



Procesadores de Lenguajes 2

Introducción a la ingeniería del software dirigida por modelos

Curso 2013-2014

Iván Ruiz Rube

Departamento de Ingeniería Informática

Escuela Superior de Ingeniería

Universidad de Cádiz



Contenidos

- Introducción
- Aplicaciones
- La visión de la OMG (MDA)



INTRODUCCIÓN A MDE

 **INTRODUCCIÓN**

Evolución del desarrollo software

1. Programación en Ensamblador
x86
2. Programación Estructurada
Pascal, C
3. Orientación a Objetos
C++, Java
4. Componentes software
J2EE, .NET
5. Orientación a Aspectos
AspectJ, Spring Framework AOP
6. Arquitecturas orientadas a Servicios
WSDL, SOAP
7. ...



Modelos

- Hasta ahora, la programación siempre es el centro de atención
- Al igual que en otras ingenierías.. ¡tenemos que aumentar el nivel de abstracción!
- Los modelos nos ayudan a construir sistemas más complejos.
- *Modelo*: descripción o especificación (de parte) de un sistema software desde un determinado punto de vista

Ejemplo de Modelo

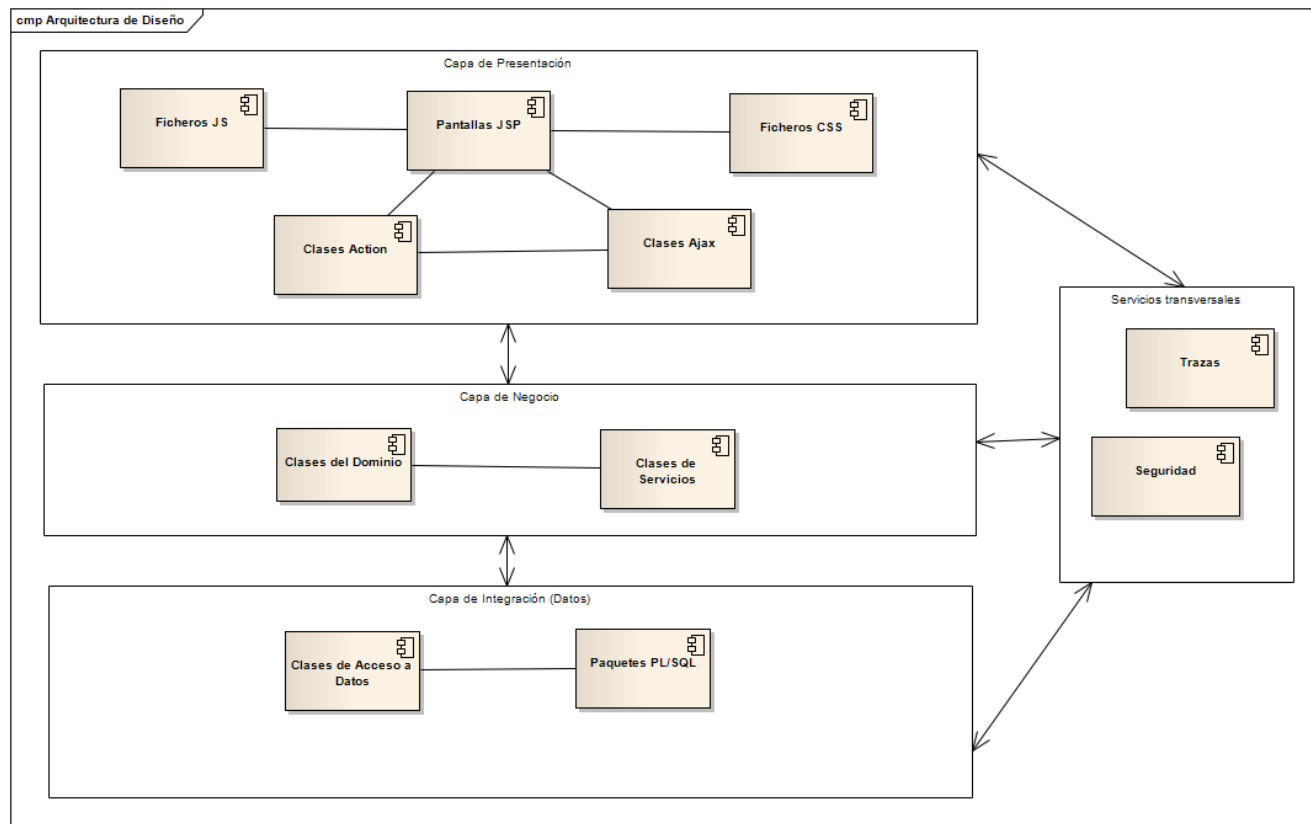


Diagrama de componentes. Arquitectura de diseño (3 capas) de un sistema web

Características y utilidades

- Características:
 - Abstracto: ocultación de elementos
 - Comprensible: lenguaje entendible
 - Preciso: correcto sintáctica y semánticamente
 - Barato: sencillo de construir
- Utilidades:
 - Especificar las características del sistema a desarrollar
 - Comprenderlo con exactitud
 - Detectar errores con antelación
 - Guiar la programación



Problemas

- Sólo suelen usarse como documentación
- En muchas ocasiones su calidad se mide “al peso”
- Falta de sincronía: entre modelos y entre modelos-código
- No podemos optimizar o validar modelos automáticamente
- ¿Cuál es el retorno de la inversión que conseguimos al desarrollar modelos?



Model-Driven Engineering

- MDE es un “nuevo” enfoque en Ingeniería del Software
- Utilización de modelos como artefactos software
- Podremos tener editores, optimizadores, validadores y compiladores (transformadores) de modelos
- Objetivo perpetuo en IS: facilitar trabajo y reducir tiempos de desarrollo y número de errores



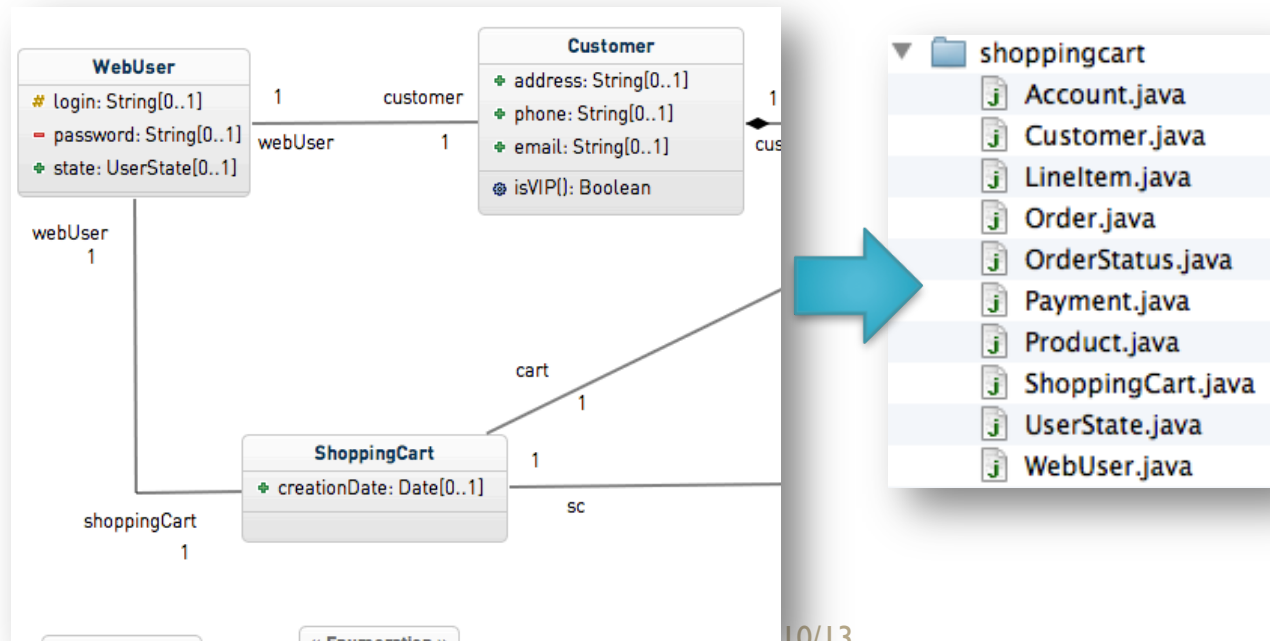
INTRODUCCIÓN A MDE



APLICACIONES

Aplicaciones

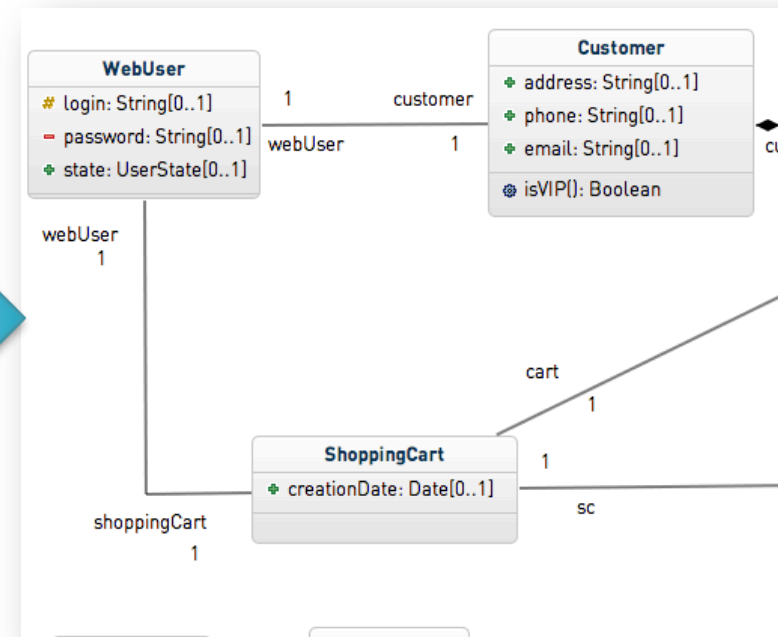
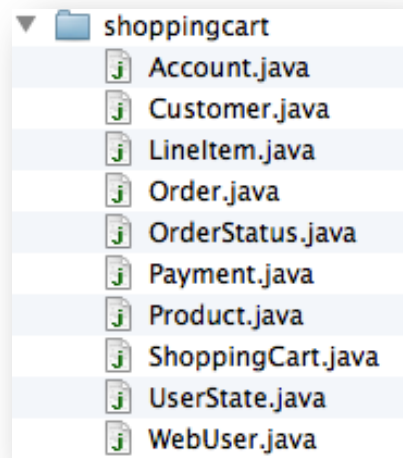
- Model-Driven Development (MDD)
 - Generación automática de software: código fuente, casos de prueba, etc.



10/13

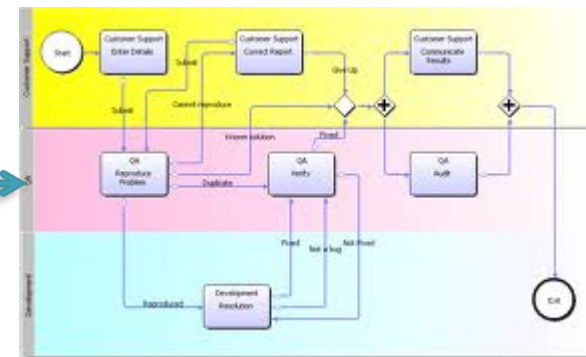
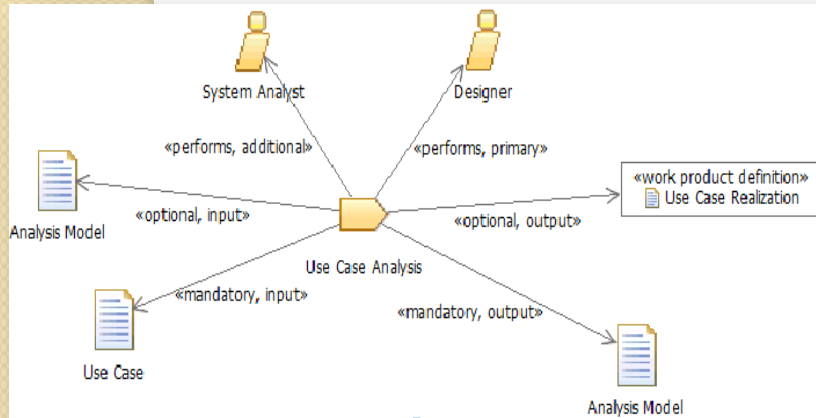
Aplicaciones

- Ingeniería Inversa
 - Obtención de métricas
 - Generar documentación
 - Migración de aplicaciones



Aplicaciones

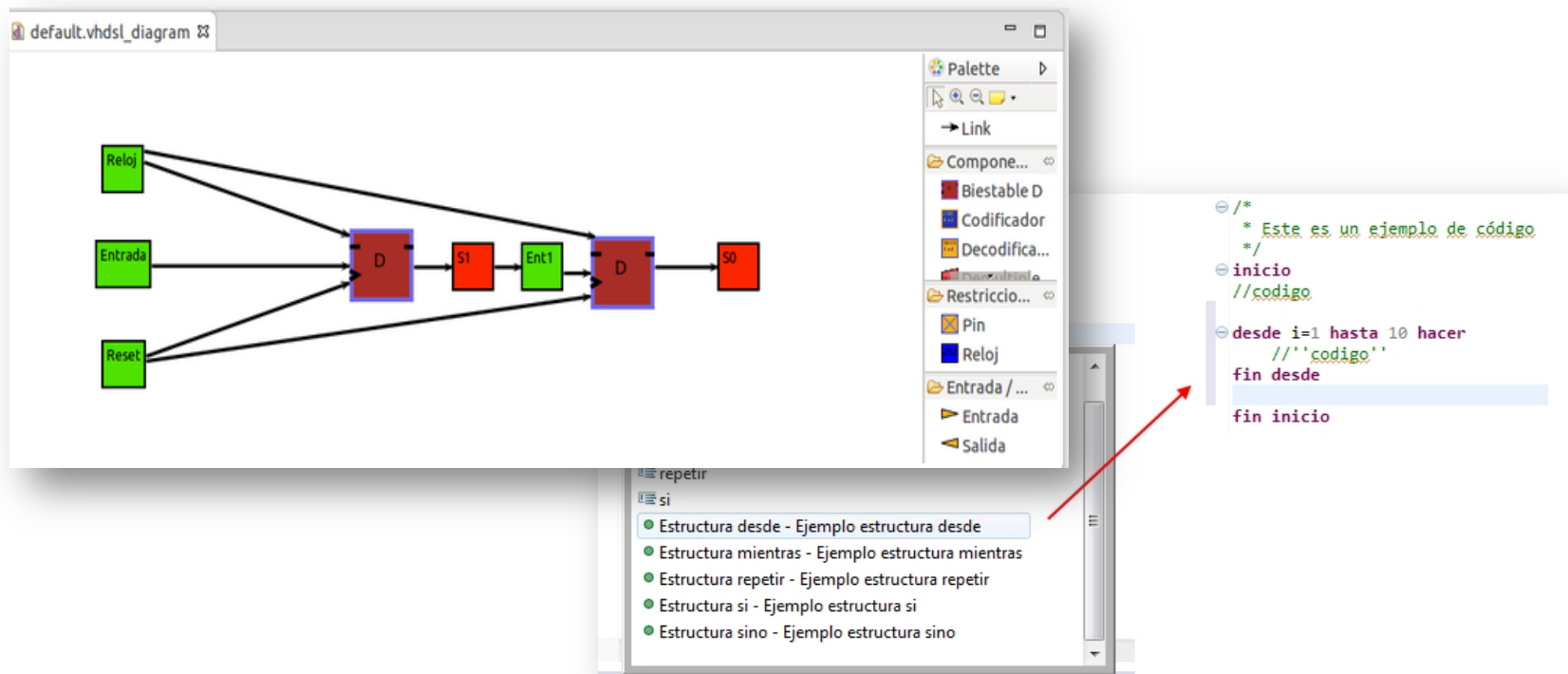
- Software Process Engineering (SPE)
 - Soporte a la gestión de procesos (metodologías) organizacionales



Id	Nazwa zadania	Cz. tryb	Poprz./Niezwy. zad.	Czas	05-cze-20	05-cze-27	05-lip-04	05-lip-11
1	Relative development	83 dni7						
2	Requirements	3 dni7						
3	Software requirements analysis	3 dni7						
4	Estimate and document software require	1 cze07						
5	Evaluate software requirements	1 cze07	4					
6	Joint review	1 cze07	5					
7	Design IT	16 dni7	2					
8	Software architectural design	8 dni7						
9	Identifies software components	1 cze07						
10	Transform requirements into architecture	1 cze07	9					
11	Evaluate architecture	1 cze07	10					
12	Develop preliminary version of user docum	1 cze07						
13	Develop top level design for interfaces	1 cze07	11					
14	Evaluate interfaces	1 cze07	13					
15	Develop top level design for database	1 cze07	11					
16	Evaluate database	1 cze07	15					
17	Define preliminary test requirements	1 cze07	14,16					
18	Define schedule for software integration	1 cze07	17					
19	Conduct Joint Review	1 cze07	18					
20	Software detailed design	8 dni7	8					
22	Coding IT	16 dni7	7					
23	Software coding and testing	8 dni7						
25	Software integration	8 dni7	23					
27	Testing IT	8 dni7	22					
28	Testing IT	16 dni7	22					
45	Coding IT	16 dni7	26					
50	Testing IT	8 dni7	45					

Aplicaciones

- **Domain Specific Languages (DSL)**
 - Desarrollo de lenguajes visuales o textuales



14/10/13

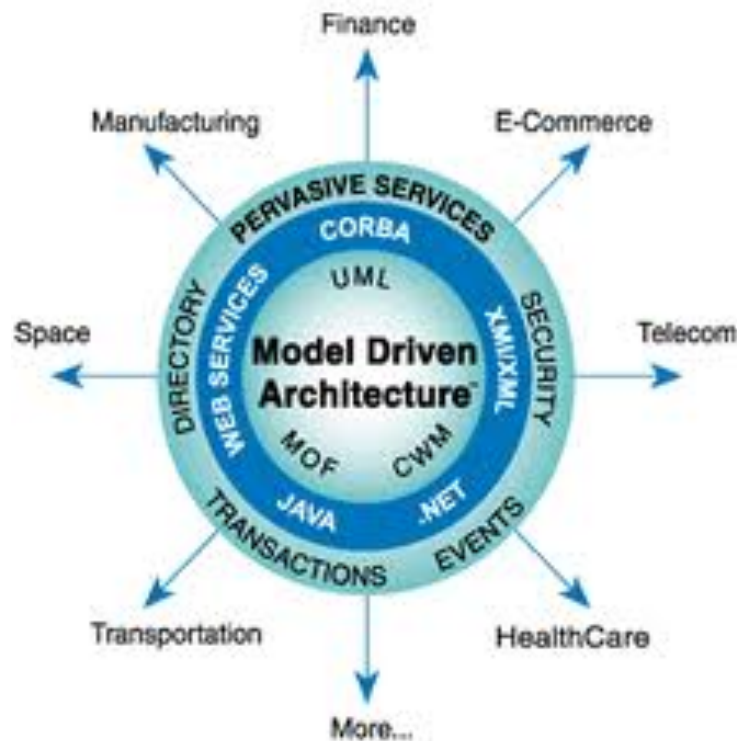
Aplicaciones

- Model-Driven Web Engineering (MDWE)
 - Generación automática de modelos de ingeniería (requisitos, análisis, etc.)
- Model-Driven Integration (MDI)
 - Interoperabilidad entre sistemas
- etc.

INTRODUCCIÓN A MDE

° **LA VISIÓN DE LA OMG
(MDA)**

Model-Driven Architecture



- MDA es la propuesta de la OMG para dar soporte al enfoque de desarrollo por modelos (MDD)
- MDA ofrece un conjunto de estándares y un enfoque de modelado basado en niveles



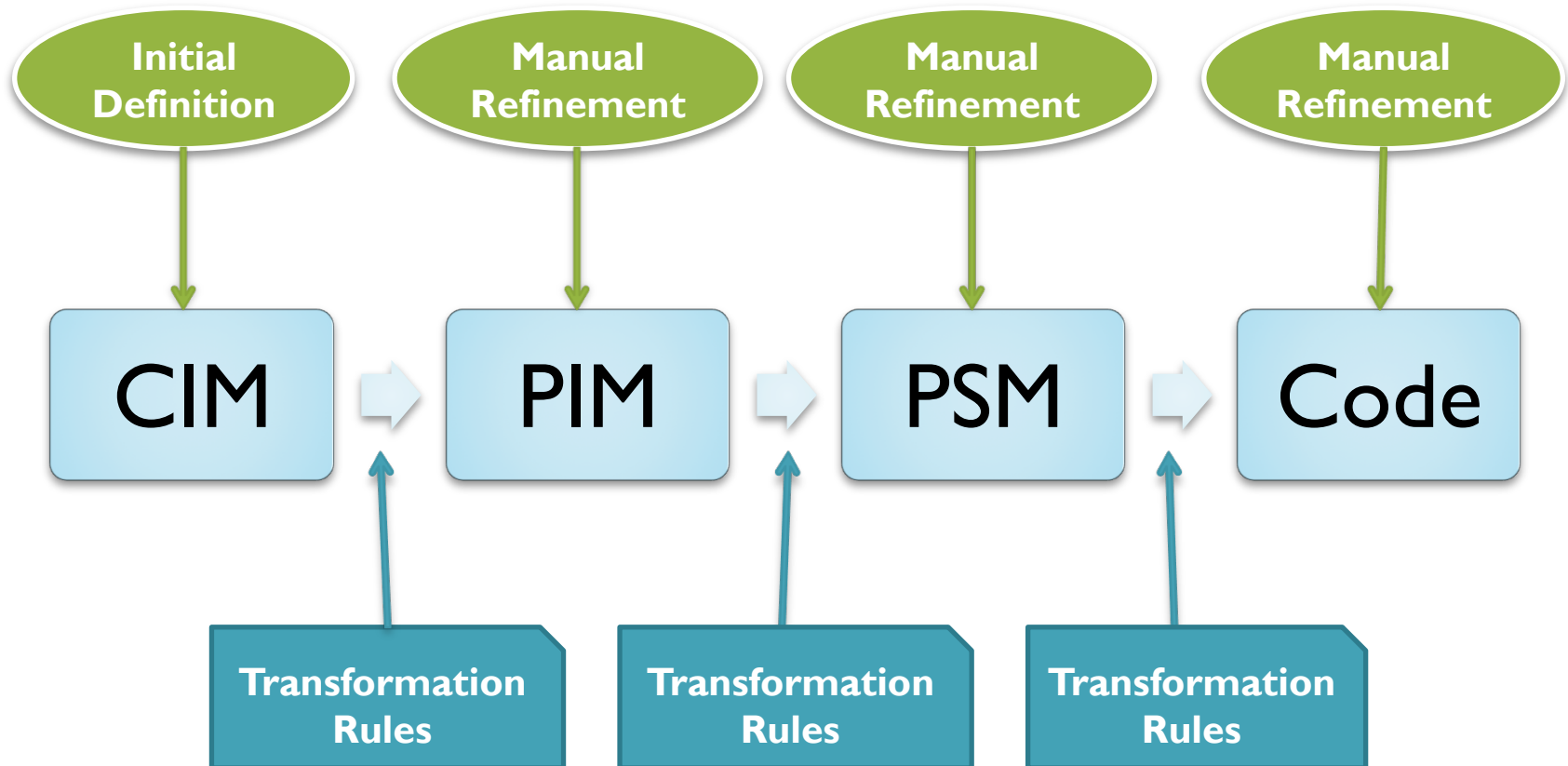
Enfoque de modelado

- MDA promueve separar la especificación de la funcionalidad de un sistema, de los detalles de implementación en cualquier plataforma tecnológica
- Proceso paso a paso de refinamiento de modelos
- Cercano a la concepción del ciclo de vida clásica del desarrollo software

Niveles en MDA

- **CIM: Computational Independent Model**
 - Ej: Modelos de procesos de negocio (ej: BPMN)
- **PIM: Platform Independent Model**
 - Ej: Modelo de casos de uso
 - Ej: Modelo entidad/relación
- **PSM: Platform Specific Model**
 - Ej: Diagrama de clases de diseño

Proceso de desarrollo MDA



Estándares

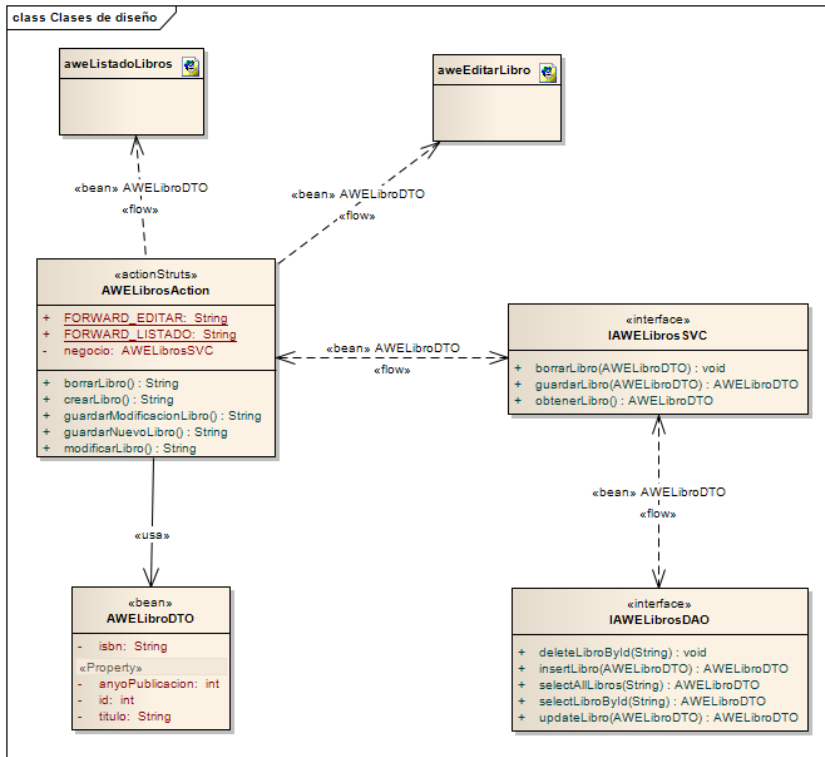
- UML: *Unified Modeling Language*
- OCL: *Object Constraint Language*
- QVT: *Query/View/Transformation*
- XMI: *XML Metadata Interchange*
- ...
- *MOF, SPEM, CWM, ODM, ...*



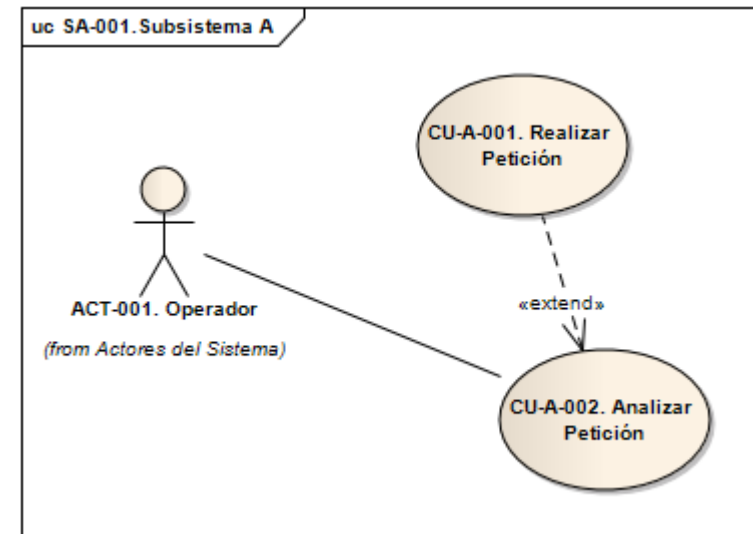
Unified Model Language (UML)

- Ofrece un conjunto de notaciones (sintaxis) gráficas para describir y desarrollar sistemas software
- Bajo UML se engloban todos los tipos de diagramas destinados a la descripción estática y dinámica del sistema
- Modelos de Estructura: *Clases, Componentes, Despliegue y Paquetes*
- Modelos de Comportamiento: *Actividad, Casos de uso, Máquinas de estado, Interacción y Secuencia*

Modelos en UML



Clases de diseño

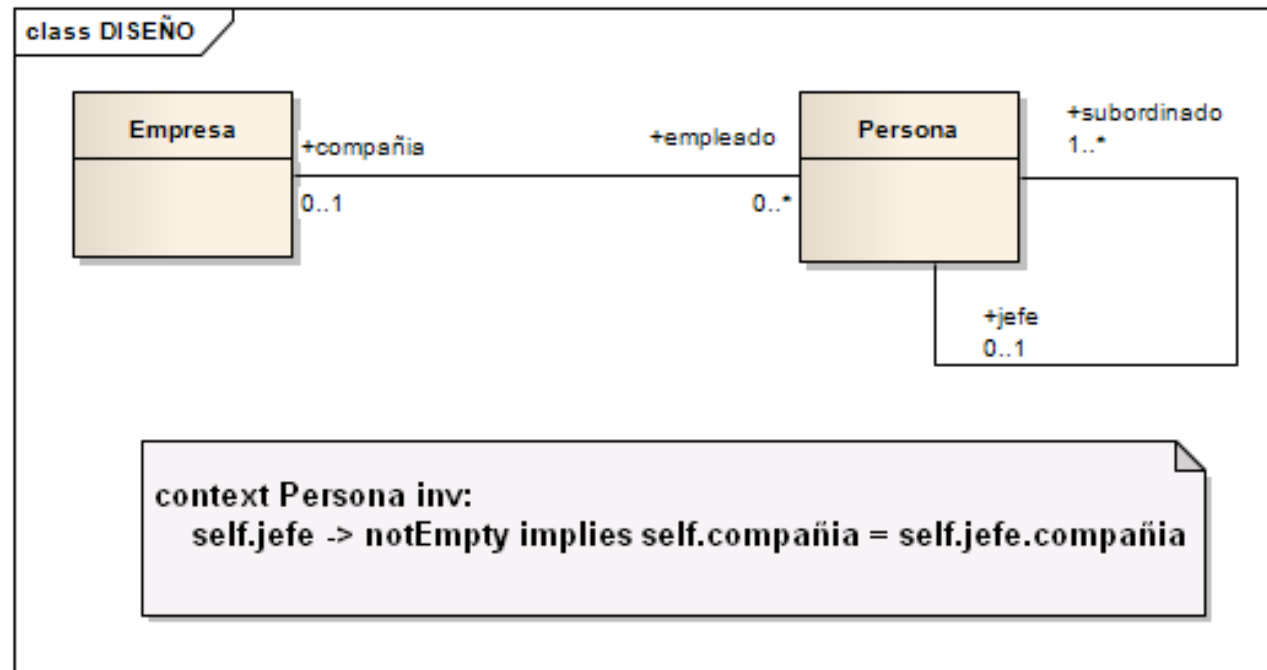


Casos de Uso

Object Constraint Language (OCL)

- No toda la semántica de un sistema puede representarse mediante la notación de UML
- OMG propone OCL, un lenguaje para enriquecer nuestros modelos
- OCL permite definir:
 - Invariantes y restricciones (pre/postcondiciones) en modelos.
 - Operaciones y acciones.
 - Consultas sobre los modelos

Ejemplo OCL

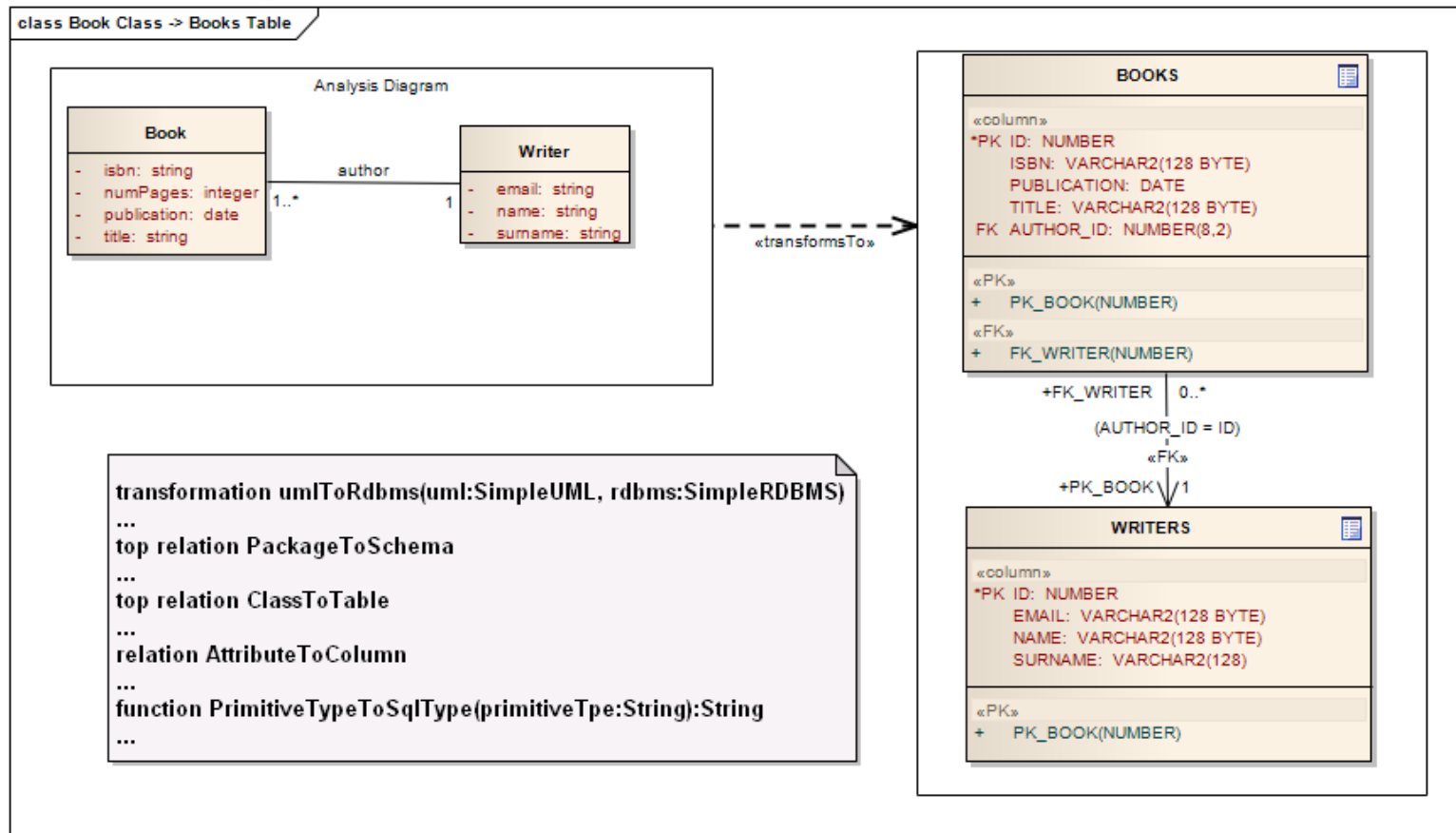


Invariante dentro de un modelo de clases UML

Query/View/Transformation (QVT)

- QVT define un conjunto de lenguajes estándar para la transformación de modelos
- QVT contempla:
 - QVT-Operational: lenguaje imperativo para escribir transformaciones unidireccionales
 - QVT-Relations: lenguaje declarativo que permite transformar modelos bidireccionalmente
 - QVT-Core: versión reducida del anterior
- ATL es la implementación de QVT por parte de la comunidad Eclipse

Ejemplo QVT



QVT nos permitiría, por ejemplo, transformar un modelo de clases de análisis en un modelo físico de datos

XML Metadata Interchange (XMI)

- Estándar para el intercambio de información de metadatos utilizando el formato XML
- El uso habitual de XMI es la serialización de modelos UML
- Los entornos basados en MDA pueden colaborar debido a que emplean UML como mecanismo estándar de representación

Ejemplo XMI

```

<packagedElement xmi:type="uml:InformationFlow" xmi:id="EAID_E301D665_5214_47e4_96CE_5F650A9EEDF8"
  source="EAID_0C2CE8E9_3DEB_4270_BBE2_22D2E48D4D00" target="EAID_27AF2527_5821_4bc9_B402_7D2D5D865D4F" />
- <packagedElement xmi:type="uml:Class" xmi:id="EAID_5F_3E298127D014" name="IAWELibrosDAO" visibility="public">
  - <ownedOperation xmi:id="EAID_33333333_3333_3333_3333_3333" name="deleteLibroById" visibility="public" concurrency="sequer
    <ownedParameter xmi:id="EAID_33333333_3333_3333_3333_3333" name="id" direction="in" type="EAJava_String" />
    <ownedParameter xmi:id="EAID_33333333_3333_3333_3333_3333" name="return" direction="return" type="EAJava_void" />
  - <ownedOperation xmi:id="EAID_33333333_3333_3333_3333_3333" name="selectLibroById" visibility="public" type="" concurrency="sequential">
    <ownedParameter xmi:id="EAID_432726A8_9FBC_4e76_B8C8_A1F7B7DCFF70" name="id" direction="in" type="EAJava_String" />
    <ownedParameter xmi:id="EAID_RT000000_DE82_473b_925C_788BE0F748D9" name="return" direction="return" type="EAJava_AWELibroDTO" />
  - <ownedOperation xmi:id="EAID_CEAC0000_0000_0000_0000_0000" name="insertLibro" visibility="public" type="" concurrency="sequential">
    <ownedParameter xmi:id="EAID_0E100000_0000_0000_0000_0000" name="return" direction="return" type="EAJava_AWELibroDTO" />
  - <ownedOperation xmi:id="EAID_0718D000_0000_0000_0000_0000" name="updateLibro" visibility="public" type="" concurrency="sequential">
    <ownedParameter xmi:id="EAID_27B00000_0000_0000_0000_0000" name="return" direction="return" type="EAJava_AWELibroDTO" />
  - <ownedOperation xmi:id="EAID_A699B887_7612_401e_8976_477B3AED8F2C" name="obtenerLibro" visibility="public" type="" concurrency="sequential">
    <ownedParameter xmi:id="EAID_RT000000_7612_401e_8976_477B3AED8F2C" name="return" direction="return" type="EAJava_AWELibroDTO" />
  </packagedElement>
<packagedElement xmi:type="uml:InformationFlow" xmi:id="EAID_722A214F_9197_4aaa_AE3D_D7878C421F1B"
  source="EAID_E2572532_0C7B_4484_B270_209648B968D6" target="EAID_9926CC60_9A6B_4a1f_825F_3E298127D014" />
<packagedElement xmi:type="uml:Class" xmi:id="EAID_27AF2527_5821_4bc9_B402_7D2D5D865D4F" name="aweEditorLibro" visibility="public" />
<packagedElement xmi:type="uml:Class" xmi:id="EAID_B91E4B60_6CD6_4dbc_9C44_1E8B796BCD81" name="aweListadoLibros" visibility="public" />
</packagedElement>

```





Modelos visuales y textuales

- Hemos visto cómo los modelos gráficos (UML) pueden representarse también en forma textual (XMI).
- Por tanto, tenemos que entender los modelos más allá de una mera representación visual.
- Los lenguajes de modelado serán visuales o textuales, según sea el modo de representación más adecuado para su propósito.



INTRODUCCIÓN A MDE



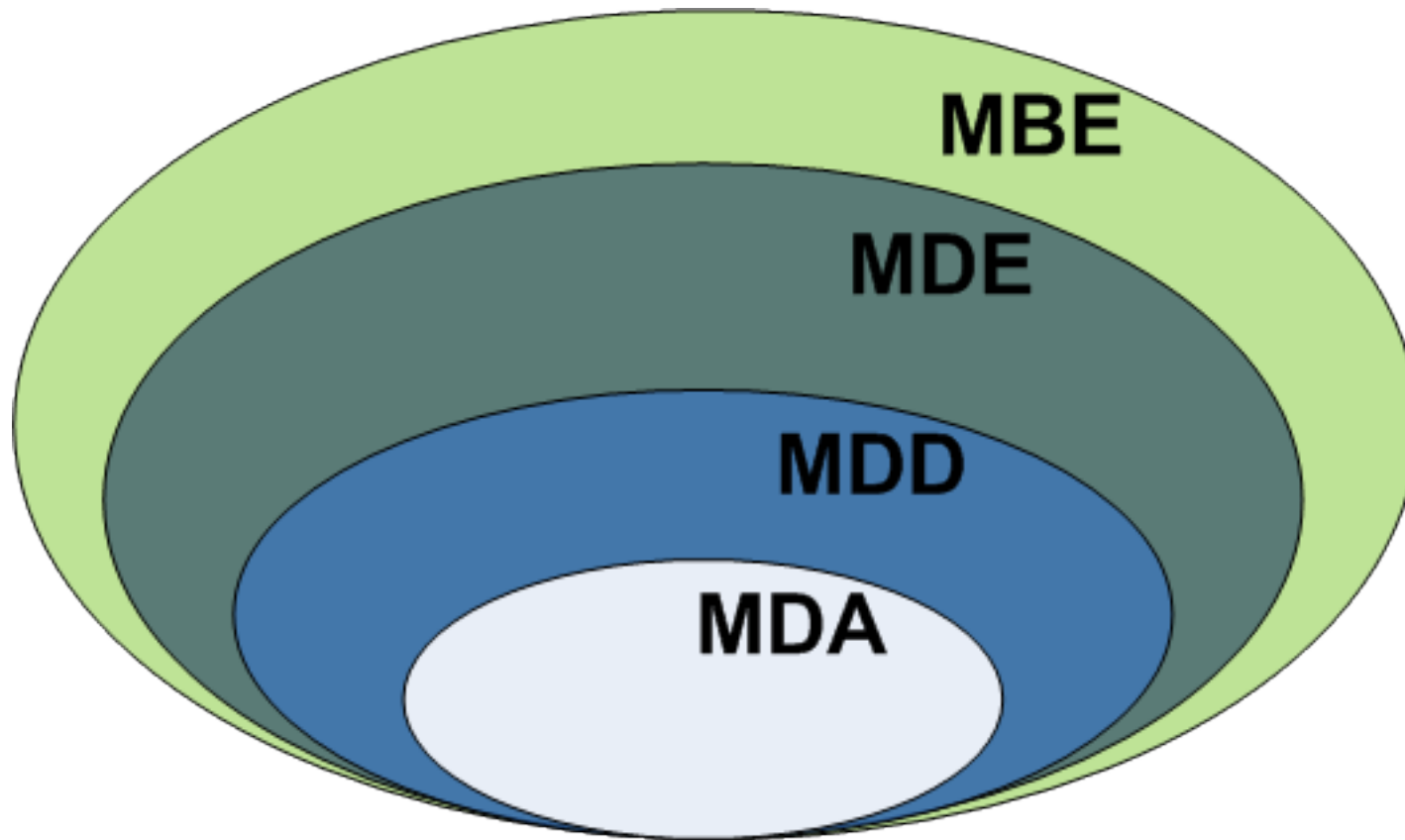
RESUMEN

Resumen

- El enfoque MDE promueve el diseño, desarrollo y transformación de modelos ‘procesables por el computador’ para dar soporte a diversas actividades en Ingeniería del Software.
- El uso más habitual en MDE es generar código fuente (semi-)automáticamente (MDD)
- MDA es un método para la construcción de software basado en modelos, haciendo uso unos estándares específicos.
- Existe gran interés en el mundo académico e investigador. Sin embargo, aún no se ha consolidado completamente en la industria.

Resumen

- La jungla *model-driven**



<http://modeling-languages.com/clarifying-concepts-mbe-vs-mde-vs-mdd-vs-mda/>



Procesadores de Lenguajes 2

Introducción a la ingeniería del software dirigida por modelos

Curso 2013-2014

Iván Ruiz Rube

ivan.ruiz@uca.es