

Evaluación de propuestas de trazabilidad en MDE

Francisca Rosique Contreras, Pedro Sánchez Palma, Diego Alonso Cáceres
 División de Sistemas e Ingeniería Electrónica (DSIE)
 Universidad Politécnica de Cartagena,
 Campus Muralla del Mar s/n, Cartagena E-30202, Spain
 paqui.rosique@upct.es, pedro.sanchez@upct.es, diego.alonso@upct.es

Resumen. *La Ingeniería Dirigida por Modelos (MDE) ofrece un escenario ideal para potenciar el papel de la trazabilidad en el desarrollo de software. El hecho de que los modelos y las transformaciones pasen a ocupar un papel principal permite contemplar las trazas como enlaces entre los elementos de dichos modelos que podrían a su vez representarse en otros modelos y por tanto transformarse, generarse, etc. Por otro lado, como artefactos software que son, resultaría deseable aplicar los principios de MDE a su desarrollo. Combinando estas dos ideas, este trabajo realiza una revisión de la literatura para estudiar las propuestas existentes en cuanto a gestión de la trazabilidad en el desarrollo dirigido por modelos.*

1. Introducción

La gestión la trazabilidad es una técnica que permite conocer cómo evolucionan los artefactos del sistema a lo largo del proceso de desarrollo y cómo se relacionan entre sí [1]. Esta información es útil en diferentes actividades como el análisis de impacto del cambio, la toma de decisiones de diseño o el mantenimiento [2]. Así, dada la importancia de la trazabilidad en el desarrollo de los sistemas, sería deseable que las metodologías de desarrollo dieran soporte para la gestión de la trazabilidad [3].

Ante las facilidades que proporciona MDE para la gestión de la trazabilidad, se plantean las siguientes cuestiones de investigación: ¿para qué se usa la información de trazabilidad? ¿Qué operaciones de gestión se llevan a cabo cuando trabajamos con información de trazabilidad? ¿Qué características presentan las aproximaciones basadas en MDE para soportar la trazabilidad? Para dar respuesta a estas cuestiones se ha llevado a cabo una revisión sistemática.

2 Propuestas basadas en MDE para trazabilidad

Esta sección responde a la cuestión de qué características presentan las aproximaciones basadas en MDE para soportar la trazabilidad. El objetivo final de esta sección es identificar características comunes o diferenciadores entre ellas.

2.1. Criterios de evaluación

En primer lugar, aclararemos que se han seleccionado aquellas características que aparecen en todos los trabajos incluidos en este grupo. Las características que se han estudiado en las propuestas son:

- El criterio de Aproximación nos permite distinguir la naturaleza del trabajo evaluado. Para este criterio se establecen dos posibles valores;

aproximaciones basadas en requisitos y aproximaciones basadas en transformaciones.

- El criterio de Transformación nos permite identificar el tipo de transformación que se lleva a cabo en la propuesta. Existen dos posibles valores para este criterio; transformaciones entre modelos (M2M) y/o transformaciones de modelo a texto (M2T).
- La Dimensión de trazabilidad evalúa si la propuesta soporta trazabilidad de modelos y/o elementos del modelo a diferentes niveles de abstracción. Consideramos la dimensión horizontal para propuestas que soportan trazabilidad entre modelos (y/o elementos del modelo) del mismo nivel de abstracción y dimensión vertical a las propuestas que soportan trazabilidad en distintos niveles de abstracción.
- El criterio de Nivel indica si la trazabilidad se lleva a cabo entre modelos (alto nivel) o entre elementos del modelo (bajo nivel).
- El criterio de Lenguajes enumera los distintos lenguajes de transformación y/o generación de código que utilizan las propuestas.
- La Representación define las principales estructuras que son usadas para representar la información de trazabilidad.
- El criterio de Metamodelo indica el propósito del metamodelo presentado en cada trabajo analizado. Hemos establecido dos posibilidades valores para el criterio de metamodelo; genérico o específico para el dominio del problema.

2.2. Trabajos evaluados

A continuación se presentan las propuestas incluidas en este documento. Han sido seleccionadas ocho propuestas para el estudio, teniendo en cuenta su grado de madurez e inclusión de soporte para gestión de la trazabilidad en un entorno MDE.

El trabajo de Valderas y Pelechano [4] se centra en la trazabilidad de requisitos para aplicaciones Web a través de transformaciones entre modelos. Se identifica una dimensión de trazabilidad vertical y para la representación de la información de trazabilidad utiliza dos herramientas; AGG Tool (Attributed Graph Grammars Tool) para la realización de gráficos y TaskTracer para la generación de informes de trazabilidad.

En [5], Dee centra su propuesta en la trazabilidad de requisitos y en transformaciones entre modelos. Utiliza los lenguajes OCL (Object Constraint Language) y QVT (Query/View/Transformation) para la dimensión de trazabilidad en vertical. Propone un metamodelo específico y para representación de la información utiliza gráficos de dependencias y tablas cruzadas.

Aleksy et al. [6] realizan una contribución basada en requisitos, a través de transformaciones de modelos y transformaciones de modelo a texto, cubriendo las dimensiones horizontal y vertical. Utiliza XML para el almacenamiento de modelos y Velocity para llevar a cabo las transformaciones de modelo a texto.

El resto de propuestas son aproximaciones basadas en transformaciones. Jouault [7] presenta una propuesta para añadir trazabilidad a programas escritos en ATL, mediante el uso de transformaciones de modelos y apoyándose en modelos de weaving AMW (Atlas Model Weaver). Presenta un metamodelo de trazas básico cubriendo la dimensión de trazabilidad horizontal y vertical.

Falleri et al. [8] proponen un framework para trazabilidad basado en el lenguaje kermeta. Parten de un metamodelo básico de trazas fundamentado en [11]. Consideran la dimensión vertical y horizontal de trazabilidad. Para la representación utiliza modelos kermeta y la especificación XMI.

Von Pilgrim et al. [9] discuten cómo una cadena de transformaciones puede ser modelada y ejecutada en un lenguaje de transformación de modo independiente. Estas transformaciones de modelos se desarrollan bajo un plugin de Eclipse llamado UniTL (Unified Transformatin Infraestructure) utilizando lenguajes como Java, ATL o MTF (Model Transformation Framework) de IBM. Proponen dos extensiones a un metamodelo básico de traza; UniTI Traceability Metamodel y MTM (Merged Traceability Metamodel). Para la representación proponen GEF3D (Eclipse Graphical Editing Framework).

Olsen y Oldevik [10] discuten sobre el uso de escenarios para trazabilidad, presentando su propuesta de implementación. Describen cómo la trazabilidad y los enlaces de traza pueden ser usados para soportar el desarrollo de sistemas. Utilizan transformaciones de modelo a texto mediante el lenguaje MofScript. Consideran la dimensión de trazabilidad horizontal y vertical.

Por último, en [11] Kolovos et al. proponen la unificación de dos enfoques de almacenamiento de información de trazabilidad a través de un modelo de fusión. El primer enfoque considera el almacenamiento de forma interna al modelo, es decir, dentro del propio modelo; mientras que el segundo enfoque propone almacenar la información en modelos externos. Se sugiere que la información de trazabilidad debe mantenerse en modelos por separado y que pueda ser combinada con el primer modelo bajo demanda. El objetivo es producir modelos anotados para fines de inspección.

Por otra parte, el trabajo se centra en transformaciones de modelos mediante el lenguaje de transformación EML (Epsilon Merging Language) incluyendo un metamodelo específico de trazas EML. Para la representación utilizan metamodelos EML y modelos de traza UML.

Ref	Aproximación	Transformación	Dimensión	Nivel	Lenguaje	Representación	Metamodelo
[4]	Requisitos	M2M	Vertical	Alto	XML	Gráficos AGG Tool / Informes TaskTracer	Específico
[5]	Requisitos	M2M	Vertical	Alto	QVT OCL	Gráficos de dependencias Tablas cruzadas	Específico
[6]	Requisitos	M2M / M2T	Horizontal Vertical	Alto	XML Velocity	Visualización con TraVis	Específico
[7]	Transformación	M2M	Horizontal Vertical	Alto	ATL	Modelo de traza ATL Enlaces de traza AMW	Genérico
[8]	Transformación	M2M	Horizontal Vertical	Bajo	Kermeta	Modelos Kermeta XMI	Genérico
[9]	Transformación	M2M	Horizontal	Alto	Java ATL MTF	Diagramas GEF3D	Específico
[10]	Transformación	M2M	Horizontal Vertical	Alto	MofScript	TAP UML (modelo de traza)	Específico
[11]	Transformación	M2M	Horizontal	Bajo	EML	Metamodelos EML UML (modelo de traza)	Específico

Tabla 1: Características de propuestas para trazabilidad en Ingeniería Dirigida por modelos

3. Conclusiones de la evaluación

La Tabla 1 resume las conclusiones sobre las distintas características estudiadas a lo largo de las ocho propuestas analizadas. El eje horizontal representa los criterios estudiados y en el eje vertical las propuestas.

Las primeras conclusiones que se obtienen, a la vista de los resultados obtenidos, es que, en las aproximaciones basadas en requisitos predomina la dimensión vertical, mientras que en las aproximaciones que basadas en transformaciones, consideran ambas dimensiones: horizontal y vertical. También llama la atención el predominio de trazabilidad de alto nivel (trazabilidad entre modelos) con respecto a la trazabilidad de bajo nivel (entre elementos del modelo) que únicamente es considerada por las propuestas [8] y [11]. El tipo de transformación más predominante en los trabajos son las transformaciones de modelos. Únicamente [11] y [10] consideran también las transformaciones de modelo a texto. Respecto al metamodelo utilizado, la tendencia general es partir de un metamodelo genérico de traza, para construir un metamodelo específico centrado en las necesidades perseguidas en cada trabajo. Por último, tanto para la representación de la información de traza, como para los lenguajes utilizados, es difícil establecer una línea de actuación común. Cada propuesta utiliza un lenguaje específico para realizar las correspondientes transformaciones o para almacenar los modelos. Lo mismo sucede con la representación de la información, que encontramos representaciones gráficas, textuales y haciendo uso de distintas herramientas muy variadas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto español de investigación EXPLORE (ref. TIN2009-08572) del CICYT y por el proyecto de investigación MISSION-SICUVA (ref. 15374/PI/10) del gobierno de la Región de Murcia.

Referencias

[1] Asuncion, H. (2008); Towards practical software traceability. 30th International Conference on Software Engineering, ICSE 2008.

[2] Naslavsky, L., Alspaugh, T. A., Richardson, D. J., Ziv, H. (2005); Using Scenarios to Support Traceability. 3rd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE'05), pp. 25-30. ACM Press.

[3] Vara, J. M. (2009); M2DAT: A Technical Solution for Model-Driven Development of Web Information Systems. Tesis Doctoral, Universidad Rey Juan Carlos, 2009.

[4] Valderas, P. and V. Pelechano, Introducing requirements traceability support in modeldriven development of web applications. *Information And Software Technology*, 2009. 51(4): p. 749-768.

[5] Dee, M.R., Traceability-based Change Management in the Model Driven Architecture, in *Electrical Engineering, Mathematics and Computer Science2007*, Twente: Twente, Holanda. p. 157.

[6] Aleksy, M., et al. A Pragmatic Approach to Traceability in Model-Driven Development. in *Process Innovation for Enterprise Software*. 2009. Mannheim, Germany: PRIMMUM'2009

[7] Jouault, F. Loosely Coupled Traceability for ATL. in *European Conference on Model Driven Architecture - Traceability Workshop*. 2005. Nuremberg, Germany: ECMDA'05.

[8] Falleri, J.R., M. Huchard, and C. Nebut. Towards a traceability framework for model transformations in kermeta. in *European Conference on Model Driven Architecture - Traceability Workshop*. 2006. Bilbao, Spain: ECMDA-TW'2006.

[9] von Pilgrim, J., et al., Constructing and Visualizing Transformation Chains, in *Model Driven Architecture – Foundations and Applications*, I. Schieferdecker and A. Hartman, Editors. 2008, Springer Berlin / Heidelberg. p. 17-32.

[10] Olsen, G.K. and J. Oldevik, Scenarios of Traceability in Model to Text Transformations, in *Model Driven Architecture- Foundations and Applications2007*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg. p. 144 - 156.

[11] Kolovos, D., R. Paige, and F. Polack, On-Demand Merging of Traceability Links with Models, in *European Conference on Model-Driven Architecture – Traceability Workshop2006*, ECMDA-TW'06: Bilbao, Spain.