

ModelSET: Soporte a Edición y Transformaciones de Modelos

Antonio Estévez García
E. Victor Sánchez Rebull
Francisco Vargas Ruiz
Orlando Avila-García

Open Canarias, S.L.
Santa Cruz de Tenerife
España

aestevez,vsanchez,francisco.vargas,orlando@opencanarias.com
<http://www.opencanarias.com>

Adolfo Sánchez-Barbudo Herrera
José Luis Roda García

Universidad de La Laguna
Santa Cruz de Tenerife
España

alu2526@etsii.ull.es, jlroda@ull.es
<http://www.taro.ull.es>

1. Introducción

Actualmente está cobrando gran importancia el Desarrollo de Software Dirigido por Modelos (DSDM) en el ámbito de la Ingeniería del Software. Object Management Group (OMG) ha propuesto Model Driven Architecture (MDA) [1], su estrategia DSDM basada en la independencia de plataforma y el uso de estándares como UML (Unified Modeling Language) y MOF (Meta-Object Facility) [2] entre otros.

Un aspecto fundamental de MDA es el concepto de transformación de modelos. Estas transformaciones permiten refinar los modelos que describen el sistema de negocio para el cual necesitamos una solución software, hasta la obtención de la aplicación final. MDA también recoge MOF QVT (Query/Views/Transformations) [3], un lenguaje estándar para la definición de transformaciones de modelos. QVT está compuesto por tres lenguajes: *Relations* y *Core* de naturaleza declarativa, y *Operational Mappings*, de naturaleza imperativa. Desgraciadamente, a pesar de que MOF QVT fue adoptado como estándar a finales de 2005, aún en la actualidad no disponemos de forma pública de herramientas que nos permitan definir transformaciones para ser ejecutadas y que se ajus-

ten enteramente a dicha especificación.

Por otro lado, en cuanto a la edición de modelos, en la actualidad existe una falta de consenso respecto a la utilización de MOF frente a los Perfiles UML para especificar y usar lenguajes de modelado específicos de dominio (DSML). Por un lado, MOF permite definir lenguajes independientes de UML y ajustados a un dominio específico de aplicación; por el otro, los Perfiles UML permiten extender/restringir los elementos del lenguaje genérico de modelado UML para que ofrezcan una semántica más precisa y poder ser usados así en dominios específicos. Desgraciadamente, no existe un cuerpo de conocimiento sobre las ventajas e inconvenientes de usar una u otra aproximación.

2. Demostración Presentada

En primer lugar, la demostración mostrará cómo somos capaces de transformar, con el motor de nuestra herramienta ModelSET, modelos en el contexto del framework de modelado EMF de Eclipse. Consistirá en partir de un fichero de texto cuyo contenido representa una transformación en el lenguaje *Operational Mappings* de QVT, y ejecutar esa misma transformación sobre unos mo-

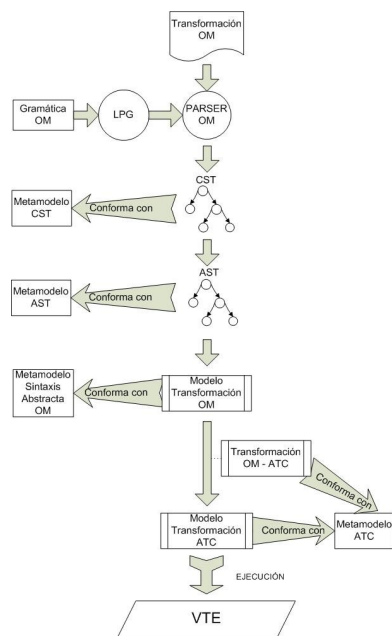


Figura 1: Esquema de todo el proceso. LPG son las siglas del generador de parsers utilizado.

delos de entrada para obtener los modelos de salida derivados, convenientemente transformados (ver Figura 1). La gran novedad que presenta esta solución es el reaprovechamiento del mismo motor, llamado VTE (Virtual Transformation Engine) para un sinfín de lenguajes de transformaciones. La compatibilización de lenguajes en VTE se canaliza mediante la traducción de los mismos a nuestro lenguaje de transformaciones de modelos intermedio, llamado ATC (Atomic Transformation Code), que funciona a modo de máquina virtual, y es el único lenguaje que el motor VTE sabe interpretar.

Añadiendo soporte específico de traducción a ATC para el resto de los lenguajes QVT se logrará por primera vez ofrecer una herramienta 100 % compatible con el estándar QVT para las transformaciones de modelos. El coste de incorporar nuevos lenguajes a este entorno se estima sensiblemente inferior al coste de reescribir un motor afin a cada lenguaje. Y gra-

cias a esta integración el usuario dispone de una mayor diversidad de opciones y acceso a repositorios de transformaciones de forma centralizada en una única herramienta.

En el ámbito del debate entre MOF frente a Perfiles UML, la demostración propondrá la utilización de los segundos como sintaxis concreta para lenguajes de modelado específicos de dominio (DSML), mientras que usamos MOF para especificar y manipular su sintaxis abstracta. En un ejemplo práctico, usaremos MSGF (ModelSET Graphical Framework) para generar, con mínimo esfuerzo, un editor gráfico basado en Perfiles UML para un DSML definido en MOF. Creemos que este ejemplo promoverá una interesante discusión sobre el papel que jugarán los perfiles UML frente a MOF.

3. Agradecimientos

Este trabajo ha sido subvencionado parcialmente por el *Ministerio de Educación y Ciencia* (PTQ2004-1495, PTR1995-0928-OP), el *Fondo Social Europeo* y la *DGUI, Consejería de Educación, Cultura y Deportes, Gobierno de Canarias* (PI042005/007). Gracias también a la *Red de Desarrollo de Software Dirigido por Modelos (DSDM)*, ref: TIN2005-25866-E.

Referencias

- [1] OMG. MDA guide version 1.0.1. Technical Report omg/2003-06-01, Jun 2003. Available at <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [2] OMG. Meta Object Facility (MOF) 2.0 core specification. Technical Report ad/2003-10-04, Apr 2003. Available at <http://www.omg.org/docs/ad/03-10-04.pdf>.
- [3] OMG. MOF 2.0 Query/Views/Transformations. Technical Report ptc/05-11-01, Nov 2005. Available at <http://www.omg.org/docs/ptc/05-11-01.pdf>.