

Manejo Visual de Transformaciones entre Modelos MOF

Autores:

- Rivera Conrado
- Rodríguez Nicolás

Director

- Giandini Roxana

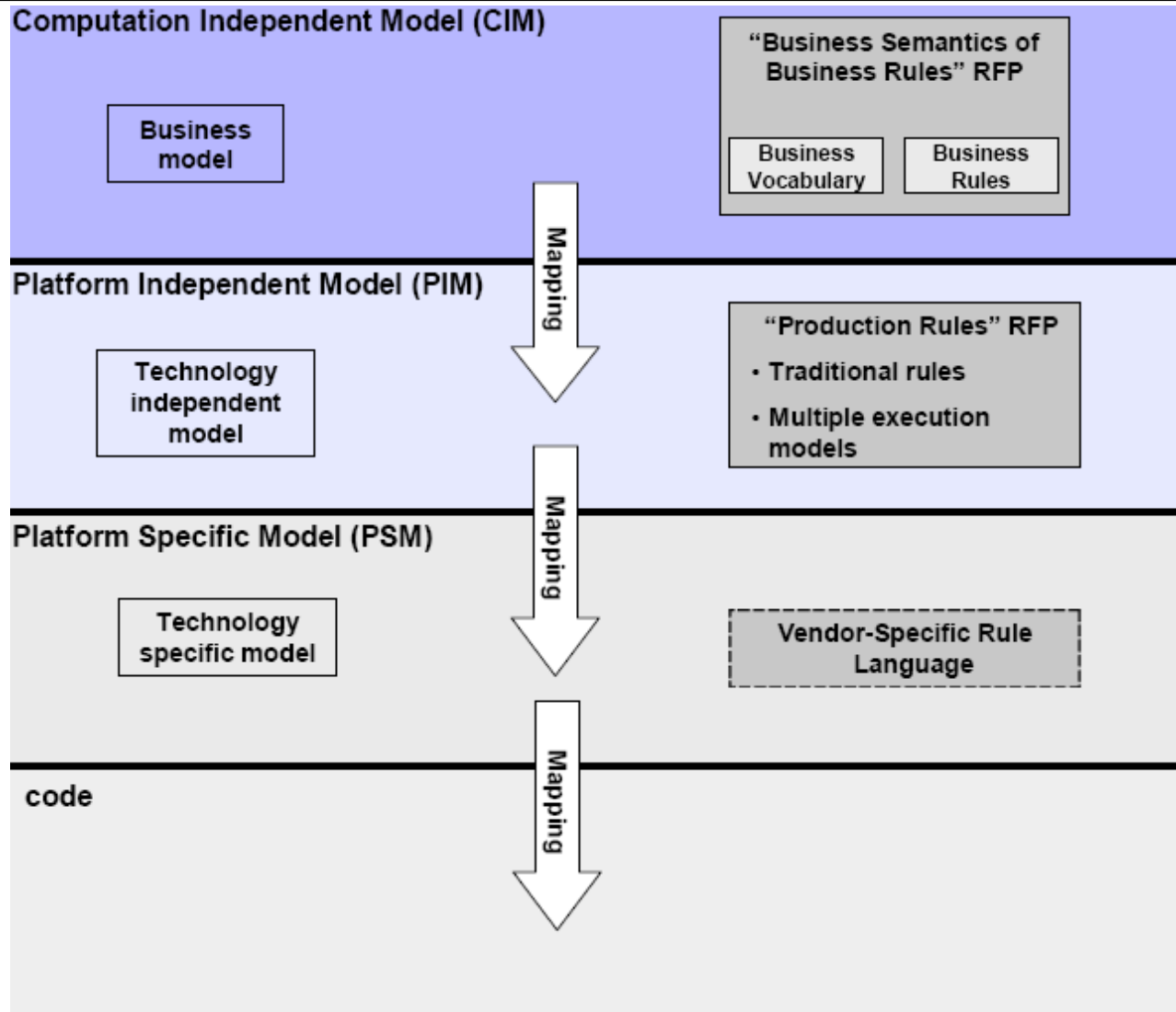
Codirector

- Pons Claudia

Introducción

- MDE (*Model Driven Engeenering*)
- Existen 4 modelos
 - *Modelo independiente de computación (CIM)*
 - *Modelo independiente de la plataforma (PIM)*
 - *Modelo específico de la plataforma (PSM)*
 - *Código*

Introducción

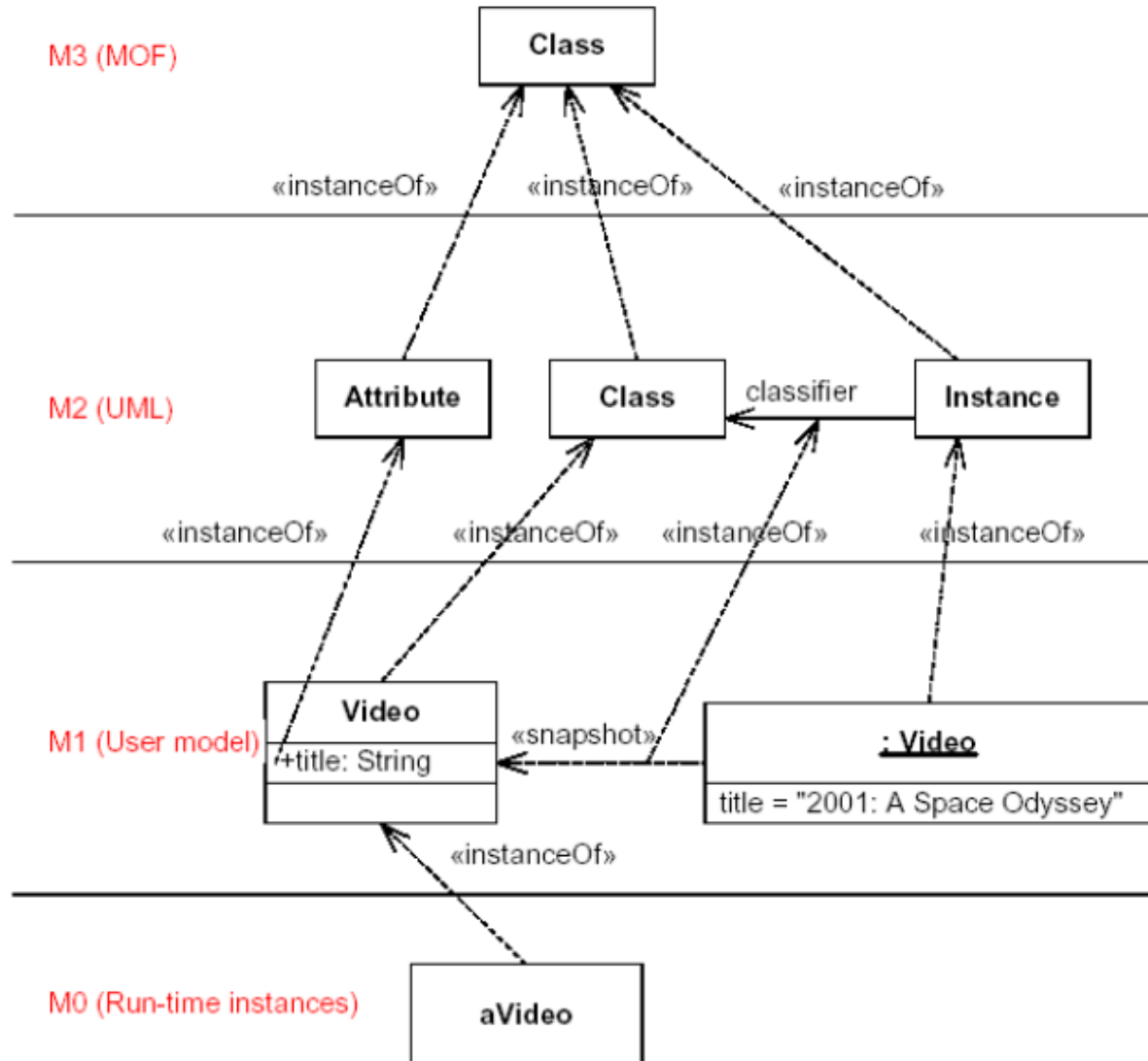




Introducción

- Definiciones
 - Meta-metamodelo
 - Metamodelo
- Arquitectura de metamodelo de cuatro capas
 - *Meta-metamodelo*
 - *Metamodelo*
 - *Modelo*
 - Objetos de usuario

Introducción



Introducción

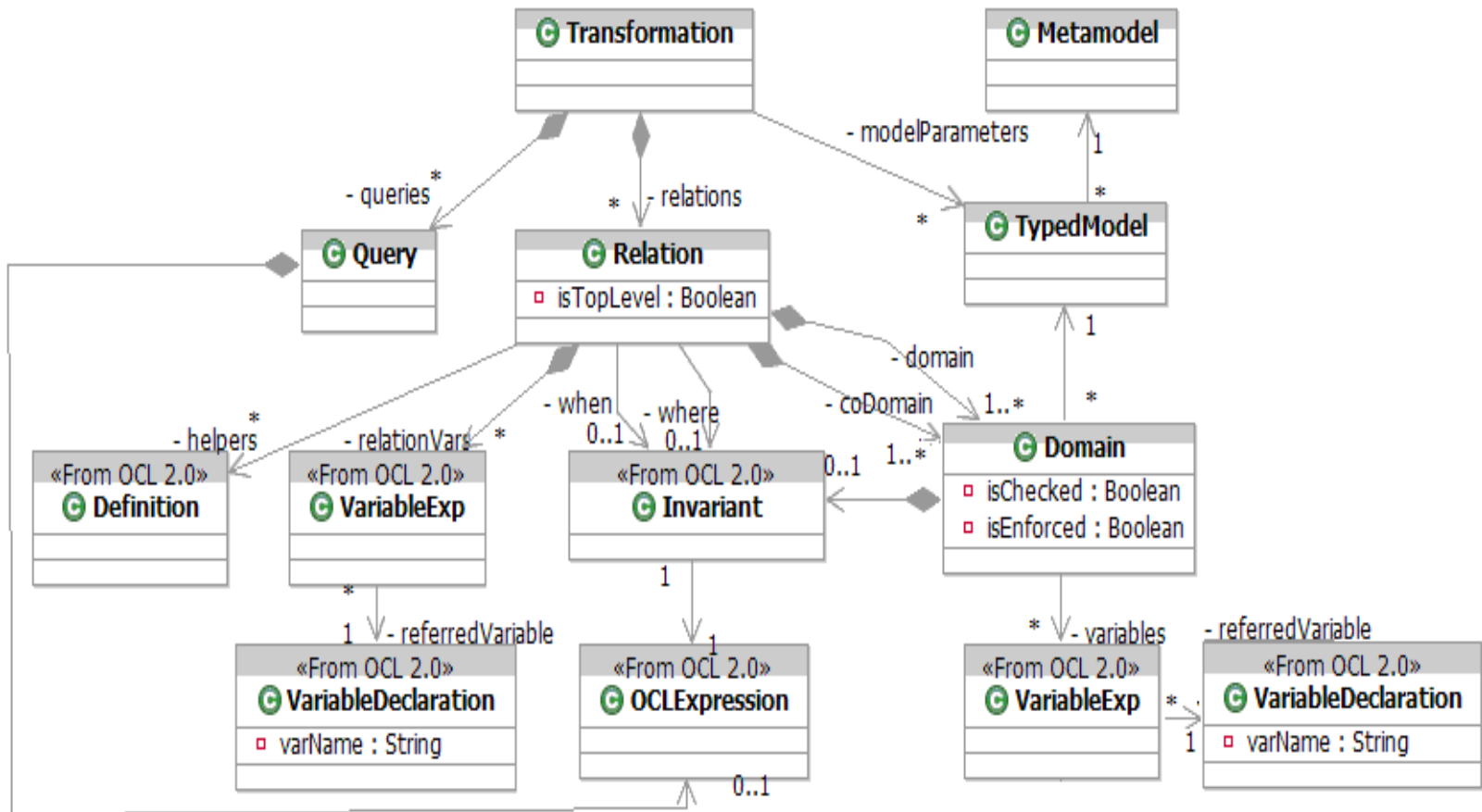
- *¿Qué es una Transformación?*
 - Definición de transformación
 - Regla de transformación
- *¿Qué es un Lenguaje de Transformación?*
- QVT (Query/View/Transformation)



Nuestra Propuesta

- SQVT (Simple QVT)
 - Lenguaje declarativo puro para transformaciones
 - Se inspira en QVT
 - Implementado como una extensión de la Infraestructura
 - Se utiliza OCL

Nuestra Propuesta



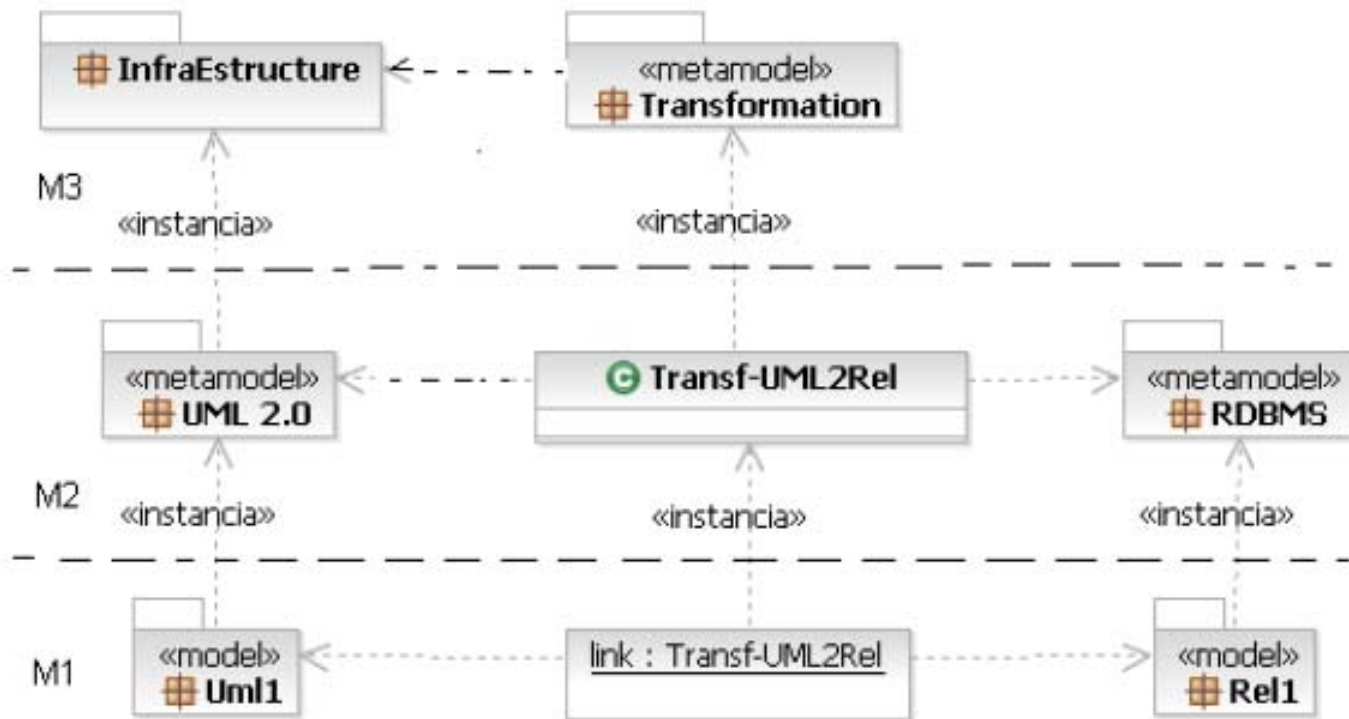


Nuestra Propuesta

- QVT vs. SQVT
 - Soporte para generación de Modelos
 - Utilización de OCL para desambiguar
 - Parámetros mas específicos

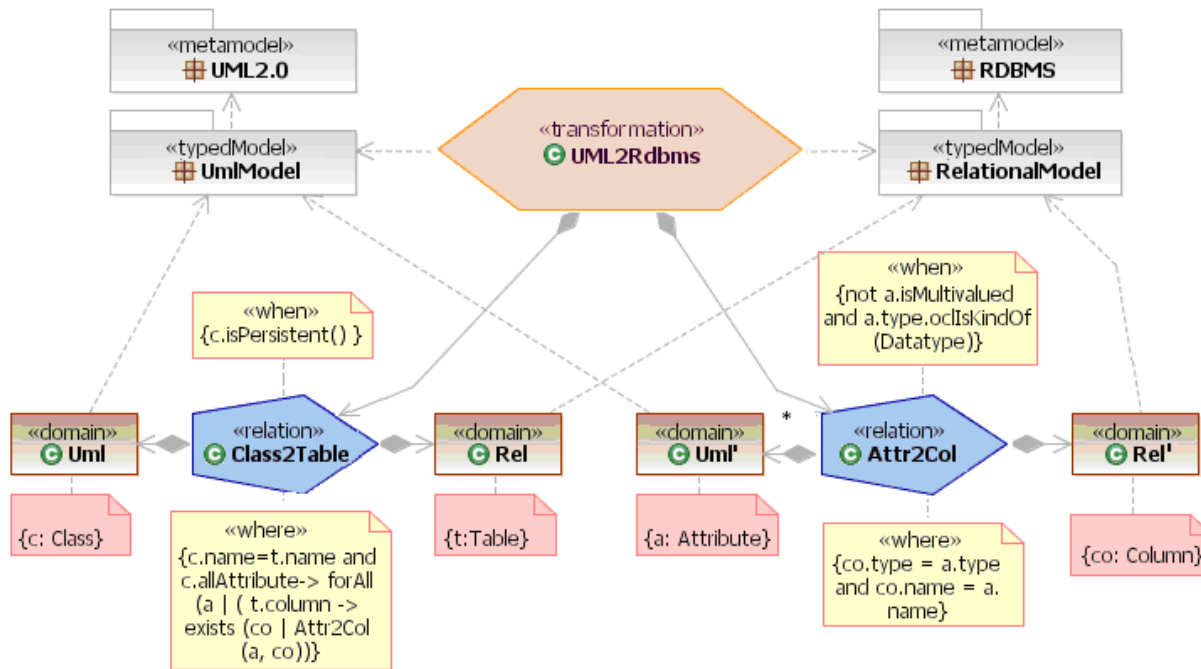
Nuestra Propuesta

- Ubicación del modelo SQVT dentro de la Arquitectura de 4 capas de Modelado



Nuestra propuesta

Ejemplo visual de una transformación en SQVT



Nuestra Propuesta

- La vista:
 - un editor que provee las herramientas gráficas para poder armar una transformación
 - un cargador de metamodelos para poder asociarle a la transformación los metamodelos de entrada y salida
 - un cargador de modelos para poder evaluar una transformación
- La lógica:
 - interpretación de una transformación.
 - evaluación de los predicados.

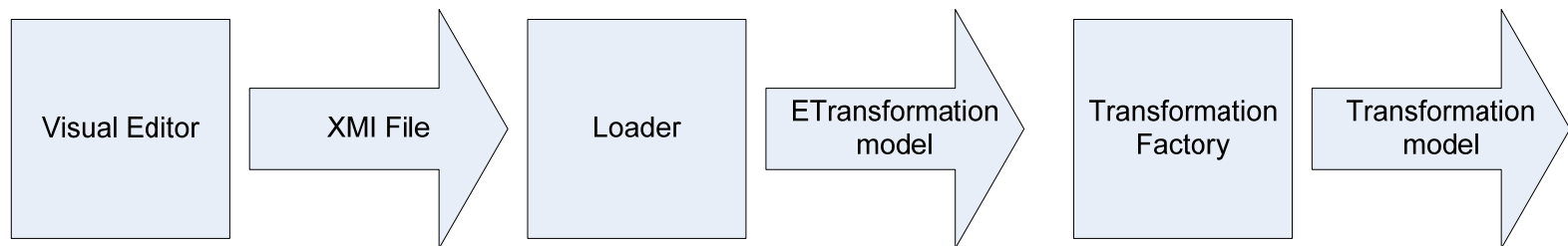
Implementación del editor (La vista)

- Creación del:

- ecore
- genmodel
- gmftool
- gmfgraph
- gmfmap
- gmfgen

Implementación del Evaluador (La lógica)

○ Construcción



○ Evaluación

- contexto de evaluación
- ambiente de evaluación de OCL
- proceso de evaluación

Conclusiones

- MDE: Software a través de un proceso guiado por modelos.
- ¿Qué logramos?:
 - Diseñador gráfico de transformaciones.
- ¿Qué problemas tuvimos?:
 - Inconvenientes tecnológicos:
 - Falta de madurez de los plugins
 - Incompatibilidad de versiones
- Ventajas:
 - Facilidad de integración entre los plugins: división de la tarea.
 - SQVT vs QVT: más fácil de desarrollar.



Futuros trabajos

- Generar modelo de salida
- Relaciones de alto nivel
- Evaluación de los Querys
- Mejoras de visualización
- Integración con e-Platero



Fin...

La Plata, 27 de Junio de 2008

Rivera, Rodriguez