

# UML, OCL y Patrones en el contexto MDA

Ana Garis

email: [agaris@unsl.edu.ar](mailto:agaris@unsl.edu.ar)



---

Maestría en Ingeniería de Software



# Agenda

- **Model Driven Architecture (MDA)**
- Unified Modeling Language (UML) y Perfiles UML
- Object Constraint Language (OCL)
- Patrones
- Conclusiones

# Modalidad del curso

- Marco teórico
- Trabajo práctico diario
- Trabajo práctico final

# **Arquitectura Dirigida por Modelos**

## **(Model Driven Architecture -MDA-)**



---

Maestría en Ingeniería de Software



# Contenido

## **Model Driven Architecture (MDA)**

- Motivaciones
  - Modelo
  - Lenguajes de especificación
  - Metamodelo
- Introducción a MDA
  - Objetivos
  - Enfoque general
  - Concepto de plataforma

# Contenido

## **Model Driven Architecture (MDA)**

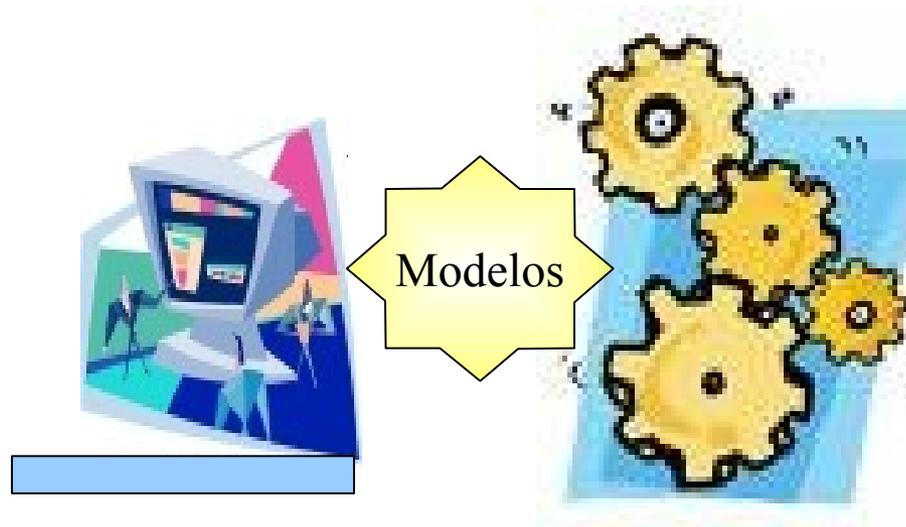
- El enfoque
  - Puntos de vista
  - Roles
- Transformación de modelos
  - Conceptos asociados
  - Descripción de la propuesta MDA

# Contenido

## **Model Driven Architecture (MDA)**

- MDA en el proceso de desarrollo de SW
  - MDA en el contexto de RUP
  - MDA en el contexto de las Met. Agiles
- MDA aplicado
  - Casos de estudio
  - Planteo de algunos beneficios
- Herramientas

# Motivaciones



**Proceso de Desarrollo de SW**



# Motivaciones

## Definiendo “Modelo”

### Diccionario

1. Arquetipo digno de ser imitado que se toma como pauta a seguir
2. Representación a escala reducida de alguna cosa.
3. Figura de barro, yeso o cera que se reproduce en un material más sólido.



### Wikipedia

Según el contexto, modelo puede referirse a: \* Modelo científico, una representación de la realidad\* Modelo matemático;\* Modelo numérico ...  
[es.wikipedia.org/wiki/Modelo](http://es.wikipedia.org/wiki/Modelo)

### Una definición

**“Un modelo de un sistema es una descripción o especificación de ese sistema y su ambiente para algún cierto propósito”**



# Motivaciones

## **¿Por qué usamos modelos?**

- Especificar un sistema de SW
- Comprender el problema que se desea resolver
- Comunicar con otros desarrolladores de SW
- Analizar posibilidades de reutilización

# Motivaciones

## ¿Cómo construimos modelos?

- Lenguaje de modelado

  - informal

  - semi-formal   -> UML

  - formal         -> OCL

- Reuso de modelos que resultaron exitosos   -> Patrones



# Motivaciones

## Características deseables de los modelos

### Un “buen” modelo

- Abstrae solo los aspectos relevantes
- Ayuda al razonamiento sobre el problema
- Mas barato de contruir que el código
- Comunica con personas
- Comunica con máquinas

[Fuente: Stephen Meller]

### Niveles de madurez

- N0 Sin especificación
- N1 Especificación textual
- N2 Texto con modelos
- N3 Modelos con texto
- N4 Modelos precisos
- N5 Solamente modelos

[Fuente: Kleppe and Warmer]

### Niveles de madurez

- N0 No UML
- N1 No UML
- N2 Poco UML
- N3 Mucho UML
- N4 Mucho UML + OCL
- N5 -----

# Motivaciones

## Conceptos previos

### **Punto de Vista**

“Un punto de vista de un sistema es una técnica de abstracción”

### **Vista**

“Una vista de un sistema es una representación de ese sistema desde una perspectiva de un punto de vista elegido”

### **Diagrama**

Elemento gráfico que forma parte de la descripción de una vista.

# Motivaciones

## **Metamodelo**

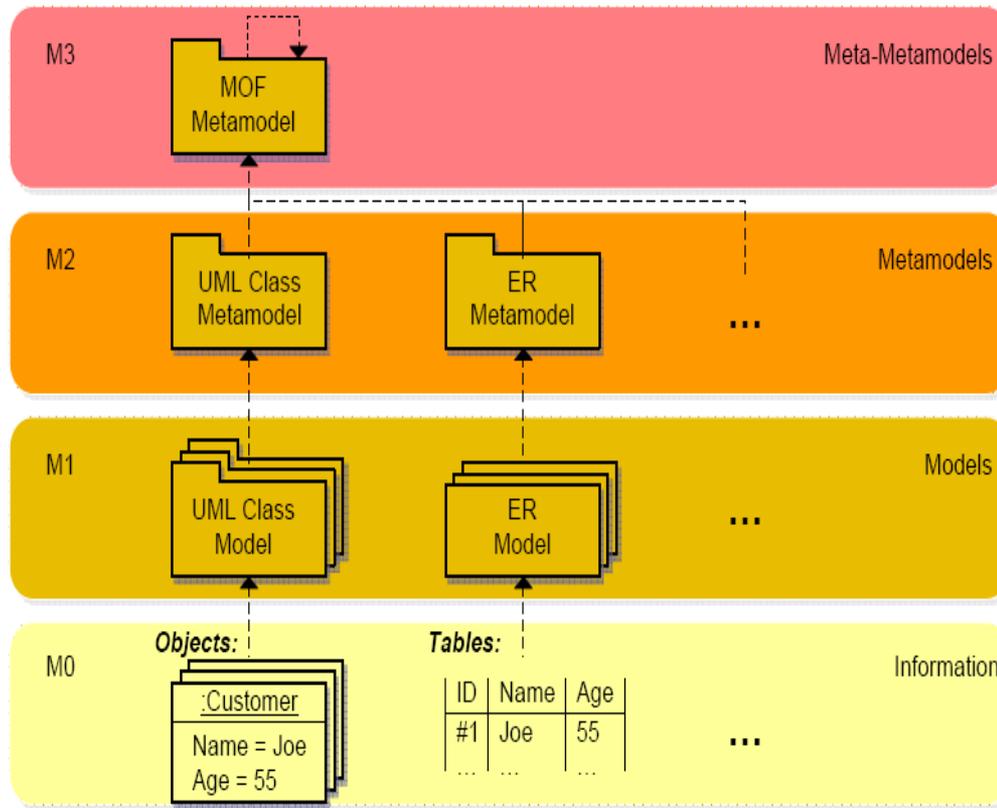
Para especificar un **modelo** usamos un **lenguaje**.

Para especificar un **lenguaje** usamos un **metamodelo**.

Para especificar un **metamodelo** usamos un **metalenguaje**

# Motivaciones

## Jerarquía de metamodelos



# Motivaciones

## Modelos

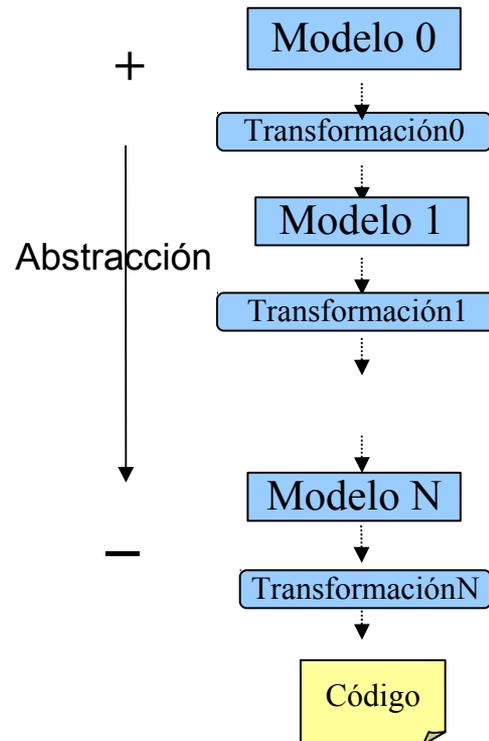
- Principalmente usados para documentación
- Cambios en el modelo no reflejados en el código y viceversa.
- Desconexión entre modelos (horizontal y vertical)
- Escasa transformación de modelos
- Modelos de software inmaduros con respecto a modelos de otras ingenierías.



Desarrollo Dirigido por Modelos

# Motivaciones

## Desarrollo dirigido por modelos (Model-Driven Development -MDD)



# Motivaciones



## Arquitectura dirigida por modelos (Model-Driven Architecture -MDA)



### La propuesta

[Object Management Group. Model Driven Architecture Guide, 2003]

Estandarizar MDD promoviendo el uso de lenguajes estándares

MOF (Meta Object Facility)

UML (Unified Modeling Language)

OCL (Object Constraint Language)

XMI (XML Metadata Interchange)

CWM (Common Warehouse Model)

# Motivaciones

## Estándares de la OMG

**MOF (Meta Object Facility):** Lenguaje abstracto para la especificación de metamodelos.

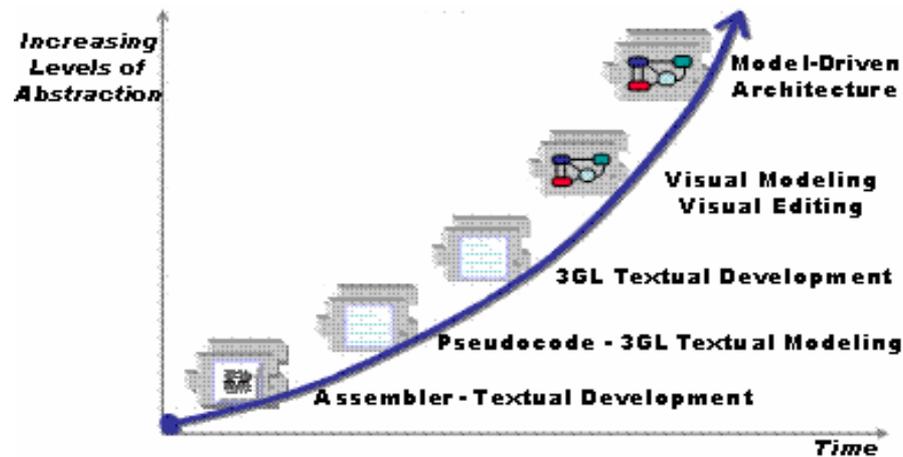
**XMI (XML Metadata Interchange):** Lenguaje para mapear MOF a XML. Proporciona reglas para transformar metamodelos MOF en DTD (o XML schema), y para transformar modelos basados en metamodelos MOF en archivos XML.

**CWM (Common Warehouse Metamodel):** Define un metamodelo, para modelar metadatos y la mayoría de objetos encontrados en un ambiente data warehousing

# Motivaciones



## Arquitectura dirigida por modelos (Model-Driven Architecture -MDA)



[Fuente: IBM]

# Introducción a MDA



## Arquitectura dirigida por modelos (Model-Driven Architecture -MDA)



### El objetivo

- Brindar un framework con soporte para la especificación de modelos
- Especificaciones permitirán creación de componentes y modelos de datos **interoperables**, **reusables** y **portables** utilizando estándares de modelado.

# Introducción a MDA



## Arquitectura dirigida por modelos (Model-Driven Architecture -MDA)



### El enfoque

- especificar un sistema independientemente de la plataforma que éste soporte,
- especificar plataformas,
- elegir una plataforma particular para el sistema,
- transformar la especificación del sistema en una de las plataformas.

[Object Management Group. Model Driven Architecture Guide, 2003]

# Introducción a MDA

## Plataforma

### Definición

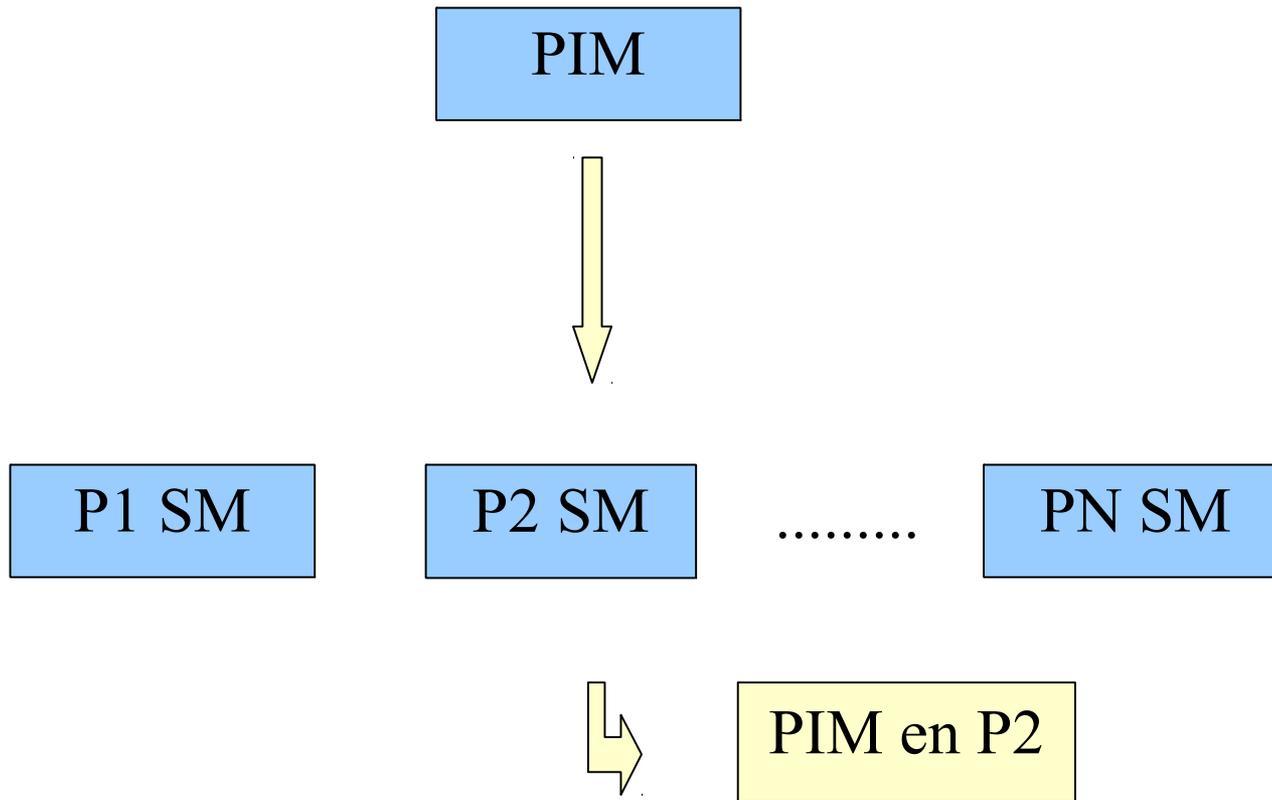
Conjunto de subsistemas y tecnologías que proveen un conjunto coherente de funcionalidades sin mostrar el detalle de cómo estas son implementadas.

### Tipos

- Genérica (Object, Dataflow, etc.)
- Específica de la tecnología (CORBA, J2EE)
- Específica del vendedor (Borland VisiBroker, IBM WebSphere)



# El enfoque

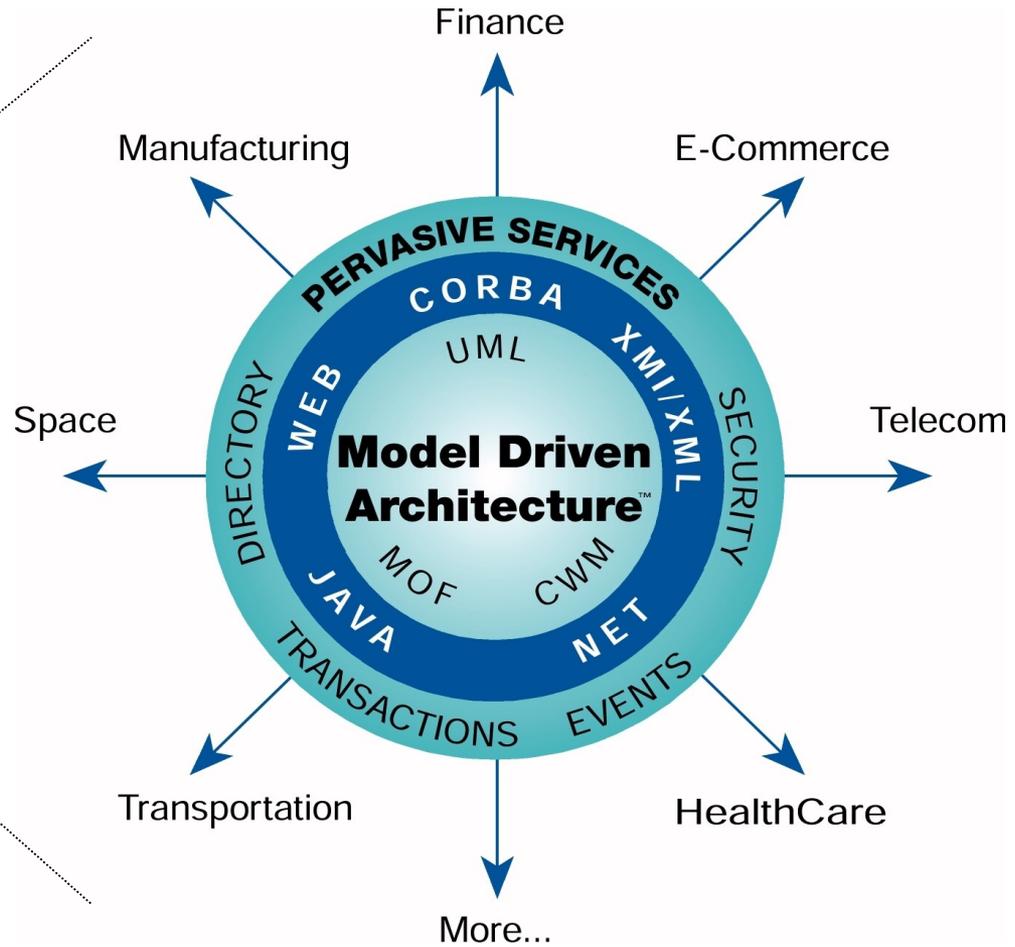


# El enfoque

## **¿Por qué separar de acuerdo a las plataformas?**

- Existen numerosas plataformas y tecnologías que deben interoperar (objetos distribuidos, aspectos, servicios web, etc).
- Evolución tecnológica de cada plataforma (surgen nuevas versiones)

# El enfoque



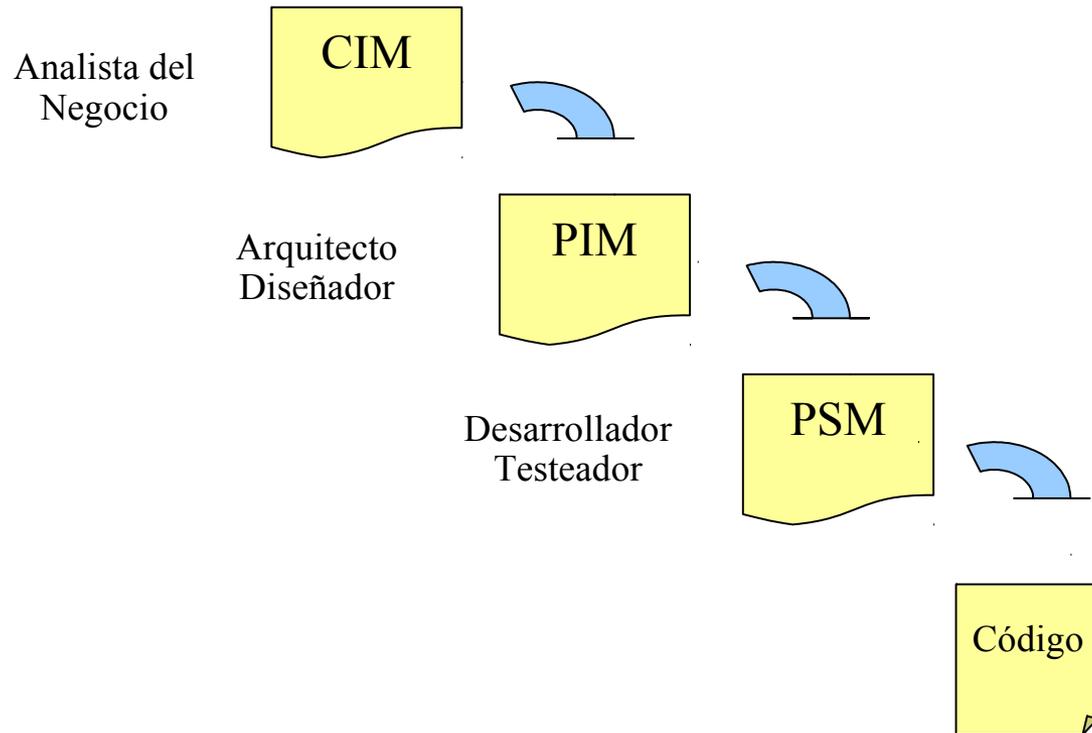
# El enfoque

## **Puntos de Vista en MDA**

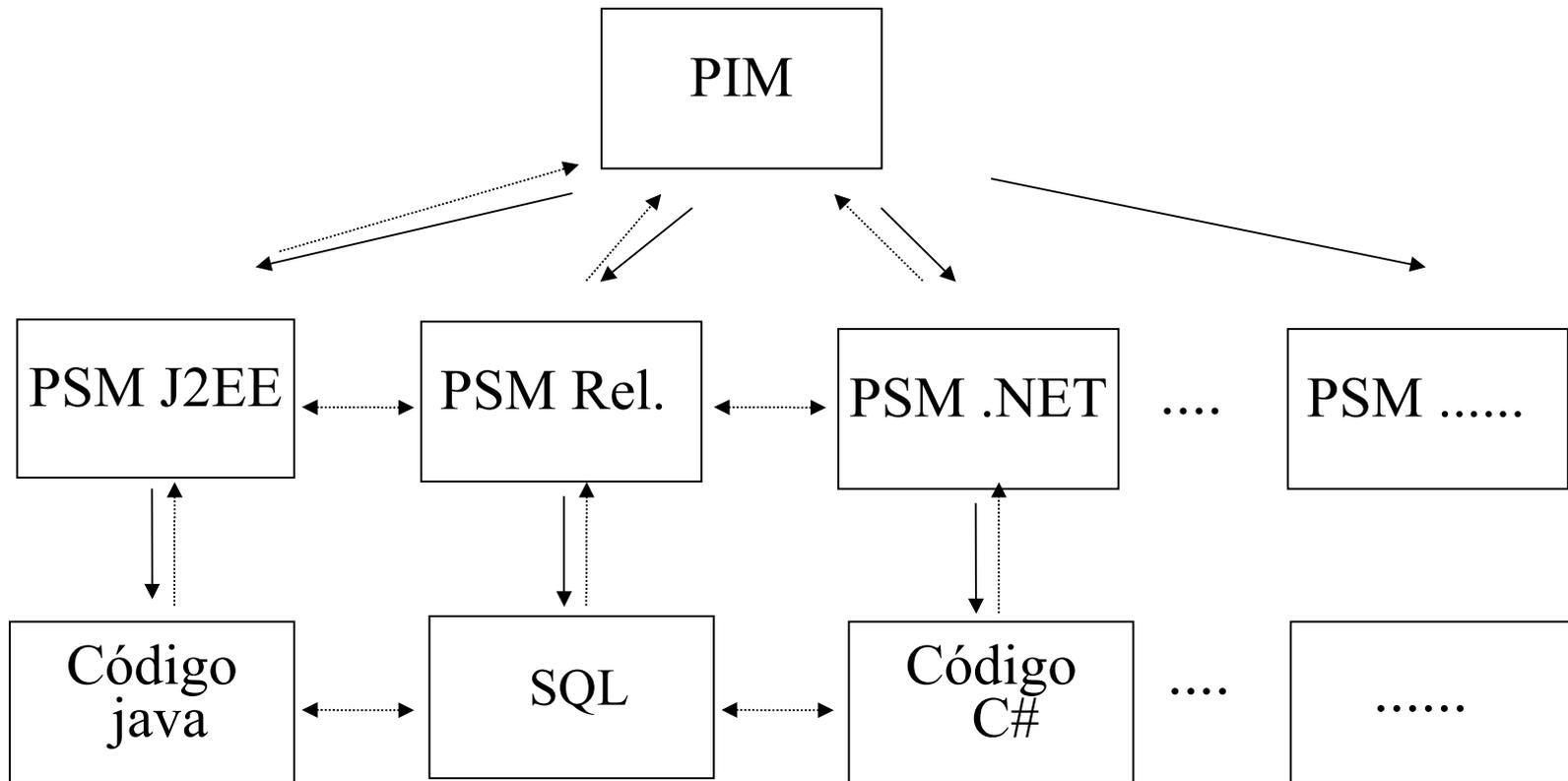
- **CIM (Computation Independent Model)**  
Descripción de la lógica del negocio desde una perspectiva independiente de la computación
- **PIM (Platform Independent Model)**  
Descripción de la funcionalidad del sistema en forma independiente de las características de plataformas
- **PSM (Platform Specific Model)**  
Descripción del sistema en términos de una plataforma

# El enfoque

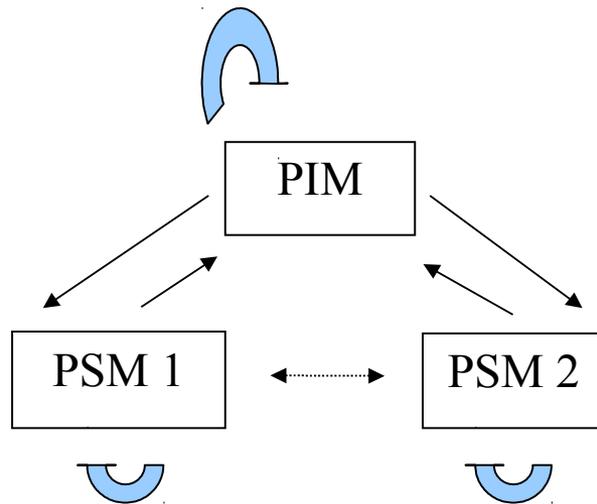
## Roles



# El enfoque



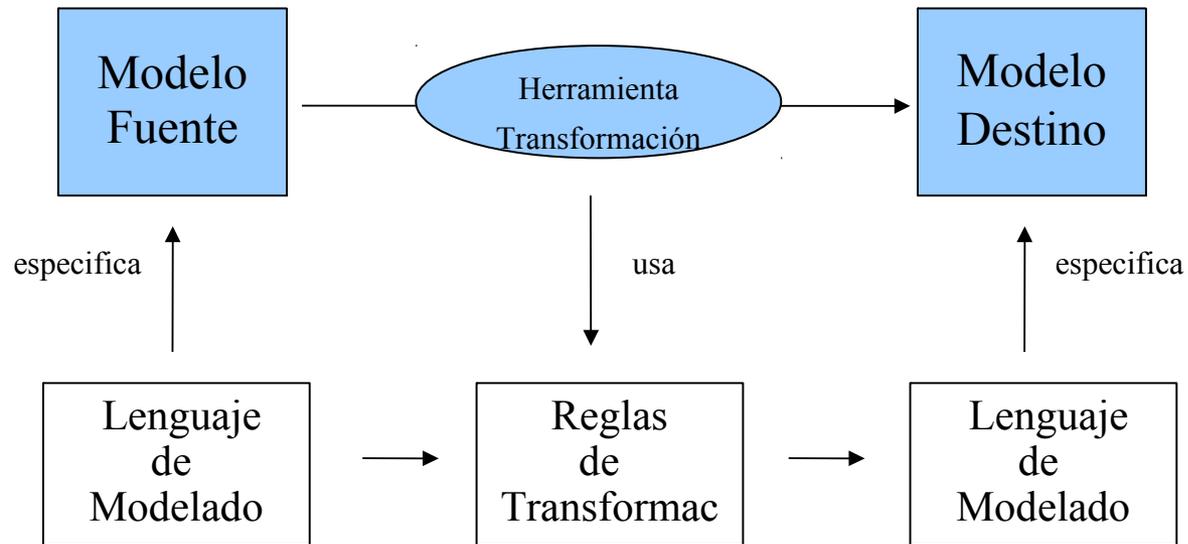
# Transformación de modelos



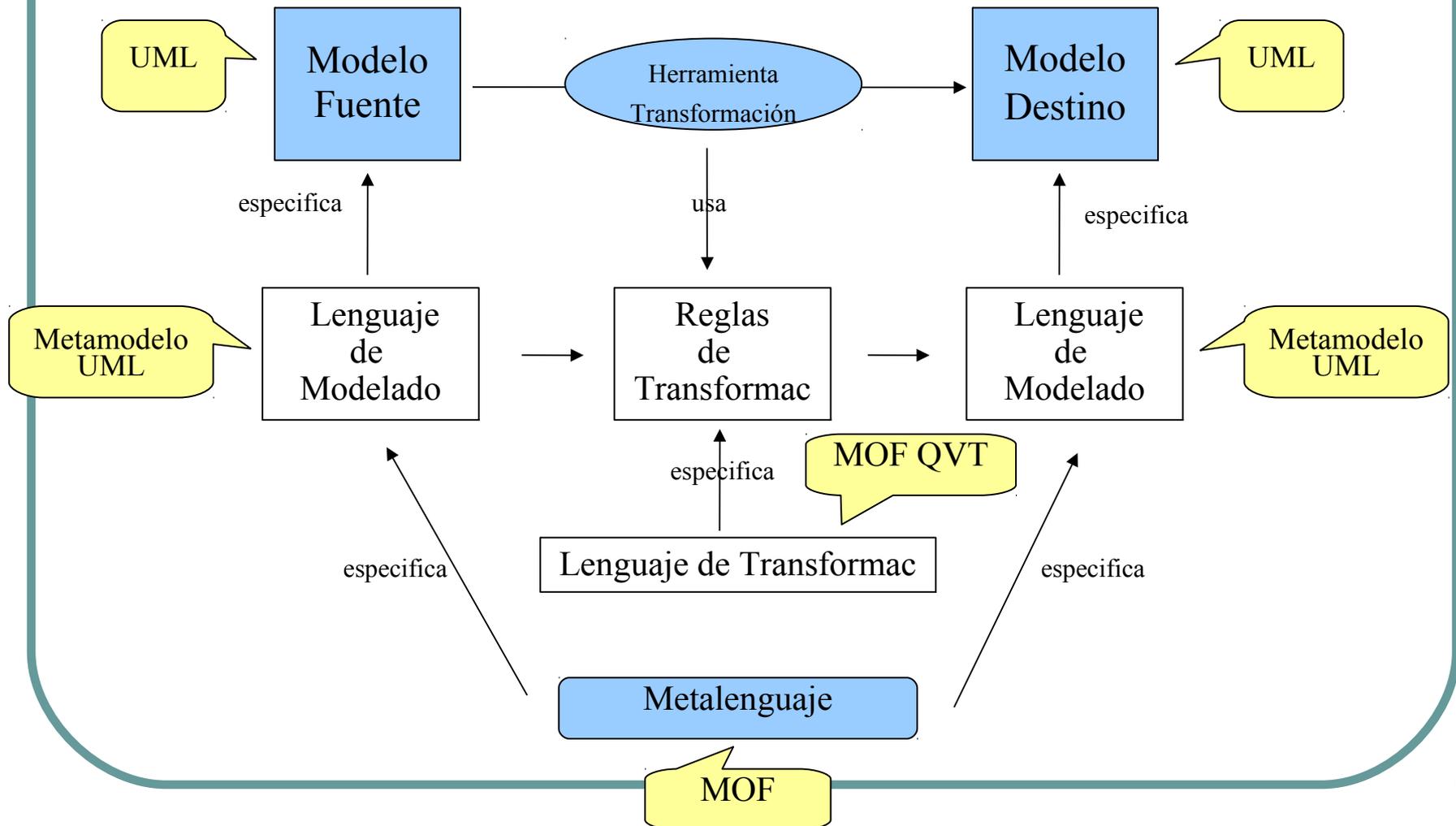
## Algunos conceptos

- Refinamiento
- Antirefinamiento
- Refactoring
- Puentes

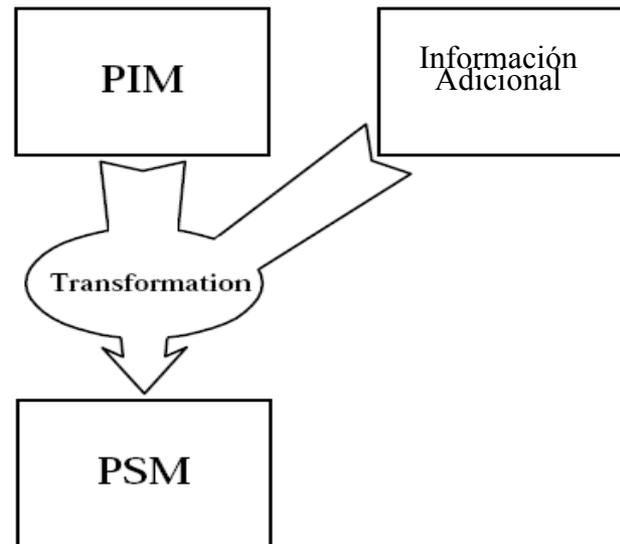
# Transformación de modelos



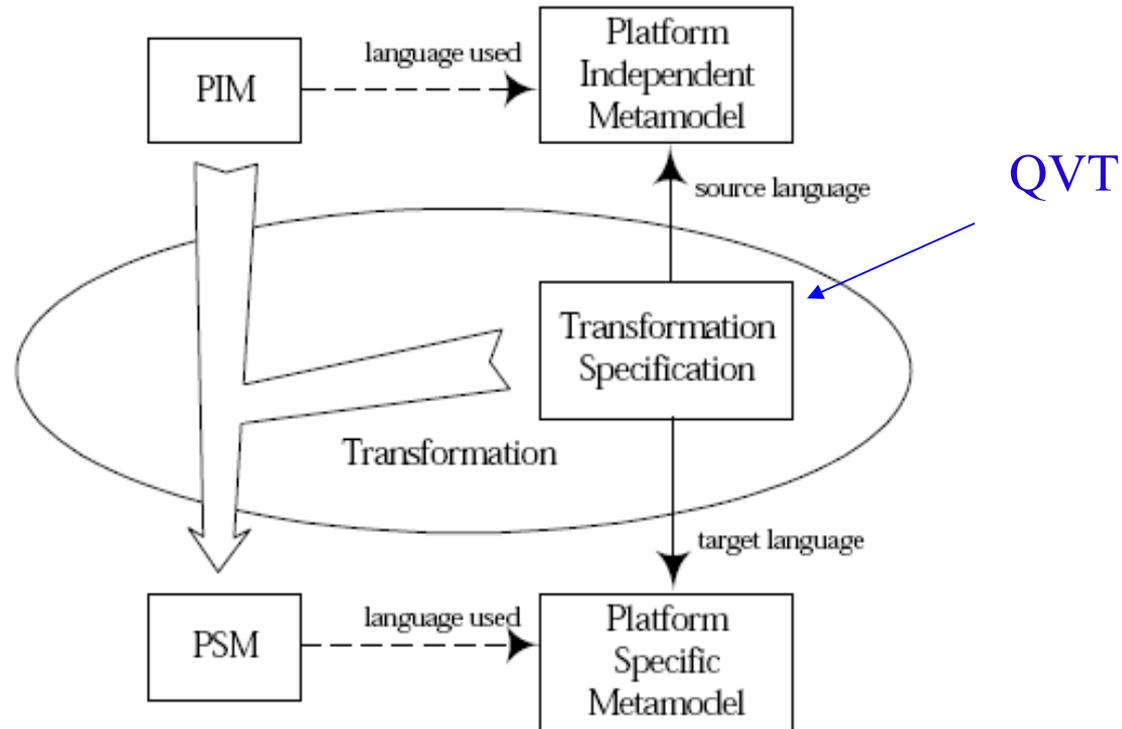
# Transformación de modelos



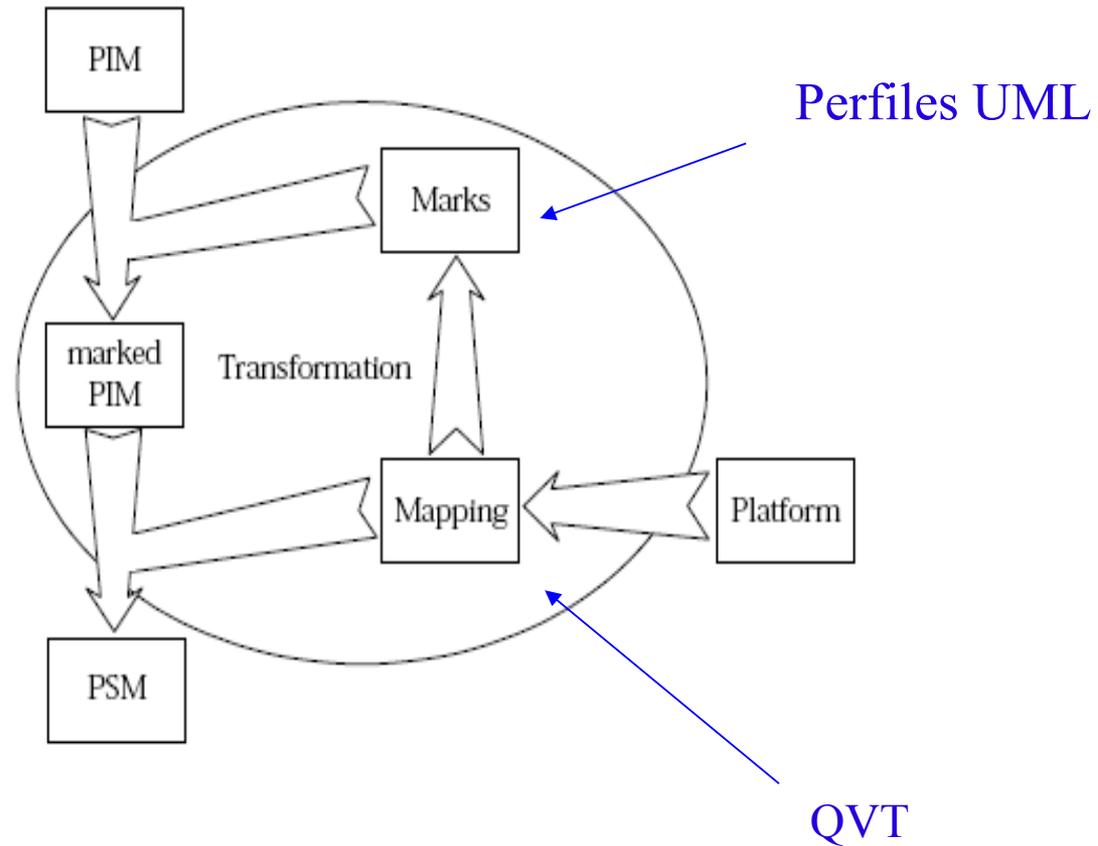
# Transformación de modelos



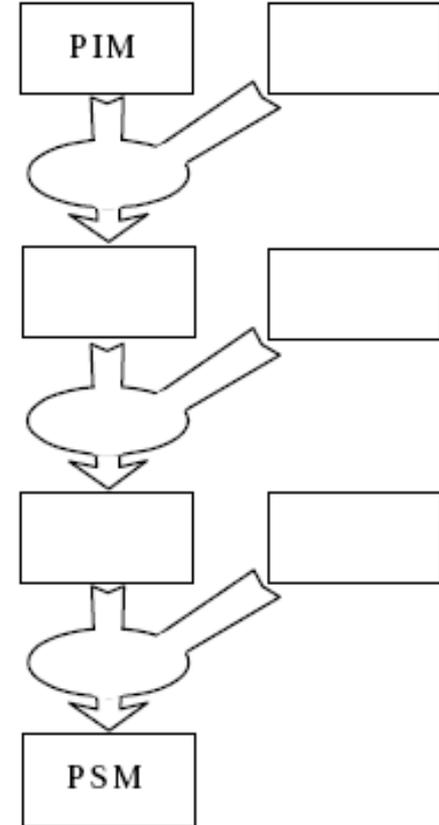
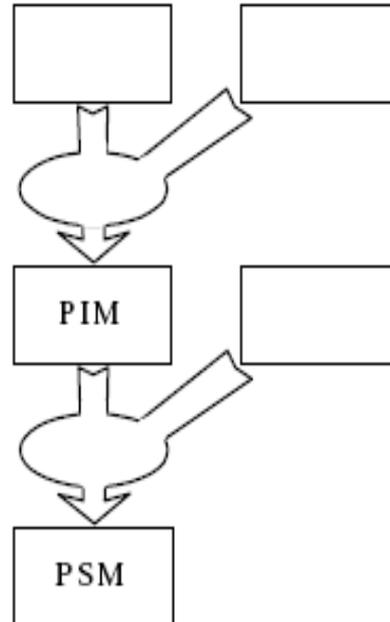
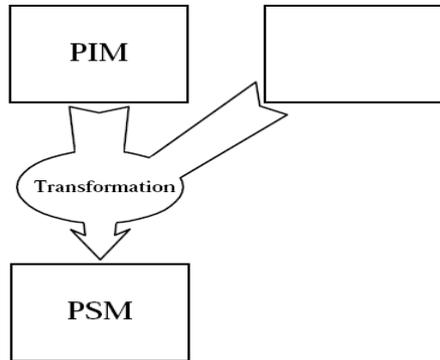
# Transformación de modelo



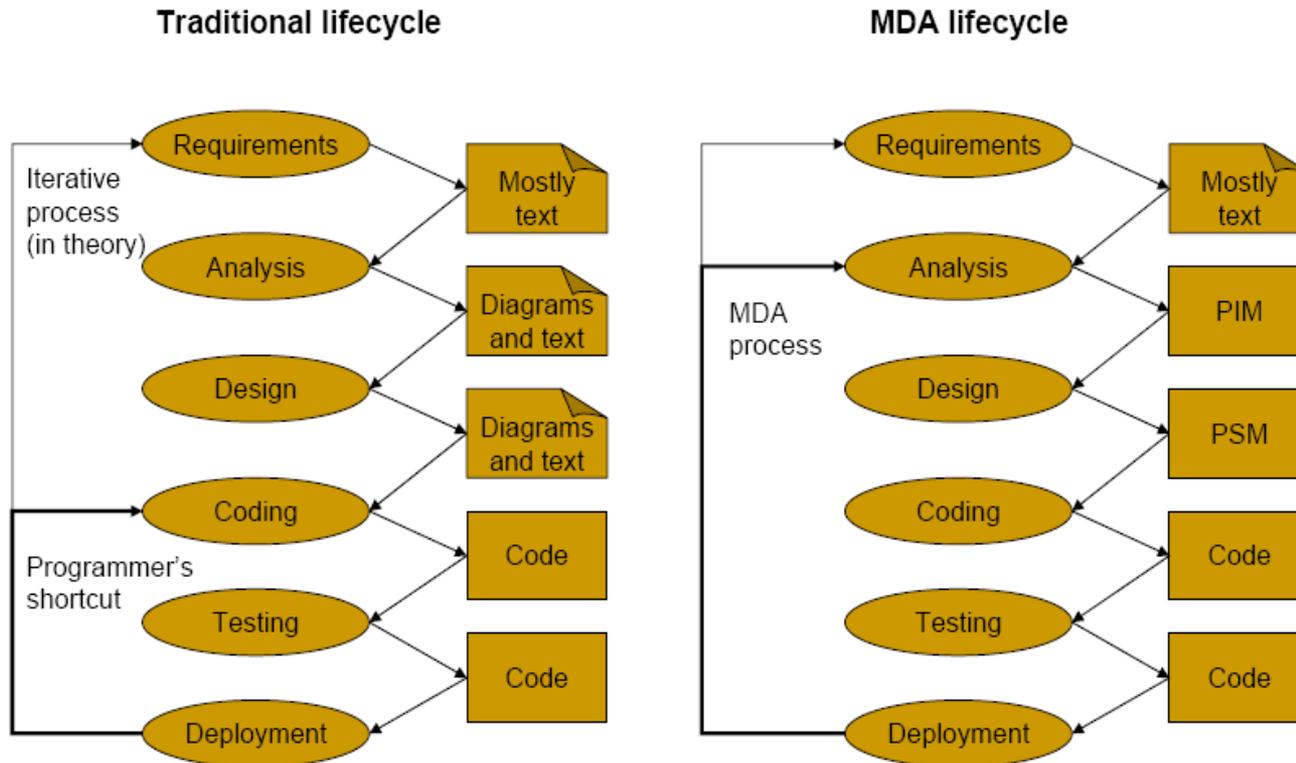
# Transformación de modelo



# Transformación de modelos



# MDA en el Proceso de Desarrollo de SW



Source: Kleppe et al 2003

# MDA en el Proceso de Desarrollo de SW

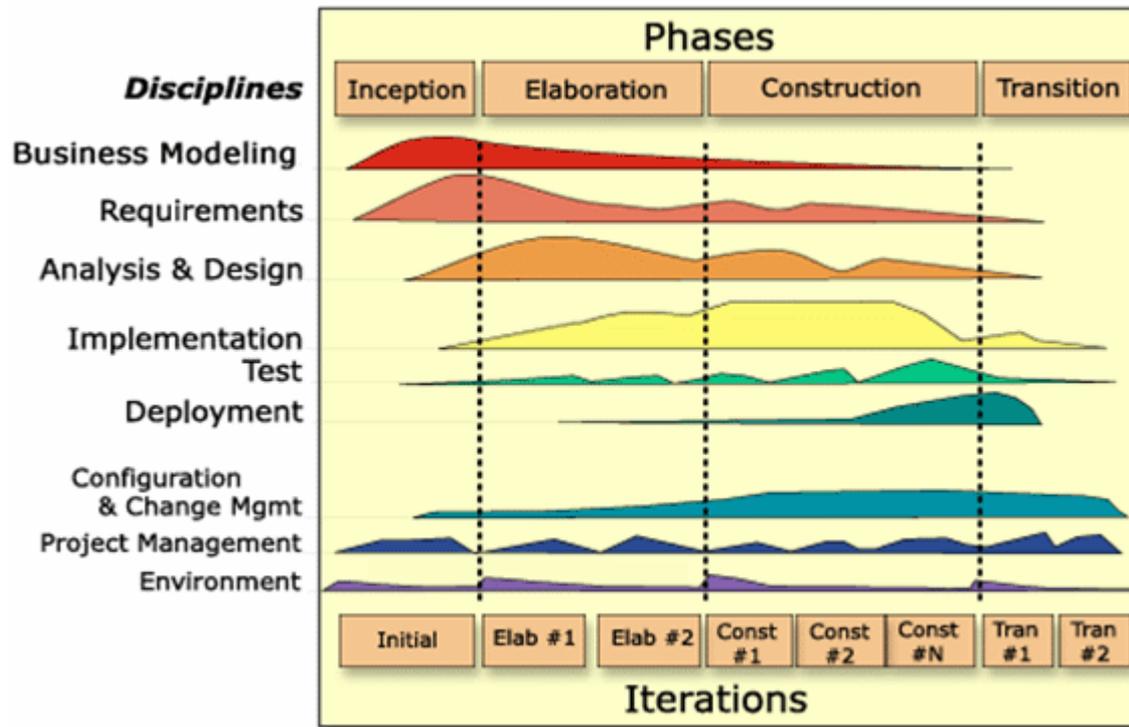
MDA **no** es un método de desarrollo de SW

MDA en contexto RUP ?

MDA en contexto Met. Agiles ?

# MDA en el Proceso de Desarrollo de SW

## MDA en el contexto de RUP



Brindar tareas adicionales para automatizar actividades RUP

# MDA en el Proceso de Desarrollo de SW

## MDA en el contexto de Met. Agiles



Agil MDA - Stephen Mellor

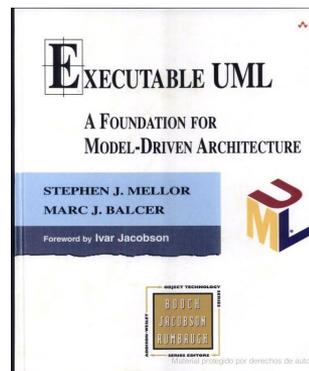


Agile Model Driven Development (AMDD) – Scott Ambler

# MDA en el Proceso de Desarrollo de SW

## Agil MDA

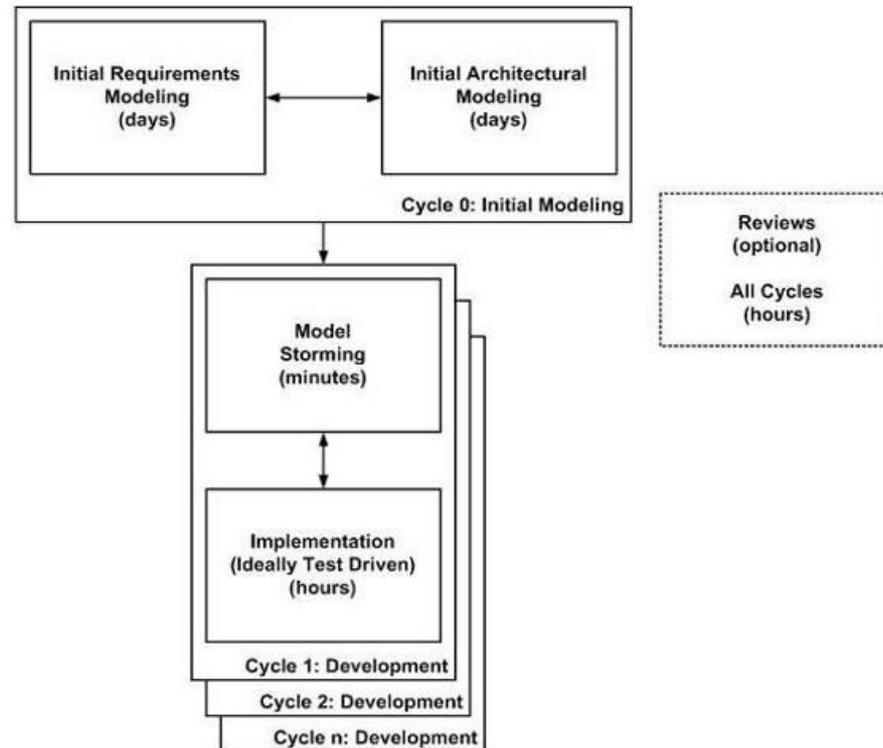
- construir casos de test
- escribir “modelos ejecutables”
- compilar modelos usando un compilador de modelos
- correr casos de test
- entregar fragmentos del sistema al cliente incrementalmente



# MDA en el Proceso de Desarrollo de SW

## Agile Model Driven Development (AMDD)

- ✓ Describe cómo desarrolladores y partes interesadas pueden trabajar cooperativamente para crear modelos “apenas lo suficientemente buenos”,
- ✓ No requiere que los participantes sean expertos en modelado,
- ✓ No depende de sofisticadas herramientas CASE.



# MDA aplicado

## Caso 1: Middleware company

- Objetivo: Analizar si el uso de MDA implica un aumento de productividad
- Contexto: 2 equipos de desarrollando una aplicación, un equipo usando MDA y el otro un enfoque tradicional centrado en el código.
- Tiempo: appox. 4 semanas (MDA) y 5 semanas (code-centric) en 2003
- Herramientas usadas: herramientas UML estándar.

### Resultados

- 35% esfuerzo ahorrado, luego de aprender MDA tool en la tarea
- El equipo MDA estimó que la próxima vez debería ser 10-20%mas rapido
- Tiempo de desarrollo mas corto que el esperado
- Amplio uso de patrones en el equipo MDA.
- Nivel de errores encontrados durante proceso de testing mayor en equipo No MDA



# MDA aplicado

## Caso 2: Volvo IT

- Objetivo: Desarrollar un framework MDA para modelado y desarrollo específico SOA
- Contexto: 5 desarrolladores.
- Tiempo: approx. 2 años (2002-2004)
- Herramientas usadas: Rational XDE con Eclipse y MS Visual Studio. NET

### Expectativas previas

- Maximizar calidad y mantenibilidad del código.
- Minimizar tiempo de entrega
- Trazabilidad desde las reglas de negocio al código fuente
- Mejorar la consistencia arquitectónica



# MDA aplicado

## Caso 2: Volvo IT (cont.)

### Resultados

- Inmadurez tecnológica
- Demasiado alta curva de aprendizaje
- Problemas para el uso de UML en sistemas de modelado basados en SOA.
- Falta de evaluación de la mejora que implica en la línea de producción

# MDA aplicado

## Beneficios

[MDA guide]

- ✓ Flexibilidad a largo plazo (implementación, integración, mantenimiento, testing y simulación)

[IBM, Rational Software Architect]

- ✓ Incrementa la productividad
- ✓ Mejora mantenimiento
- ✓ Permite reuso de componentes legacy
- ✓ Facilita adaptabilidad de nuevos componentes
- ✓ Facilita la consistencia
- ✓ Repetición
- ✓ Mejora la comunicación
- ✓ Captura experiencia
- ✓ Modelos como recursos a largo plazo
- ✓ Decisiones de tecnología demorada

# Discusión

## **MDA Manifesto**

**Grady Booch  
Alan Brown  
Sridhar Iyengar  
James Rumbaugh  
Bran Selic**

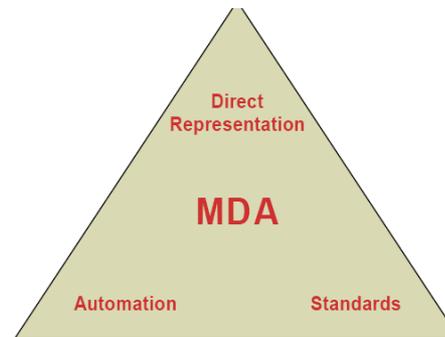
# Discusión

## Estrategia para lograr los objetivos MDA

**Representación directa:** Usar el principio de abstracción separando aspectos relevantes del problema de las decisiones de tecnología.

**Estándares abiertos.** El uso de estándares posibilita la integración de herramientas.

**Automatización.** La propuesta de MDA fortalece el papel que las herramientas CASE, aunque surgen nuevas funcionalidades que deben ser soportadas. El rol de las herramientas es esencial en MDA.



# Discusión

## Herramientas

### → CASE-UML

Ej: Rational Rose, ArgoUML, MagicDraw, Poseidon, Together, etc

### → CASE-MDA

- OptimalJ
- Objecteering
- ArcStyler
- AndromDA
- OpenModel
- Together Architect
- Codagen Architect
- Rational Architect
- iUML

# Discusión

## Herramientas

- ATL (ATLAS Transformation Language): Lenguaje para definir transformaciones con framework MDA.
- GMT (Generative Model Transformer): Creador de herramientas MDA

### **Integración de herramientas:**

*“Para modelar, necesitamos una infraestructura con la cual construir e integrar herramientas. Eclipse Modeling Framework (EMF) es un framework y proyecto de integración de herramientas open-source que soporta y usa estándares de modelado (UML, MOF y XMI)”*

[Fuente: Manifiesto MDA]