

# El Desarrollo Dirigido por Modelos y Ontologías

**Autor:**

Karen Mariel Nájera Hernández

**Publicado en :**

[SG #39: \(Febrero - Abril 2013\)](#)

**Sección:**

[Código Innovare](#)

El surgimiento y evolución de plataformas y tecnologías de desarrollo de software, la evolución del hardware, las nuevas formas de interacción humano-computadora, los avances en el cómputo móvil, el incremento de los usuarios de los sistemas y de los datos que deben gestionar y el surgimiento de la Web de datos, son algunas de las razones por las cuales los sistemas de software son cada vez más complejos.

Un medio para manejar esta complejidad y continuar proporcionando sistemas que soporten las nuevas y crecientes necesidades del mercado, es el llevar a cabo el desarrollo de software utilizando métodos apropiados de abstracción [1].

En este contexto, un paradigma de desarrollo de software que se ha vuelto popular en la academia y la industria en los últimos años es el Desarrollo Dirigido por Modelos (Model Driven Development – MDD). MDD basa el desarrollo de software en la construcción de modelos conceptuales que describen el sistema a desarrollar (estructura del sistema, comportamiento, estructura de datos, aspectos de presentación, etc.). Es parte de un modelo que conceptualiza el dominio del problema, y de manera sistemática este modelo se va transformando en otros modelos de niveles de abstracción más bajos, hasta generar implementaciones concretas.

Mientras más completa y precisa sea la especificación de las funcionalidades de un sistema en los modelos, mayor será el porcentaje del código fuente del sistema que se podrá generar automáticamente.

Para la construcción de modelos se cuenta con:

- Lenguajes de modelado de propósito general (GPLs), que soportan el modelado de cualquier dominio. Un ejemplo de GPL sería UML (Unified Modeling Language).
- Lenguajes de modelado de dominio específico (DSLs), los cuales se definen específicamente para un dominio, como el proporcionado por EAGLE (Easily Applicable Graphical Layout Editor), un software de modelado, creación y simulación de circuitos electrónicos.

Para llevar a cabo las transformaciones entre modelos se definen conjuntos de reglas de transformación que describen cómo es que los elementos de un modelo pueden ser transformados en elementos de otro modelo de menor nivel de abstracción. Las reglas pueden ser reusadas para transformar nuevos o distintos modelos y además pueden ser implementadas para realizar el proceso de transformación de manera automática.

El flujo común de transformaciones se presenta en la figura 1. Se inicia con el modelo de más alto nivel de abstracción que generalmente corresponde a la conceptualización del dominio del problema, el cual es transformado a modelos de más bajo nivel de abstracción mediante transformaciones Modelo a Modelo (M2M) y finaliza con una transformación de Modelo a Texto (M2T) que como resultado arroja el código fuente del sistema.

**Figura 1.** Flujo de transformación de modelos.

Dentro de las ventajas de MDD se encuentra la generación automática de código, mayor calidad en implementación (menos defectos), desarrollo ágil: aplicación de cambios rápido y fácil, permitiendo la evolución y el mantenimiento del sistema de una forma más simple; reúso de código y/o modelos.

El proyecto de modelado de Eclipse [2] se enfoca en la evolución y promoción de tecnologías de desarrollo basadas en modelos dentro de la comunidad de Eclipse. Utiliza como base la Arquitectura Dirigida por Modelos (Model Driven Architecture - MDA) [3] una iniciativa presentada por el Object Management Group (OMG) que soporta el paradigma de MDD. MDA se enfoca en el uso de tecnologías interoperables que soportan el desarrollo dirigido por modelos con transformaciones automáticas y para ello define un conjunto de estándares (UML, MOF, OCL, QVT, XMI). El proyecto de modelado de Eclipse proporciona: frameworks de modelado y de generación de código automático para desarrollar aplicaciones basadas en modelos de datos estructurados, componentes e infraestructura en tiempo real para el desarrollo de editores gráficos, un lenguaje para definir reglas de transformación (ATL), etcétera.

Las ontologías como modelos en el proceso de desarrollo

Una de las tendencias actuales en la ingeniería de software es el uso de ontologías como parte del proceso del desarrollo dirigido por modelos, es decir, como medio para representar los modelos conceptuales que definen al sistema a desarrollar en el paradigma MDD. Una ontología representa un modelo abstracto que identifica los componentes más relevantes de cierto fenómeno del mundo real, y los describe en un lenguaje formal, es decir, que puede ser entendido por una computadora, a través de conceptos, propiedades, relaciones, funciones, restricciones y axiomas.

La ventaja principal del uso de ontologías como modelo en MDD es que en una sola ontología es posible representar la estructura del sistema, su comportamiento y el modelo de datos, además de describir la lógica de negocio y restricciones de software en el mismo modelo.

Entre otras ventajas del uso de las ontologías, además de las que por defecto se heredan de MDD, se encuentra que proporcionan un entendimiento compartido del dominio del problema y una especificación formal de los requisitos de un sistema, lo que permite el reúso de definiciones existentes de conocimiento del dominio; además proporcionan mayor soporte para inferencias lógicas, y la integración e interoperabilidad con otros componentes o aplicaciones.

SemanticWebBuilder (SWB) [4] es una plataforma para el desarrollo de software implementada en INFOTEC. SWB parte de una ontología (definida en el lenguaje estándar de la Web Semántica OWL - Web Ontology Language) y a partir del conocimiento modelado en la ontología se genera automáticamente código en lenguaje Java que representa la arquitectura base del sistema a desarrollar. Los datos del sistema resultante son representados en el formato de la Web semántica (RDF) por lo que pueden ser procesados por computadoras y consumidos a través de la Web.

La plataforma SWB ha servido de base para la construcción de sistemas de diferentes dominios. Entre los productos creados con SWB están:

- SWBPortal, un sistema de gestión de contenidos con licencia de código abierto para desarrollar y administrar portales de internet con tecnología semántica.
- SWBProcess, un sistema de gestión de procesos de negocio semántico que parte de modelos de procesos descritos con el estándar BPMN 2.0 y a través de transformaciones de BPMN 2.0 a ontologías y de ontologías a código en lenguaje Java proporciona la implementación necesaria para la ejecución de los procesos.

Estos sistemas, además de ser generados bajo el paradigma de MDD, también proporcionan el mecanismo bajo el mismo paradigma, para ser extendidos con funcionalidades específicas a partir de requisitos especificados en una ontología.

## Conclusión

En conclusión, el desarrollo dirigido por modelos es un paradigma visto, por muchos, como la mejor opción para incrementar el nivel de abstracción en el cual construimos, mantenemos y razonamos sobre el software. Su aplicación se encuentra en constante crecimiento debido a las ventajas que ofrece al ciclo de vida del software principalmente la automatización del desarrollo de sistemas y la reutilización de componentes desde los niveles de abstracción más altos hasta la codificación de la solución.

El uso de ontologías como punto de partida para el desarrollo de software permite la representación de la información contenida en el sistema en un formato definido por un lenguaje formal que puede ser interpretado y procesado por personas y/o sistemas. Por lo tanto, es posible aplicar técnicas de inferencia a los datos para por ejemplo, generar nuevo conocimiento y obtener predicciones, útiles para toma de decisiones. Además, se pueden aplicar mecanismos Web para exponer la información del sistema en la Web Semántica, y explotar los datos con los nuevos paradigmas de la Web Semántica como Linked Data.

## Referencias

- [1] B. Anda, et al. "Experiences from Introducing UML-based Development in a Large Safety-Critical Project". Empirical Software Engineering, vol. 11, pp. 555-581, 2006.
- [2] Eclipse Modeling Project. <http://www.eclipse.org/modeling>
- [3] OMG MDA. <http://www.omg.org/mda>
- [5] SemanticWebBuilder. <http://www.webbuilder.com.mx>

## Bio:

M.C. Karen Mariel Nájera Hernández es egresada de la Universidad Autónoma de Puebla de la carrera Ingeniería en Ciencias de la Computación. Obtuvo el grado de Maestra en Ciencias en en la especialidad de Tecnologías Web por el Centro de Investigación y Desarrollo Tecnológico (CENIDET) en donde en conjunto con la Fundación Bruno Kessler (Italia) desarrolló trabajos de investigación enfocados al modelado organizacional y ontologías. Actualmente colabora en proyectos para la web semántica en la Gerencia de Desarrollo de Nuevos Productos en INFOTEC. [karen.najera@infotec.com.mx](mailto:karen.najera@infotec.com.mx)