relación de transformación. (10)

Aunque UML es el lenguaje de modelado central de MDA, no todos los modelos tienen por qué estar especificados en dicho lenguaje. Para ello entra en juego el concepto de *metamodelo*. Un *metamodelo* es un modelo para definir modelos. UML es un metamodelo que especifica cómo crear modelos UML. Es decir, un modelo UML es una instancia del metamodelo UML.

1.3.1.2 Transformaciones de modelos

La transformación de modelos es el proceso de convertir un modelo en otro modelo del mismo sistema. (10)

En el paradigma MDA, la transformación de modelos puede ser horizontal o vertical. Una transformación horizontal consiste en pasar un modelo de un nivel de abstracción M_x a otro modelo del mismo nivel M_x , pero ambos basados en un modelo del nivel inmediato superior M_{x-1} diferente. Una transformación

vertical ocurre cuando los modelos pertenecen a dos niveles inmediatos diferentes, M_x y M_{x-1} . Por ejemplo, se puede usar la transformación vertical de modelos para pasar de un modelo PIM a un modelo PSM, o de un modelo CIM a un modelo PIM; o una transformación horizontal para pasar de un PIM1 a otro PIM2 basado en un metamodelo diferente que el primero. (10)

Las distintas etapas del ciclo de vida del software pueden ser representadas en función de los distintos tipos de modelos que MDA propone y de la transformación de dichos modelos con la transición de una etapa a otra. (10)

Para el caso en que se aplique el paradigma MDA a un sistema heredado (legado o antiguo), se puede aplicar la reingeniería para soportar su evolución, para lo que es necesario la transformación inversa de modelos (de PSM a PIM), lo cual también es sugerido por MDA. (10)

El lenguaje estándar que OMG propone para la definición de transformaciones de modelos es el lenguaje QVT, que a su vez se basa en el lenguaje de restricciones OCL (*Object Constraint Language*). Con QVT se pueden definir transformaciones genéricas entre metamodelos, así cualquier instancia del metamodelo fuente puede ser transformado en una instancia del metamodelo destino. (10)

1.3.1.3 El enfoque Model Driven Architecture (MDA)

Model Driven Architecture (MDA), es un enfoque de desarrollo de software que no plantea de forma explícita la realización del modelado del negocio como requerimiento para el desarrollo, ni la orientación a servicios para el diseño de las aplicaciones, pero si los permite y promueve. (8)

Plantea realizar tres vistas del desarrollo de software como modelos: Computation Independent Model (CIM) o modelo independiente de la computación en el cual especificar los requerimientos del desarrollo con artefactos como Modelo de Casos de Uso, de dominio, entre otros; Platform Independent Model (PIM) o modelo independiente de la plataforma como modelo de diseño del software en el cual incluir diagramas de subsistemas y clases, entre otros; Platform Specific Model (PSM) o modelo específico de la plataforma, donde se transforma el PIM para obtener un modelo para una plataforma en particular o el código asociado en forma directa. (8)

Este enfoque sigue el principio básico de la Ingeniería de Software de separación de intereses, donde en cada vista se plantea la obtención de distintos intereses asociados al desarrollo. Como meta principal se plantea la portabilidad, interoperabilidad y reusabilidad de las aplicaciones obtenidas. El aspecto central del enfoque es la transformación de modelos, que según la definición provista por el estándar, es el proceso de convertir un modelo en otro modelo del mismo sistema, la cual se realiza especificando la

transformación de un objeto desde un modelo origen a uno o más objetos en un modelo destino, siguiendo distintos enfoques. (8)

Para permitir estas transformaciones se proveen también los mapeos entre modelos y el marcado de modelos. Un mapeo brinda las especificaciones para realizar las transformaciones de un PIM en un PSM para una plataforma específica, mediante mapeos o equivalencias de elementos en el modelo origen al modelo destino, en un lenguaje como QVT. Las marcas de modelos permiten marcar elementos en un modelo de forma de identificar la transformación que se desea realizar sobre el mismo, pueden ser por ejemplo estereotipos de un perfil UML. (8)

La automatización de la transformación de un PIM hacia uno o más PSM permite crear desde una misma solución conceptual especificada en el PIM, aplicaciones que se ejecutan en plataformas distintas, como J2EE o .NET, simplemente generando desde el PIM los PSM o el código asociado a cada plataforma elegida. Esto permite entonces que los modelos constituyan la base del desarrollo, donde los cambios requeridos se realicen en el PIM asociado y sean luego impactados en las plataformas correspondientes. (8)

1.4 Ingeniería de Sistemas Dirigidos por Modelos (MDSE)

La metodología MDSE (del inglés Model-Driven Systems Engineering) tiene como objetivo integrar a las actuales prácticas comerciales de modelado dentro de una empresa, lo que permite el aprovechamiento de las prácticas existentes y la ampliación de modelos en lugar de reemplazarlos. A partir de estos modelos vamos a conducir la especificación de los servicios como un conjunto de artefactos de modelos SoaML. (11)

La metodología MDSE proporciona directrices para el uso de SoaML para definir y especificar una arquitectura orientada a servicios, tanto desde el negocio y una perspectiva de Tecnologías de la Información. La metodología establece la construcción de un conjunto de objetos siguiendo el modelo iterativo y paradigma de proceso incremental. La Figura 7 muestra el proceso general e identifica el conjunto de artefactos del modelo para especificar. La figura muestra el conjunto de productos de trabajo establecidos por la metodología y el flujo de trabajo general. Los iconos indican el Modelo de Motivación de Negocio (BMM), Notación de Modelamiento de Procesos de Negocio (BPMN) o esquema(s) SoaML asociados para cada producto del trabajo y las flechas indican el camino más común a través del conjunto de productos de trabajo dentro de una iteración. (11)

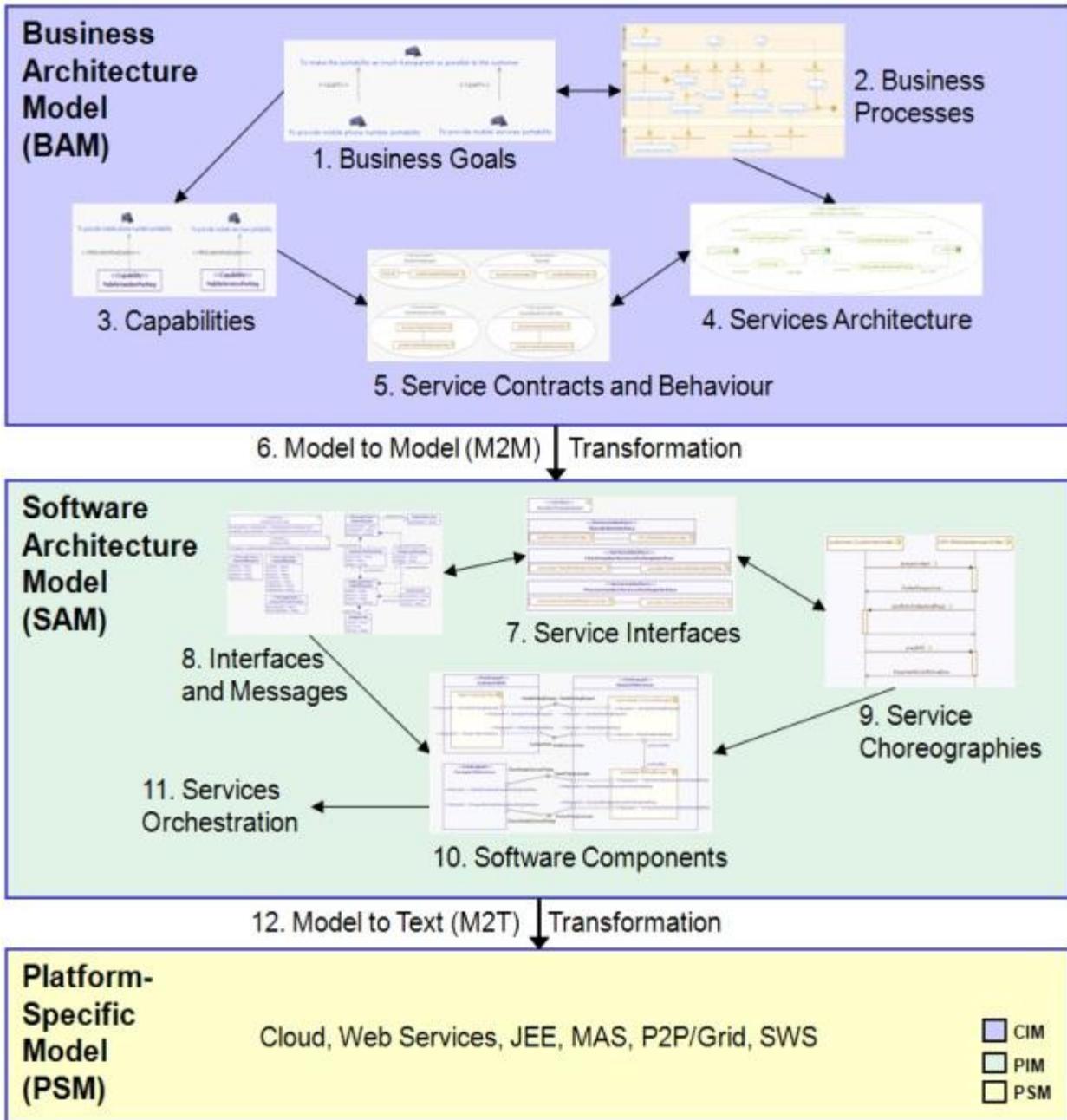


Figura 7. El proceso general basado en modelos. (11)

A partir de la parte superior izquierda tenemos el *Modelo de Arquitectura de Negocio (BAM)*, que incluye los objetivos de negocio, procesos de negocio, las capacidades, la arquitectura de servicios, contratos de

servicios y el comportamiento. *El Modelo de Arquitectura de Software (SAM)* especifica las interfaces y tipos de mensajes, interfaces de servicios y el comportamiento, componentes y puertos. La *transformación de modelo a modelo (M2M)* se compone de reglas de transformación y directrices de procedimiento para apoyar una asignación semi-automática del BAM para SAM. (11)

El *Modelo Específico de la Plataforma (PSM)*, contiene los objetos de diseño e implementación de la especificada arquitectura orientada a servicios en las plataformas tecnológicas elegidas, por ejemplo, Nube, Servicios Web, Java Enterprise Edition (JEE), sistemas multi-agente (MAS), igual a igual (P2P), la red y los Servicios Web Semánticos (SWS). Consideramos las directrices de modelado a nivel del PSM fuera del alcance de la presentación de la metodología en este capítulo, por lo que el mismo se centrará en la arquitectura empresarial y modelado de arquitectura de servicios. Las extensiones de la metodología a nivel del PSM implicaría la definición de nuevas pautas de modelado y de las reglas de *transformación de modelo a texto (M2T)* de las plataformas tecnológicas. (11)

1.4.1 Modelo de Arquitectura de Negocio (BAM)

Esta parte de la metodología cubre las áreas seleccionadas de los modelos que resultan a nivel del CIM en un *Modelo de Arquitectura de Negocio (BAM)* que describe el punto de vista comercial de una arquitectura orientada a servicios. El Modelo de Arquitectura de Negocios se utiliza para expresar las operaciones comerciales y el medio ambiente que la arquitectura orientada a servicios consiste en apoyar. El Modelo de Arquitectura de Negocios incluye los objetivos de negocio, procesos de negocio asociados con las funciones de la organización y los elementos de información y capacidades que son relevantes para la captura de requisitos del negocio e identificar los servicios dentro de una arquitectura orientada a servicios. EL Modelo de Arquitectura de Negocios además describe la arquitectura de servicios de la comunidad empresarial y los contratos de servicios entre las entidades empresariales que participan en la comunidad. La siguiente figura muestra un diagrama de actividad que muestra las tareas de modelado involucrados en la especificación del Modelo de Arquitectura de Negocios. (11)

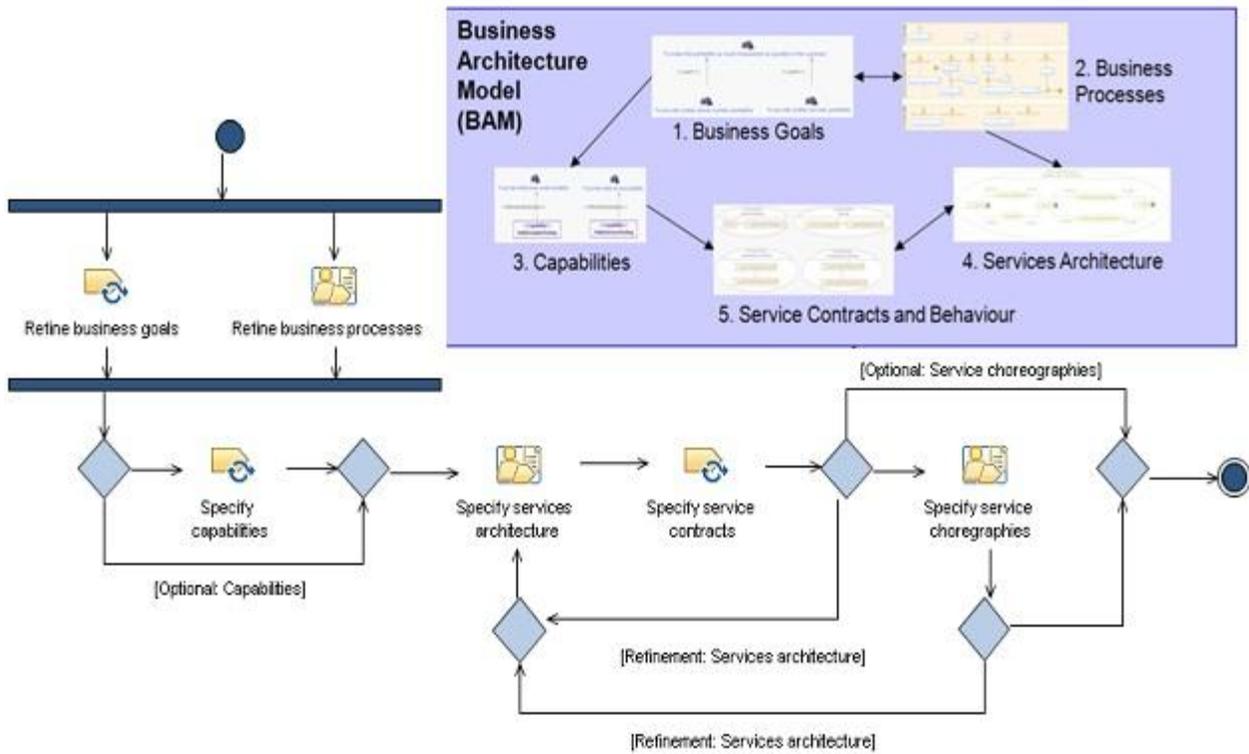


Figura 8. Actividades del modelado de arquitectura de negocio. (11)

1.4.2 Modelo de Arquitectura de Software (SAM)

El Modelo de Arquitectura de Software (SAM) describe la arquitectura general del sistema a nivel PIM. Representa las perspectivas de las Tecnologías de la Información de una Arquitectura Orientada a Servicios y particiona el sistema en componentes de software e interfaces. Un modelo estructural describe los componentes, sus dependencias y sus interfaces: las interfaces de servicios, interfaces, mensajes y los componentes de software. Un modelo dinámico describe las interacciones de los componentes y protocolos: coreografías de servicios y orquestaciones de servicios. (12)

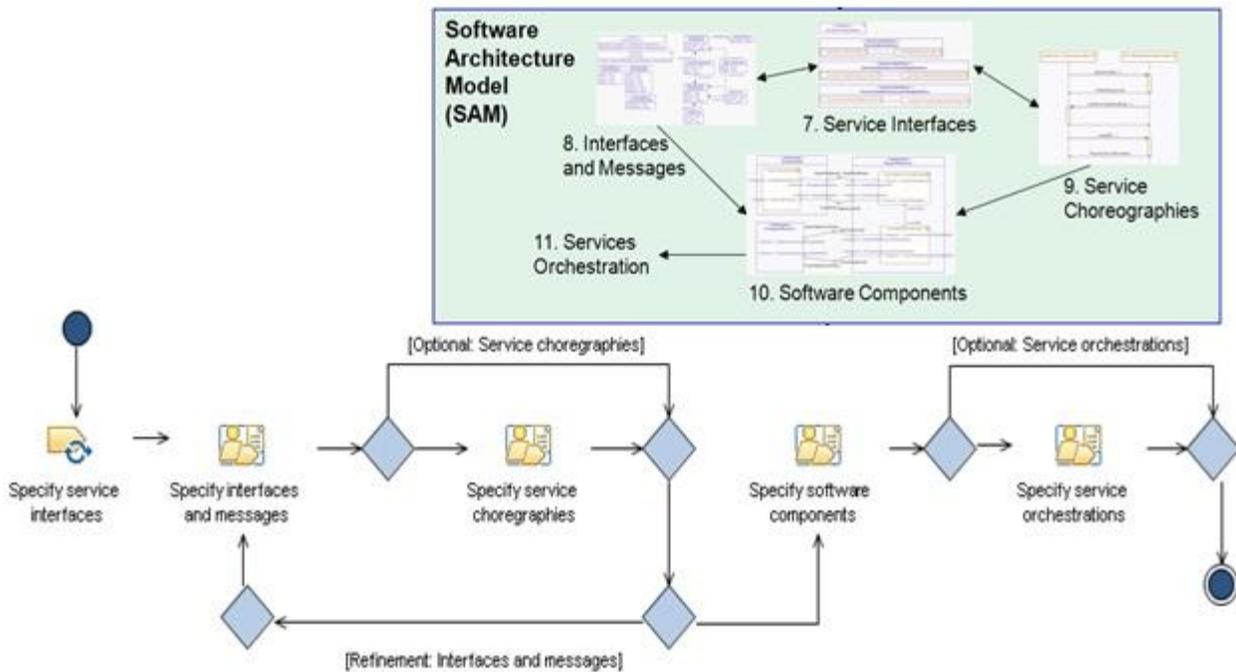


Figura 9. Actividades del modelado de arquitectura de software. (11)

1.4.3 Modelo Específico de la Plataforma (PSM)

El *Modelo Específico de la Plataforma (PSM)*, contiene los objetos de diseño e implementación de la especificada arquitectura orientada a servicios en las plataformas tecnológicas elegidas, por ejemplo, Nube, Servicios Web, Java Enterprise Edition (JEE), sistemas multi-agente (MAS), igual a igual (P2P), la red y los Servicios Web Semánticos (SWS). (11)

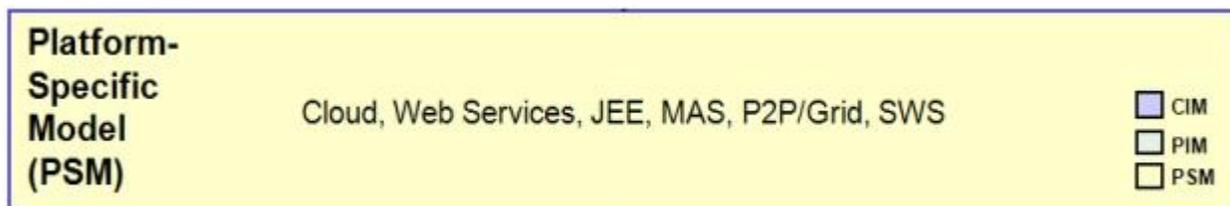


Figura 10. Modelo Específico de la Plataforma (PSM). (11)

1.5 Implementación de las Soluciones SoaML con MDA

Por su diseño, SoaML no es una solución completa de SOA. SoaML ofrece la posibilidad de un modelo de arquitectura orientada a servicios en la empresa, el sistema y los sistemas de los niveles de los sistemas. La mayoría de los usuarios requieren servicios de tecnología para apoyar y poner en práctica su arquitectura. Usando las técnicas del Model Driven Architecture ® (MDA ®) de OMG, los modelos SoaML se pueden utilizar para implementar soluciones SOA en la parte superior de la popular tecnología de estándares SOA, tales como Servicios Web, Enterprise Service Buses, servidores de aplicaciones y suites de Ejecución de Procesos Empresariales. (3)

Cuando se combina con herramientas MDA e infraestructuras de apoyo, servicios y componentes de los servicios definidos en SoaML pueden ser parte del "código fuente" para las implementaciones de servicio, proporcionando un ambiente de alto nivel en el desarrollo, donde las soluciones de nivel empresarial se pueden desarrollar de forma rápida y eficiente desde la arquitectura. (3)

Mediante la integración de los procesos de arquitectura, diseño e implementación utilizando MDA, las soluciones se pueden crear con mayor rapidez, son más fáciles de mantener y duran más. Debido a que las arquitecturas de SoaML son independientes de la tecnología, la "próxima gran tecnología" puede ser adoptada más rápidamente, basada en la arquitectura misma. (3)

1.6 Conclusiones

Producto que el desarrollo de software se puede caracterizar como la sucesión de incrementos en el nivel de abstracción, la cual es caracterizada por el uso de lenguajes de desarrollo con un mayor nivel construidos sobre otros ya existentes y las respectivas herramientas encargadas de automatizar la transformación de programas escritos en un lenguaje de alto nivel en programas escritos en lenguaje de menor nivel. Todo esto hace que la evolución se dirija hacia la utilización de lenguajes de modelado y la iniciativa MDA cuenta con las bases necesarias para conseguir ese paso evolutivo, al proponer como se expuso anteriormente, la definición de las aplicaciones mediante modelos independientes de la plataforma que luego se refinan en modelos específicos para una plataforma a partir de los cuales se obtendría el código de la aplicación. Esta iniciativa, sin embargo, debe realizarse mediante la provisión de herramientas que permitan la creación, edición y validación de dichos modelos y que automaticen las transformaciones entre ellos.

2. Capítulo 2: Propuesta de Solución

En este capítulo se realiza la propuesta de solución mediante análisis de los principales marcos tecnológicos para soluciones SoaML y su integración con MDA de tal manera que contribuya a la generación automática de artefactos derivados, teniendo en cuenta los aspectos más importantes a la hora de seleccionar una herramienta, lo que de conjunto con el estudio de los casos de éxito existentes en cuanto a la utilización de las mismas permitirá contar con los elementos necesarios para plantear los escenarios de negocio propicios para implementar dicha solución conjuntamente con el marco tecnológico a utilizar.

2.1 Selección de la Herramienta de Modelado

De las herramientas de modelado que soportan SoaML con la que se va a trabajar es con Modelio, de la empresa **SOFTEAM**.

SOFTEAM proporciona una extensión de Modelio llamada SoaMLDesigner, dicho componente es opensource. Incluye las últimas especificaciones del lenguaje SoaML e implementa una interfaz gráfica (GUI) que incluye 6 diagramas: diagrama de Capacidades, de Contrato del Servicio, de Arquitectura del Servicio, de Mensajes y de Participantes. SoaML está integrado con la metodología de SOFTEAM para arquitecturas empresariales. Esta solución integra Modelos de Negocio (Objetivos, Reglas de Negocio, Requisitos), Procesos de Negocio (BPMN), SoaML y las transformaciones a varios modelos de implementación que incluye XSD, WSDL, BPEL y Java. Dichas transformaciones son realizadas con el módulo SoaML Engine de Modelio, permitiendo la integración satisfactoria de SoaML con MDA. (5)

El SoaML Designer de Modelio proporciona los editores dedicados que ayudan al diseñador en la construcción de modelos SoaML (29):

- ✓ Modelado de la arquitectura SOA
- ✓ Especializados editores gráficos
- ✓ Generación de modelos de implementación de la arquitectura

Otras herramientas también permiten modelar servicios con SoaML, pero el aspecto principal a la hora de hacer la selección lo constituye la transformación de los modelos de diseño de SoaML a los modelos de implementación usando MDA. En este aspecto Modelio tiene una gran ventaja pues permite hacerlo satisfactoriamente brindando muchas facilidades a través de sus módulos para el diseño de modelos SoaML y su posterior transformación. Debido a esto la mayor experiencia acumulada es con dicha herramienta, la cual es ampliamente utilizada y recomendada por el estándar SoaML en casi toda la documentación asociada y una gran mayoría de ejemplos existentes en dicha documentación son con estereotipos y artefactos generados con dicha herramienta. Por todo lo anteriormente dicho la decisión es por la herramienta de modelado Modelio.

2.2 Modelado de Servicios en la Metodología del CDAE

Actualmente en el CDAE existe una línea de investigación y desarrollo que se interesa por las soluciones orientadas a servicios. Para el desarrollo de proyectos con este enfoque se está elaborando una metodología de desarrollo que tiene varios procedimientos y artefactos definidos, los cuales se encuentran asociados a la documentación de un proceso de desarrollo SOA. Dicha metodología consta de una serie de disciplinas de las cuales se han identificado las dos disciplinas donde SoaML permite realizar el modelado de servicios, en conjunto con su posterior integración con las técnicas de generación de MDA, dándole solución al problema en cuestión.

Las disciplinas identificadas en la metodología son: *Arquitectura y diseño de las soluciones* y *Especificación y Diseño de Servicios*.

A continuación se muestra en la Figura 11, la metodología de desarrollo SOA del CDAE con sus respectivas disciplinas asociadas y las relaciones que se establecen entre las mismas. En la figura se señalan además las disciplinas identificadas para el modelado de servicios. Más adelante se hará referencia a cada una de las disciplinas identificadas en particular, detallando la utilización del estándar SoaML en cada actividad definida, los diagramas a realizar así como los diferentes estereotipos a utilizar y artefactos generados.

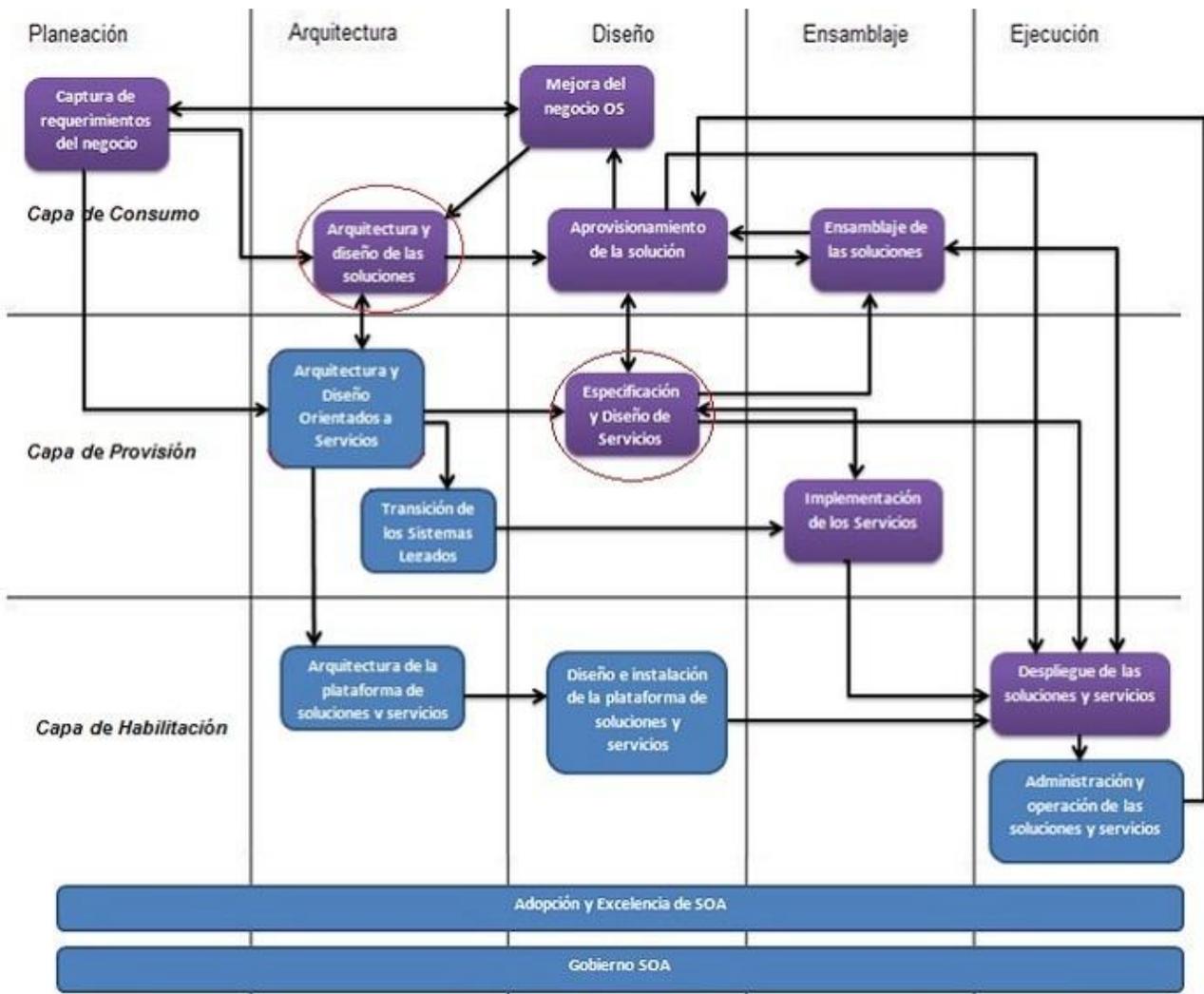


Figura 11. Metodología de Desarrollo SOA del CDAE.

2.2.1 Arquitectura y diseño de las soluciones

Identificar servicios

En esta actividad se identifican los servicios necesarios para realizar los PN en desarrollo. 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 piscina (Pool) en el PN.

Esta actividad se modela con la especificación Arquitectura de Servicios (Services Architecture, SA) de SoaML, 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 Figura 12 muestra las diferentes piscinas (Pool) en un diagrama BPMN.

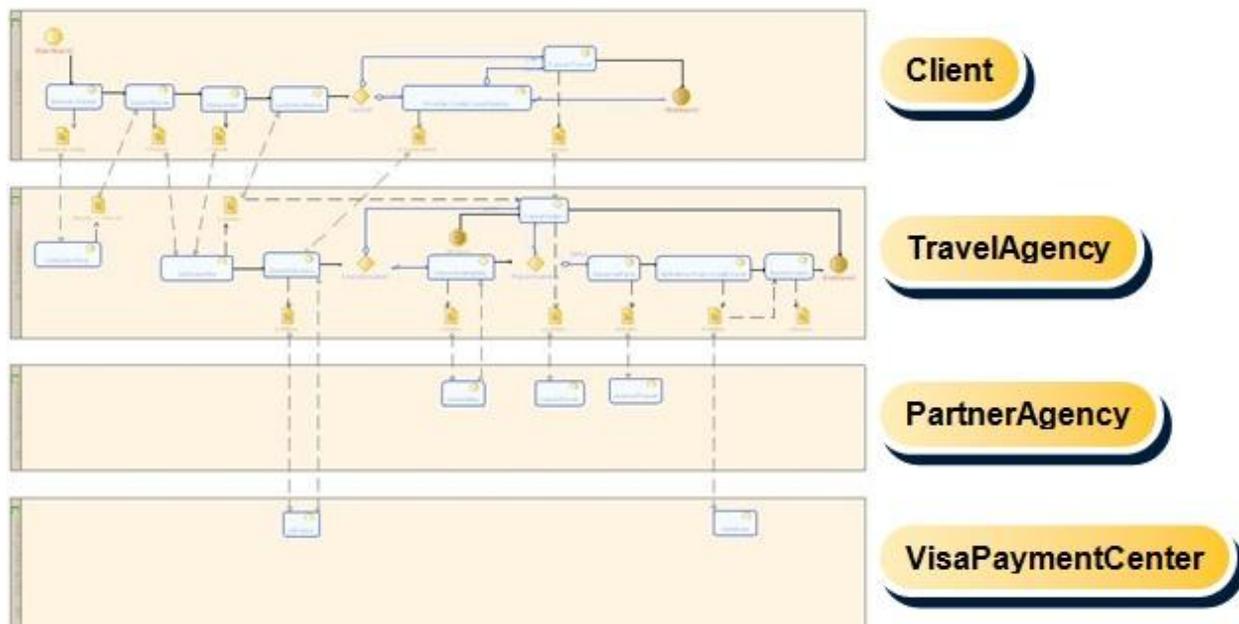


Figura 12. Piscinas en un diagrama BPMN.

Las piscinas del modelo de PN (definición de Pool en BPMN) se corresponden con Participantes en SoaML, cada piscina puede ser asignada a un participante en el diagrama de Arquitectura de Servicios.

Los servicios se identifican con las actividades identificadas para soportar servicios, a las que se les indica su tipo como *Service*, o sea, que sean automatizables, y con los mensajes intercambiados entre las piscinas, es decir, las interacciones o intercambio de mensajes entre las piscinas en el PN, aquellas que se puedan automatizar se clasifican en contratos de servicios. 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.

La Figura 13 muestra como ocurre el intercambio de mensajes entre piscinas en un diagrama BPMN, estas interacciones son las que se clasifican en contratos de servicio.

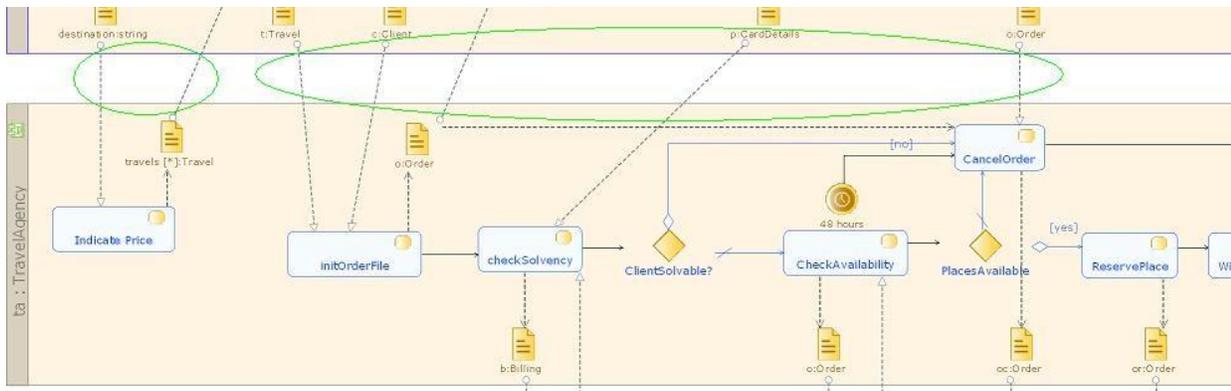


Figura 13. Intercambio de mensajes entre piscinas en un diagrama BPMN.

Teniendo en cuenta la dirección de los mensajes podemos decir 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 piscina 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.

Arquitectura de Servicios: Propósito

Una **arquitectura de servicios** en SoaML es una descripción de alto nivel de cómo los participantes trabajar juntos por un fin, el suministro y uso de los servicios expresados en los **contratos de servicios**.

La arquitectura de servicios define los requisitos de los tipos de participantes y realizaciones de servicios que cumplen funciones específicas.

Un rol define la función básica (o conjunto de funciones) que una entidad puede llevar a cabo en un contexto particular.

Tanto los contratos de servicios y los participantes pueden ser reutilizados para componer los diferentes servicios en las arquitecturas de otros servicios.

Arquitectura de Servicios: Pasos de modelización

Arquitectura de Servicios:

- Colaboración UML estereotipada «ServicesArchitecture».
- Identificados a partir de los procesos BPMN.

Los participantes:

- Clases UML estereotipadas «Participant».
- Identificados a partir de las piscinas, los participantes y los carriles específicos en los procesos BPMN.

Los contratos de servicios:

- Colaboraciones UML estereotipados «ServiceContract».
- Identificados a partir de las posibles interacciones entre los diferentes participantes en los procesos BPMN.

Especifican la arquitectura de servicios:

- Utilizar los **contratos de servicios** y los **participantes** para construir la arquitectura de servicios.
- Los **roles** en la colaboración de UML se escriben por los **participantes** identificados.
- Enlazar los diferentes roles a los **usos de colaboración** apropiados, especificando cómo los participantes van a interactuar.

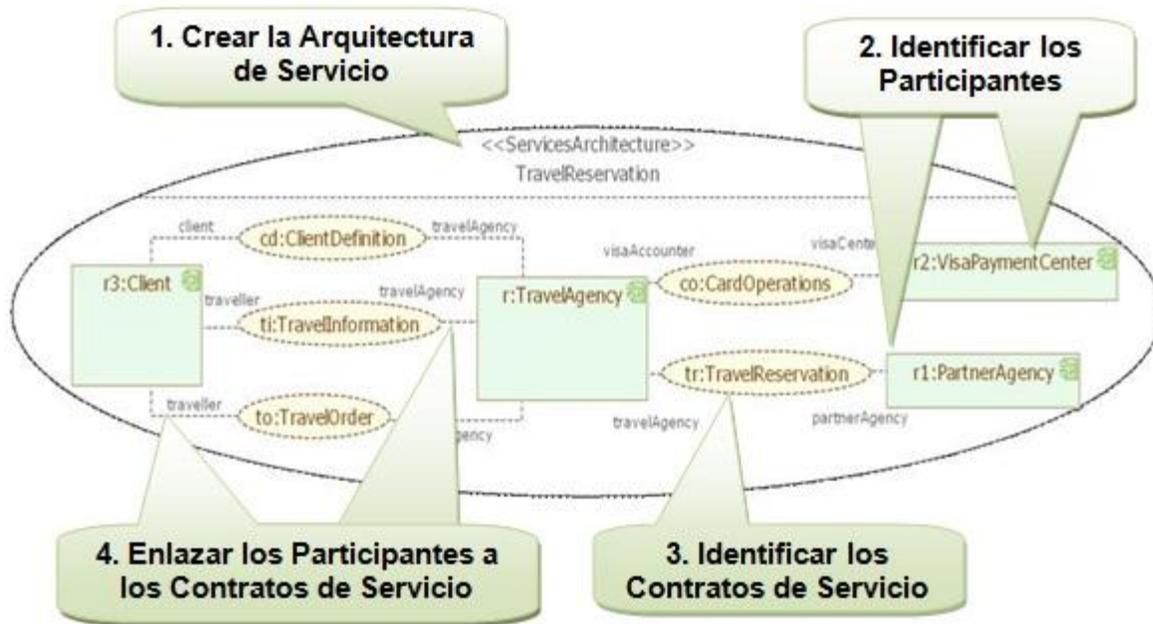


Figura 14. Modelado de una Arquitectura de Servicios.

2.2.2 Especificación y Diseño de Servicios

Especificar servicios

En esta actividad se 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. 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 ejemplo, 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.

SoaML soporta diferentes enfoques para SOA. La especificación establece una distinción entre tres enfoques diferentes para la especificación de un servicio:

- El enfoque basado en interfaz simple utiliza una interfaz UML para especificar una interacción de servicio de un solo sentido.
- El enfoque basado en contrato de servicio extiende a una colaboración UML para especificar un servicio de interacción binaria.
- El enfoque basado en interfaz de servicio extiende a una clase UML para especificar un servicio de interacción binaria.

Ambos enfoques, el basado en contrato de servicio y en interfaz de servicio implican la especificación de interfaces simples, por lo general una para cada uno de los roles que participan en la interacción del servicio. Así, un contrato de servicio o una interfaz de servicio pueden ser vistos como una extensión del enfoque basado en interfaz simple.

El enfoque basado en interfaz simple centra la atención en una interacción unidireccional proporcionada por un participante en un puerto representado como una interfaz UML. El participante, en este caso el proveedor, recibe las operaciones en este puerto y puede proporcionar los resultados a la persona que llama, en este caso el consumidor.

La Figura 15 muestra la especificación de este servicio, que consiste en la interfaz de proveedor *TravelOrder* (modelado como una interfaz UML «Provider») y los tipos de mensaje *Order*, *CardDetails*, *Travel* y *Client* (modelado como clases UML «MessageType»). Los tipos de mensajes representan los tipos de los parámetros de entrada de las operaciones *cancelOrder* y *payOrder*, así como los tipos de los parámetros de entrada y el tipo de retorno de la operación *createOrder*, todas estas operaciones definidas en la interfaz del proveedor.

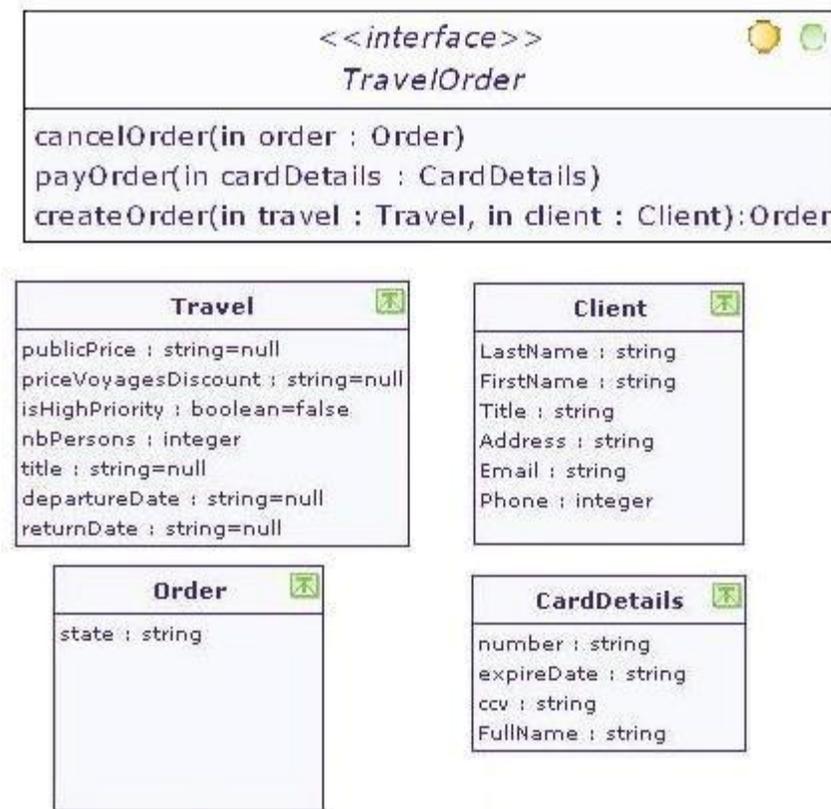


Figura 15. Interfaz del Proveedor y Tipos de Mensaje de un servicio.

Contrato de Servicios: Propósito

El enfoque basado en contrato de servicio define las especificaciones del servicio que definen los roles que cada participante desempeña en el servicio (como el proveedor y el consumidor) y las interfaces que implementan para desempeñar ese papel en ese servicio. Estas interfaces son entonces los tipos de puertos en el participante, que obliga al participante a ser capaz de desempeñar ese papel en ese contrato de servicio.

La especificación de los contratos de servicios puede ser visto como el perfeccionamiento de una arquitectura de servicios. El enfoque basado en contrato de servicio extiende una colaboración UML para modelar la parte estructural de la interacción de servicio.

La figura 16 muestra la especificación del contrato de servicios *TravelOrder*, con los dos roles, el consumidor y el proveedor y la interfaz de tipo proveedor *TravelOrder*.



Figura 16. Modelado de un Contrato de Servicios con Interfaz del Proveedor.

Contrato de Servicios: Pasos de modelización

Analizar los diagramas BPMN para identificar a los contratos de servicios:

- Para cada contrato de servicio identificado crear una colaboración UML con el estereotipo <<ServiceContract>>.

Identificar los roles:

- Especificar los proveedores y los consumidores del contrato de servicio.
- Se especifican los roles escritos por una interfaz.
- Los roles están conectados por un canal de servicio.

Especificar las interfaces a nivel de negocio:

- Especificar las operaciones de negocio utilizados en la interfaz. En esta etapa de modelado que sólo identifican los nombres y, posiblemente, algunas operaciones de alto nivel en las interfaces.

Interfaz de Servicio: Propósito

El enfoque basado en interfaz de servicio es muy similar al enfoque basado en el contrato de servicio que también se centra en las interacciones de servicio binario, obligándonos a especificar una serie de interfaces relacionadas como una especificación de servicio.

Una interfaz de servicio define las interfaces y las responsabilidades de un participante para proporcionar o consumir un servicio. Las interfaces de servicios perfeccionan los contratos de servicios.

Un enfoque basado en interfaz simple centra la atención en una interacción unidireccional proporcionada por un participante en un puerto representado como una interfaz UML «Proveedor».

El enfoque basado en interfaz de servicio introduce el concepto de una interfaz de servicio que permite servicios bidireccionales, con "devoluciones de llamada" desde el proveedor hasta el consumidor y el concepto de una interfaz de servicio conjugado para escribir los puertos en el proveedor y el lado del consumidor, respectivamente.

La figura 17 muestra la especificación del servicio TravelOrder previamente modelada como un contrato de servicio utilizando ahora el enfoque basado en interfaz de servicio en su lugar.

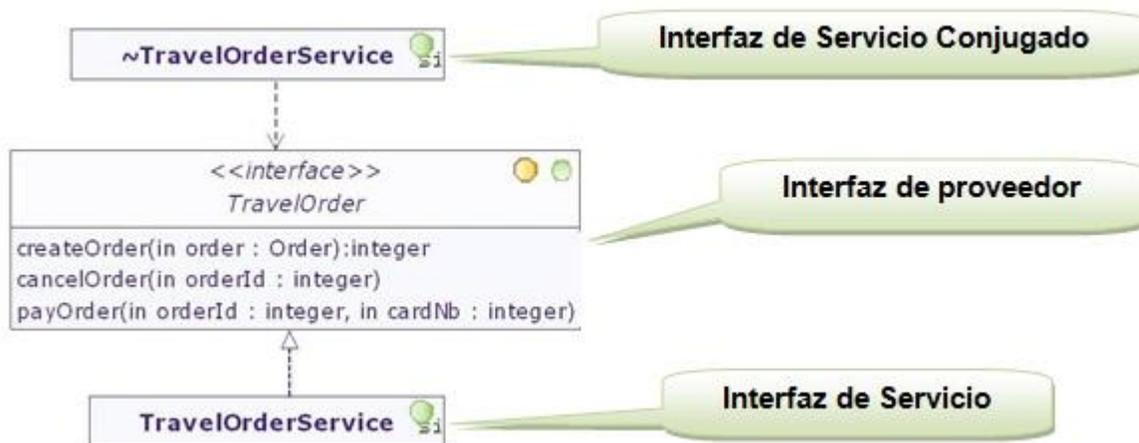


Figura 17. Modelado de una Interfaz de Servicio.

Interfaz de Servicio: Pasos de modelización

- Definir las **interfaces de servicio**. Interfaz UML estereotipada «ServiceInterface». Mapeo uno-a-uno entre los contratos de servicios y las interfaces de servicio.
- Definir la **interfaz proporcionada**. Interfaz UML estereotipada «Provider».

- Vincular la **interfaz prevista** a la **interfaz de servicio**. Crear una **interfaz de realización** entre la interfaz de proveedor y la interfaz de servicio.
- Crear y vincular una **interfaz de servicio conjugado**. Crear una interfaz de servicio conjugado como un «ServiceInterface» que **utiliza** la **interfaz del proveedor**.

El nombre de la interfaz de servicio conjugado se inicia con "~", y representa a la contra-parte de una interfaz de servicio y se utiliza para escribir el puerto en el que el participante desempeña el papel de los consumidores.

Generar WSDL de servicios

En esta actividad constituye otro pilar de la disciplina Especificación y Diseño de Servicios y representa la fase final para la descripción y obtención de los artefactos generados en el diseño. Su objetivo principal es la obtención del Lenguaje de Descripción de Servicio Web (WSDL) de cada servicio identificado y especificado en dichas disciplinas para su posterior implementación.

La figura 18 muestra la interfaz del servicio *TravelOrder* previamente modelada y la relación con su WSDL correspondiente generado usando la herramienta escogida Modelio. La herramienta permite a partir de las interfaces de los diferentes servicios, generar sus WSDL correspondientes utilizando las técnicas de generación de MDA y posteriormente permite exportar ese WSDL generado en la herramienta a un fichero de tipo .wsdl en una dirección física del ordenador. En la figura se ven las diferentes partes del WSDL generado para el servicio *TravelOrder*. Tener en cuenta que el vocabulario en el WSDL viene directamente de la arquitectura utilizando la automatización de MDA.

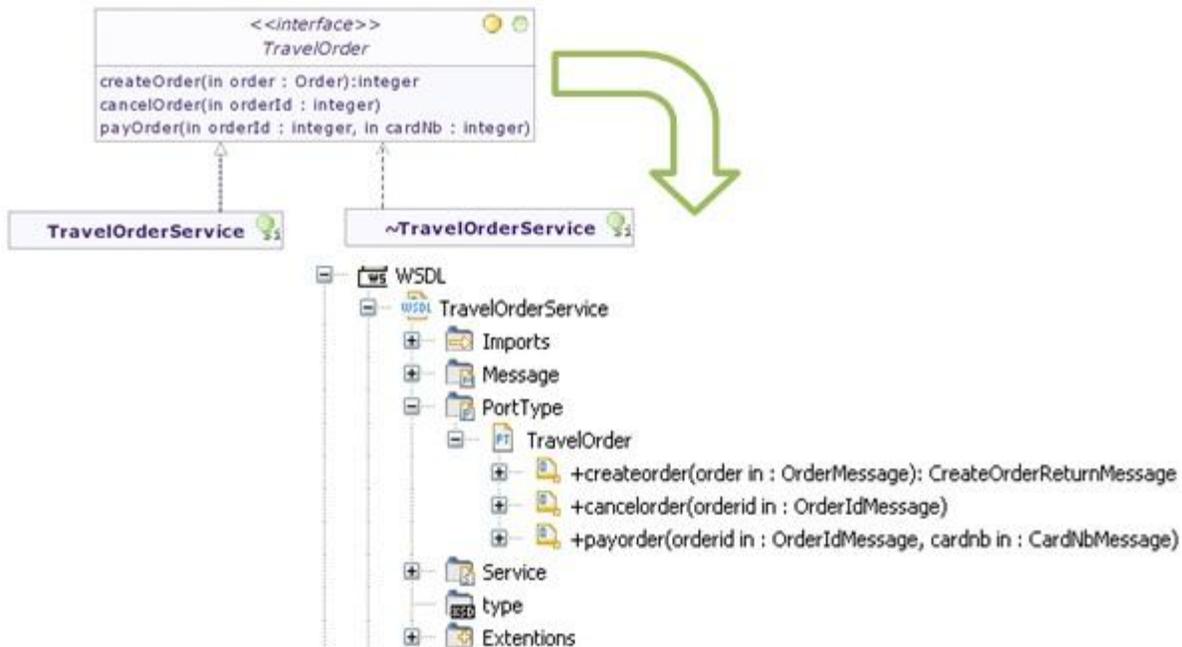


Figura 18. WSDL generado a partir de la Interfaz de Servicio.

2.3 Conclusiones

Para el desarrollo de la propuesta de solución, se propuso una serie de actividades para el modelado de servicios, que abarcaron la identificación, especificación y generación automática del WSDL de los servicios web, definiendo así un procedimiento de modelado de servicios utilizando el estándar SoaML para el modelado de los servicios y las técnicas de MDA para las transformaciones de estos modelos, proporcionando un ambiente de alto nivel en el desarrollo, donde las soluciones se pueden desarrollar de forma rápida y eficiente desde la arquitectura, describiendo así los artefactos generados en el diseño y contribuyendo de esa forma a minimizar los retrasos existentes, dándole cumplimiento al objetivo principal de la investigación.

3. Capítulo 3: Validación de la Solución Propuesta

Como cumplimiento a una de las tareas propuestas para la presente investigación, se decide someter la propuesta a la valoración de un conjunto de expertos para su aprobación y/o sugerencias. Se emplearon las técnicas propuestas por el método Delphi. El cual es el método subjetivo de pronosticación más confiable, su origen se remonta a los inicios de los años 50 por RAND Corporation (Centro de Investigación y Desarrollo de las Fuerzas Armadas), ideado por Olaf Helmer y Theodore J. Gordon como un instrumento para realizar predicciones sobre un caso de catástrofe nuclear, desde entonces ha sido utilizado para obtener información sobre el futuro (13). Se dice que su nombre se inspira en el oráculo de Delphos, al que se acudía para realizar preguntas a una sacerdotisa; muy famosa por aquella época por la certeza de sus predicciones a pesar de ser ambigua en sus respuestas.

Linstone y Turoff (14) definen la técnica Delphi como un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo.

3.1 Método Delphi

El Método Delphi es un método experto que se basa en la consulta a personas que tienen grandes conocimientos sobre el entorno en el que la organización desarrolla su labor. Estas personas exponen sus ideas y finalmente se redacta un informe en el que se indican cuáles son, en su opinión, las posibles alternativas o sucesos que se obtendrán en el futuro.

La calidad de los resultados obtenidos por este método experto dependen de:

- La elaboración de los cuestionarios.
- Las predicciones de los expertos consultados.

El Delphi es pronosticado como uno de los métodos más fiables porque se basa en la interrogación a expertos con la ayuda de cuestionarios sucesivos, con el fin de encontrar convergencias en las opiniones y así deducir eventuales consensos. Se basa en la consulta de un grupo de expertos de forma individual

por medio de un conjunto de preguntas bien conformadas, que apoyadas por los resultados promedio de la ronda anterior, genera coincidencia de opiniones.

Algunas de las ventajas que ofrece el Método Delphi son:

- Permite la formación de un criterio con mayor grado de objetividad y el consenso logrado sobre la base de los criterios es muy confiable.
- La tarea de decisiones sobre la base de los criterios de expertos, obtenido por éste, tiene altas probabilidades de ser eficiente.
- Permite valorar alternativas de decisión.
- Un requisito imprescindible para garantizar el éxito del método, evitar conflictos entre expertos y crear un clima favorable a la creatividad, es ser anónimo.
- El experto se siente involucrado plenamente en la solución del problema y facilita su implantación.
- De ello es importante el principio de voluntariedad del experto en participar en la investigación y la confidencialidad de su opinión.

El método presenta 4 características principales:

- **Anonimato:** Ningún experto conoce la identidad de los otros que componen el grupo de debate.
- **Iteración y retroalimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Como se van presentando los resultados obtenidos de los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- **Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.
- **Heterogeneidad:** Pueden participar expertos de determinadas ramas sobre las mismas bases.

El criterio de expertos puede ser tomado mediante encuestas o entrevistas y empleado en cualquier momento de la investigación, cuya experiencia y opiniones pueden ser de una valiosa contribución, resultando fundamental durante el estudio exploratorio.

Por lo anteriormente mencionado, se determinó usar el uso del método Delphi, en este caso la variante propuesta por Silvia Colunga y Georgina Amayuela (15) a su vez empleada por el Lic. Carlos Álvarez Martínez de Santelices (16) en su tesis de maestría: "Experimentos virtuales para la enseñanza del Electromagnetismo" y el Ing. Rolando Quintana Aput en su tesis de maestría "Propuesta de Indicadores para medir competencias del personal según el rol en proyectos multimedia" (17); donde aparecen las conclusiones del estudio de numerosas tesis de maestría y doctorado para ese tipo de investigación. Dichos investigadores no utilizan el método clásico Delphi para la valoración de expertos, sino algunas características para propiciar mayor objetividad a los criterios de los especialistas a partir de la introducción de escalas valorativas.

Básicamente consta de tres fases:

- Selección de los expertos.
- Elaboración del cuestionario para validación de la propuesta.
- Desarrollo práctico y explotación de los resultados.

A continuación se explica cómo fue desarrollado el método en la presente investigación.

3.1.1 Selección de los expertos

Se considerará un experto aquella persona que sea capaz de brindar valoraciones conclusivas del proceso de creación de una Arquitectura Orientada a Servicios y de hacer recomendaciones con un determinado coeficiente de competencia.

En base a esto y bajo los siguientes criterios se realizó la selección de expertos:

- Graduado de nivel superior.
- Vinculación a la investigación y desarrollo de la línea BPM/SOA.

- Un año de experiencia como mínimo.
- Capacidad de análisis y pensamiento lógico.
- Disposición para participar en la validación.

Otro aspecto importante en la selección de los expertos es el número de expertos que debe tener el grupo. No existe una norma generalizada para determinar el número óptimo de expertos, sin embargo hasta siete expertos el error disminuye exponencialmente, después de treinta, aunque el error disminuye lo hace de manera poco significativa y no compensa el incremento de costos y esfuerzo, por lo que se sugiere utilizar un número de expertos en el intervalo de siete a treinta (18).

Los posibles candidatos se buscaron en la UCI, teniendo en cuenta su experiencia en este campo y además con experiencia en la investigación de BPM/SOA, por lo que todos fueron especialistas pertenecientes al CDAE. El siguiente paso fue la obtención del consentimiento de los expertos a participar en la validación, confeccionándose así un listado de siete expertos. Luego se les aplicó una encuesta de autovaloración para determinar el coeficiente de competencia, el listado con los expertos seleccionados se muestra en el Anexo 3. La encuesta de autovaloración aplicada puede encontrarse en el Anexo 1.

Para la selección de los expertos es muy acertado emplear la valoración por competencias. Este método consiste en hallar el coeficiente de competencia del experto a partir de la autovaloración sobre su nivel de conocimiento sobre el tema y el coeficiente de argumentación o valoración, mediante el uso de la siguiente ecuación:

$$k = \frac{(kc + ka)}{2} \quad (1)$$

Dónde:

k: es el coeficiente de competencia del experto.

kc: es el nivel de conocimiento del experto sobre el tema.

ka: es el coeficiente de argumentación o valoración del experto.

Para el cálculo del coeficiente de competencia se procedió de la siguiente manera:

Capítulo 3: Validación de la Solución Propuesta

1. Se envió a cada experto el cuestionario de autovaloración.
2. Se procedió a la tabulación de los datos del formulario.

El coeficiente de conocimiento sobre el tema (k_c) se obtiene de la primera tabla que aparece en el formulario del Anexo 1, recoge una autoevaluación del posible experto.

Tabla 1. Tabla para obtener el coeficiente de conocimiento.

0	1	2	3	4	5	6	7	8	9	10
							x			

En la misma, el presunto experto marcará en una de las casillas el grado de conocimiento que presenta sobre la temática que se ha puesto a su consideración, en una escala del 0 al 10 y después para ajustarla a la teoría de las probabilidades se multiplica por 0,1. De esta manera una evaluación de 0 implica que el experto no tiene ningún conocimiento sobre la materia y una de 10 que significa que el experto tiene pleno conocimiento. En la Tabla 1 el experto $1\ k_c = 0,7$.

Para calcular el coeficiente de argumentación (k_a), se procede de la siguiente manera: En el formulario se ofrece una tabla con la siguiente información:

Tabla 2. Tabla para calcular el coeficiente de argumentación.

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.			
2.-	Experiencia.			
3.-	Trabajos de autores nacionales.			
4.-	Trabajos de autores extranjeros.			
5.-	Su propio conocimiento del tema.			
6.-	Su intuición.			

En esta tabla el experto debe marcar, según su consideración, cuáles fueron sus fuentes para la obtención del conocimiento que le permite argumentar su evaluación del nivel de conocimiento que

Capítulo 3: Validación de la Solución Propuesta

especificó en la Tabla 1. Las marcas de los expertos se traducen a puntos según la siguiente escala:

Tabla 3. Escala de puntos para la determinación del (ka).

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.	0,3	0,2	0,1
2.-	Experiencia.	0,5	0,4	0,2
3.-	Trabajos de autores nacionales.	0,05	0,05	0,05
4.-	Trabajos de autores extranjeros.	0,05	0,05	0,05
5.-	Su propio conocimiento del tema.	0,05	0,05	0,05
6.-	Su intuición.	0,05	0,05	0,05
	Totales	1,0	0,8	0,5

Con estos elementos se obtiene el coeficiente de argumentación (ka). Si el presunto experto del ejemplo anterior, seleccionó los siguientes aspectos:

Tabla 4. Ejemplo ilustrativo de las selecciones de un experto dado.

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.	x		
2.-	Experiencia.		x	
3.-	Trabajos de autores nacionales.	x		
4.-	Trabajos de autores extranjeros.		x	
5.-	Su propio conocimiento del tema.	x		
6.-	Su intuición.	x		

En este ejemplo, el coeficiente de argumentación sería: $k_a = 0,3 + 0,4 + 4(0,05) = 0,9$

Para este caso, el coeficiente de competencia (k), quedaría de la siguiente manera:

$$k = \frac{(k_c + k_a)}{2}$$

$$k = \frac{(0,7 + 0,9)}{2}$$

$$k = 0,8$$

Para interpretar los resultados de dicho test, es necesario tener en cuenta que:

- Si $0,8 < k < 1,0$ el coeficiente de competencia es Alto
- Si $0,5 < k < 0,8$ el coeficiente de competencia es Medio
- Si $k < 0,5$ el coeficiente de competencia es Bajo

Se seleccionaron los que presentaban un coeficiente Medio y Alto.

En la siguiente tabla se muestran los resultados obtenidos en la encuesta de autovaloración:

Tabla 5. Resultados obtenidos en la Encuesta de Autovaloración.

Experto No.	kc	ka	k	Grado
1	0,4	0,8	0,6	Medio
2	0,7	0,6	0,65	Medio
3	0,6	0,8	0,7	Medio
4	0,7	0,9	0,8	Alto
5	0,6	0,8	0,7	Medio
6	0,5	0,7	0,6	Medio
7	0,6	0,7	0,65	Medio

Dado a que los siete expertos estuvieron de acuerdo en participar y todos tuvieron un coeficiente entre medio y alto, se decidió que el número de expertos del Comité de Expertos sería siete.

3.1.2 Elaboración del cuestionario para validación de la propuesta

Para validar la propuesta de Método de Modelado de Servicios basado en SOAML y MDA en el Proceso de Desarrollo para una Arquitectura Orientada a Servicios (SOA), se utilizó la encuesta mostrada en el

Anexo 2, entre sus objetivos se encontraba:

- Determinar la utilidad de la propuesta para dar solución a la problemática planteada en la presente investigación. Para ello se propusieron los puntos uno y dos.
- Determinar la efectividad de las actividades centrales del proceso propuesto. Para ello se propuso el punto dos.
- Identificar aspectos erróneos y/o recomendaciones que permitan mejorar la propuesta. Para ello, en cada pregunta formulada los expertos podían hacer sus observaciones y el punto tres estaba dedicado a dar una valoración general del proceso.

El cuestionario se elaboró de tal manera que las preguntas fuesen categorizadas (Muy Adecuado (C1), Bastante Adecuado (C2), Adecuado (C3), Poco Adecuado (C4), No Adecuado (C5)).

3.1.3 Desarrollo práctico y explotación de los resultados

Se confeccionarán tablas para ir recogiendo los resultados aportados por los expertos. Para ello se utilizó el programa Excel 2010. Dichos resultados se recogen a continuación:

Tabla 6. Tabla de Frecuencias Absolutas.

Tabla de Frecuencias Absolutas:							
No.	Elementos	C1	C2	C3	C4	C5	Total
1	Identificar servicios	3	4				7
2	Especificar servicios	2	5				7
3	Generar WSDL de servicios	4	3				7
Suma		9	12				

Tabulados los datos, se siguieron los siguientes pasos para la obtención de los resultados:

Primer paso: Se construye una tabla de frecuencias acumuladas. Cada número de fila, excepto la primera, se obtiene sumándole la anterior.

Tabla 7. Tabla de Frecuencias Absolutas Acumuladas.

Tabla de Frecuencias Absolutas Acumuladas:						
No.	Elementos	C1	C2	C3	C4	C5
1	Identificar servicios	3	7			
2	Especificar servicios	2	7			
3	Generar WSDL de servicios	4	7			

Segundo paso: Se construye la tabla de frecuencias relativas acumulativas.

Tabla 8. Tabla de Frecuencias Relativas Acumuladas.

Tabla de Frecuencias Relativas Acumuladas:						
No.	Elementos	C1	C2	C3	C4	C5
1	Identificar servicios	0,428571428	0,9999			
2	Especificar servicios	0,285714285	0,9999			
3	Generar WSDL de servicios	0,571428571	0,9999			

Los resultados de esta tabla se obtienen dividiendo por el número total de expertos, en este caso 7, el número que se encontraba en la tabla anterior.

Tercer paso: Se buscan las imágenes de los elementos de la tabla anterior por medio de la función (Dist. Normal. Estándar Inv).

A la misma tabla se le adicionan 3 columnas y una fila para colocar los resultados que se explican a continuación:

- Suma de las columnas.
- Suma de filas.
- Promedio de las columnas.
- Los promedios de las filas se obtienen de forma similar, en este caso también se divide por

cuatro porque quedan 4 categorías ya que la última se eliminó.

- Para hallar N, se divide la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.
- El valor N-P da el valor promedio que otorgan los expertos para cada indicador propuesto.

La tabla siguiente muestra cómo se procedió:

Tabla 9. Puntos de corte y Grado de Adecuación para cada criterio.

Puntos de corte:									
No	Aspectos	C1	C2	C3	C4	Suma	P	N-P	
1	Identificar servicios	-0,18	3,72			3,54	1,77	-1,06	Muy adecuado
2	Especificar servicios	-0,57	3,72			3,15	1,58	-0,87	Muy adecuado
3	Generar WSDL de servicios	0,18	3,72			3,90	1,95	-1,24	Muy adecuado
Suma		-0,57	11,16			10,59			
P.de corte		-0,19	3,72						

Las sumas obtenidas en las cuatro primeras columnas dan los puntos de corte. Estos se utilizan para determinar el grado de adecuación de los indicadores según los criterios de los expertos seleccionados. Para ello se opera del modo siguiente:

Tabla 10. Grado de Adecuación de los indicadores.

Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
-0,19	3,72			

Observación: El análisis de los indicadores propuestos, puede verse en el Anexo 4.

Después de la tabulación de los resultados obtenidos con la encuesta, todos los indicadores catalogados por el autor resultaron de muy adecuados. Por lo tanto todos se incluyen dentro del método propuesto y se valida así la adecuación de los mismos.

3.2 Conclusiones

En este capítulo se validó la solución propuesta, utilizando el método de validación Delphi. Se seleccionaron 7 expertos que categorizaron una serie de aspectos relacionados con el método planteado. Después de haber analizado los resultados, se puede afirmar que el método propuesto para el Modelado de Servicios basado en SOAML y MDA en el Proceso de Desarrollo para una Arquitectura Orientada a Servicios (SOA), es útil, correcto y efectivo.

En general, el procedimiento propuesto de modelado de servicios para describir los artefactos generados en el diseño, fue calificado por los expertos como muy adecuado que tiene en cuenta aspectos estudiados en importantes fuentes bibliográficas. De forma general las actividades propuestas en el mismo fueron calificadas de muy adecuadas.

Conclusiones Generales

Un procedimiento bien definido de modelado de servicios es de vital importancia para que el desarrollo e implantación de una Arquitectura Orientada a Servicios se ejecute de manera exitosa. En el presente trabajo se desarrolló una investigación, ofreciendo información referente a los principales conceptos tratados y definiciones asociadas al modelado de servicios, especificando en lo referente al Lenguaje de Modelado de Arquitecturas Orientadas a Servicios (SoaML) y la Arquitectura Dirigida por Modelos (MDA) y su uso para dicho modelado, realizándose así un estudio del estado del arte del tema.

Para el desarrollo de la propuesta de solución, se propuso una serie de actividades para el modelado de servicios, que abarcaron la identificación, especificación y generación automática del WSDL de los servicios web, definiendo así un procedimiento de modelado de servicios utilizando el estándar de modelado de servicios SoaML, describiendo así los artefactos generados en el diseño y contribuyendo de esa forma a minimizar los retrasos existentes, dándole cumplimiento al objetivo principal de la investigación.

Mediante el Método Delphi se procedió a la validación del método propuesto. Para ello se consultaron siete expertos los cuales luego del desarrollo del mismo, lo valoraron de muy adecuado para el modelado de servicios en el proceso de desarrollo para una Arquitectura Orientada a Servicios, brindando sus puntos de vista, los cuales se tuvieron en cuenta para el enriquecimiento del mismo.

Recomendaciones

Se recomienda para un trabajo futuro, crear un metamodelo que ayude al modelado de servicios REST y elaborar un mecanismo para su posterior implementación a partir del modelo antes creado. La intención de esta recomendación es crear una solución para las herramientas utilizadas por el centro, porque las herramientas que contienen un mecanismo para el modelado e implementación de servicios REST, son de difícil acceso para el país. Una de ellas es el caso de la herramienta de IBM Rational Software Architect.

Referencias Bibliográficas

1. **ISO (2000)**. *"Norma Internacional ISO 9001 - Sistemas de gestión de la calidad – Requisitos"*. Ginebra, Suiza : Impreso en la Secretaría Central de ISO.
2. **Woods, D., Mattern, T.** *"Enterprise SOA: Designing IT for Business Innovation"*. Abril 2006.
3. **Casanave, Cory.** *"Enterprise Service Oriented Architecture Using the OMG SoaML Standard"*. s.l. : A Model Driven Solutions, Inc. White Paper, December, 2009.
4. **Andrea Delgado, Ignacio García-Rodríguez de Guzmán, Francisco Ruiz.** *"Desarrollo de servicios con SoaML desde procesos de negocio en BPMN: metodología y automatización"*. Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, 1300 Montevideo, Uruguay : s.n.
5. **OMG.** Sitio wiki colaborativo de OMG para SoaML. [Online] <http://www.omgwiki.org/SoaML/doku.php>.
6. **Revised Submission OMG document.** *"Service oriented architecture Modeling Language(SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS)"*. 2008-08-04.
7. **Casanave, Cory, CEO.** *"Enterprise-SOA with SoaML by Example"*. Presentation given at SOA Consortia : s.n., 09 January 2009.
8. **Delgado, Andrea.** *"Desarrollo de Software con enfoque en el Negocio"*. Instituto de Computación Facultad de Ingeniería Universidad de la República 11300, Montevideo, Uruguay : s.n.
9. **Juan Carlos Molina, Oscar Pastor.** *"MDA, OO-Method y la Tecnología OLIVANOVA Model Execution"*. CARE Technologies; Universidad Politécnica de Valencia : s.n.
10. **Jose Manuel Pérez, Francisco Ruiz, Mario Piattini.** *"Model Driven Engineering Aplicado a Business Process Management"*. Informe Técnico UCLM-TSI-002; Departamento de Tecnologías y Sistemas de Información; Universidad de Castilla-La Mancha : s.n., Marzo 2007.

11. **Brian Elvesæter, Cyril Carrez, Parastoo Mohagheghi, Arne-Jørgen Berre, Svein G. Johnsen and Arnor Solberg.** *"Model-Based Development with SoaML"*.
12. **Arne J. Berre, Brian Elvsaeter, SINTEF ICT, Dima Panfilenko, DFKI IWI, Andrey Sadovykh, SOFTEAM.** *"SHAPE Tutorial"*. Paris, France : s.n., 16 June 2010.
13. **Astigarraga, Eneko. 2006.** Prospectiva, Estrategia y Planificación. [Online] 2006. [Cited: Mayo 13, 2009.] http://www.prospectiva.eu/zaharra/Metodo_delphi.pdf.
14. **Linstone, H. y Turoff, M. 1975.** *"The Delphi Method. Techniques and Applications."* s.l. : Addison-Wesley, 1975.
15. **Colunga, S y Amayuela, G. 2003.** *"La Psicología Educativa, su objeto, métodos y problemas principales."* Universidad de Camaguey : s.n., 2003.
16. **Martínez de Santelices, C. 2004.** *"Experimentos virtuales para la Enseñanza del Electromagnetismo."* Universidad de Camaguey : s.n., 2004.
17. **Quintana Aput, Rolando. 2007.** *"Propuesta de Indicadores para medir competencias del personal según el rol en proyectos multimedia."* Universidad de las Ciencias Informáticas : s.n., 2007.
18. **Jaramillo, Carlos Mario Pérez. 2008.** [Online] 2008. [Cited: Mayo 25, 2008.] www.escuelagobierno.org/v1/archivos.php?descargar=78.
19. **Weske, M. 2007.** *"BPM Concepts, Languages, Architectures."* s.l. : Springer.
20. **Smith, H., Fingar, P. 2003.** *"Business Process Management: The third wave, Meghan-Kieffer."*
21. **van der Aalst, W.M.P., ter Hofstede, A., Weske, M.** *"Business Process Management: A Survey."* International Conference on Business Process Management : s.n., 2003.
22. **Papazoglou, M., Traverso, P., Dustdar, S. and Leymann, F.** *"Service-Oriented Computing: State of the Art and Research Challenge."* IEEE Computer Society : s.n., 2007.
23. **Mellor, S., Clark, A., Futagami, T.** *"MDD, Guest editors intro."* IEEE CS : s.n., 2003.

24. **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 : s.n., Octubre 2010.
25. **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 : s.n., 2009.
26. **Object Management Group.** SoaML, UML, Middleware and MDA standards. [Online] <http://www.omg.org>.
27. **Supported by OMG.** SoaML Web Site. [Online] <http://www.SoaML.org>.
28. **Model Driven Solutions.** Business, SOA, & BPM Architectures and Solutions. [Online] <http://www.modeldriven.com>.
29. **Modeliosoft.** [Online] <http://www.modeliosoft.com>.
30. **Group, Object Management.** *"Business process modeling notation (BPMN) 1.2."*. 2009-01-03. document formal.
31. **Brian Elvesæter, Arne-Jørgen Berre, Andrey Sadovykh.** *"SPECIFYING SERVICES USING THE SERVICE ORIENTED ARCHITECTURE MODELING LANGUAGE (SOAML): A baseline for Specification of Cloud-based Services."*
32. **Amsden, J., 2010.** *"Modeling with SoaML, the Service-Oriented Architecture Modeling Language, IBM, Technical article series."*
33. **Elvesæter, B., Carrez, C., Mohagheghi, P., Berre, A.-J., Johnsen, S. G., Solberg, A., 2011.** *"Model-driven Service Engineering with SoaML, in Service Engineering - European Research Results."*. s.l. : Wien, Springer. pp. 25-54.
34. **OMG, 2009.** *"Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0."*. Object Management Group, Document dtc : s.n., 2009-08-14.
35. **Graham., Ian.** *"Requirements Modelling and Specification for Service Oriented Architecture."*. s.l. : A John Wiley and Sons, Ltd., Publication, 2008.
36. **Andrews, D. 2009.** *"Service Oriented Architecture: The Catalysis II Approach, in preparation."*

37. **Erl, T. 2004.** *"Service Oriented Architecture: A Field Guide to Integrating XML and Web Services."* Upper Saddle River NJ: Prentice Hall : s.n.
38. **Erl, T. 2005.** *"Service Oriented Architecture: Concepts, Technology and Design."* Upper Saddle River NJ: Prentice Hall : s.n., 2005.
39. **Bell, Michael.** *"SERVICE-ORIENTED MODELING: SERVICE ANALYSIS, DESIGN, AND ARCHITECTURE."* s.l. : John Wiley & Sons, Inc., Hoboken, New Jersey, 2008.
40. **(OMG), Object Management Group.** *"Model Driven Architecture (MDA)."* 2003.
41. **Object Management Group (OMG).** Service Oriented Architecture (SOA). [Online] <http://www.omg.org/soa>.
42. **(OMG), Object Management Group.** *"SOA Modeling Language (SoaML)."* 2009.
43. **Object Management Group (OMG).** Model Driven Architecture (MDA). [Online] <http://www.omg.org/mda>.
44. **Krafzig, D. Banke, K. Slama, D.** *"Enterprise SOA, Service Oriented Architecture Best Practices."* Prentice Hall : s.n., 2005.
45. **IBM RedBooks.** *"Patterns: Service Oriented Architecture and Web Services."* IBM : s.n., 2004.
46. **Oscar Pastor, Juan Carlos Molina.** *"Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling."* s.l. : Springer-Verlag Berlin Heidelberg New York, 2007.
47. **Brown, A. 2004.** *"Model driven architecture: Principles and practice."* Journal on Software and Systems Modelling : Springer, Berlin Heidelberg, Volume 3, Number 4, December, 2004. pp 314–327.
48. **MDA (2006).** Model Driven Architecture. [Online] <http://www.omg.org/mda>.
49. **Miller J, Mujerki J (eds) (2003).** MDA Guide version 1.0.1. [Online] [Cited: Junio 12, 2003.] <http://www.omg.org/docs/omg/03-06-01.pdf>.
50. **Object Management Group (OMG).** Object Management Group. [Online] <http://www.omg.org>.

51. **Raistrick C, Francis P, Wright J, Carter C, Wilkie I (2004).** *"Model driven architecture with executable UML."* Cambridge University Press, Cambridge : s.n.
52. **UML (2004) The Object Management Group.** *"Unified Modeling Language: Super-structure, version 2.0."* OMG doc formal : s.n., 05-07-04.
53. **Bézivin, J.** *"MDA: From Hype to Hope, and Reality."* Guest talk at the 6th International Conference on the Unified Modeling Language-UML'2003, San Francisco, California, USA : s.n., October 20-24, 2003.
54. **Harmon, P.** *"The OMG's Model Driven Architecture and BPM, Business Process Trends."* Mayo 2004. pp.1 – 12.
55. **Delgado, Andrea.** *"Tesis de Maestría en Informática: Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)."* Montevideo, Uruguay : s.n., 2007.
56. **Arsanjani, A.** Service-oriented modeling and architecture, How to identify, specify, and realize services for your SOA. [Online] [Cited: Julio 31, 2007.] <http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/,noviembre2004>.
57. **Endrei M ., Ang J., Arsanjani A., et. al.** Patterns: Service-oriented Architecture and Web Services, IBM Redbook, SG24-6303-00. [Online] 2004. [Cited: Julio 31, 2007.] <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>.
58. **Kleppe A., Warmer J., Bast W.** *"MDA Explained. The Model Driven Architecture: Practice and Promise."* s.l. : Pearson Education, 1st. edition, 2003, 4th. printing 2005.
59. **Delgado, A.** *"Model Driven Architecture (MDA): enfoque para el desarrollo de software basado en modelos."* Biblioteca PEDECIBA, Instituto de Computación , Facultad de Ingeniería, Universidad de la República. : s.n., 2005. TR-05-18.
60. **Mellor S., Scott K.,Uhl A.,Weise D.** *"MDA Distilled: Principles of Model-Driven Architecture."* s.l. : Addison Wesley, 1st. edition, 2004.
61. **W3C.** Web Services Description Language (WSDL). [Online] [Cited: Julio 31, 2007.] <http://www.w3.org/TR/ws-arch/http://www.w3.org/TR/wsdl>.

Anexo 1. Encuesta de Autovaloración

Estimado Compañero(a):

En la ejecución de la presente investigación deseamos poner a consideración de un grupo de expertos la propuesta Método de Modelado de Servicios basado en SOAML y MDA en el Proceso de Desarrollo para una Arquitectura Orientada a Servicios (SOA). Para ello necesitamos conocer el grado de dominio que Ud. posee sobre el proceso para la identificación e especificación de los servicios en una SOA. Para ello deseamos que responda lo que a continuación le solicitamos.

Nombre y apellidos: _____

Centro de trabajo: _____

Labor que realiza: _____

Años de experiencia: _____ Especialidad: _____

Categoría docente: _____ Categoría científica: _____

País: _____

1.- Marque con una cruz (X) el grado de conocimiento que Ud. tiene sobre la temática que se investiga:

0	1	2	3	4	5	6	7	8	9	10

2.- Marque con una cruz (X) las fuentes que le han servido para argumentar el conocimiento que tiene Ud. de la temática que se investiga. Señale con otro color la que más ha influido.

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.			
2.-	Experiencia.			
3.-	Trabajos de autores nacionales.			
4.-	Trabajos de autores extranjeros.			
5.-	Su propio conocimiento del tema.			
6.-	Su intuición.			

Anexo 2. Encuesta a Expertos

Estimado Compañero(a):

La presente tesis propone un Método de Modelado de Servicios basado en SOAML y MDA en el Proceso de Desarrollo para una Arquitectura Orientada a Servicios (SOA). Dicho método posee una serie de actividades para el modelado de servicios.

Puntúe el grado de factibilidad para cada pregunta de acuerdo con la siguiente escala:

- 5 – Muy Adecuado
- 4 – Bastante Adecuado
- 3 – Adecuado
- 2 – Poco Adecuado
- 1 – No Adecuado

1. ¿Cómo considera Ud. la propuesta de tres actividades (Identificar servicios, Especificar servicios y Generar WSDL de servicios) para la concreción de un procedimiento de modelado de servicios utilizando SoAML que describa los artefactos generados en el diseño de servicios?

- 1.1. Considera esas tres actividades :

- necesarias: _____ sí _____ no _____ no se
- suficientes: _____ sí _____ no _____ no se

a) De considerarlo conveniente proponga otras:

2. El proceso general propuesto consta de las siguientes Actividades. Puntúelas de acuerdo al grado de factibilidad que Ud. considere.

No	Actividades del Proceso General	Calif.
1.	Identificar servicios	
2.	Especificar servicios	
3.	Generar WSDL de servicios	

3. Por último, deseáramos que usted expresara sus criterios y/o recomendaciones que pudieran ayudar para perfeccionar nuestra investigación.

Anexo 3. Listado de Expertos que colaboraron

Expertos:

Ing. Jorge Hernández Roselló (UCI-Cuba)

Ing. Leslier López Nicot (UCI-Cuba)

Ing. Luis Enrique Hernández Vega (UCI-Cuba)

Ing. Noel Marquez Batista (UCI-Cuba)

Ing. Raimundo Llerena Ferrer (UCI-Cuba)

Ing. Ubalquis Montenegro Moreno (UCI-Cuba)

Ing. Yainelys Rivero Gutierrez (UCI-Cuba)

Anexo 4. Factibilidad de las actividades contenidas en el Proceso General del Método propuesto.

Tabla de Frecuencias Absolutas:							
No.	Elementos	C1	C2	C3	C4	C5	Total
1	Identificar servicios	3	4				7
2	Especificar servicios	2	5				7
3	Generar WSDL de servicios	4	3				7

Tabla de Frecuencias Absolutas Acumuladas:							
No.	Elementos	C1	C2	C3	C4	C5	
1	Identificar servicios	3	7				
2	Especificar servicios	2	7				
3	Generar WSDL de servicios	4	7				

Tabla de Frecuencias Relativas Acumuladas:							
No.	Elementos	C1	C2	C3	C4	C5	
1	Identificar servicios	0,428571428	0,9999				
2	Especificar servicios	0,285714285	0,9999				
3	Generar WSDL de servicios	0,571428571	0,9999				

Puntos de corte:									
No	Aspectos	C1	C2	C3	C4	Suma	P	N-P	
1	Identificar servicios	-0,18	3,72			3,54	1,77	-1,06	Muy adecuado
2	Especificar servicios	-0,57	3,72			3,15	1,58	-0,87	Muy adecuado
3	Generar WSDL de servicios	0,18	3,72			3,90	1,95	-1,24	Muy adecuado
Suma		-0,57	11,16			10,59			
P.de corte		-0,19	3,72						

Anexo 5. Caso de Estudio en la Herramienta Modelio

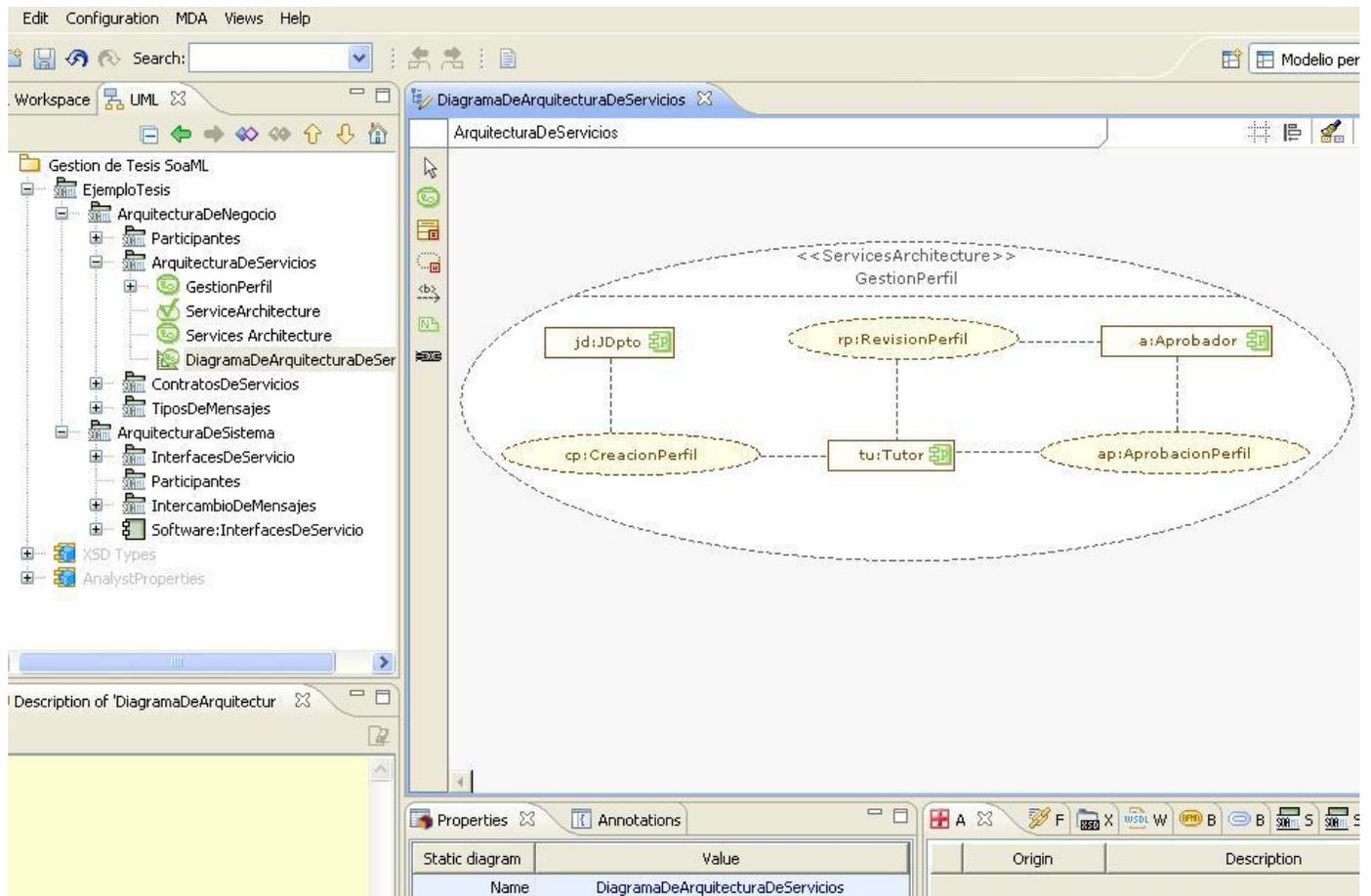


Figura 19. Modelado de Arquitectura de Servicios (Modelio).

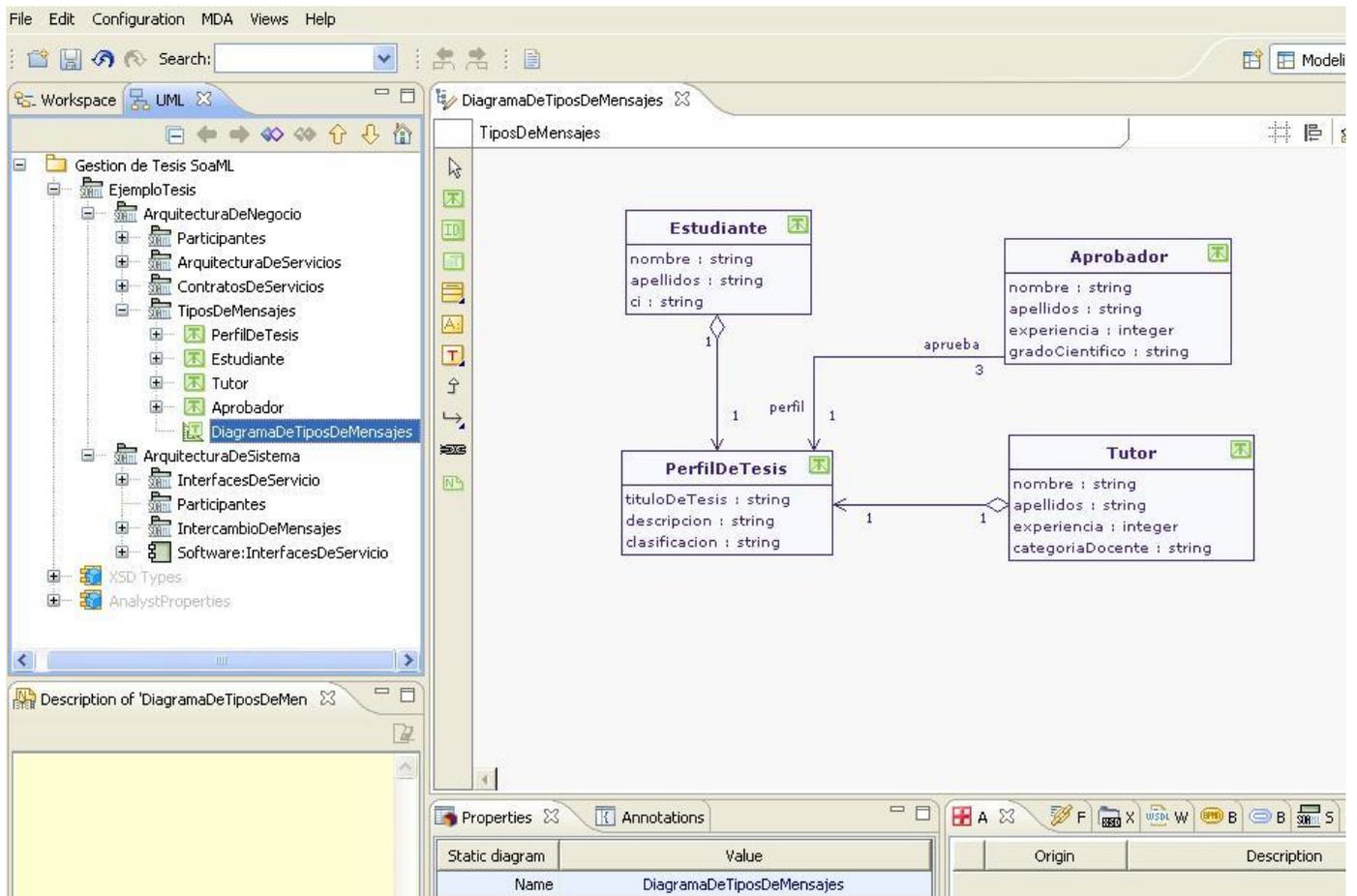


Figura 20. Modelado de los Tipos de Mensaje (Modelo).

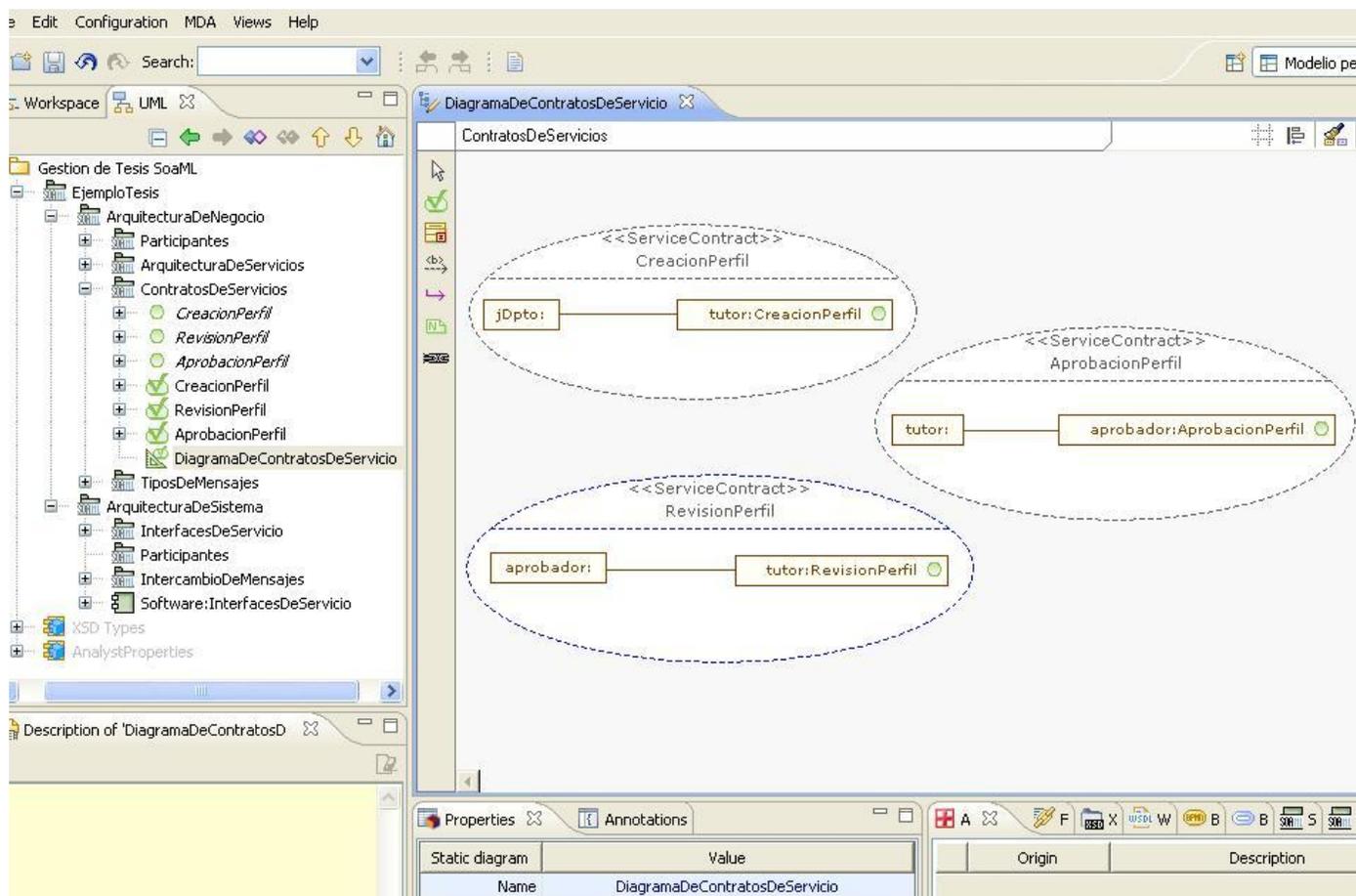


Figura 21. Modelado del Contrato de Servicio (Modelo).

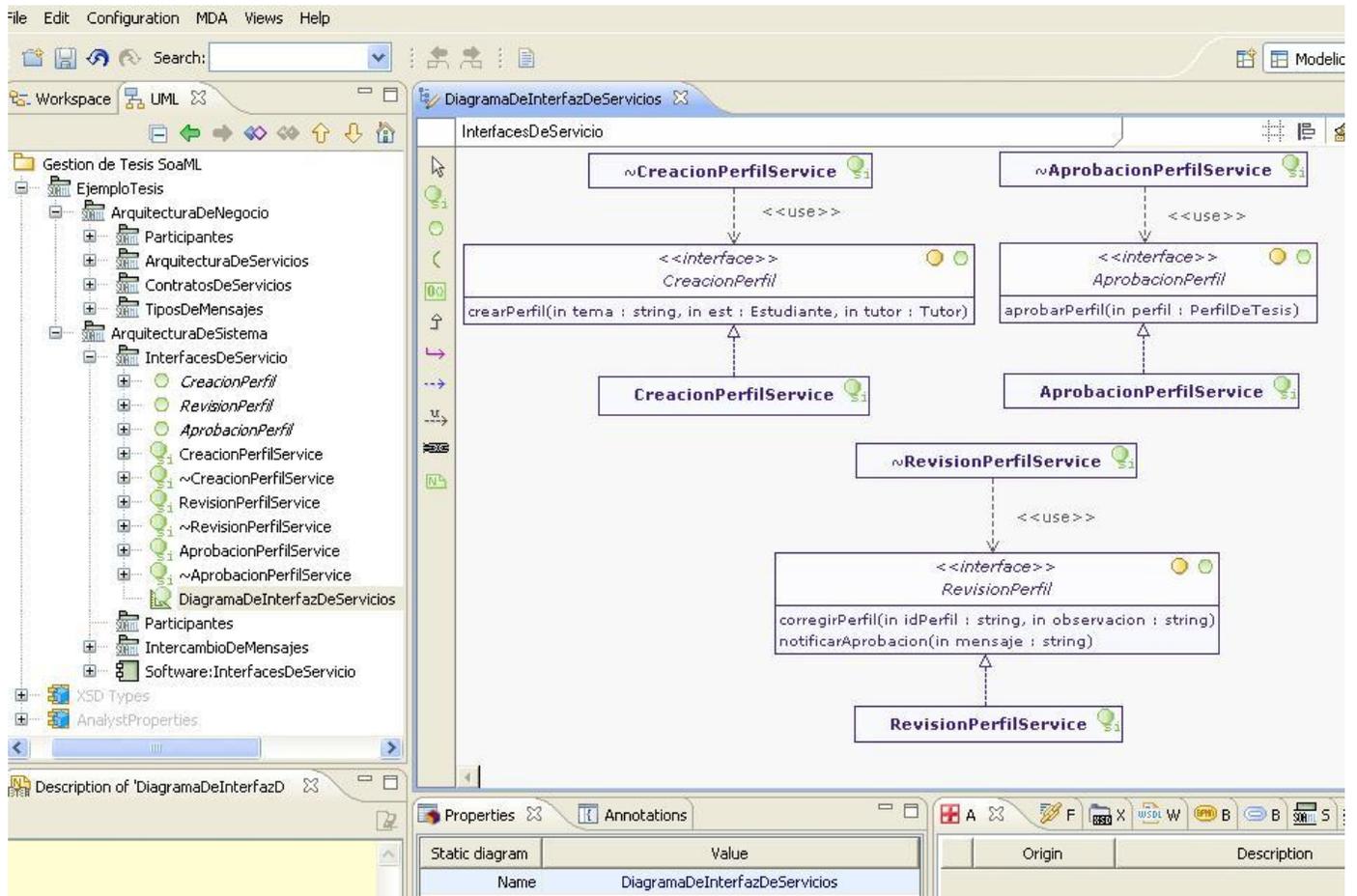


Figura 22. Modelado de la Interfaz de Servicio (Modelio).

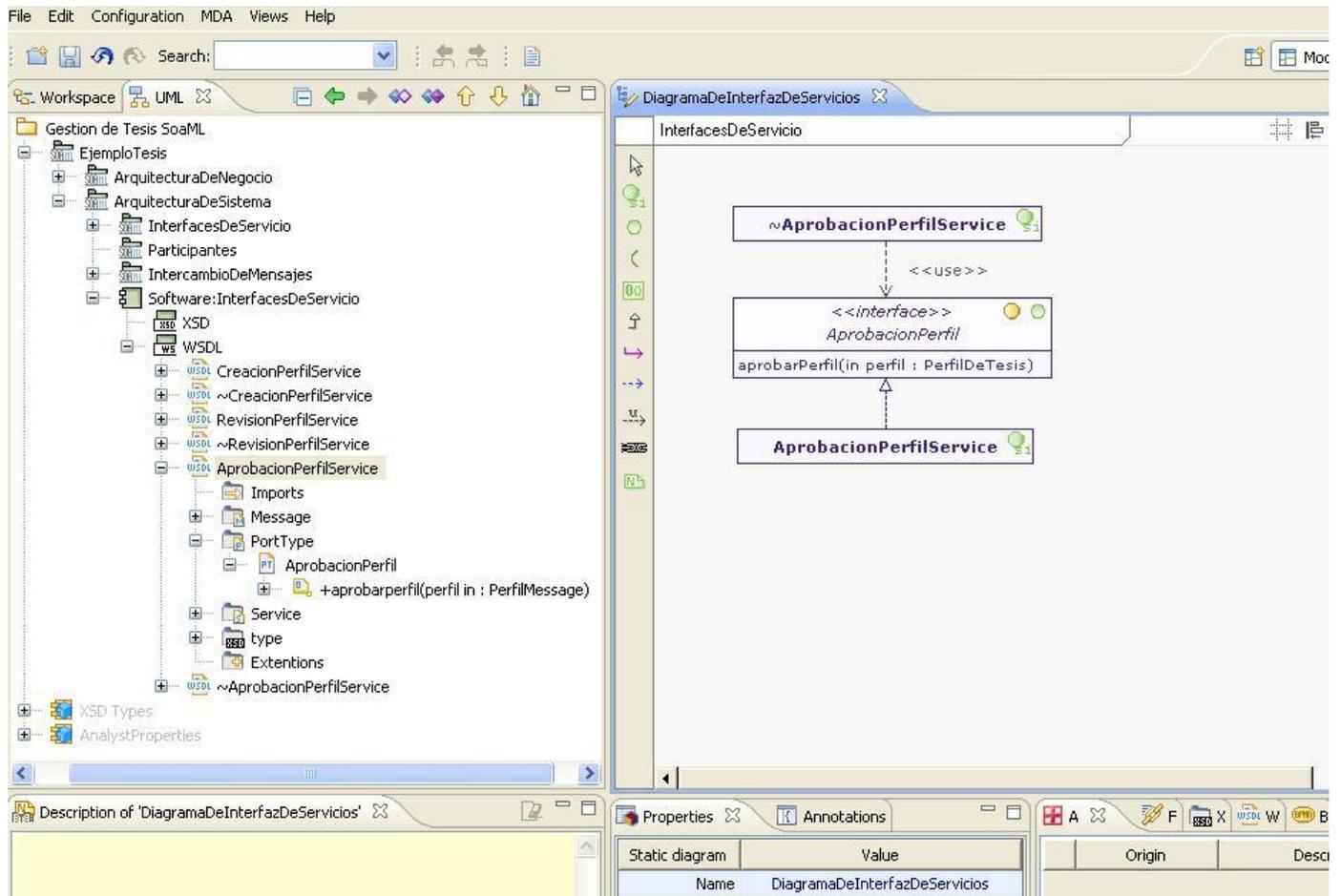


Figura 23. Transformación a los WSDL a partir de las Interfaces de Servicio (Modelo).

```

<?xml version="1.0" encoding="UTF-8"?>
<ws:definitions xmlns:ws="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://SOATransformer/wsdl/AprobacionPerfilService.wsdl" >
  <ws:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:Perfiledetesis="http://modeliosoft/xsddesigner/37c466b0-bf1f-11e1-8610-000000000000" >
      <element name="perfiledetesis" type="tns:PerfilDeTesis"/>
      <complexType name="PerfilDeTesis">
        <sequence>
          <element name="titulodetesis" type="string"/>
          <element name="descripcion" type="string"/>
          <element name="clasificacion" type="string"/>
          <element name="idestudiante" type="string"/>
          <element name="idtutor" type="string"/>
          <element name="idperfil" type="string"/>
        </sequence>
      </complexType>
    </schema></ws:types>
    <ws:message name="PerfilMessage">
      <ws:part name="Part" element="tns:perfiledetesis"/>
    </ws:message>
    <ws:portType name="AprobacionPerfil">
      <ws:operation name="aprobarperfil">
        <ws:input name="perfil" message="tns:PerfilMessage"/>
      </ws:operation>
    </ws:portType>
    <ws:binding name="IAprobacionPerfilServiceBinding" type="tns:AprobacionPerfil">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" ws:required="false"/>
      <ws:operation name="aprobarperfil">
        <soap:operation soapAction="http://SOATransformer/wsdl/AprobacionPerfilService.wsdl/aprobarperfil" ws:required="false"/>
        <ws:input name="perfil">
          <soap:body use="literal" ws:required="false"/>
        </ws:input>
      </ws:operation>
    </ws:binding>
    <ws:service name="AprobacionPerfilServiceService">
      <ws:port name="AprobacionPerfilServicePort" binding="tns:IAprobacionPerfilServiceBinding">
        <soap:address location="http://localhost/discountvoyagesoamltutorial" ws:required="false"/>
      </ws:port>
    </ws:service>
  </ws:definitions>

```

Para obtener Ayuda, presione F1

Figura 24. Fichero WSDL generado del servicio AprobacionPerfil (Modelio).