

Título: Testing-Based Conceptual Schema Validation in a Model-Driven Environment

Autor: María Fernanda Granda Juca

Directores: Nelly Condori-Fernández, Tanja Vos, Oscar Pastor

1. Resumen

A pesar del escepticismo y dificultades en su adopción, el Desarrollo Orientado por Modelos (MDD, por sus siglas en inglés) está siendo usado y mejorado para proveer muchos beneficios inherentes a la industria. Uno de sus mayores beneficios es la capacidad de manejar la complejidad del desarrollo de software elevando el nivel de abstracción. Los modelos se expresan utilizando conceptos que no están relacionados con una tecnología de implementación específica (por ejemplo, Lenguaje de Modelado Unificado -UML, Lenguaje de Restricción de Objetos -OCL, Lenguaje de Acción para el Foundational UML - ALF), lo que significa que los modelos pueden ser más fáciles de especificar, mantener y documentar. Debido a que en una Ingeniería dirigida por modelos (MDE), los artefactos primarios son los modelos conceptuales, los esfuerzos se centran en su creación, prueba y evolución a diferentes niveles de abstracción a través de transformaciones, porque si un esquema conceptual tiene defectos, éstos se pasan a las siguientes etapas, incluida la codificación. Por lo tanto, uno de los retos para los investigadores y desarrolladores in MDD es poder identificar los defectos temprano, a nivel de esquemas conceptuales, ya que esto ayudaría a reducir los costos de desarrollo y mejorar la calidad del software.

Durante la última década, pocos trabajos de investigación se han realizado en esta área. Algunas de las causas de esta realidad son la alta complejidad teórica de probar esquemas conceptuales y la falta de soporte de software adecuado. Por lo tanto, este área de investigación admite nuevos métodos y técnicas, enfrentando retos como la generación de casos de prueba utilizando información externa a los esquemas conceptuales (es decir, los requisitos), la medición de una posible automatización, selección y priorización de casos de prueba, la necesidad de una herramienta de soporte eficiente que utilice una semántica estándar, la retroalimentación oportuna para apoyar el proceso de aseguramiento de la calidad del software y facilitar la toma de decisiones basadas en el análisis y la interpretación de los resultados.

El objetivo de esta tesis es mitigar algunos de los problemas que afectan la validación de los esquemas conceptuales, proporcionando un nuevo marco de validación basado en pruebas que fue construido usando un desarrollo dirigido por modelos. El uso de MDD permite un aumento en la abstracción, automatización y reutilización que nos permite aliviar la complejidad de nuestro marco de validación. Además, al aprovechar las técnicas MDD (como el metamodelado, las transformaciones de modelos y los modelos en tiempo de ejecución), nuestro marco soporta cuatro fases del proceso de prueba: diseño de pruebas, generación de casos de prueba, ejecución de casos de prueba y la evaluación de los resultados. Esto es diferente a los enfoques de pruebas tradicionales, que, en general, sólo admiten algunas de estas fases.

Con el fin de proporcionar soporte de software para nuestra propuesta, hemos desarrollado CoSTest, un entorno de pruebas basado en el lenguaje ALF. Para asegurar que CoSTest ofrece la funcionalidad necesaria, primero identificamos un conjunto de requisitos funcionales. Luego, después de identificar estos requisitos, definimos la arquitectura y el ambiente de pruebas de nuestro marco de validación y, finalmente, implementamos la arquitectura en el contexto de Eclipse. CoSTest ha sido desarrollado para probar varias propiedades sobre el modelo ejecutable como la corrección sintáctica (es decir, todos los elementos del modelo se ajustan a la sintaxis del lenguaje en el que se describe), consistencia entre la parte estructural y el comportamiento

(sus restricciones de integridad) y completitud (es decir, todos los cambios posibles en el estado del sistema se pueden realizar a través de la ejecución de las operaciones definidas en el modelo ejecutable). Para los modelos defectuosos, el informe de CoSTest devuelve una retroalimentación significativa que ayuda a localizar y reparar los defectos detectados.

El trabajo involucrado en la tesis fue validado mediante seis estudios usando casos encontrados en la literatura, así como un caso industrial. Los cuatro primeros fueron experimentos de laboratorio para validar y evaluar algunos componentes de CoSTest tales como la generación dirigida por modelos de los casos de prueba, el generador de mutantes usado para priorizar y seleccionar casos de prueba, así como también el generador de un esquema conceptual ejecutable basado en ALF. En el quinto estudio, se aplicó el análisis de mutaciones para evaluar la efectividad y la adecuación de los casos de prueba de CoSTest al detectar defectos en esquema conceptuales mutados con diferentes tipos de defectos. En el último estudio, CoSTest fue evaluado con la participación de usuarios finales a través del Modelo de Aceptación de Tecnología (TAM) y el método de entrevista. Mientras que el TAM nos permitió medir la utilidad y facilidad de uso de una manera subjetiva, el método de entrevista nos permitió identificar las limitaciones y posibles mejoras que se pueden implementar en la herramienta. En general, los resultados fueron favorables. CoSTest fue altamente valorado en la utilidad percibida y facilidad de uso; también obtuvimos resultados positivos con respecto a la efectividad de los casos de prueba.