Transformaciones Genéricas para la Implementación de Web Services en Diferentes Plataformas

Ariel Arsaute, Marcela Daniele, Mariana Frutos, Paola Martellotto, Fabio Zorzan
Departamento de Computación
Facultad de Ciencias Exactas, Físico-Químicas y Naturales - Universidad Nacional de Río Cuarto

CONTEXTO

La línea de investigación presentada en este trabajo se desarrolla en el marco del proyecto "Definición de Transformaciones Genéricas para la implementación de Web Services en diferentes plataformas", presentado en el Programa de Apoyo a Grupos de Reciente Formación en la Convocatoria 2008 del Ministerio de Ciencia y Tecnología de la provincia de Córdoba.

RESUMEN

Esta línea de investigación pretende aportar al desarrollo de aplicaciones basadas en Web Services, y propone la definición de diferentes transformaciones que reciban como entrada un modelo genérico para Web Services, y retornen modelos que representan a diferentes plataformas/lenguajes de implementación para Web Services. Dichas transformaciones se enmarcan dentro de la metodología de desarrollo de Arquitecturas Dirigidas por Modelos (Model Driven Architecture-MDA) el cual sostiene que si el desarrollo está guiado por los modelos de software, se obtienen beneficios importantes en aspectos fundamentales como la productividad, portabilidad, interoperabilidad y mantenimiento. La definición de la transformación se realiza por medio del lenguaje Relations que forma parte de Query/Views/Transformations (QVT) o con Reglas de Transformación de Grafos, para los cuales existen herramientas de software que permiten, tanto la definición, como la ejecución de transformaciones.

Palabras Clave: Web Services, QVT, Reglas de Transformación de Grafos, MDA

1. INTRODUCCIÓN

Desde los inicios de la era de la información la reutilización ha supuesto una práctica habitual para el desarrollo de aplicaciones de software. En este momento, la reutilización de software continúa siendo un aspecto esencial en los sistemas de información, a través del uso de componentes de software.

La tendencia actual en el desarrollo de componentes es reutilizarlos eficientemente en diferentes proyectos, tanto como sea posible. Esta reutilización está limitada a un lenguaje de programación o a una plataforma en particular.

Con la aparición de los Web Services (Servicios Web) [1], esta limitación fue superada, proporcionado un modelo diferente para el desarrollo de aplicaciones, ofreciendo la capacidad de acceder a servicios heterogéneos de forma unificada e interoperable a través de Internet. Los Web Services son servicios autónomos e independientes que se ofrecen mediante la web. Principalmente, permiten que las aplicaciones sean más modulares y desacopladas, facilitando su reutilización en distintas plataformas o lenguajes de programación.

En la actualidad, la construcción de software se enfrenta a continuos cambios en las tecnologías de implementación, lo que implica realizar esfuerzos importantes tanto en el diseño para integrar las diferentes tecnologías que incorpora, como en el mantenimiento para adaptar la aplicación a los

cambios en los requisitos y en las tecnologías de implementación. Es difícil satisfacer requisitos de escalabilidad, seguridad y eficiencia. La idea clave que subyace a las arquitecturas dirigidas por modelos MDA [2] es que, si el desarrollo está guiado por los modelos de software, se obtendrán beneficios importantes en aspectos fundamentales como son la productividad, la portabilidad, la interoperabilidad y el mantenimiento.

La necesidad de compartir información entre sistemas implementados en distintos lenguajes de programación, las ventajas que provee la técnica MDA y los importantes beneficios aportados por la incorporación de Web Services a las aplicaciones, conduce a pensar en la necesidad de contar con una solución genérica que aproveche las fortalezas de estas tecnologías. Esta línea de investigación tiene como objetivo proveer una solución genérica y eficiente para el tipo de problemas planteados, permitiendo transformaciones, ya sea empleando QVT [3] o Reglas de Transformación de Grafos [4], para transformar el metamodelo de Web Services a diferentes metamodelos de lenguajes de implementación.

1.1. WEB SERVICES

Los Web Services proveen esencialmente un medio estándar de comunicación entre diferentes aplicaciones de software. Su uso en Internet se ha extendido rápidamente debido a la creciente necesidad de comunicación e interoperabilidad entre aplicaciones. Se definen como aplicaciones auto-contenidas y auto-descritas, que pueden ser publicadas, localizadas e invocadas a través de la Web. Una vez desarrolladas, otras aplicaciones (y otros Web Services) pueden descubrirlas e invocar el servicio dado [1].

Los Web Services se basan en estándares abiertos, lo que permite a las aplicaciones comunicarse más libre e independientemente de las plataformas en que se están ejecutando. Las tecnologías subyacentes que permiten su implementación son: Web Service Description Language (WSDL) [5] [6], Simple Object Access Protocol (SOAP) [7], Hiper Text Transport Protocol (HTTP) y Universal Description, Discovery and Integration (UDDI) [8]. El uso de estos protocolos estándares es necesario para lograr la interoperabilidad en ambientes heterogéneos, con independencia del Sistema Operativo, el lenguaje de programación, etc.

1.2. MDA

En la actualidad, la construcción de software se enfrenta a continuos cambios en las tecnologías de implementación, lo que implica realizar esfuerzos importantes tanto en el diseño, para integrar las diferentes tecnologías que incorpora, como en el mantenimiento, para adaptar la aplicación a los cambios en los requisitos y en las tecnologías de implementación. Para conseguir los beneficios de MDA se plantea el siguiente proceso de desarrollo: de los requisitos se obtiene un modelo independiente de la plataforma (Platform Independent Model-PIM), luego este modelo es transformado con la ayuda de herramientas en uno o más modelos específicos de la plataforma (Platform Specific Model-PSM), y finalmente cada PSM es transformado en código. Por lo tanto, MDA incorpora la idea de transformación de modelos (PIM a PSM, PSM a código) y se necesitan herramientas para automatizar esta tarea. Estas herramientas constituyen uno de los elementos básicos de MDA.

1.3. TRANSFORMACIONES DE MODELOS

La transformación de modelos es una de las bases en las que se fundamenta la metodología MDA, en particular la transformación automática de modelos. Para lograr esta automaticidad en la transformación de modelos se pueden utilizar diferentes herramientas, para esta propuesta se plantea la

utilización de dos enfoques para la definición de las transformaciones antes planteadas, el primero es el uso del lenguaje Relations que forma parte de QVT [3], y el segundo el uso de técnicas de transformación de grafos [4]. A continuación se describe brevemente cada uno de estos enfoques.

1.4. QVT

El planteamiento QVT se basa principalmente en la definición de un lenguaje para las consultas (Queries) sobre los modelos Meta Object Facility (MOF) [9], la búsqueda de un estándar para generar vistas (Views) que revelen aspectos específicos de los sistemas modelados, y finalmente, la definición de un lenguaje para la descripción de transformaciones (Transformations) de modelos MOF.

En este trabajo se utiliza el componente de QVT que tiene como objetivo definir transformaciones. Estas transformaciones describen relaciones entre un metamodelo fuente F y un metamodelo objetivo O, ambos metamodelos deben estar especificados en MOF. Luego esta transformación definida se utiliza para obtener un modelo objetivo, el cual es una instancia del metamodelo O, a partir de un modelo fuente que es una instancia del metamodelo F. Una característica muy importante de estas transformaciones es que pueden ser bidireccionales (multidimencionales también).

1.4.1. EL LENGUAJE RELATIONS

La especificación de QVT que se utiliza tiene una naturaleza híbrida declarativa/imperativa. El lenguaje Relations es una especificación declarativa de relaciones entre metamodelos MOF. Este lenguaje permite realizar “pattern matching” de objetos complejos y definir “templates” de creación de objetos.

1.5. LAS REGLAS DE TRANSFORMACIÓN DE GRAFOS

Para la construcción y evolución de modelos es necesario asegurar la consistencia entre los mismos, a través de la transformación de modelos. Para llevar a cabo esta transformación de modelos se propone usar las reglas de transformación de grafos y aplicarlas a la transformación de modelos en el campo de MDA. La transformación de grafos es una técnica gráfica, formal, declarativa y de alto nivel muy usada para transformación y simulación de modelos, chequeo de consistencia entre modelos o vistas y optimización en diversos contextos. Los artefactos producidos para conceptualizar un sistema son llamados modelos, y los diagramas son usados para visualizar su estructura compleja de una forma más intuitiva y natural. A través de los grafos pueden ser definidas las estructuras de esos modelos, entonces la transformación de grafos puede ser explorada para especificar cómo deben ser construidos y cómo deben evolucionar. Esta técnica formal para la manipulación de grafos está basada en reglas. Los modelos gráficos se interpretan como grafos y se usan las reglas de transformación de grafos para construir diagramas correctos.

2. LINEAS DE INVESTIGACION Y DESARROLLO

Una de las alternativas para implementar la transformación propuesta es usar QVT. A continuación se presenta esta alternativa.

2.1. TRANSFORMACIÓN DE MODELOS WEB SERVICES BASADA EN QVT

El esquema general de la transformación, de Web Services genéricos a la definición de estos en un lenguaje de implementación, puede ser visto en tres niveles: metamodelo, definición/modelo y ejecución, como lo muestra la Figura 1.

En el nivel *metamodelo* se encuentran los metamodelos involucrados en la transformación, estos son: el metamodelo de Web Services y el metamodelo de algún lenguaje de implementación, por ejemplo Java. Entre estos modelos se define la transformación, en este caso, mediante el lenguaje de transformación Relations de QVT. Pasando al nivel de *modelo/definición* se encuentran los modelos específicos que definen un Web Service. Por ejemplo, a partir de un modelo Web Service para reservas de hoteles de una agencia de viajes, y por aplicación de la transformación a nivel metamodelo, se obtiene el modelo Java que implementa dicho Web Service. Por último, en el nivel de *ejecución*, se encuentran los sistemas desarrollados en Java que implementan el Web Service de la agencia de viajes.

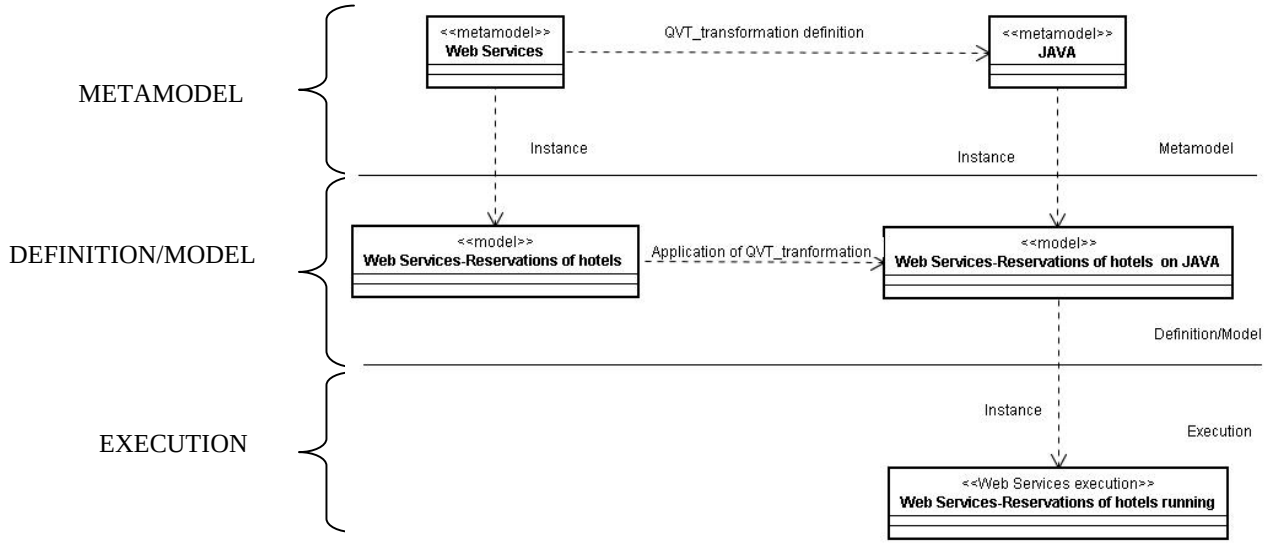

Figura 1: Vista general de la transformación.

3. RESULTADOS OBTENIDOS/ESPERADOS

Los resultados de la investigación aportan a la mejora del desarrollo de aplicaciones que utilizan Web Services, esto se logra con la utilización de herramientas (transformaciones, metamodelos) para el desarrollo de este tipo de aplicaciones en el marco de MDA.

El beneficio de estas transformaciones queda reflejado teniendo en cuenta los constantes cambios en los requerimientos de un sistema. Esto se debe a que cualquier cambio en la especificación de un Web Service de una aplicación, podrá ser propagado a la especificación en el lenguaje de implementación de la aplicación, y así, adaptar rápidamente los sistemas (escritos en diferentes lenguajes) a los cambios en los Web Services que implementan.

En síntesis, los principales objetivos planteados en esta investigación son:

- Aplicar, adaptar y mejorar las técnicas utilizadas en arquitecturas orientadas a modelos, tanto para el desarrollo de software como aquellas orientadas a mejorar la calidad del software.
- Definir y formalizar el metamodelo para Web Services.
- Definir y formalizar metamodelos para lenguajes de programación específicos.
- Especificar las transformaciones entre metamodelos, que tengan como origen el metamodelo Web Services y como destino un metamodelo de un lenguaje/plataforma específico.
- Desarrollar un caso de estudio, implementando los módulos necesarios para soportar la interoperabilidad entre sistemas a través de Web Services.

4. FORMACION DE RECURSOS HUMANOS

Los estudios realizados en esta línea de investigación sirven como marco para la elaboración de trabajos finales de grado y tesis de posgrado. En este punto, una estudiante de la carrera Licenciatura en Ciencias de la Computación ha iniciado su trabajo final y consiste en la implementación de los Web Services necesarios para compartir información de alumnos entre dos sistemas utilizados de la UNRC, denominados SIAL (Sistema Integral de Alumnos) y SIAT (Sistema Informático de Apoyo a la Teleformación). El primero permite la gestión de alumnos de grado y es desarrollado por el Centro de Cómputos de la UNRC, y el segundo permite la gestión de alumnos de cursos y carreras a distancia y es desarrollado por la Secretaría de Extensión de la UNRC. La aplicación de las transformaciones especificadas posibilitará el desarrollo de este trabajo final utilizando la metodología MDA, alcanzando las ventajas de su aplicación al desarrollo de software, como son la portabilidad, interoperabilidad, productividad y mantenimiento.

Los temas abordados en esta línea de investigación brindan un fuerte aporte al proceso de perfeccionamiento continuo de los autores de este trabajo, que se desempeñan como docentes de las carreras de computación que se dictan en la Universidad Nacional de Río Cuarto y participan en asignaturas relacionadas a dichos temas.

Además, este proyecto aporta a la construcción de estructuras genéricas orientadas a la reutilización de modelos en diferentes plataformas de implementación. Al mismo tiempo, se proveen herramientas para la construcción de software en el marco de MDA, que podrán ser usadas por otros desarrolladores de aplicaciones que involucran el uso de Web Services.

5. BIBLIOGRAFIA

[1] World Wide Web Consortium. Web Service Architecture. http://www.w3.org/TR/ws-arch/.

[2] OMG, Model Driven Architecture (MDA) Guide. Miller, J., Mukerji, J. (eds) 2003. document number ormsc/2001-07-01. Obt. en Mayo de 2007 de http://www.omg.org/docs/omg/03-06-01.pdf

[3] Object Management Group, “Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification” Final Adopted Specification ptc/05-11-01, http://www.omg.org/docs/ptc/05-11-01.pdf, último acceso Febrero 2008.

[4] Baresi L., Heckel R. Tutorial Introduction to Graph Transformation: A Software Engineering Perspective. First International Conference on Graph Transformation. Spain. v. 2505, pp. 402-429. (ICGT 2002). 2002.

[5] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. http://www.w3.org/TR/wsdl20. Último acceso Mayo 2007.

[6] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. http://www.w3.org/TR/wsdl20-primer. Último acceso Mayo 2007.

[7] World Wide Web Consortium. SOAP Version 1.2 Part 1: Messaging Framework. http://www.w3.org/TR/soap12-part1/. Último acceso Mayo 2007.

[8] OASIS. UDDI Version 3.0.2. http://uddi.org/pubs/uddi_v3.htm. Último acceso Mayo 2007.

[9] Object Management Group “Meta Object Facility (MOF) Core Specification” OMG Available Specification. Vers 2.0. formal/06-01-01, http://www.omg.org/docs/formal/06-01-01.pdf. (Nov. 2006.