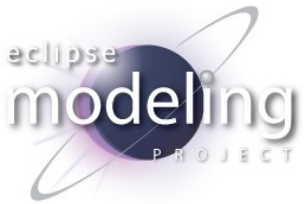
Introducción a EMF (Eclipse Modeling Framework)

Lady Viviana Garay González
Jeyson Stith Arévalo Sandoval
Universidad Distrital
Grupo de investigación ITI
www.itiud.org



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- Introducción
- ¿Por qué usar modelado?
- ¿Cómo funciona?
- Núcleo
- Niveles de generación de código
- Ejemplo

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- **Introducción**
- ¿Por qué usar modelado?
- ¿Cómo funciona?
- Niveles de generación de código
- Ejemplo

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Introducción

El proyecto EMF es un marco de modelado que facilita generación de código para construir herramientas y otras aplicaciones basadas en un modelo de datos estructurado. A partir de una especificación de modelo descrita en XMI, EMF proporciona herramientas y soporte en tiempo de ejecución para producir un conjunto de clases Java para el modelo, junto con un conjunto de clases de adaptador que permiten su edición basadas en comandos del modelo en un editor básico.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- Introducción
- **¿Por qué usar modelado?**
- ¿Cómo funciona?
- Niveles de generación de código
- Ejemplo

¿Por qué usar modelado?

La desconcertante complejidad del software moderno pide un nuevo enfoque centrado en el diseño de alto nivel, la delegación de tareas manuales a las herramientas y marcos. A partir de una descripción concisa de su dominio del problema, se puede inferir una solución completa.

- Producir rápidamente resultados de alta calidad.
- Para reutilizar eficazmente las soluciones probadas.
- Especificar concisamente información estructurada compleja.
- Diseñar fácilmente anotaciones gráficas.
- Implementar eficientemente soluciones potentes en tiempo de ejecución.
- Explotar los estándares industriales interoperablemente.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- Introducción
- ¿Por qué usar modelado?
- **¿Cómo funciona?**
- Niveles de generación de código
- Ejemplo

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

¿Cómo funciona?

EMF es un estándar común para los modelos de datos, muchas tecnologías y framworks se basan en el. Esto incluye soluciones de servidor, marcos de persistencia, frameworks de interfaz de usuario y soporte para transformaciones.

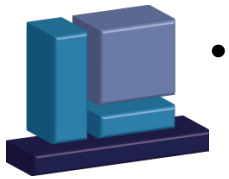
Servidor y Persistencia

Las siguientes tecnologías proporcionan la infraestructura necesaria para atender diferentes escenarios en los proyecto.

- **CDO**: (Connected Data Objects) El repositorio de modelos, es una estructura distribuida de modelo compartido para modelos EMF y metamodelos.
- **EMFStore**: Es un repositorio de modelos para el Marco de Modelado de Eclipse (EMF) y cuenta con la edición y el versionado colaborativo de modelos.
- **Net4j:** Es un sistema cliente-servidor extensible basado en Eclipse Runtime y Spring Framework.
- **Teneo:** Es una solución de persistencia de bases de datos para EMF usando Hibernate o EclipseLink.
- **Modelo de Transacción:** El componente de transacción proporciona una capa de gestión de modelo construida en la parte superior de EMF para la gestión de recursos EMF.

Interfaz de usuario

Las siguientes tecnologías proporcionan soporte para implementar de manera eficiente interfaces de usuario para modelos de dominio EMF en su proyecto.



- **EMF Client Platform:** Es un marco para crear aplicaciones cliente basadas en EMF. Todos los componentes se pueden utilizar de forma independiente y se pueden integrar en su propia aplicación.

- **EMF Forms:** Proporciona una nueva forma de desarrollar interfaces de usuario basadas en formularios. En lugar de codificar manualmente diseños basados en formularios, le permite describir la interfaz de usuario con un modelo simple en lugar de con código.

- **Extended Editing Framework:** Es un marco de presentación para el marco de modelado de Eclipse. Permite al usuario crear interfaces de usuario ricas para editar los modelos EMF.

Modelo de transformación

Existen marcos y herramientas dedicadas que permiten desarrollar transformaciones de modelo a modelo por un lado y, por otro lado, transformaciones de modelo a texto.

- **Acceleo:** Es una implementación pragmática del Object Management Group (OMG) para transformaciones Modelo a texto (MOF).
- **ATL:** (lenguaje de transformación ATL) Proporciona maneras de producir un conjunto de modelos objetivo de un conjunto de modelos de origen.
- **Epsilon:** Es una familia de lenguajes y herramientas para la generación de código, transformación de modelo a modelo, validación de modelos, comparación y migración.

Modelo de transformación

- **JET:** Se utiliza típicamente en la implementación de un "generador de código". Un generador de código es un componente importante de MDD (Model Driven Development).
- **Model-to-Model Transformation:** El proyecto MMT aloja lenguajes de transformación Modelo a Modelo. Las transformaciones son ejecutadas por los motores de transformación que se conectan a la infraestructura de Eclipse Modeling.
- **Xpand:** Es un lenguaje de plantilla de tipo estático que incluye: invocación de plantillas polimórficas, programación orientada a aspectos, extensiones funcionales, abstracción de sistemas de tipo flexible, transformación de modelos, validación de modelos y mucho más.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- Introducción
- ¿Por qué usar modelado?
- ¿Cómo funciona?
- **Niveles de generación de código**
- Ejemplo

Niveles de generación de código

Se admiten tres niveles de generación de código:

- **Modelo**: Proporciona interfaces Java y clases de implementación para todas las clases del modelo, además de una clase de implementación de fábrica y de paquete (metadatos).
- **Adapters** Genera clases de implementación (denominadas ItemProviders) que adaptan las clases de modelo para su edición y visualización.
- **Editor:** Produce un editor correctamente estructurado que se ajusta al estilo recomendado para los editores de modelos EMF de Eclipse y sirve como punto de partida para comenzar a personalizar.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Agenda

- Introducción
- ¿Por qué usar modelado?
- ¿Cómo funciona?
- Niveles de generación de código
- **Ejemplo**

Ejemplo

- Configuración del entorno
- Creación de un modelo simple
- Restricciones en EMF
- Prueba manual del modelo