

Generación automática de API REST a partir de API Java, basada en transformación de Modelos (MDD).

Ariel Arsaute, Fabio Zorzan, Marcela Daniele, Ariel González, Mariana Frutos

Departamento de Computación, Facultad de Ciencias Exactas, Físico-Químicas y Naturales,
Universidad Nacional de Río Cuarto
Ruta 36 Km. 601 –CP 5800 - Río Cuarto – Córdoba - Argentina Tel. (0358) 4676235
{aarsaute, fzorzan, marcela, agonzalez, mfrutos}@dc.exa.unrc.edu.ar

RESUMEN

En la actualidad existe una gran cantidad de proyectos o aplicaciones que disponen de una API REST, una nueva opción o estilo de uso de los Servicios Web (Web Services, WS), para la creación de servicios profesionales. Twitter, YouTube, los sistemas de identificación con Facebook, y cientos de empresas generan negocio gracias a REST y las APIs REST. REST es una interfaz entre sistemas que usa HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML, JSON, HTTP, etc. En los últimos años logró un gran impacto en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de utilizar y sobre todo por su eficiencia.

La línea de investigación propone un mecanismo de generación de API REST a partir de versiones existentes de API Java, en el contexto del desarrollo dirigido por modelos (Model-Driven Development, MDD), para la construcción de WSs. Aplicar esta técnica mediante la transformación de modelos se diferencia de otras formas convencionales, las cuales se basan en generar un AST (Abstract Syntax Tree) mediante algún parser de JAVA. Además, nuestra propuesta permitirá generar código hacia distintas implementaciones de WS REST a partir de un modelo JAVA.

Palabras clave: MDD, QVT Operacional, API REST

CONTEXTO

La línea de investigación presentada en este trabajo se desarrolla en el marco del proyecto “Ingeniería de Software. La Transformación de Modelos aplicada a la Mejora continua de Procesos de Desarrollo de Software”, perteneciente a los Proyectos y Programas de Investigación (PPI) de la Secretaría de Ciencia y Técnica de la Universidad Nacional de Río Cuarto. Además esta línea de investigación se enmarca dentro de un trabajo de tesis de Maestría en Ingeniería de Software.

1. INTRODUCCIÓN

El desarrollo dirigido por modelos (MDD) [1] es una metodología de desarrollo de software que se centra en la creación y explotación de modelos de dominio. Es decir, representaciones abstractas de los conocimientos y las actividades que rigen un dominio de aplicación particular. MDD tiene como objetivo aumentar la productividad mediante la maximización de la compatibilidad entre los sistemas, a través de la reutilización de modelos estandarizados, simplificando el proceso de diseño (a través de modelos de patrones de diseño que se repiten en el dominio de aplicación), y promoviendo la comunicación entre los individuos y equipos que trabajan en un sistema por medio de una estandarización de la terminología y las mejores prácticas utilizadas en el dominio de aplicación. Los modelos son desarrollados contemplando comunicaciones entre los gerentes de producto, diseñadores, miembros del equipo

de desarrollo y usuarios del dominio de la aplicación.

El Object Management Group (OMG) [2] tiene como iniciativa de MDD la arquitectura dirigida por modelos (Model Driven Architecture, MDA) [3], que es un acercamiento al diseño de software. Bajo la metodología MDA la funcionalidad de un sistema es definida en primer lugar como un modelo independiente de la plataforma (Platform-Independent Model, PIM). Dado un modelo de definición de la plataforma (Platform Definition Model, PDM), el modelo PIM puede traducirse entonces a uno o más modelos específicos de la plataforma (Platform-Specific Models, PSM) para la implementación correspondiente, usando diferentes lenguajes específicos del dominio, o lenguajes de propósito general como Java, C#, Python, entre otros. La traducción entre un modelo PIM y modelos PSM se realiza normalmente utilizando herramientas automatizadas, como lo son las herramientas de transformación de modelos, por ejemplo, aquellas que cumplen con el estándar OMG denominado Query/View/Transformation, (QVT) [4]. Además, se propone las automatizaciones de las transformaciones entre modelos y de la generación de código, centrando el proceso de desarrollo de software en las tareas esencialmente de modelado

MDA está relacionado con múltiples normas, incluyendo el lenguaje Unificado de Modelado (Unified Modeling Language, UML) [5], que se ha convertido en un estándar. UML sigue el paradigma de orientación a objetos y permite la descripción de aspectos tanto estáticos como dinámicos de sistemas de software. Más que un lenguaje es un conjunto de lenguajes, en su mayoría notaciones gráficas, soportados por un número importante de herramientas propietarias y de código abierto.

Por otro lado, existe una necesidad importante de permitir la interacción con sistemas y aplicaciones remotas, como por ejemplo aplicaciones móviles, por ello la

definición de las Interfaces de Programación de Aplicaciones (Application Programming Interface - API) toman un rol importante en el diseño e implementación dado que permiten interacciones con sistemas o aplicaciones existentes o nuevas heterogéneas.

Los WS, brindan un soporte adecuado y de manera transparente a la interacción de sistemas remotos. El consorcio W3C [6] define los WS como sistemas de software diseñados para soportar una interacción interoperable máquina a máquina sobre una red. Los WS suelen ser APIs Web que pueden ser accedidas dentro de una red (principalmente Internet) y son ejecutados en el sistema que los aloja.

Una clase de WS son los basados en llamada a procedimientos remotos (Remote Procedure Calls, RPC), los cuales son familiares a muchos desarrolladores. La interfaz es definida a través un archivo WSDL (Web Services Description Language WSDL) el cual es un formato XML. Las primeras herramientas para WS estaban centradas en esta visión y tuvo un auge importante. Sin embargo, ha sido algunas veces criticado por no ser débilmente acoplado, ya que suele ser implementado por medio del mapeo de servicios directamente a funciones específicas del lenguaje o llamadas a métodos.

Los WS pueden ser implementados siguiendo los conceptos de la Arquitectura Orientada a Servicios (Service-Oriented Architecture, SOA), donde la unidad básica de comunicación es el mensaje. Esto es típicamente referenciado como servicios orientados a mensajes. Los WS basados en SOA son soportados por una gran cantidad de desarrolladores de software y analistas. Al contrario que los Servicios Web basados en RPC, este estilo es débilmente acoplado, lo cual es preferible ya que se centra en el “contrato” proporcionado por el documento WSDL, más que en los detalles de implementación subyacentes.

En los últimos años se ha popularizado un estilo de arquitectura de Software conocido como REST (Representational State Transfer). Este nuevo estilo ha supuesto una nueva opción de estilo de uso de los Servicios

Web. Los WS basados en REST intentan emular al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar (GET, PUT, POST, DELETE, etcétera), por lo tanto, este estilo se centra más en interactuar con recursos con estado, que con mensajes y operaciones. En particular, el lanzamiento de REST como protocolo de intercambio y manipulación de datos en los servicios de internet produjo un gran cambio en el desarrollo de software en los últimos años. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por Roy Fielding, el padre de la especificación HTTP y uno de los referentes internacionales en todo lo relacionado con la Arquitectura de Redes [7].

En la actualidad existe una gran cantidad de proyectos o aplicaciones que disponen de una API REST para la creación de servicios profesionales. Twitter, YouTube, los sistemas de identificación con Facebook, y cientos de empresas generan negocio gracias a REST y las APIs REST. REST es una interfaz entre sistemas que usa HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML, JSON, HTTP, etc. En los últimos años logró un gran impacto en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de utilizar y sobre todo por su eficiencia [8].

El presente trabajo propone un mecanismo de generación de API REST a partir de versiones existentes de API Java, en el contexto de MDD, para la construcción de WS's. Aplicar esta técnica mediante la transformación de modelos se diferencia de otras formas convencionales, las cuales se basan en generar un Árbol de Sintaxis Abstracta (Abstract Syntax Tree AST) mediante algún parser de JAVA. Además, nuestra propuesta permite generar código hacia distintas implementaciones de WS REST a partir de un modelo JAVA.

2. LÍNEA DE INVESTIGACIÓN y DESARROLLO.

Para lograr la generación automática de API REST, aplicando transformación de modelos, es muy importante partir de un modelo origen bien definido.

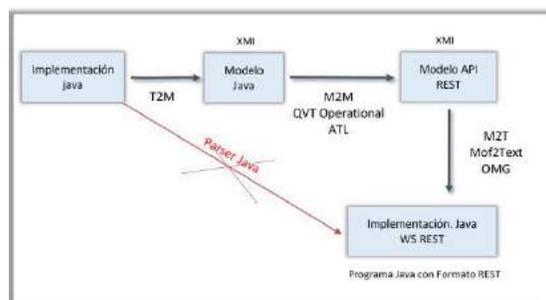


Figura 1: Etapas de transformación propuesta

En la figura 1, se muestra esquemáticamente nuestra propuesta, donde se define un proceso de transformaciones de la siguiente manera:

1. A partir de un programa java, con una interfaz bien definida, generar un modelo java (XMI) mediante una transformación texto a modelo (T2M).
2. Luego del modelo java (XMI), utilizando un lenguaje de transformación de modelo convencional (QVT Operacional, ATL; etc.), generar un modelo API REST (XMI) mediante una transformación de modelo a modelo (M2M).
3. Finalmente, del modelo API REST (XMI), generaremos el código java que implementará una API REST, a través de una transformación de modelo a texto (M2T).

Con la presente propuesta es sencillo diseñar aplicaciones que permitan la interacción entre sistemas heterogéneos tal como lo muestra la figura 2.

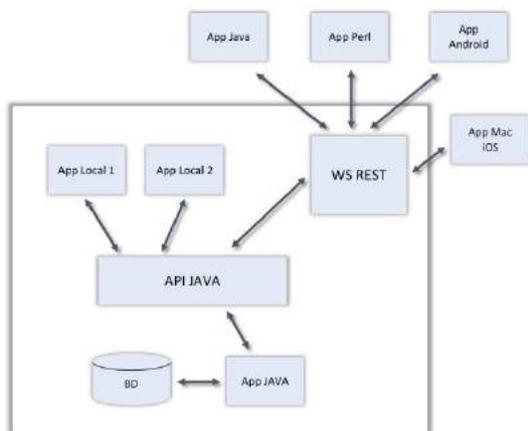


Figura 2: Diseño de un sistema utilizando API REST

En la actualidad existen trabajos que se vinculan. EMF-REST [9] es una herramienta que aprovecha las técnicas MDE para generar APIs Web RESTful a partir de los modelos EMF, promoviendo así la gestión de modelos en entornos distribuidos. La API Web RESTful generada se basa en bibliotecas y estándares bien conocidos con el fin de facilitar su comprensión y mantenibilidad. A diferencia de otros enfoques basados en el MDE para la generación de servicios web, EMF-REST proporciona un mapping directo para acceder a los modelos de datos mediante servicios Web siguiendo los principios de REST. Además, EMF-REST aprovecha características del modelo y de la Web, como el modelo Validación y seguridad, respectivamente.

En [10] los autores definen un mecanismo de diseño y descripción de API REST basado en formalismos de redes de Petri coloreadas, lo cual permite escalabilidad, extensibilidad e interoperabilidad. Además, permite una mayor facilidad para el testing por estar basado en un formalismo

3. RESULTADOS ESPERADOS

El proceso de transformaciones inicia con un programa escrito en el lenguaje JAVA, el cual es transformado a una representación de formato XMI, precisamente ECORE, conforme al metamodelo de dicho lenguaje de

programación. Este proceso es llevado a cabo utilizando la herramienta MoDisco.

Posteriormente, sobre dicho modelo se aplica una transformación Model-To-Model definida en QVT Operacional que nos permitirá generar una versión API REST del programa original. Finalmente, se obtiene el código JAVA que implementará la API REST a través de una transformación Model-To-Text definida con MOF2Text vía la herramienta Acceleo [11].

La contribución principal de este trabajo es proponer un mecanismo de transformación de programas JAVA, haciendo uso de estándares y herramientas existentes en el contexto de desarrollo dirigido por modelos, obteniendo como resultado una aplicación API REST que permite la interacción entre sistemas heterogéneos, promoviendo así la gestión de modelos en entornos distribuidos.

El mecanismo de generación de código es muy particular, dado que el mismo está basado en el tratamiento de los modelos JAVA (XMI), y no sobre el código fuente JAVA, dada la existencia de nuevas herramientas que permiten obtener dicho modelo, en caso contrario sería necesario manipular un parser del lenguaje.

4. FORMACION DE RECURSOS HUMANOS

Durante el desarrollo de esta línea de investigación, integrantes del grupo de trabajo están actualmente trabajando es su tesis de Licenciatura, Magister en Ingeniería de Software y Tesis de Doctorado.

También, se han formado ayudantes de segunda en las asignaturas de Análisis y Diseño de Sistemas, Ingeniería de Software, Base de Datos y Proyecto.

Los temas abordados en esta línea de investigación brindan un fuerte aporte al proceso de perfeccionamiento continuo de los autores de carreras de computación en Universidades Nacionales como del exterior.

5. BIBIOGRAFÍA

- [1] B. Selic, The pragmatics of model-driven development, IEEE Softw., vol. 20, no. 5, pp. 19-25, 2003.
- [2] Object Management Group, Object Management Group Std., Last access: May 2015. [Online]. Available: <http://www.omg.org>
- [3] J. Miller and J. Mukerji, Mda guide version 1.0.1, Object Management Group (OMG), Tech. Rep., 2003.
- [4] OMG, Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Version 1.1, Object Management Group Std., Rev. 1.1, 2011. [Online]. Available: <http://www.omg.org/spec/QVT/1.1/>
- [5] OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, Object Management Group Std., Rev. 2.4.1, 2011. [Online]. Available: <http://www.omg.org/spec/UML/2.4.1>
- [6] World Wide Web Consortium (W3C), World Wide Web Consortium (W3C) Std., último acceso: junio 2017. [Online]. Available: <http://www.w3.org/>
- [7] Architectural Styles and the Design of Network-based Software Architectures, UNIVERSITY OF CALIFORNIA, IRVINE Std., último acceso: junio 2017. [Online]. Available: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [8] L. Richardson and S. Ruby, Restful Web Services, 1st ed. O'Reilly, 2007.
- [9] H. Ed-Douibi, J. L. C. Izquierdo, A. Gómez, M. Tisi, and J. Cabot, "EMF-REST: generation of restful apis from models", CoRR, vol. abs/1504.03498, 2015. [Online]. Available: <http://arxiv.org/abs/1504.03498>
- [10] L. Li and W. Chou, "Design and describe rest api without violating rest: A petri net based approach." in ICWS. IEEE Computer Society, 2011, pp. 508-515. [Online]. Available: <http://dblp.uni-trier.de/db/conf/icws/icws2011.html#LiC11>
- [11] Acceleo, último acceso: mayo 2018. [Online]. Available: <https://www.eclipse.org/acceleo/>