

Añadiendo soporte para la generación de trazas en transformaciones ATL: impacto sobre la calidad

Iván Santiago, Juan M. Vara, Valeria de Castro, and Esperanza Marcos

Grupo de Investigación Kybele, Universidad Rey Juan Carlos,
Avda. Tulipán S/N, 28933 Móstoles, Madrid, España
{ivan.santiago,juanmanuel.vara,valeria.decastro,esperanza.marcos}@urjc.es
<http://www.kybele.es>

Resumen Los beneficios que aporta la gestión y el análisis de la información de trazabilidad, hacen que se convierta en una característica deseable en cualquier propuesta de ingeniería software. Más aún si su incorporación se hace de forma autónoma y sin incrementar el trabajo de los desarrolladores. No obstante, la incorporación de trazabilidad a un proyecto software siempre supone unos costes. En el contexto de la ingeniería dirigida por modelos, existen varias propuestas que permiten generar modelos de trazas a partir de transformaciones de modelos. Sin embargo ¿cuál es el coste de este proceso? ¿cómo afecta a la calidad de las transformaciones? En este trabajo tratamos de responder cuantitativamente a estas cuestiones para un caso concreto: el enriquecimiento de transformaciones ATL con capacidades de producción de modelos de trazas que soporta el framework *iTrace*.

Keywords: Trazabilidad, Ingeniería Dirigida por Modelos, Transformaciones de Modelos, Métricas de Calidad

1. Introducción

Los beneficios que aporta la gestión y el análisis de la información de trazabilidad, hacen que se convierta en una característica deseable en cualquier propuesta de ingeniería software. Una correcta identificación permite conocer cómo evolucionan los elementos de un sistema y cómo se relacionan entre sí a lo largo de su ciclo de vida [1]. Además, facilita el desempeño de actividades como el impacto del cambio, pruebas de regresión, validación de requisitos, y en general todas las tareas relativas al mantenimiento del software [2]. Sin embargo, se trata de una tarea tediosa, costosa y propensa a errores si no se disponen de herramientas adecuadas para tal fin [3].

Por tanto, sería deseable que cualquier propuesta para el desarrollo software ofreciera soporte a la gestión de la trazabilidad [4]. La llegada de la Ingeniería Dirigida por Modelos (*Model-Driven Engineering*, MDE) [5] ofrece un nuevo escenario para la gestión de la trazabilidad [6], lo que ha provocado la aparición de distintas herramientas que proponen soluciones. Un ejemplo de esta tendencia es *iTrace* [7], una herramienta para la gestión y el análisis de información de trazabilidad en el contexto de la MDE. No obstante, para poder aprovechar las ventajas que ofrece la trazabilidad, el primer paso es poder generar modelos de

trazas que almacenen las relaciones entre los distintos artefactos software. *iTrace* permite la generación automática de trazas a partir de motores de transformación que no generan trazas, o que al menos no las generan de forma explícita.

Sin embargo, ¿cuál es el coste del proceso de enriquecer las transformaciones? ¿cómo afecta a la calidad de las transformaciones resultantes? Para poder estimar este coste, necesitamos disponer de métricas que nos permitan cuantificar la calidad de las transformaciones. Desafortunadamente, los trabajos en este área se reducen casi exclusivamente a las aportaciones de van Amstel y van den Brand [8] [9] que proponen un conjunto de métricas para evaluar la calidad de las transformaciones escritas en lenguaje ATL (*Atlas Transformation Language*) [10] y presentan un estudio empírico donde tratan de relacionar las métricas con un conjunto de atributos de calidad, como completitud, entendibilidad, reusabilidad, etc.

No obstante, aunque estos trabajos relacionan la influencia de una determinada métrica o de un conjunto de métricas con un determinado atributo de calidad, no permiten cuantificar la calidad de una transformación ATL. Por ejemplo, podemos obtener los valores de una determinada métrica para dos transformaciones ATL y comparar sus valores, pero no podemos obtener un indicador global que permita determinar qué transformación tiene más calidad.

Partiendo de las métricas de van Amstel y van den Brand, este trabajo presenta una evaluación que mide el impacto en la calidad del proceso de enriquecimiento con capacidades de producción de modelos de trazas de transformaciones ATL, mediante la herramienta *iTrace*. Para poder estimar la calidad en las transformaciones ATL mediante un indicador, ha sido necesario seguir una heurística que dado un conjunto de transformaciones permita obtener sus respectivos indicadores de calidad. Concretamente, en la evaluación presentada en este trabajo, se mide la calidad de 7 transformaciones ATL con diferentes tamaños y complejidades en sus reglas de transformación; antes y después de ser enriquecidas.

El resto del trabajo se estructura de la siguiente forma. La sección 2 detalla el proceso de enriquecimiento de transformaciones ATL con capacidades de producción de modelos de trazas implementado en *iTrace*. La sección 3 introduce los trabajos de van Amstel y van den Brand en torno a la definición y obtención de métricas de calidad para transformaciones ATL. La sección 4 presenta la evaluación realizada para medir el impacto en la calidad del proceso de enriquecimiento de transformaciones ATL, describiendo la heurística aplicada para la obtención de indicadores globales de calidad. Finalmente, la sección 5 presenta las principales conclusiones derivadas de este trabajo.

2. Generación de modelos de trazas con *iTrace*

El primer paso para aprovechar las ventajas que aporta la gestión de trazabilidad en la ingeniería del software es disponer de trazas que hagan explícitas las relaciones entre los distintos artefactos. Por ello, es necesario disponer de los medios para generar estas trazas de forma automática. Desafortunadamente existen muchos proyectos que no proporcionan estas trazas. Incluso cuando estas son generadas, son completamente omitidas del proceso de desarrollo.

La herramienta *iTrace* permite la generación automática de trazas a partir de dos escenarios distintos. Por una parte; permite enriquecer las transformaciones existentes para aquellos motores de transformación que no generan trazas

(o al menos no lo hacen de forma explícita). Por otra parte; para los motores de transformación que generan sus propias trazas, ofrece un conjunto de transformaciones que permiten normalizarlos utilizando un meta-modelo común: el meta-modelo de trazabilidad *iTrace*. En este trabajo consideramos únicamente el primer escenario, es decir, la producción de trazas mediante el enriquecimiento de transformaciones existentes. Más concretamente, nos centraremos en la generación de trazas a partir de transformaciones ATL.

Desafortunadamente, ATL ofrece acceso limitado a los elementos de destino que generan las reglas de transformación. Por ejemplo, en la actual implementación de la máquina virtual de ATL (ATL-VM), no es posible seleccionar elementos de salida por su tipo. Además ATL-VM descarta la información de trazado después de la ejecución de las reglas de transformación. Esta situación obliga a los usuarios de ATL a extender sus reglas de transformación para construir y almacenar sus modelos de trazas [11]. No obstante, la extensión de reglas de transformación para generar modelos de trazas puede realizarse de forma automática mediante la utilización de Transformaciones de Orden Superior (*Higher-Order Transformations*, HOTs [12]), como demostró Jouault en [13]. En este prototipo inicial, Jouault muestra cómo añadir automáticamente elementos de salida adicionales a las reglas de transformación originales para generar el modelo de trazas deseado. Aunque inicialmente se implementó esta aproximación en *iTrace*, se comprobó que ante transformaciones suficientemente complejas el resultado no era el esperado. Además, optar por esta aproximación añadía una dependencia sobre distintas herramientas para su correcto funcionamiento como inyectores/extractores. Adicionalmente, el meta-modelo de *iTrace* es más rico que el meta-modelo manejado en la propuesta de Jouault, lo cual dificultaba aún más utilización de dicha técnica.

Por todo ello, en *iTrace* se ha optado finalmente por la implementación de un refactorizador que enriquezca las transformaciones con capacidades de producción de modelos de trazas. Aunque la aproximación pueda parecer más artesanal, es la que ofrece un mayor control sobre el resultado del proceso. La figura 1 ilustra el funcionamiento del refactorizador¹:

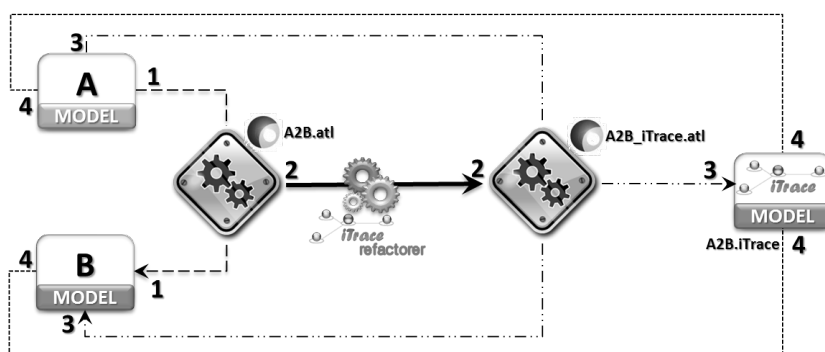


Figura 1. Proceso de enriquecimiento de transformaciones ATL con *iTrace*

¹ Puede verse un ejemplo de funcionamiento de la herramienta en: <http://www.kybele.etsii.urjc.es/itracetool/index.php/examples/>

- (1) Se parte de una transformación ATL (A2B.atl) que produce un modelo B consumiendo un modelo A.
- (2) El refactorizador añade a la transformación original las construcciones necesarias para la generación de trazas, dando lugar a la transformación enriquecida (A2B_iTrace.atl).
- (3) La ejecución de esta transformación enriquecida producirá, no solo el correspondiente modelo B de destino, sino también uno modelo de trazas (A2B_iTrace.atl).
- (4) Las trazas almacenadas en el modelo, explicitan las relaciones entre los elementos del modelo A y el modelo B.

Finalmente, el resultado del proceso de enriquecimiento de la transformación original puede observarse en la figura 2. En la parte izquierda, bajo el rótulo *Original Transformation* se muestra el código de la transformación original que implementa la regla de transformación `Package2Schema` de la transformación UML2SQL2003. A la derecha (bajo el rótulo *Enriched Transformation*) se muestra la misma regla de transformación después del proceso. El recuadro (1) resalta las líneas de la transformación original que son conservadas en la transformación enriquecida. El recuadro (2) identifica el conjunto de líneas de código que han sido generadas con el propósito de producir el modelo de trazas conjuntamente con el modelo de salida.

Original Transformation	Enriched Transformation
<pre> 42 rule Package2Schema { 43 from 44 p : UML!Package 45 to 46 s : SQL2003!Schema (47 name <- p.name 48) 49 } 50 </pre>	<pre> 120 rule Package2Schema { 121 from 122 p : UML!Package 123 to 124 s : SQL2003!Schema (125 name <- p.name 126) 127 128 -- Begin Added by iTrace 129 130 131 TraceLink : iTrace!M2MLink (132 ruleName <- 'Package2Schema', 133 comment <- 'Automatic generation by iTrace', 134 createdOn <- '17-11-2012', 135 mode <- 'Automatic', 136 technicalBinding <- 'ATL', 137 createdBy <- 'iTrace Tool', 138 type <- 'Transformation', 139 fromFileName <- '', 140 141 iTraceModel <- thisModule.getTraceModelRoot 142), 143 elementSource_p : iTrace!SourceElement (144 type <- p.ocltType().toString(), 145 traceLink <- TraceLink, 146 model <- thisModule.getModel_UML 147), 148 elementTarget_s : iTrace!TargetElement (149 type <- s.ocltType().toString(), 150 traceLink <- TraceLink, 151 model <- thisModule.getModel_SQL2003 152) 153 } 154 do { 155 elementSource_p.refSetValue('object', p); 156 elementTarget_s.refSetValue('object', s); 157 } 158 } 159 160 -- End Added by iTrace </pre>

Figura 2. Resultado de adición de capacidades de trazabilidad a la regla `Package2Schema` de la transformación UML2SQL2003

3. Métricas para transformaciones de modelos

No son demasiados trabajos que podemos encontrar en la literatura que proponen métricas para transformaciones de modelos. Y mucho menos centradas en calidad. Algunas propuestas en este sentido las encontramos en [14] [15] [16] [8] [9]. Tal vez, los trabajos más maduros en este área, son los propuestos por van Amstel y van den Brand [8] [9]. En [8] los autores proponen un conjunto de métricas con el objetivo de evaluar la calidad de las transformaciones ATL. Estas métricas son divididas en cuatro categorías: métricas de reglas, métricas de *helpers*², métricas de dependencia y métricas de miscelánea. Además implementan una herramienta llamada **ATL2Metrics** que permite la recolección automática de los valores de cada métrica para una transformación ATL dada. La arquitectura de **ATL2Metrics** puede observarse en la figura 3. Partiendo de una transformación ATL se obtiene un modelo de la transformación (ATL model) mediante un *parser* de ATL. A continuación, mediante una transformación (**Metrics extractor**) se produce un modelo de métricas a partir del modelo ATL. Finalmente se obtiene un fichero de métricas (en formato csv) a partir del modelo de métricas.

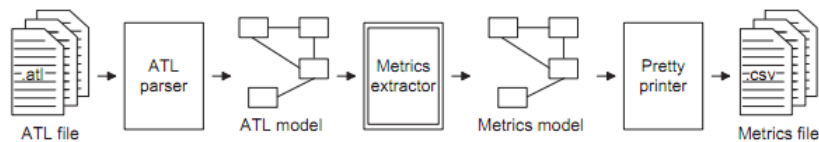


Figura 3. Arquitectura de la herramienta de extracción de métricas (obtenida de [8])

En un trabajo posterior [9] van Amstel y van den Brand utilizan las métricas definidas previamente para realizar un estudio cuyo objetivo principal es establecer una forma de medir la calidad de las transformaciones codificadas en lenguaje ATL. Para ello, trataron de establecer la relación entre el valor de las métricas extraídas de un conjunto de transformaciones ATL y un conjunto de atributos de calidad como entendibilidad, modificabilidad, completitud, consistencia, reusabilidad y concesión.

Para identificar esta relación, realizaron una encuesta a un grupo de desarrolladores que respondían a preguntas sobre la calidad de las transformaciones. De los resultados de la encuesta obtuvieron valores cuantitativos para los atributos de calidad de cada transformación. Para buscar una relación entre esos valores y los que obtuvieron al extraer las métricas de las transformaciones, utilizaron el coeficiente de correlación por rangos de Kendall.

Este test devuelve dos valores: el coeficiente de correlación *cc* y el valor de significancia *sig*. El coeficiente de correlación indica la fuerza y la dirección de la correlación. Es decir, un coeficiente de correlación positivo significa que hay una relación positiva entre el atributo de calidad y el valor devuelto por la métrica, y un coeficiente de correlación negativo implica una relación negativa. La

² Un *helper* puede ser visto como el equivalente a un método en Java, haciendo posible definir código ATL que puede ser llamado desde diferentes puntos de una transformación ATL

significancia indica la probabilidad de que esa correlación no exista en realidad. Es decir, la probabilidad de que no sea una correlación real sino algo casual. La tabla 1 muestra las correlaciones de Kendall obtenidas en el trabajo [9].

Métrica	Entend.		Modifi.		Comple.		Consis.		Reusab.		Concis.	
	cc	sig	cc	sig	cc	sig	cc	sig	cc	sig	cc	sig
# Elements per output pattern	-,375	,026	-,215	,202	-,228	,180	,124	,472	-,146	,389	-,122	,474
# Calls to resolveTemp()	-,352	,049	-,358	,045	-,159	,380	-,088	,632	-,236	,189	-,179	,323
# Calls to resolveTemp per rule	-,326	,068	-,306	,087	-,106	,558	-,061	,741	-,236	,189	-,153	,399
# Parameters per called rule	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Unused parameters per called rule	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
Called rule fan-in	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
Unit fan-in	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
Unit fan-out	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Input models	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Ouput models	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Units	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Unused helpers	-,407	,029	-,407	,029	-,391	,038	-,122	,524	-,345	,066	-,218	,249
# Times a unit is imported	-,318	,088	-,138	,459	-,379	,045	,086	,654	-,084	,656	-,247	,192
Lazy rule fan-in	-,356	,037	-,143	,404	-,397	,021	,005	,976	-,021	,905	-,062	,719
Helper cyclomatic complexity	-,248	,142	-,154	,364	-,357	,037	-,026	,882	-,175	,304	,126	,461
# Direct copies	,227	,197	,040	,822	,322	,070	-,059	,745	-,125	,478	-,196	,271
# Imported units	-,252	,164	-,080	,661	-,323	,078	,110	,554	-,027	,883	-,223	,225
Rule fan-out	-,223	,175	-,124	,453	-,333	,046	-,157	,351	-,235	,157	,024	,885
Helper fan-out	,020	,907	,183	,278	-,105	,537	,302	,081	,264	,120	,136	,426
# Transformation rules	,024	,885	-,086	,604	-,082	,623	-,364	,031	-,273	,099	-,092	,582
# Called rules	-,128	,482	-,263	,149	-,158	,389	-,308	,098	-,365	,046	-,347	,060
# Unused called rules	-,128	,482	-,263	,149	-,158	,389	-,308	,098	-,365	,046	-,347	,060
# Rules with filter	-,005	,977	-,038	,818	-,005	,977	-,049	,771	-,129	,435	-,402	,016
# Rules with local variables	-,013	,944	-,051	,780	,032	,861	-,137	,460	-,178	,328	,302	,100
# Rules per input pattern	-,109	,507	-,114	,489	-,130	,434	,029	,861	-,014	,931	,315	,059
# Unused input pattern elements	-,032	,854	,059	,737	-,033	,854	-,056	,758	,033	,854	,297	,097
# Variables per helper	,006	,972	,063	,727	,013	,944	-,111	,550	,178	,328	,328	,074
# Non-lazy matched rules	,014	,931	,000	1,000	,034	,839	-,029	,861	-,129	,435	-,383	,022
# Helpers per helper name (overloadings)	-,033	,855	-,066	,714	-,054	,769	,220	,237	,060	,741	,007	,971
# Variables per rule	-,038	,834	-,076	,676	,070	,700	-,111	,550	-,242	,184	,225	,220
Helper fan-in	,138	,402	,000	1,000	-,024	,885	-,236	,162	-,196	,236	-,005	,977
# Helpers	-,178	,296	-,229	,180	-,201	,243	-,047	,786	-,246	,152	,187	,281
# Unused lazy matched rules	-,025	,893	,138	,460	-,152	,419	,026	,892	,076	,687	,217	,251
# Rules with do-section	,000	1,000	-,153	,385	-,057	,747	,018	,922	-,108	,540	-,173	,332
# Lazy matched rules	-,081	,633	,041	,812	-,201	,243	-,058	,741	,051	,765	,239	,168
# Helpers per unit	-,178	,296	-,229	,180	-,201	,243	-,047	,786	-,246	,152	,187	,281

Tabla 1. Correlaciones de Kendall (obtenido de [9])

4. Evaluación

Esta sección presenta la evaluación que mide el impacto sufrido por la calidad de una transformación ATL, cuando esta es enriquecida con capacidades de producción de modelos de trazas. Para ello, partiendo de las métricas propuestas por van Amstel y van den Brand introducidas en la sección anterior, ha sido necesario aplicar una heurística que nos permite cuantificar el valor de calidad de cada transformación de la evaluación, antes y después de ser enriquecida por iTrace.

Para mejorar el rigor de esta evaluación, se ha definido un proceso a partir de las directrices propuestas por Runeson y Höst [17], compuesto por las siguientes etapas: selección de casos de estudio, diseño y ejecución, recolección de datos y, finalmente, análisis e interpretación de los resultados.

4.1. Selección de los casos de estudio

A continuación se presentan los 7 casos de estudio seleccionados para la evaluación. Vale la pena mencionar que hemos seleccionado estos casos de estudio ya que proporcionan transformaciones ATL con diferentes tamaños y niveles de complejidad.

La tabla 2 muestra información relativa a las transformaciones. La primera columna (*ID*) representa el identificador de la transformación, mientras que la segunda y tercera indican el nombre y el propósito respectivamente. La cuarta columna (*LOC*) indica el número de líneas de código y la quinta (*TR*) el número de reglas de la transformación. Finalmente la sexta columna (*IN/OUT*), indica el número de modelos de entrada y el número de modelos de salida. Tenga en cuenta que las tres últimas columnas de la tabla permiten hacernos una idea aproximada del tamaño de cada transformación.

ID	Transformación	Propósito	LOC	TR	IN/OUT
T1	ASD2WSDL	Produce un modelo WSDL (<i>Web Services Description Language</i>) a partir de un modelo ASD (<i>Abstract Service Description</i>).	236	13	1/1
T2	Class2Relational	Produce un modelo de base de datos relacional a partir de un modelo de clases UML.	112	6	1/1
T3	Families2Persons	Produce un modelo de personas a partir de un modelo de familias.	46	2	1/1
T4	SQL20032ORDB4ORA	Produce un modelo de bases de datos objeto relacional para el producto Oracle a partir del modelo SQL2003.	1247	51	1/1
T5	UML2SQL2003	Produce un modelo estándar SQL2003 a partir de un diagrama de clases UML y un modelo de anotación AMW (<i>Atlas Model Weaver</i>) que introduce decisiones de diseño del proceso de transformación.	2181	66	2/1
T6	UML2XMLSchema	Produce un esquema XML a partir de un modelo UML y un modelo de anotación AMW.	459	13	2/1
T7	WSDL2ASD	Produce un modelo ASD a partir de un modelo WSDL.	190	9	1/1
TOTAL			4471	160	9/7

Tabla 2. Transformaciones ATL seleccionadas para la evaluación

4.2. Diseño y ejecución

La evaluación ha diseñado siguiendo los pasos que se enumeran a continuación:

- (1) La transformación original es analizada y ejecutada recogiendo el número de líneas y el tiempo de ejecución.
- (2) Mediante la herramienta *ATL2Metrics* se obtienen los valores de las distintas métricas para la transformación.
- (3) La transformación es enriquecida por *iTrace* para producir modelos de trazas. En este paso se recoge el tiempo empleado por *iTrace* en enriquecer la transformación.

- (4) La transformación enriquecida es analizada para determinar el número de líneas de código y ejecutada para recoger el tiempo de ejecución.
- (5) Se obtienen las métricas de la transformación enriquecida mediante **ATL2-Metrics**.
- (6) Los pasos 1, 2, 3, 4 y 5 se repiten por cada una de las 7 transformaciones que componen la evaluación.
- (7) Finalmente, la información recogida es analizada.

Una vez recogidos los valores obtenidos por cada transformación en cada una de las métricas, pasamos a describir la heurística utilizada para obtener un indicador global de calidad, que nos permita determinar cómo afecta el proceso de enriquecimiento a la calidad.

El punto de partida de la heurística, es la tabla correlaciones de Kendall (ver tabla 1). Nótese que existen algunas celdas sombreadas en la tabla. Estas celdas representan los valores de significancia menores o iguales que 0,100 ya que establecemos un nivel de confianza del 90%. Por lo tanto, de las 36 métricas iniciales, en esta evaluación únicamente se utilizarán 28. Para estimar el valor de los atributos de calidad de cada transformación, hemos seguido la siguiente heurística:

Sea n el número de métricas utilizadas, p el número de atributos que permiten medir la calidad y k el número de transformaciones cuyos atributos queremos estimar. Sea $X \in [-1, -1]^{n \times p}$ la matriz que contiene los coeficientes de correlación de Kendall para cada par de métricas y atributos. Sea $Y \in \mathbb{R}^{n \times k}$ la matriz que contiene las métricas para cada transformación. El objetivo es estimar una matriz $\tilde{Z} \in [0, -1]^{p \times k}$ con los atributos para cada transformación.

Por tanto, calculamos Z como el promedio ponderado de las correspondientes métricas donde los pesos vienen dados por los coeficiente de correlación, siguiendo la siguiente ecuación:

$$Z_{jl} = \frac{\sum_{i=1}^n Y_{il} \cdot X_{ij}}{\sum_{i=1}^n |X_{ij}|} \quad (1)$$

Finalmente, queremos normalizar la escala de valores de la matriz Z , para que cada rango de la columna (valor de los atributos) este comprendido entre 0 y 1. Para ello, sea \tilde{Z} la matriz normalizada mediante la siguiente ecuación:

$$\tilde{Z}_{jl} = \frac{Z_{jl} - \min(Y_{.l})}{\max(Y_{.l}) - \min(Y_{.l})} \quad (2)$$

4.3. Recolección datos

En esta sección, presentamos los datos obtenidos tras aplicar la heurística a los diferentes valores obtenidos en las métricas por las transformaciones objeto de evaluación. Para el cálculo de estos valores se ha utilizado la herramienta estadística R³.

La tabla 3 recoge los resultados de la evaluación. La primera columna indica la transformación evaluada. Nótese que para cada par de filas aparece consecutivamente la transformación original (Tx) y la transformación enriquecida (Tx').

³ The R Project for Statical Computing. Disponible en: <http://cran.r-project.org>

Por ejemplo, T1 representa la transformación ASD2WSDL, mientras que T1' representa la misma transformación enriquecida con producción de modelos de trazas, es decir, ASD2WSDL_iTrace.

Las seis columnas siguientes recogen la puntuación obtenida por cada transformación en los distintos atributos de calidad. Nótese que la fila que representa la transformación enriquecida incorpora un valor adicional. Este valor indica la diferencia de puntuación obtenida entre la transformación original y la transformación enriquecida. Por ejemplo, para el atributo de *entendibilidad* la transformación T1 obtiene 0,93 puntos, mientras que la transformación enriquecida (T1') obtiene 0,75 puntos. Nótese que el valor que acompaña a esta columna es de -18,43 % (0,18) que determina la variación porcentual de los puntos obtenidos por la transformación enriquecida respecto a la transformación original.

La última columna recoge el indicador global de calidad, calculado como la media aritmética de los puntos obtenidos por cada transformación en los diferentes atributos. Al igual que antes, la puntuación para la transformación enriquecida se acompaña de un valor adicional que indica la variación entre la transformación original y la transformación enriquecida. La última fila de la tabla, muestra la media aritmética de los valores por columnas, representado las puntuaciones (en negrita) y las variaciones (en cursiva) medias por atributo, y por indicador de calidad (las dos últimas celdas).

Transf.	Entend.	Modifi.	Comple.	Consis.	Reusab.	Conces.	Calidad
T1	0,93	1,00	0,96	0,88	0,88	0,96	0,93
T1'	0,75 <i>-18,43 %</i>	0,90 <i>-9,89 %</i>	0,60 <i>-36,64 %</i>	0,71 <i>-17,04 %</i>	0,70 <i>-18,10 %</i>	0,71 <i>-24,47 %</i>	0,73 <i>-20,76 %</i>
T2	0,97	1,00	1,00	0,96	0,96	0,97	0,98
T2'	0,80 <i>-16,79 %</i>	0,90 <i>-9,89 %</i>	0,69 <i>-30,95 %</i>	0,78 <i>-17,04 %</i>	0,78 <i>-18,10 %</i>	0,73 <i>-24,47 %</i>	0,78 <i>-19,54 %</i>
T3	1,00	1,00	0,84	1,00	1,00	0,99	0,97
T3'	0,88 <i>-12,36 %</i>	0,90 <i>-9,89 %</i>	0,65 <i>-19,05 %</i>	0,83 <i>-17,04 %</i>	0,82 <i>-18,10 %</i>	0,75 <i>-24,47 %</i>	0,80 <i>-16,82 %</i>
T4	0,93	1,00	0,55	0,38	0,40	0,66	0,65
T4'	0,71 <i>-21,26 %</i>	0,91 <i>-8,90 %</i>	0,34 <i>-21,13 %</i>	0,21 <i>-17,04 %</i>	0,22 <i>-17,95 %</i>	0,40 <i>-25,85 %</i>	0,47 <i>-18,69 %</i>
T5	0,22	0,09	0,71	0,22	0,22	0,31	0,29
T5'	0,00 <i>-22,10 %</i>	0,00 <i>-9,44 %</i>	0,33 <i>-37,18 %</i>	0,00 <i>-21,58 %</i>	0,00 <i>-22,47 %</i>	0,00 <i>-30,59 %</i>	0,06 <i>-23,89 %</i>
T6	0,47	0,37	0,32	0,89	0,85	0,90	0,63
T6'	0,26 <i>-20,29 %</i>	0,27 <i>-9,82 %</i>	0,00 <i>-32,02 %</i>	0,67 <i>-21,73 %</i>	0,62 <i>-22,60 %</i>	0,59 <i>-30,60 %</i>	0,40 <i>-22,84 %</i>
T7	0,97	1,00	1,00	0,92	0,92	1,00	0,97
T7'	0,80 <i>-16,67 %</i>	0,90 <i>-9,89 %</i>	0,64 <i>-35,95 %</i>	0,75 <i>-17,04 %</i>	0,74 <i>-18,10 %</i>	0,76 <i>-24,47 %</i>	0,76 <i>-20,35 %</i>
Promedio	0,69 <i>-18,27 %</i>	0,73 <i>-9,68 %</i>	0,62 <i>-30,42 %</i>	0,66 <i>-18,36 %</i>	0,65 <i>-19,34 %</i>	0,69 <i>-26,42 %</i>	0,67 <i>-20,41 %</i>

Las abreviaturas de las transformaciones puede consultarse en la tabla 2

Tabla 3. Evaluación de calidad de las transformaciones

4.4. Análisis e interpretación de los resultados

Para facilitar el análisis e interpretación de los resultados obtenidos, se han agrupado en varios apartados. En primer lugar se presenta un análisis general de resultados y posteriormente se analizan por los atributos de calidad.

Comentarios generales. A la vista de los resultados recogidos en la tabla 3 la primera conclusión que podemos obtener es que la pérdida media de calidad en las transformaciones enriquecidas se estima en torno al 20 % (20,41 %). Este valor, se acentúa en transformaciones que ya presentaban un peor valor de calidad

en la transformación original. Por ejemplo, la transformación *UML2SQL2003* (T5) presentaba el peor indicador de calidad 0,29 antes del proceso de enriquecimiento, y es la transformación que mayor pérdida sufre (23,89%). En cambio, la transformación con mejor indicador de calidad *Families2Persons* (T3) obtiene un valor de 0,97, reduciendo su calidad en un 16,82% tras el proceso. Podríamos intuir de antemano un decremento en la calidad de las transformaciones enriquecidas, ya que este proceso conlleva la incorporación de varias líneas de código adicionales, así como de funciones auxiliares que permitan a la transformación original producir los correspondientes modelos de trazas. Por ello, este valor parece bastante razonable. Además la pérdida de calidad es más acusada en transformaciones más grandes (como el caso de las transformaciones T5 y T6), ya que es necesario incorporar más instrucciones a la transformación enriquecida pues están compuestas de más reglas de transformación.

Complejidad. Por otra parte, si analizamos por separado los atributos de calidad, podemos destacar que el atributo que más acusa la incorporación de mecanismos para la generación de trazas es la *complejidad*⁴, con una pérdida media del 30,42%.

La complejidad comparte varias métricas con otros atributos como entendibilidad, modificabilidad y reusabilidad, como puede observarse en la tabla de correlaciones de Kendall (ver tabla 1). Sin embargo, los valores de complejidad, no siguen la misma tendencia que los valores de estos atributos que a continuación analizaremos. Este hecho nos hace pensar que las métricas que determinan el valor de complejidad son aquellas métricas que únicamente están relacionadas con este atributo: *Helper cyclomatic complexity*, *# Direct copies*, *# Imported units* y *Rule fan-out*. Más concretamente, podríamos atribuir esta pérdida casi de forma exclusiva a la métrica *Rule fan-out*, que indica el promedio de veces que una regla de transformación llama a otra regla de transformación o *helper*.

El proceso de enriquecimiento de *iTrace* añade una llamada a una regla auxiliar por cada elemento del patrón origen y/o destino de la regla de transformación original. Esto hace que, las reglas de transformación originales que antes del proceso de enriquecimiento eran completas (podían producir elementos de salida a partir de los elementos de entrada sin necesidad de reglas externas) pierdan esta capacidad.

Modificabilidad. Al contrario de lo que sucede con la complejidad, el atributo menos afectado por el proceso de enriquecimiento es la *modificabilidad*⁵ con una pérdida media del 9.68%. Este atributo está condicionado por el número unidades, modelos de entrada y/o salida, así como el uso de construcciones que aumenten el nivel de acoplamiento (por ejemplo el uso de operaciones `resolveTemp()`) y llamadas a estas. El enriquecimiento de transformaciones únicamente conlleva la producción de un modelo adicional de salida, el modelo de trazas, pero no

⁴ La complejidad determina la capacidad de una transformación para relacionar por si sola los elementos origen con sus correspondiente elementos de destino sin acudir a otras transformaciones

⁵ La modificabilidad determina la facilidad con la que una transformación puede acomodar cambios

hace uso de unidades adicionales ni de construcciones que aumentan el nivel de acoplamiento. Esto hace que la modificabilidad no se vea demasiado afectada por el proceso de incorporación de trazabilidad.

A diferencia de la completitud, la modificabilidad no depende del tamaño de la transformación, como puede observarse en los resultados (todas las transformaciones sufren aproximadamente la misma pérdida, en torno al 9,68 %). Este dato es justificado por el hecho de que todas las transformaciones evaluadas producen únicamente un modelo de salida, mientras que las transformaciones enriquecidas producen dos modelos de salida.

Consistencia, reusabilidad y concisión. Una tendencia común observada en los resultados es la constancia en la pérdida de puntos para ciertos atributos tras el proceso de enriquecimiento. Por ejemplo, para la *consistencia*⁶ observamos dos grupos de valores: transformaciones con una pérdida del 17,04 % y transformaciones con pérdida entorno al 21,65 %. Las transformaciones incluidas en el primer grupo son T1, T2, T3, T4 y T7, mientras que las transformaciones del segundo grupo son T5 y T6. Si buscamos un criterio que nos permita agrupar de la misma manera las transformaciones, tenemos que utilizar el número de modelos de entrada. Para el primer grupo, todas las transformaciones tienen un único modelo de entrada, mientras que las transformaciones del segundo grupo, tienen dos modelos de entrada. Este comportamiento puede ser extendido para los atributos *reusabilidad*⁷ y *concisión*⁸, donde también podríamos establecer estos mismos grupos.

Sin embargo, como podemos observar en las correlaciones de Kendall (tabla 1) la métrica # *Input models*, que mide el número de modelos de entrada, no afecta a la puntuación de los atributos, ya que tiene una significancia del 0,524. Para explicar esta tendencia en los datos, acudimos de nuevo a las correlaciones de Kendall para buscar las métricas comunes a los tres atributos. Según se observa en la tabla 1 se trata de las métricas # *Called rules* y # *Unused called rules*.

Entonces, ¿cómo se explica esta tendencia? Esta tendencia es explicada por el proceso implementado por *iTrace* para enriquecer las transformaciones. *iTrace* crea por cada modelo de entrada una *Called Rule* que se utiliza para asignar cada elemento del modelo con su correspondiente modelo. Lógicamente el número de *Called Rules* depende del número de modelos de entrada. Como decíamos anteriormente, las transformaciones del experimento pueden agruparse en transformaciones con un modelo de entrada (T1, T2, T3, T4 y T7), y transformaciones con dos modelos de entrada (T5 y T6). Por tanto, las transformaciones pertenecientes al segundo grupo serán más penalizadas (en cuanto a los atributos de consistencia, reusabilidad y concisión) cuando son enriquecidas con capacidades de producción de modelos de trazas.

En resumen, podemos decir que el número de modelos de entrada en una transformación ATL determina significativamente su calidad (al menos, basán-

⁶ La consistencia determina el uso de un diseño uniforme y de técnicas de documentación en la transformación

⁷ La reusabilidad determina la capacidad de poder volver a usar parte de una transformación en otra

⁸ La concisión determina lo compacta que resulta ser la transformación en términos de líneas de código.

donos en las métricas propuestas por van Amstel y van den Brand) y de manera particular en los atributos de consistencia, reusabilidad y concisión.

Entendibilidad. La **entendibilidad**⁹ es el atributo evaluado que presenta pérdidas más dispares cuando las transformaciones son enriquecidas. Como puede observarse en la tabla de correlaciones de Kendall (ver tabla 1) comparte un amplio conjunto de métricas con los atributos de modificabilidad, completitud y reusabilidad. Sin embargo estos atributos, como hemos visto anteriormente, no siguen el mismo patrón de pérdidas. Esto nos lleva a pensar que la métrica que justifica estos valores es **# Elements per output pattern**, al tratarse de una métrica que únicamente determina el valor del atributo entendibilidad. Esta métrica mide el número de elementos por patrones de salida.

El proceso de enriquecimiento de transformaciones implementado por **iTrace** necesita añadir a cada regla de transformación un elemento en el patrón de salida por cada elemento que exista tanto en el patrón origen como en el patrón destino del transformación original, a fin de almacenar su referencia, su tipo, el enlace de traza que lo conecta y el modelo al que pertenece. Además se genera en el patrón de salida de cada regla un elemento enlace de traza, para conectar los elementos del patrón de entrada con los elementos del patrón de salida.

Volvemos a visitar el ejemplo presentado en la figura 2 para ilustrar esta explicación. El código mostrado representa el resultado del proceso de enriquecimiento de la regla de transformación **Package2Schema** correspondiente a la transformación **UML2SQL2003**. Como puede observarse en la parte izquierda de la figura, esta regla de transformación tiene dos elementos: el elemento **p** en el patrón de entrada y el elemento **s** en el patrón de salida. En cambio, la transformación enriquecida (mostrada en la parte derecha de la figura) conserva el elemento **p** como elemento en el patrón de entrada y el elemento **s** como elemento en el patrón de salida, pero además se han añadido el elemento **TraceLink** para conectar los elementos de entrada y salida, el elemento **elementSource_p** correspondiente al elemento **p** del patrón de entrada y el elemento **elementTarget_s**, correspondiente al elemento **s** del patrón de salida. Es decir, hemos pasado de tener un elemento en el patrón de salida en la transformación original a cuatro elementos en la transformación enriquecida.

En conclusión, podemos decir que el proceso de enriquecimiento de transformaciones de **iTrace** hace que estas sean menos comprensibles debido a que necesita incrementar el número de elementos en el patrón de salida.

5. Conclusiones

En este trabajo hemos presentado una evaluación que mide la pérdida de calidad que supone incorporar mecanismos de generación de modelos de trazas en transformaciones ATL mediante la herramienta **iTrace**.

Apoyándonos en las métricas propuestas por van Amstel y van den Brand, y en su estudio que relaciona dichas métricas con un conjunto de atributos de calidad, hemos seguido una heurística que permite dado un conjunto de transformaciones ATL obtener su indicador global de calidad, así como indicadores

⁹ La entendibilidad determina la facilidad para comprender el código de una transformación

parciales de sus atributos: entendibilidad, modificabilidad, completitud, consistencia, reusabilidad y concisión.

El proceso de enriquecimiento de **iTrace** conlleva la incorporación de instrucciones adicionales a la transformación original que disminuyen su calidad en torno al 20 %. Lógicamente cuando la transformación original es más grande (por ejemplo, tiene mayor número de reglas de transformación) esta pérdida es más acusada. De igual modo, el proceso implica la incorporación de nuevos elementos en el patrón de salida de las reglas de transformación y las hace depender de reglas auxiliares que añade **iTrace**, lo cual hace que las transformaciones tengan menos completitud y entendibilidad.

El número de modelos (de entrada y/o salida) involucrados en la transformación original, es un factor determinante en el impacto de la calidad, al menos respecto a los atributos de consistencia, reusabilidad y concisión. En realidad estos atributos están directamente relacionados con el número de reglas que son explícitamente invocadas. **iTrace** por su parte crea, para enriquecer las transformaciones, una regla por cada modelo involucrado que tiene que ser explícitamente invocada para vincular cada elemento con su modelo correspondiente. Por lo tanto, cuanto mayor sea el número de modelos involucrados en la transformación, mayor será el número de reglas explícitamente invocadas, lo cual implica que mayor será la penalización en los atributos de consistencia, reusabilidad y concisión.

La modificabilidad es el atributo menos afectado por el proceso de enriquecimiento, ya que está condicionado entre otros, por el número de modelos de salida y el uso construcciones que aumentan el nivel de acoplamiento. El proceso de enriquecimiento de **iTrace** únicamente conlleva la producción de un modelo adicional de salida, el modelo de trazas, pero no hace uso de construcciones que aumentan el nivel de acoplamiento. Merece la pena destacar que la modificabilidad no depende del número de reglas de transformación que presenta la transformación original, ya que las 7 transformaciones evaluadas obtienen la misma puntuación de modificabilidad y presentan distinto número de reglas de transformación.

Agradecimientos

Este trabajo se ha llevado a cabo en el marco del proyecto MASAI (Ref. TIN-2011-22617) financiado por el Ministerio de Ciencia y Tecnología de España.

Referencias

1. Hazeline U. Asunción. Towards practical software traceability. In *Companion of the 30th international conference on Software engineering, ICSE Companion '08*, pp. 1023–1026, New York, NY, USA, 2008. ACM.
2. N. Aizenbud-Reshef, B.T. Nolan, J. Rubin, and Y. Shaham-Gafni. Model traceability. *IBM Systems Journal*, 45(3): pp. 515–526, 2006.
3. R Oliveto. Traceability Management meets Information Retrieval Methods - Strengths and Limitations. In *12th European Conference on Software Maintenance and Reengineering, 2008. CSMR 2008.*, pp. 302–305, 2008.
4. George Spanoudakis and Andrea Zisman. Software Traceability: A Roadmap. S. K. Chang, editor, *Handbook of Software Engineering and Knowledge Engineering*, volume 3, pp. 395–428. World Scientific Publishing Co., 2005.

5. Douglas C Schmidt. Model-Driven Engineering. *IEEE Computer*, 39(2): pp. 25–31, 2006.
6. Iván Santiago, Álvaro Jiménez, Juan Manuel Vara, Valeria De Castro, Verónica Bollati, and Esperanza Marcos. Model-Driven Engineering As a New Landscape For Traceability Management: A Systematic Review. *Information and Software Technology*, 54(12): pp. 1340–1356, 2012.
7. Iván Santiago, Juan M. Vara, Valeria de Castro, and Esperanza Marcos. *iTrace: un framework para soportar el análisis de información de trazabilidad en proyectos de Desarrollo Software Dirigidos por Modelos*. In XVII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2012), pp. 491–504, 2012.
8. Marcel F van Amstel and Mark G van den Brand. Quality assessment of ATL model transformations using metrics. In *Proceedings of the 3rd International Workshop on Model Transformation with ATL*, volume 711, pp. 19–33, 2010.
9. Marcel F van Amstel and Mark G van den Brand. Using metrics for assessing the quality of ATL model transformations. In *Proceedings of the 3rd International Workshop on Model Transformation with ATL*, volume 742, pp. 20–34, 2011.
10. Frédéric Jouault, Freddy Allilaire, Jean Bézivin, and Ivan Kurtev. ATL: A model transformation tool. *Science of Computer Programming*, 72(1-2): pp. 31–39, 2008.
11. Andrés Yie and Dennis Wagelaar. Advanced traceability for ATL. In *Proceedings of the 1st International Workshop on Model Transformation with ATL (MtATL 2009)*, pp. 78–87, Nantes, France, 2009.
12. Massimo Tisi, Jordi Cabot, and Frédéric Jouault. Improving higher-order transformations support in ATL. In *International Conference on Model Transformation (ICMT)*, pp. 1–10, 2010.
13. Frédéric Jouault. Loosely coupled traceability for ATL. In *Proceedings of the Traceability Workshop of the 1st European Conference on Model Driven Architecture - Foundations and Applications*, volume 91, pp. 29–37, Nuremberg, Germany, 2005.
14. Parastoo Mohagheghi and Vegard Dehlen. An overview of quality frameworks in model-driven engineering and observations on transformation quality. In *Workshop on Quality in Modeling*, pp. 3–17, 2007.
15. Motoshi Saeki and Haruhiko Kaiya. Measuring model transformation in model driven development. In *Proceedings of the CAiSE*, volume 7, 2007.
16. Andrés Vignaga. Metrics for measuring ATL model transformations. Technical report, MaTE, Department of Computer Science, Universidad de Chile, 2009.
17. Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2): pp. 131–164, 2009.