

una aproximación imperativa o bien complementar transformaciones relacionales (escritas en lenguaje QVT Relations) con operaciones imperativas (enfoque híbrido), cuando es difícil proveer una especificación completamente declarativa de una relación. Cada relación define una clase que será instanciada para la traza entre elementos de modelo que están siendo transformados, y esta tiene un mapeo unívoco con una operación que el mapeo operacional implementa.

Bajo este esquema, la extracción de información de trazabilidad ocurre en dos pasos: en primer lugar, la transformación de modelos es ejecutada permitiendo la conversión de cualquier instancia del metamodelo de trazabilidad interno de QVT en una instancia de Trace-DSL mediante la definición de un mapeo operacional entre ambos metamodelos. El segundo paso consiste en la importación de la instancia Trace-DSL generada en el correspondiente repositorio. Aunque exigua, esta descripción nos permite observar lo siguiente:

- Al igual que con la propuesta de Jouault, esta opera a nivel de transformación. Es decir, requiere la ejecución de la transformación para la generación de las trazas.
- Permite la interacción con arbitrarios lenguajes de transformación de modelos, aunque depende fuertemente del conector que posibilita el enlace con el motor de transformación.
- En el contexto de QVT, utiliza el lenguaje Operational Mappings.
- Define un metamodelo de trazabilidad muy interesante, tipificando los distintos escenarios de traza (generación de nuevos elementos, actualización, eliminación y consulta).
- Es independiente de los modelos involucrados.
- No agrega información adicional que pudiera alterar los modelos o la lógica de la transformación.

4.3. Resumen de las principales diferencias

Luego de repasar los detalles de los dos trabajos de referencia seleccionados y analizadas las características del enfoque propuesto, contrastaremos las soluciones según los criterios definidos anteriormente.

Metamodelo de trazabilidad A lo largo de la Sección hemos detallado las principales características de los metamodelos puestos en contraste. Por un lado hemos visto un esquema de Jouault muy genérico, simple, y flexible, cuyo énfasis está puesto en el mantenimiento de la información de trazabilidad desde el punto de vista de la relación entre elementos de los modelos origen y destino. En segundo lugar, repasamos el metamodelo de Grammel, con un enfoque por escenarios que caracteriza distintos tipos de traza en base al impacto en el modelo destino. Finalmente, como una propuesta intermedia el esquema de QVTrace, el cual es más parecido al de Jouault, aunque más restrictivo respecto de los participantes de la relación de traza y cuyo eje es el significado de la relación, no solo los integrantes de la misma, esto es la manera en que un elemento del modelo origen se transforma en uno del modelo destino.

Si bien es posible seguir enumerando diferencias, cada metamodelo tiene sus fortalezas y debilidades, y las mismas surgen de la aplicación y el contexto en que han sido definidas. Claramente no existe “el” metamodelo de trazabilidad, sino distintos enfoques que enfatizan aspectos puntuales en función de la utilización y los resultados que busquen obtenerse.

Obtención de la información de trazabilidad Si el metamodelo de trazabilidad define qué información de rastreabilidad obtener, el segundo criterio de comparación considerado ha sido el “cómo” obtenerla. Como hemos visto, tenemos por un lado un mecanismo de Jouault netamente conducido por modelos, definido con las mismas herramientas del lenguaje de transformación, automatizado, implementado a nivel de transformación, que agrega información extra a la definición de dicha transformación, alterando no la lógica pero sí la legibilidad de la misma. En contraste repasamos el trabajo de Grammel, que está basado en un DSL desarrollado *ad hoc*, genérico, adaptado a un número arbitrario de lenguajes de transformación de modelos, aunque dependiente de las posibilidades y herramientas que éste pueda proveerle para la construcción de un componente de conexión fundamental, implementado a nivel de transformación. Finalmente, la propuesta de estos autores que sugieren que desde el conocimiento de la definición de la transformación y los modelos involucrados es posible identificar ciertos patrones o construcciones del lenguaje que permiten reconocer trazas de manera independiente a la transformación, caracterizando no sólo los participantes de la misma, sino también el significado o forma, que no altera de ninguna manera los modelos ni la lógica de la transformación, automatizada, implementada por fuera de las posibilidades del lenguaje de transformación de modelos, pero integrada con uno de los principales entornos de desarrollo utilizados en el marco del paradigma conducido por modelos.

5. Conclusiones y trabajos futuros

A lo largo del presente trabajo se repasaron los aspectos más destacados del paradigma de desarrollo conducido por modelos o MDSD y se abordó el concepto de trazabilidad como característica deseable en toda transformación de modelos. En este contexto, se presentó una técnica de análisis de trazabilidad basado en variables que permite identificar en la definición de una transformación escrita en lenguaje QVT ciertos patrones que posibilitan la inferencia automática de trazas. Al margen de ser una propuesta teórica, esta idea ha sido implementada en un prototipo llamado QVTrace, el cual está diseñado como plugin de Eclipse, y pensado para que interactúe con otras herramientas de desarrollo conducido por modelos diseñadas para dicho framework.

A continuación, se detallaron las principales características de QVTrace y se realizó un análisis comparativo de éste junto a dos esquemas de similar índole: la propuesta de trazabilidad escasamente acoplada de Jouault y el framework de trazabilidad genérico de Grammel para extracción de trazas basada en *facets*. Tomando como eje de comparación el metamodelo de trazabilidad definido y

el mecanismo de obtención de trazas, hemos podido comprobar que lejos de existir “el” metamodelo, la representación de las trazas está fuertemente ligada a la implementación, tanto del mecanismo de obtención de trazas, como a la representación de los modelos, y a la información que se desee mantener. Por el lado de los mecanismos de generación, a diferencia de los trabajos de los otros autores, la propuesta propia se independiza completamente del proceso de transformación de modelos y trabaja sobre la definición de la misma con la convicción que ésta encierra información de trazabilidad de manera implícita.

Por último, es importante mencionar posibles pasos a seguir en esta línea propuesta por el presente trabajo. En particular, sería de mucho interés poder determinar nuevas construcciones o casos de traza que permitan inferir nueva información de rastreabilidad no identificada hasta el momento, o en su defecto, en caso de encontrar el límite de los posibles tipos de trazas que la técnica permite generar, poder demostrarlo formalmente.

Referencias

1. mediniQVT. <http://projects.ikv.de/qvt>.
2. N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, and Y. Shaham-Gafni. Model traceability. *IBM Syst. J.*, 45(3):515–526, 2006.
3. Frank Budinsky, Stephen A. Brodsky, and Ed Merks. *Eclipse Modeling Framework*. Pearson Education, 2003.
4. Nikolaos Drivalos, Dimitrios Kolovos, Richard Paige, and Kiran Fernandes. Engineering a DSL for software traceability. In Dragan Gasevic, Ralf Lammel, and Eric Van Wyk, editors, *Software Language Engineering*, volume 5452 of *Lecture Notes in Computer Science*, pages 151–167. Springer Berlin / Heidelberg, 2009.
5. Birgit Grammel and Stefan Kastenholz. A generic traceability framework for facet-based traceability data extraction in model-driven software development. In *Proceedings of the 6th ECMFA Traceability Workshop*, ECMFA-TW '10, pages 7–14, New York, NY, USA, 2010. ACM.
6. Object Management Group. *MOF 2.0 Query/Views/Transformations RFP*, omg document edition, October 2002.
7. F. Jouault and I. Kurtev. Transforming models with ATL. In *Satellite Events at the MoDELS 2005 Conference*, volume 3844 of *Lecture Notes in Computer Science*, pages 128–138, Berlin, 2005. Springer Verlag.
8. Frédéric Jouault. Loosely coupled traceability for ATL. In *Proceedings of the European Conference on Model Driven Architecture (ECMDA) workshop on traceability*, Nuremberg, Germany, 2005.
9. Anneke G. Kleppe, Jos Warmer, and Wim Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
10. OMG. Meta object facility (MOF) 2.0 Query/View/transformation specification version 1.0. <http://www.omg.org/spec/QVT/1.0/PDF/>, April 2008.
11. The Institute of Electrical and Electronics Engineers. *IEEE Standard Glossary of Software Engineering Terminology*. New York, USA, September 1990.
12. Bert Vanhooff, Stefan Van Baelen, Wouter Joosen, and Yolande Berbers. Traceability as input for model transformation. In *Proceedings of the European Conference on Model Driven Architecture (ECMDA) workshop on traceability*, Nuremberg, Germany, 2007.