

Evaluación de Transformaciones de Modelos

Corina Abdelahad, Enrique Miranda, Daniel Riesco

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950 – San Luis – Capital – Argentina

C.P.: 5700

Tel.: +54-0266- 4520300 – Int. 2102

[cabdelah, eamiranda, driesco]@unsl.edu.ar

Resumen

La Ingeniería Dirigida por Modelos está en auge desde hace algunos años y junto con ella la especificación de transformaciones se ha vuelto una de las tareas más importantes llevando a la creación de muchos lenguajes de transformación de modelos. ATL y QVT son algunos de los lenguajes de transformación más utilizados en la actualidad. Teniendo en cuenta el incremento en la utilización de estos tipos de lenguajes, es importante comenzar a considerar aspectos relacionados con la compresión, reutilización y mantenimiento de las transformaciones especificadas en estos lenguajes. Las estrategias para evaluar y medir transformaciones ayudan al Ingeniero de Software en las tareas de compresión, reutilización y posterior mantenimiento en las mismas.

El presente trabajo propone una línea de investigación que tiene como principal lineamiento el estudio, análisis y definición de frameworks de evaluación en el contexto de transformación de modelos.

Palabras claves: Métricas, Lenguajes de Transformación de Modelos, Framework de Evaluación, Mantenimiento.

Contexto

El presente trabajo se enmarca en el Proyecto de Investigación: Ingeniería de Software: Conceptos, Prácticas y Herramientas para el desarrollo de software de Calidad – Facultad de Ciencias Físico,

Matemáticas y Naturales, Universidad Nacional de San Luis. Dicho proyecto, es reconocido por el programa de incentivos, y es la continuación de diferentes proyectos de investigación de gran éxito a nivel nacional e internacional.

Introducción

La Ingeniería Dirigida por Modelos (Model-Driven Engineering-MDE) [1, 2] ofrece un escenario ideal para fortalecer el papel de la trazabilidad en el desarrollo de software. MDE propone un proceso de desarrollo de software en el cual la clave son los modelos y las transformaciones entre ellos. En este proceso, el software se desarrolla construyendo uno o más modelos, y a través de un refinamiento, mediante transformaciones, se generan otros modelos o código ejecutable. La transformación de modelos se refiere al proceso de transformación (relaciones y mapeo) de elementos de un modelo a elementos correspondientes de otro modelo. Los modelos están definidos por sus metamodelos, los cuales son los encargados de brindar la sintaxis para la construcción de los mismos. Es decir, un metamodelo define el lenguaje con el cual se construyen modelos. Una transformación de modelos posee como entrada un modelo acorde a un determinado metamodelo y produce como salida otro modelo acorde también a un determinado metamodelo. Más precisamente, y de acuerdo con la definición de Kleppe et al [3]:

“Una transformación es la generación automática de un modelo destino a partir de un modelo origen de acuerdo a una definición

de transformación".

Especificar transformaciones se ha vuelto una de las tareas más importantes en MDE. Esto se ve reflejado en la creación de muchos lenguajes de transformación de modelos como por ejemplo ATL y QVT[5, 7]. Debido al importante crecimiento en la utilización de dichos lenguajes, es imprescindible comenzar a contemplar tareas de mantenimiento y evolución en dichas transformaciones. En este contexto, resulta sumamente útil comenzar a contemplar y medir el esfuerzo que conlleva la comprensión, reutilización y mantenimiento [4] en las transformaciones. La definición de métricas en conjunto con su evaluación brindan información valiosa para el ingeniero de software a la hora de realizar estas actividades.

Líneas de Investigación, Desarrollo e Innovación

En general, al momento de medir y evaluar cualquier tipo de artefacto (ya sea, un programa, una técnica, un proyecto, etc.) de forma sistemática, es necesario definir una estrategia que contemple un conjunto de principios, métodos, técnicas y herramientas que permitan especificar, definir y recolectar métricas e indicadores y sus respectivos valores [9]. Además, con el objetivo de llevar a cabo el análisis y el proceso de toma de decisión, es necesario asegurar que las medidas y los valores de los indicadores sean repetibles y comparables. Por lo tanto, es mandatorio guardar no solo la información de la medición, sino también ciertos metadatos como procedimiento de medición, escala, tipo de escala, modelo de indicador elemental, niveles de aceptabilidad, entre otros.

Una métrica es la especificación de un proceso de medición que transforma un atributo de una entidad en una medida. Un indicador elemental es la especificación de un

proceso de evaluación, el cual recibe como entrada los valores correspondiente a una métrica en particular y produce un valor indicador (es decir, información contextual). Sin embargo en los distintos trabajos relacionados con la temática, en general, se le ha dado poca relevancia a la especificación de este tipo de conceptos. Así mismo, se puede observar una falta de consenso entre los diferentes estándares y manuscritos relativos a la medición y evaluación.

Esta línea de investigación está orientada al estudio, profundización y desarrollo de métricas y criterios los cuales pueden ayudar al ingeniero de software en la medición y evaluación de la comprensión, mantenimiento y reutilización de las transformaciones.

Resultados y Objetivos

A continuación se mencionan los resultados obtenidos hasta el momento dentro del marco de la línea de investigación planteada.

En primera instancia, se especificó un conjunto de métricas en el cual se ha utilizado un enfoque similar al propuesto por Olsina et. Al [6, 9]. En dichos trabajos, los autores proponen modelos y frameworks de calidad en conjunto con estrategias de evaluación que apuntan a un enfoque integrado para medir y evaluar diferentes áreas de las nuevas aplicaciones que se utilizan en la web.

Primeramente se definió un conjunto de atributos (o requerimientos) agrupados en categorías, subcategorías, etc. De esta manera se obtiene una estructura de árbol que pasa a ser el elemento principal del proceso de medición y evaluación. Para esta instancia de la investigación se desarrolló el árbol exhibido en la figura 1. El mismo ha sido desarrollado con el principal objetivo de medir y evaluar transformaciones en lenguaje QVT-R, desde el punto de vista del mantenimiento de software [10].

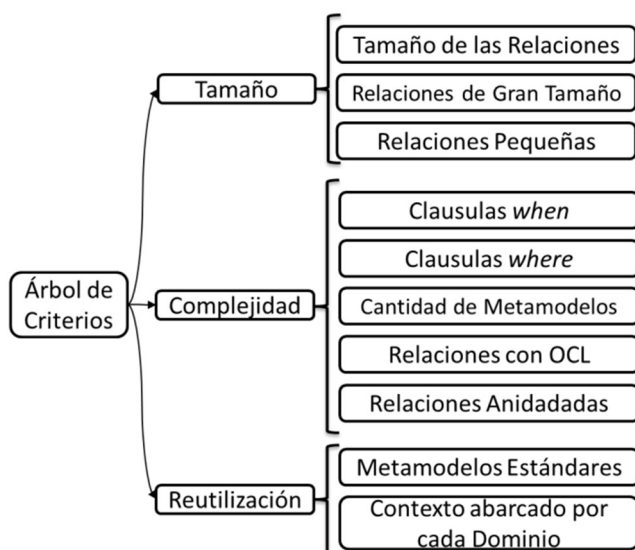


Figura 1: Árbol de Características y Atributos para evaluar transformaciones

El mismo está compuesto de 3 categorías de más alto nivel las cuales, a su vez, contienen atributos atómicos. Dichas categorías son *Tamaño*, *Complejidad* y *Reutilización*. En la categoría *Tamaño* se ha definido 3 atributos que pretenden medir la dificultad que conlleva la comprensión y mantenimiento de la transformación bajo estudio, desde el punto de vista del tamaño que presente la misma. Por otra parte, también se ha incluido la categoría *Complejidad*. La misma está compuesta de atributos que apuntan a medir desde ciertos aspectos sintácticos, la complejidad de una transformación. La última categoría se denomina *Reutilización*. Esta posee dos atributos que pretenden reflejar cuan reutilizable es el código que compone la transformación.

Una vez definido el árbol de atributos, es necesario establecer las métricas que permitirán cuantificar dichos atributos. Para diseñar una métrica, se debe definir el método de medición y el procedimiento de cómputo, en conjunto con la escala en donde se refleja la medición. El tipo de la escala depende de la naturaleza de las relaciones entre los valores de la misma; es decir, algunos aspectos que se pueden ver reflejados por medio de ciertas preguntas, como: los valores en la escala, ¿mantienen el orden y/o las distancias entre categorías? ¿Existe el cero (ausencia de

atributo medido)? entre otras. Entre los tipos de escalas más utilizadas en Ingeniería de Software se encuentran: nominal, ordinal, intervalos, porcentaje y absolutas. Esto es importante, ya que cada tipo de escala, determina el uso de operaciones matemáticas y técnicas estadísticas adecuadas para analizar la información.

Por motivos de extensión de este artículo se especifica un ejemplo de definición de la métrica: *Relaciones con OCL* [8](Categoría: *Complejidad*).

Atributo: Relaciones con OCL.

Definición: La métrica está orientada a medir la proporción de relaciones que usan expresiones OCL, sobre el total. Las relaciones que poseen expresiones OCL dificultan el entendimiento por parte del ingeniero de software, ya que el mismo no sólo debe comprender la semántica de las sentencias en el lenguaje de transformación, sino también la semántica de OCL. Claramente, si una transformación posee muchas relaciones con OCL, dificultará el proceso de comprensión y mantenimiento de la misma.

Métrica Indirecta: Proporción de Relaciones con Expresiones OCL (PROCL).

Objetivo: determinar la cantidad de relaciones de la transformación que poseen expresiones OCL en proporción a la cantidad total de relaciones.

Método de Cálculo:

$$P\ ROCL = (\#RelOCL / \#TotalRel) * 100$$

Escala Numérica: Continua.

Tipo de Valor: Real.

Tipo de Escala: Proporción.

Métricas Relacionadas:

- 1) Número Total de Relaciones con Expresiones OCL (#RelOCL);
- 2) Número Total de Relaciones (#TotalRel).

El conjunto de métricas definidas en la etapa previa será útil si se definen los indicadores adecuados en el proceso de evaluación. Un indicador permite especificar como calcular e interpretar los atributos especificados en el árbol en la Figura 1. A continuación se define el indicador para el

atributo *Relaciones con OCL*.

Atributo: Relaciones con OCL.

Indicador Elemental:

Nombre: Nivel de Desempeño en Relaciones con OCL (D_PROCL).

Modelo Elemental:

Especificación: el mapeo es de la siguiente manera:

$$D_PROCL = \begin{cases} 100 - PROCL & PROCL \leq 100 \\ 0 & PROCL \geq 100 \end{cases}$$

Donde $PROCL$ es la métrica indirecta definida anteriormente.

Criterios de Decisión:

Criterio 1: Insatisfactorio;

Rango: $0 \leq D_PROCL \leq 40$;

Criterio 2: Marginal;

Rango: $40 < D_PROCL \leq 60$;

Criterio 3: Satisfactorio;

Rango: $60 < D_PROCL \leq 100$;

Escala Numérica: Continua.

Tipo de Valor: Real.

Tipo de Escala: Proporción.

Finalmente, mientras el proceso de medición produce datos, el de evaluación retorna información contextual respecto de lo que se está evaluando. El resultado de este análisis se refleja en distintos tipos de reportes, documentos, sugerencias, de acuerdo al objeto de estudio. De esta manera se puede proveer un punto de partida para intervenir los puntos débiles que se detecten en el conjunto de requerimientos de calidad.

Utilizando el enfoque presentado previamente, se realizó un caso de estudio en donde se evalúa una transformación definida para la empresa especializada en optimización multi-objetivo ESTECO [11]. Dicha transformación permite la conversión de muchos flujos de trabajo ingenieriles definidos en el formato propietario de ESTECO al estándar de procesos de negocio BPMN2. Las reglas de transformación, han sido validadas experimentalmente aplicándolas en la empresa mencionada anteriormente.

Para este estudio, se han utilizado 3

intervalos de aceptabilidad en una escala de porcentaje para todos los indicadores elementales. Los valores pueden caer en los intervalos: i) Insatisfactorio, indicado con color rojo; ii) Marginal, en color amarillo; o iii) Satisfactorio determinado por el color verde.

Característica / Atributo	I.E %
Tamaño	
Tamaño de las Relaciones	0
Relaciones Grandes	78
Relaciones Pequeñas	86
Complejidad	
Cláusulas when	61
Cláusulas where	81
Cantidad de Metamodelos	14
Relaciones con OCL	100
Relaciones Anidadas	80
Reutilización	
Metamodelos Estándares	0
Contexto Abarcado por cada Dominio	65

Tabla 1. Indicadores Elementales para la transformación analizada. Lo valores están especificados en %.

Como resultado del análisis del caso de estudio se puede concluir los siguientes aspectos: 1) desde el punto de vista del *Tamaño*, las relaciones poseen, en promedio, más líneas de código que las estimadas para una relación grande, sin embargo la transformación posee pocas relaciones “pequeñas” y “extensas”; 2) desde un enfoque más general, la *Complejidad* de la transformación no muestra muchos atributos como *Insatisfactorios*, sin embargo, se puede notar que hace uso de muchos metamodelos, aspecto que desfavorece el proceso de comprensión; 3) con respecto a la *Reutilización*, es posible destacar lo siguiente: i) la transformación posee un metamodelo no estándar y esto perjudica su reutilización en otro contexto por parte de los ingenieros de software; ii) el contexto del dominio, teniendo en cuenta todos los metamodelos, que debe tener el ingeniero de software para comprender la transformación es *Marginal*, es decir, no está dentro de lo óptimo pero tampoco se considera voluminoso.

Dentro de los objetivos planteados a corto plazo se pretende: i) expandir el árbol de atributos abarcando otras características que reflejen aspectos del mantenimiento no contemplados en esta instancia de la investigación; ii) desarrollar estrategias de extracción automática de métricas en conjunto con la integración en un entorno de desarrollo como Eclipse o Netbeans; iii) integrar frameworks de evaluación de transformaciones de modelos con Ingeniería de Requerimientos.

Formación de Recursos Humanos

Las tareas realizadas en esta línea de investigación sirve como base para el desarrollo de tesis de posgrado, ya sea de doctorado o maestrías en Ingeniería de Software y desarrollo de trabajos finales de la carrera Licenciatura en Ciencias de la Computación e Ingeniería en Informática de la Universidad Nacional de San Luis, en el marco del Proyecto de Investigación.

Referencias

- [1] J.-M. Favre, J. Estublier, and M. Blay. Beyond MDA: Model Driven Engineering (L'Ingénierie Dirigée par les Modeles: audéla du MDA) Edition Hezmes-Lavoisier. Technical report, ISBN 2-7462-1213-7. Février, 2006.
- [2] M. Vo lter, T. Stahl, J. Bettin, A. Haase, S. Helsen, and K. Czarnecki. Model-Driven Software Development: Technology. Engineering, Management. Wiley, 5:6, 2006.
- [3] A. G. Kleppe, J. Warmer, and W. Bast. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [4] M. Petrenko, V. Rajlich, and V. R. Partial Domain Comprehension in Software Evolution and Maintenance. International Conference on Program Comprehension (ICPC08), pages 13–21, 2008.
- [5] QVT: <http://www.omg.org/spec/QVT/1.1>, 2017.
- [6] P. Becher, P. Lew, and L. Olsina. Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy. *Advances in Software Engineering*, 2012:2, 2012.
- [7] Jouault, F., Kurtev, I.: Transforming models with ATL. In: *Satellite Events at the MoDELS 2005 Conference*, pp. 128–138. Springer, Berlin (2006)
- [8] OCL: <http://www.omg.org/spec/OCL/>
- [9] L. Olsina, P. Lew, A. Dieser, and B. Rivera. Updating Quality Models for Evaluating New Generation Web Applications. *Journal of Web Engineering*, 11(3):209–246, 2012.
- [10] C. Abdelahad, E. Miranda, y N. Perez. Medición y Evaluación de Transformaciones a Nivel Metamodelo: un Enfoque Orientado al Mantenimiento. Congreso Nacional de Ingeniería Informática y Sistemas de Información – CoNaIISI'14: pag. 1083-1094.
- [11] ESTECO. <http://www.esteco.com>,