

## 4. Arquitectura Dirigida por Modelos (MDA)

## Índice

- Referencias
- Introducción
- Conceptos fundamentales
- Arquitectura OMG
- Camino a seguir
- Conclusiones

## Referencias

- OMG MDA Guide v1.0.1  
<http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- S.J. Mellor, K. Scott, A. Uhl, and D. Weise, *MDA Distilled. Principles of Model-Driven Architecture*, Addison-Wesley, 2003
- IEEE Std. 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology

## Referencias

- OMG Meta Object Facility (MOF) 2.0 Query/View/Transformation V1.1  
<http://www.omg.org/spec/QVT/1.1/>

## Introducción

- La Arquitectura Dirigida por Modelos (MDA, *Model-Driven Architecture*) es una aproximación definida por el OMG (*Object Management Group*) que promueve el desarrollo de sistemas software basado en modelo de diseño

## Introducción

- Los términos MDA y MDD (*Model-Driven Development*) se suelen utilizar de manera indistinta:
  - MDD se suele referir a las actividades llevadas a cabo por los desarrolladores de software
  - MDA es la definición formal OMG centrada en crear un marco en el que se pueda llevar a cabo el MDD

## Introducción

- Uno de los principales objetivos de MDA es hacer la especificación de un sistema de manera independiente de los detalles de implementación
- MDA proporciona un marco que permite:
  - Dar una especificación de un sistema con independencia de los detalles de implementación
  - Especificar plataformas de implementación

## Introducción

- Elegir una de estas plataformas
  - Transformar la especificación del sistema en una plataforma de implementación
- La independencia se logra utilizando distintos tipos de modelos
- Estos modelos permiten proporcionar distintas especificaciones de sistemas desde distintos puntos de vista

## Introducción

- Estos modelos son:
  - CIM (*Computation Independent Model*): caracterización del sistema desde el punto de vista del dominio (modelo del dominio/negocio)
  - PIM (*Platform Independent Model*): vista de diseño omitiendo detalles específicos de la plataforma
  - PSM (*Platform Specific Model*): vista de diseño considerando detalles específicos de una plataforma

## Introducción

- Un componente fundamental de la aproximación MDA es el uso de *transformaciones*, sobre todo en el paso de PIM a PSM

## Conceptos fundamentales

- **Sistema**: colección de componentes organizados para llevar a cabo una función o conjunto de funciones. Dicho sistema puede incluir casi cualquier cosa:
  - Un programa
  - Un sistema basado en un único ordenador
  - Combinación de distintos sistemas
  - Personas

## Conceptos fundamentales

- **Modelo** (de un sistema):
  - Abstracción semánticamente completa de un sistema
  - Representación abstracta de un sistema
  - Descripción o especificación de ese sistema y su entorno para cierto propósito

## Conceptos fundamentales

- *Arquitectura* (de un sistema): estructura organizativa de un sistema que incluye su descomposición en partes, conectividad, mecanismos de interacción y principios de guía que proporcionan información sobre el diseño del mismo

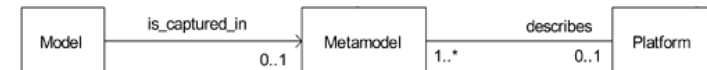
## Conceptos fundamentales

- *Plataforma*: conjunto de subsistemas y tecnologías que proporcionan un conjunto coherente de funcionalidad mediante interfaces y patrones de uso, y que cualquier aplicación soportada por esa plataforma puede utilizar sin preocuparse por los detalles específicos de cómo está implementada la funcionalidad proporcionada por la plataforma

## Conceptos fundamentales

- *Aplicación*: funcionalidad que es desarrollada. Así un sistemas es una o más aplicaciones soportadas por una o más plataformas
- *Metamodelo*: modelo de un lenguaje de modelo. Define la estructura, semántica y restricciones para una familia de modelos (modelos que comparten una sintaxis y semántica común)

## Conceptos fundamentales



Modelos, metamodelos y plataformas

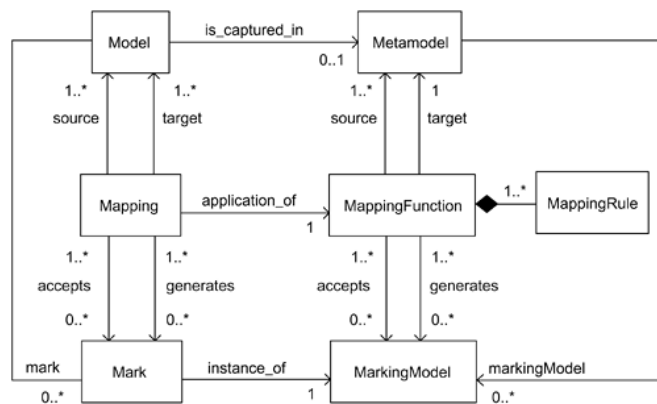
## Conceptos fundamentales

- *Transformación de modelo*: proceso de convertir un modelo en otro modelo del mismo sistema
- *Mapping*: especificación de una transformación de un modelo en otro (normalmente un PIM en un PSM)

## Conceptos fundamentales

- Las transformaciones pueden estar definidas:
  - Entre metamodelos (tipos de modelos)
  - Entre modelos (normalmente utilizando marcas)
  - Combinadas
  - Basadas en plantillas
- Provenzan de metamodelos, o de modelos decorados específicamente, las transformaciones pueden aceptar marcas

## Conceptos fundamentales



Mappings definidos entre metamodelos y que aceptan marcas

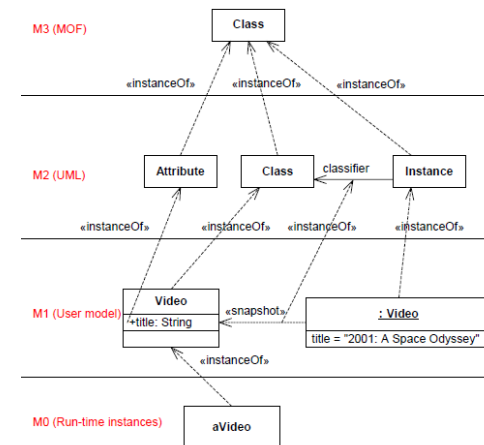
## Conceptos fundamentales

- *Elaboración del modelo*: modificaciones en el modelo generado.

# Arquitectura OMG

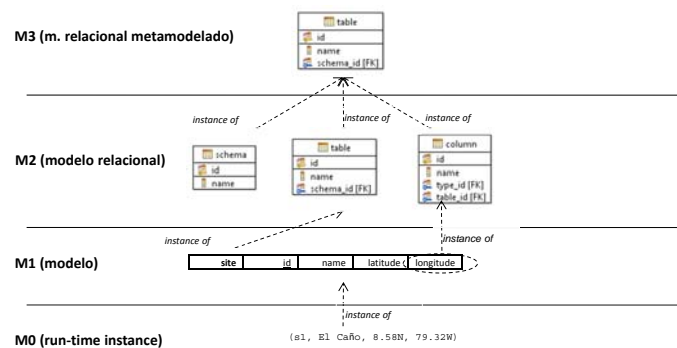
- OMG ha definido una arquitectura de cuatro niveles para organizar estos conceptos

# Arquitectura OMG



Ejemplo de arquitectura de cuatro niveles de metamodelado OMG

# Arquitectura OMG

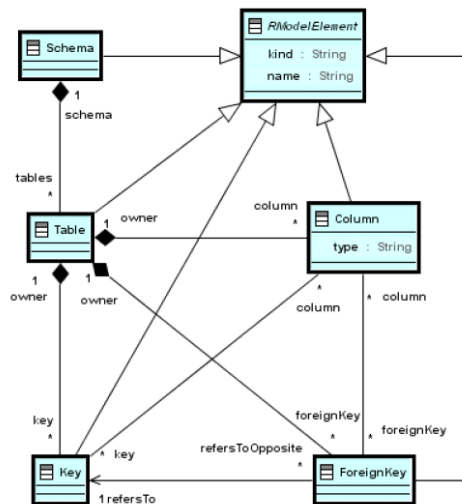


Ejemplo de arquitectura de cuatro niveles de metamodelado OMG

# Arquitectura OMG

- M3: meta-metamodelo. Describe las propiedades que pueden tener los metamodelos
- M2: metamodelo. Describen la propiedades que pueden tener los modelos
- M1: Modelo. Describe un modelo de un sistema
- M0: Instancias. Valores del modelo ejecución

## Arquitectura OMG



Modelado de Software  
Antonio Navarro... Metamodelo básico para el modelo relacional

25

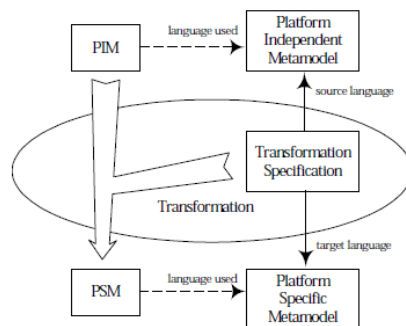
## Camino a seguir

- Vamos a definir un mecanismo de meta-metamodelado
- Lo utilizaremos para definir metamodelos
- Definiremos también transformaciones entre estos metamodelos
- Así, podremos traducir modelos concretos en modelos concretos

Modelado de Software  
Antonio Navarro

26

## Camino a seguir



Camino a seguir

Modelado de Software  
Antonio Navarro

27

## Conclusiones

- MDA/MDD: desarrollo de aplicaciones basado en modelos
  - CIM
  - PIM
  - PSM
- Transformaciones: concepto clave
- **Cuatro niveles de metamodelado**

Modelado de Software  
Antonio Navarro

28