

Proceso de verificación de reglas de transformación basado en métricas*

Fernando Macías, Roberto Rodríguez-Echeverría, Víctor M. Pavón, José M. Conejero, and Fernando Sánchez-Figueroa

Universidad de Extremadura,
Quercus Software Engineering Group
{fernandomacias, rre, victorpavon, chemacm, fernando}@unex.es
<http://quercusseg.unex.es>

Resumen. La Ingeniería Dirigida por Modelos (IDM) se basa fundamentalmente en la definición de metamodelos, la edición de modelos y la especificación de transformaciones entre éstos (MMT, *Model-to-Model Transformations*). En muchos casos el desarrollo, evolución y adaptación de estas transformaciones se sigue realizando sin apoyarse en métodos o herramientas que permitan reducir el esfuerzo y los costes asociados a estas actividades. En este trabajo se presenta un proceso que permite medir el grado en que las reglas que implementan dichas transformaciones se adecúan a su especificación. Para ello, se plantea el proceso de verificación de MMT como un proceso de extracción de información aplicándose métricas ampliamente utilizadas en este tipo de escenarios. Este trabajo de verificación de MMT se ha desarrollado y probado en la adaptación de reglas de transformación dentro del proyecto MIGRARIA.

Palabras clave: MDD, transformación, métricas, MDS.

1 Introducción

El presente trabajo se enmarca dentro del proyecto MIGRARIA [1], que define un proceso de modernización de aplicaciones Web heredadas (AWHs) hacia clientes móviles y *Rich Internet Applications* (RIA). En la fase de ingeniería inversa de este proceso, se define un análisis estático de los artefactos software de la AWH mediante MMT con el fin de abstraer su información y estructurarla conforme a un metamodelo propio independiente de la tecnología: MIGRARIA MVC. Con el objetivo de reutilizar este análisis estático en otros casos de estudio, ha sido necesario adaptar o extender las reglas de transformación definidas para que sean capaces de soportar nuevos patrones de código, puesto que las guías de estilo de código utilizadas en el desarrollo de la AWH diferían de las contempladas previamente. En este contexto, surgió la necesidad de disponer

* Trabajo financiado por el Ministerio de Ciencia e Innovación, Proyecto TIN2011-27340; la Consejería de Empleo, Empresa e Innovación del Gobierno de Extremadura, Ayuda GR10129; y el Fondo Europeo para el Desarrollo Regional (FEDER)

de un proceso que ayudase a realizar la adaptación de la reglas señalando, por un lado, qué reglas faltaban o no estaban funcionando como se esperaba y, por otro lado, permitiendo comprobar de manera rápida que las reglas adaptadas funcionaban conforme a su especificación.

En este trabajo utilizamos el concepto de verificación definido en [2] como “la correspondencia entre la especificación y la implementación en un producto software”, para definir el problema presentado como un problema de verificación, en este caso de reglas de transformación modelo a modelo. La aproximación seguida en este trabajo parte de la premisa de que el problema planteado es asimilable a un problema de extracción de información y, por lo tanto, se pueden utilizar métricas bien conocidas dentro de ese ámbito [3]. Las tres métricas que se utilizarán en este planteamiento son: *precisión*, *exhaustividad* y *valor-F*, obtenidas para cada tipo de elemento que aparezca en el modelo de salida de las MMT. Para calcularlas, se partirá de datos estadísticos obtenidos mediante consultas sobre los modelos de entrada y salida. A partir de estos datos, se deducirán una serie de valores que, finalmente, se podrán utilizar para calcular las métricas. El análisis de estas métricas permitirá realizar la verificación de las MMT.

En resumen, en este trabajo planteamos un proceso de verificación de reglas de transformación mediante métricas, el cual sirve como apoyo a la hora de realizar la adaptación de reglas de transformación. Este proceso puede dividirse en dos fases: (1) la obtención de las métricas, que permiten la verificación de reglas de transformación mediante su interpretación, y (2) el uso de esta información para guiar el proceso de adaptación de las reglas. Este artículo se centra en la primera fase, ofreciendo una vista general del proceso de obtención y haciendo hincapié en la interpretación de los valores de las métricas que se obtienen como resultado de la aplicación de dicho proceso. El escenario de aplicación de partida, como ya se ha expuesto, es el proceso de ingeniería inversa definido en MIGRARIA, sin descartar su aplicación en otros escenarios. En este contexto, nos interesa verificar el correcto funcionamiento de las MMT, así como comprobar que el proceso de ingeniería inversa ha extraído correctamente toda la información que se esperaba. Las métricas planteadas nos permiten satisfacer el segundo objetivo y, en esta propuesta, queremos valorar su adecuación para la verificación de MMTs, como alternativa a las propuestas realizadas hasta ahora.

El resto de este artículo se divide de la siguiente forma. La sección 2 comenta algunos trabajos relacionados con éste. La sección 3 presenta la propuesta. Y la sección 4 apunta las conclusiones de este artículo y el trabajo futuro a realizar.

2 Trabajos relacionados

En esta sección se comentan otras técnicas para la verificación de las reglas de transformación con respecto a su especificación. La terminología utilizada para clasificarlas se ha obtenido de [4] y [5].

En primer lugar, encontramos técnicas de verificación formal basadas en el uso de restricciones OCL. En esta línea se encuentra [6], donde se combinan los modelos de entrada y salida, las transformaciones ejecutadas y las restricciones

OCL sobre ellos en un único modelo, sobre el que pueden definirse postcondiciones OCL que sirvan para encontrar contraejemplos en caso de error en la especificación. También definen planteamientos similares, haciendo uso de OCL, los trabajos de [7], que define restricciones en forma de *Tracts*, como una alternativa de carácter más generalista que la presentada en [8], la cual se centra en el *testing* de transformaciones mediante oráculos. De forma similar, en la propuesta presentada en [9] también se opta por un cambio de representación: se traducen reglas ATL y restricciones OCL sobre los metamodelos implicados a *First Order Logic*, que luego son verificados mediante solucionadores SMT.

En segundo lugar, también se han encontrado trabajos que definen técnicas informales para verificar la correspondencia estructural de los modelos de entrada y salida, como [10], centrado en generar avisos de error en el código de las reglas de transformación a partir de la especificación.

Según los criterios de los surveys mencionados al principio de esta sección, nuestra propuesta se clasifica como un método directo e informal para verificar la correspondencia estructural. Cabe destacar que, en este área, ninguno de los trabajos revisados aborda el problema de la verificación de reglas de transformación utilizando métricas, por lo que en la siguiente sección se presenta un método que parte de este planteamiento.

3 Propuesta

A continuación se ilustran las dos partes del proceso presentado: (1) la obtención de las métricas y (2) su interpretación.

3.1 Proceso de obtención de las métricas

Este proceso se ha implementado de forma manual en ATL sobre los casos de estudio utilizados en MIGRARIA. En este trabajo, por cuestiones de espacio, se ilustra mediante el conocido ejemplo *Families2Persons* [11]. La implementación en ATL de este ejemplo se encuentra accesible públicamente¹.

Los cinco pasos que conforman el proceso son:

1. **Selección de elementos del modelo de destino:** Los valores de las métricas se calculan para cada uno de los tipos de elemento en el modelo destino. En el ejemplo *Families2Persons*, estos tipos son *Male* y *Female*.
2. **Establecimiento de relaciones entre elementos de ambos modelos:** El segundo paso consiste en identificar todos los elementos en el modelo de entrada a partir de los cuales se genera cada uno de los tipos de elemento destino seleccionados en el paso anterior, información que puede obtenerse de la especificación de las reglas. Las dos relaciones detectadas en este ejemplo son entre *Male* y *Member* y entre *Female* y *Member*, ambas con cardinalidad 1:1 (un elemento de entrada genera un elemento de salida en ambos casos).

¹ <https://bitbucket.org/migraria/families2persons/src>

3. **Definición de criterios de exclusión:** Estos criterios deben servir para identificar los elementos del modelo de entrada que no deben ser transformados. En el ejemplo *Families2Persons*, se puede definir, por ejemplo, la exclusión de “todos los *Members* del modelo de entrada que no pertenezcan a ninguna familia”. En escenarios con modelos y transformaciones de mayor volumen y complejidad, estos criterios deben extraerse de fuentes documentales concretas, como por ejemplo guías de estilo de código en MIGRARIA.
4. **Definición de criterios de detección:** Estos criterios deben permitir comprobar si un elemento se ha transformado, es decir, encontrar los elementos que deberían haberse generado en el modelo destino a partir de un elemento de origen dado. Por ejemplo, para identificar un *Female* generado a partir de un *Member*, una comparación de nombre y apellido resulta suficiente. Esta información es posible extraerla también de la traza de la ejecución de las transformaciones, si está disponible.
5. **Cálculo de las métricas:** Finalmente, puede procederse a calcular los valores de precisión, exhaustividad y valor-F para cada elemento del modelo destino. Para ello, deben utilizarse los conjuntos de elementos (1) *positivos*, que cumplen los criterios de detección y no los de exclusión; (2) *falsos positivos*, que cumplen los criterios de exclusión y de detección; y (3) *falsos negativos*, que no cumplen los criterios de exclusión ni de detección. Las fórmulas para realizar este cálculo son:

$$Precisión = \frac{p}{p + fp} \quad Exhaustividad = \frac{p}{p + fn} \quad F_1 = 2 \cdot \frac{Pr \cdot Ex}{Pr + Ex}$$

La Tabla 1 muestra los valores obtenidos en el ejemplo utilizado. En este caso, se detectan falsos positivos en la generación de elementos *Male* que afectan a la precisión, puesto que en el modelo de entrada existe un elemento *Member* de género masculino sin un elemento *Family* asociado. En el resto de casos, los valores de las métricas son perfectos.

Tabla 1. Resumen del proceso de obtención de las métricas

Elemento origen	Elemento destino	Criterios exclusión	Excluidos	P	FP	FN	Precisión	Exhaustividad	Valor-F
<i>Male</i>	<i>Member (father/son)</i>	Sin familia	1	5	1	0	83.33 %	100.00 %	0.91
		Total	1						
<i>Female</i>	<i>Member (mother/daughter)</i>	Sin familia	0	4	0	0	100.00 %	100.00 %	1.00
		Total	0						

3.2 Interpretación de los resultados

Una vez calculadas las métricas se realiza el análisis de los resultados, el cual permite detectar errores o carencias en las reglas de transformación utilizadas. Puesto que éste es un paso fundamental de la propuesta, a continuación se incluye una relación de algunos de los resultados que más información pueden aportar en este sentido, junto con su interpretación:

- **Errores que afectan a la precisión.** Son causados por un falso positivo. Apuntan a un error en la selección de los elementos origen: se ha transformado un elemento que no debería haberlo hecho, como el caso comentado anteriormente entre *Male* y *Member*. Los valores de las métricas sugieren adecuar el filtro en la regla de transformación que genera elementos *Male* para evitarlo.
- **Errores que afectan a la exhaustividad.** Son causados por un falso negativo. Pueden indicar que la regla de transformación encargada de generar un tipo de elemento está incompleta o excluye indebidamente algunos elementos del origen, puesto que no se ha ejecutado una transformación que sí debería haberlo hecho.
- **Errores que afectan a ambas métricas simultáneamente.** Las dos métricas son independientes, por lo que la existencia de errores simultáneos en ambas indica simplemente que sus causas también ocurren a la vez: errores de filtrado y transformaciones incompletas. También puede indicar falta de expresividad del metamodelo destino.
- **Imposibilidad de calcular precisión.** En algunos casos, puede ser imposible calcular el valor de la precisión (y por tanto del valor-F), en los cuales además el valor de la exhaustividad es 0%. Se debe a que no existen elementos clasificados como positivos ni como falsos positivos, por no haberse transformado ningún elemento (exhaustividad 0%). La explicación más factible es que no existen reglas de transformación definidas para el caso dado.
- **Imposibilidad de calcular exhaustividad.** No existen elementos clasificados como positivos ni como falsos negativos: ningún elemento de entrada del tipo dado debe transformarse, de acuerdo con la especificación. No se necesitan, por tanto, reglas de transformación. Una combinación de este caso con la posibilidad de calcular la precisión, señalaría que las reglas de transformación están definidas, pero sólo generan falsos positivos. Por lo tanto, se sugiere eliminar dichas reglas.

4 Conclusiones y trabajo futuro

Este trabajo presenta un proceso de verificación de reglas de transformación mediante métricas propias de la extracción de información. Este proceso ha sido utilizado con éxito en procesos de adaptación de reglas de transformación, que especifican análisis estáticos de código, dentro de MIGRARIA. En este contexto de adaptación, se ha propuesto una interpretación de los resultados de las métricas que se estima puede ser de utilidad para la verificación de reglas de transformación.

Una de las conclusiones de su aplicación en MIGRARIA es que nuestra propuesta parece ser más adecuada para la verificación de MMTs cuando se trabaja con modelos de gran tamaño y se requiere asegurar que el modelo destino contiene toda la información esperada. Otras propuestas relacionadas, como por ejemplo el uso de oráculos, pueden conseguir los mismos objetivos. Sin embargo,

creemos que nuestra propuesta debería ser más eficiente en términos de esfuerzo y beneficio. En este sentido, dentro de las líneas de trabajo futuro, el primer paso que planteamos es realizar experimentos, a partir de los casos de estudio de MIGRARIA, que nos permitan comparar nuestra propuesta con las ya existentes y ver si se satisface esta hipótesis. Otra de las líneas de trabajo futuro consiste en estudiar la necesidad y adecuación de definir un DSL para simplificar la especificación del cálculo de las métricas (definir los elementos, sus relaciones y los criterios de exclusión y detección) e independizarla del lenguaje final. También se pretende analizar diferentes lenguajes ya existentes que puedan jugar este rol o servir como lenguaje final.

Referencias

1. Roberto Rodríguez-Echeverría, José M. Conejero, Pedro J. Clemente, and Juan Carlos Preciado. Migración dirigida por modelos de sistemas web heredados a rich internet applications. In *XVI Jornadas de Ingeniería del Software y Base de Datos*, 2011.
2. Barry W Boehm. Verifying and validating software requirements and design specifications. In *IEEE software*. Citeseer, 1984.
3. George Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.
4. Moussa Amrani, Levi Lucio, Gehan Selim, Benoit Combemale, Jürgen Dingel, Hans Vangheluwe, Yves Le Traon, and James R Cordy. A tridimensional approach for studying the formal verification of model transformations. In *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on*, pages 921–928. IEEE, 2012.
5. Lukman Ab Rahim and Jon Whittle. A survey of approaches for verifying model transformations. *Software & Systems Modeling*, pages 1–26, 2013.
6. Gehan MK Selim, Fabian Büttner, James R Cordy, Juergen Dingel, and Shige Wang. Automated verification of model transformations in the automotive industry. In *Model-Driven Engineering Languages and Systems*, pages 690–706. Springer, 2013.
7. Martin Gogolla and Antonio Vallecillo. Tractable model transformation testing. In *Modelling Foundations and Applications*, pages 221–235. Springer, 2011.
8. J-M Mottu, Benoit Baudry, and Yves Le Traon. Model transformation testing: oracle issue. In *Software Testing Verification and Validation Workshop, 2008. ICSTW'08. IEEE International Conference on*, pages 105–112. IEEE, 2008.
9. Fabian Büttner, Marina Egea, and Jordi Cabot. *On verifying ATL transformations using 'off-the-shelf' SMT solvers*. Springer, 2012.
10. Nikolaos Drivalos Matragkas, Dimitrios S Kolovos, Richard F Paige, and Athanasios Zolotas. A traceability-driven approach to model transformation testing. In *AMT@ MoDELS*, 2013.
11. ATL Team. *ATL/Tutorials - Create a simple ATL transformation*. Online: https://wiki.eclipse.org/ATL/Tutorials_-_Create_a_simple_ATL_transformation. Accessed: 10-04-2014.