

# MDA: Uma Abordagem Prática

Paulo F. Pires  
DIMAp/UFRN

<http://www.dimap.ufrn.br>  
[paulo.pires@dimap.ufrn.br](mailto:paulo.pires@dimap.ufrn.br)

Porto Alegre, Agosto de 2008

## Tópicos

- **Parte 1 – Conceitual**
  - O que é MDA?
  - Arquitetura
  - Metamodelagem
    - *Modelo em Níveis*
    - *MOF*
    - *Mapeamentos*
    - *Metamodelagem & MDA*
  - Transformação entre modelos
    - *Conceitos básicos*
    - *Modelo para Modelo: MOV-QVT & ATL*
    - *Modelo para Texto: MOFScript*

## Tópicos

- **Parte 1 (Continuação)**
  - Processo de desenvolvimento com MDA
- **Parte 2 – Prática**
  - Ferramentas MDA/MDD
  - OpenArchitectureWare
    - *Exemplo*
  - [AndroMDA 4
    - *Exemplo*]



Minicurso 2: MDA: Uma abordagem Prática

## Evolução da visão OO

- **“Tudo é um objeto”**
  - Jean Bézivin (Conferência UML 2003)
    - *Um dos princípios mais fortes que levaram a melhorias tecnológicas nos últimos 20 anos*
    - *Enquanto esse princípio foi seguido houve progresso constante*



- **“Tudo é um modelo” é o princípio motor de MDA**
  - Enquanto esse princípio for seguido haverá progresso constante



Minicurso 2: MDA: Uma abordagem Prática

## Promessa da tecnologia de Objetos

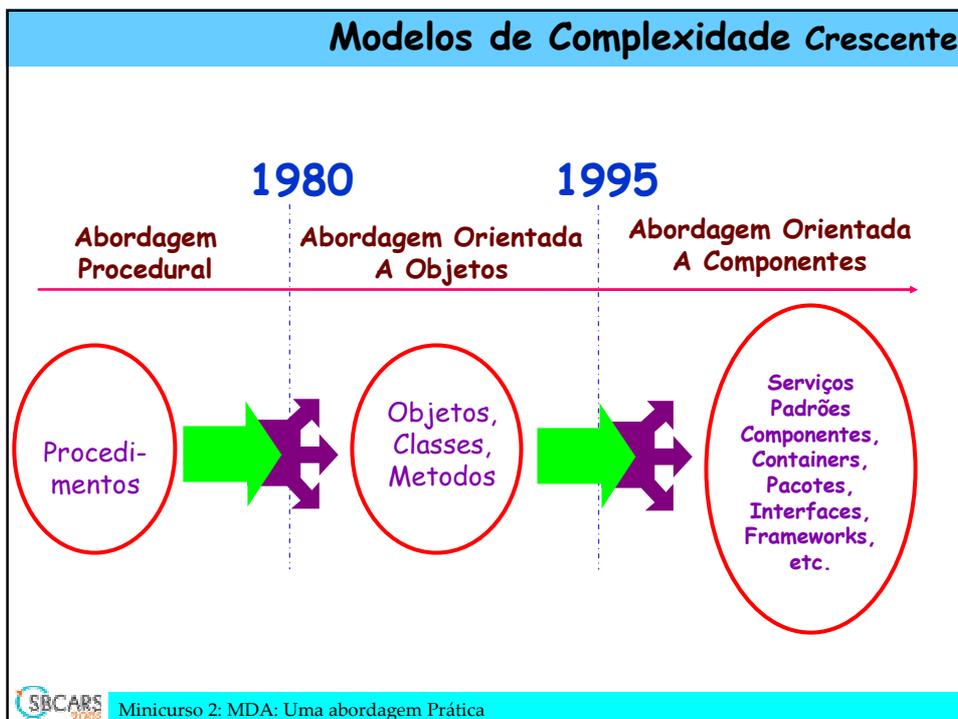
"Because of the wonderful unifying properties of the object paradigm, the transition from procedural technology to object technology will bring **huge conceptual simplification** to the software engineering field. Since everything will be considered as an object, we shall observe a dramatic **reduction in the number of necessary concepts**."

Todos nós ☺, década de 80

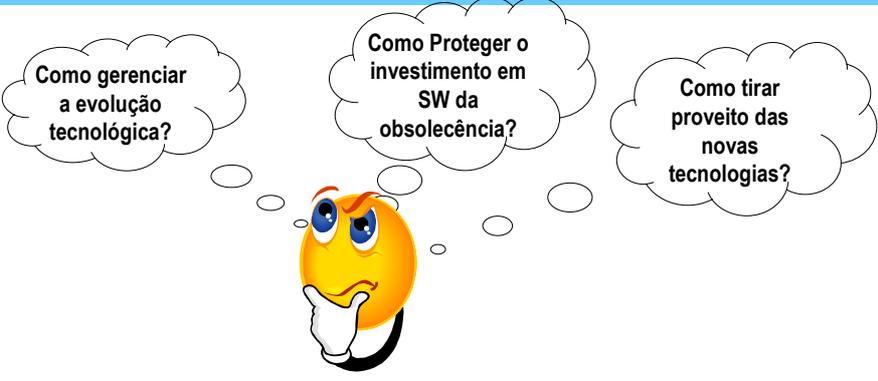


Minicurso 2: MDA: Uma abordagem Prática

## Modelos de Complexidade Crescente



## Evolução Tecnológica



Como gerenciar a evolução tecnológica?

Como Proteger o investimento em SW da obsolescência?

Como tirar proveito das novas tecnologias?

**Não existe (e nunca existirá) uma única solução tecnológica padrão**

↓

**Separar a tecnologia das funcionalidades**

↓

**Separar as funcionalidades dos conceitos**

 Minicurso 2: MDA: Uma abordagem Prática

## O que é MDA?

***“An approach to IT system specification that separates the specification of **system functionality** from the specification of the **implementation of that functionality** on a particular **technology platform**”***

Especificação MDA, Comitê de Arquitetura da OMG

- **“Design once, build it on any platform”**

 Minicurso 2: MDA: Uma abordagem Prática 8

### Separação Vertical de responsabilidades de um sistema:

Dependente de Plataforma x Independente de Plataforma

Alto custo de migração entre diferentes plataformas de middleware (COM, CORBA, Java, HTML, XML, DotNet, etc.)

Construção de modelos abstratos de negócio e serviços que garantam o isolamento com relação a evolução tecnológica.

Os provedores de plataformas tecnológicas devem prover soluções de mapeamento para os modelos de negócio e serviços



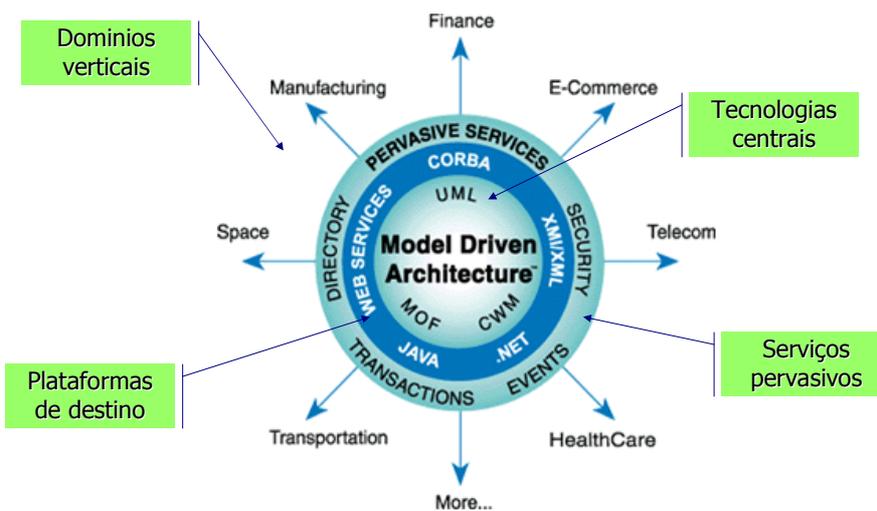
- **Longevidade**
  - Menor esforço para portar de uma plataforma para outra
  - Estratégia → Interoperabilidade entre diferentes ferramentas MDA
    - *Reúso de transformações*
- **Qualidade**
  - Derivada do aumento no nível de abstração
  - Testes/Validação/Verificação de modelos independentes de plataforma
- **Queda nos custos de desenvolvimento**
  - Um elemento de modelo → muitas linhas de código



## O que MDA oferece para o desenvolvimento de SW?

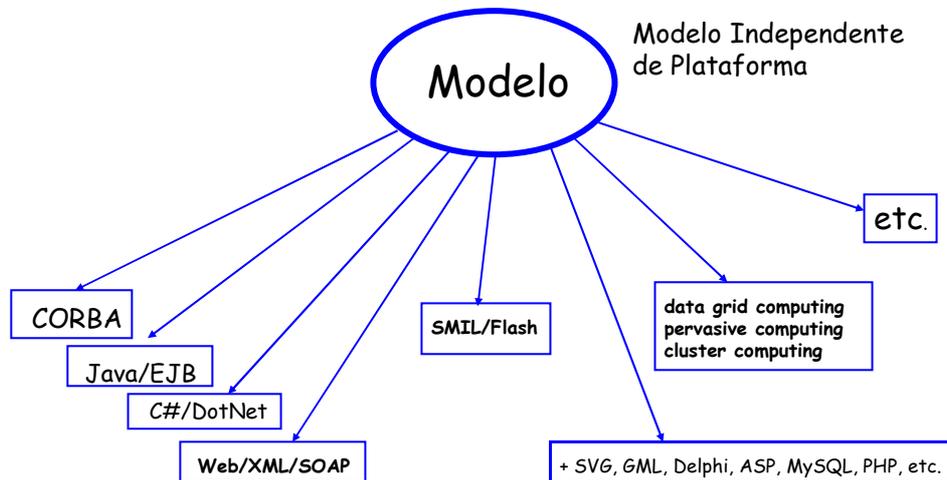
- **MDA fornece uma abordagem para:**
  - Especificar um sistema independentemente da plataforma de software de execução;
  - Especificar plataformas de execução;
  - Escolher uma plataforma específica de execução para um sistema e
  - Transformar a especificação do sistema em uma especificação voltada para uma plataforma de execução específica

## Visão Geral de MDA



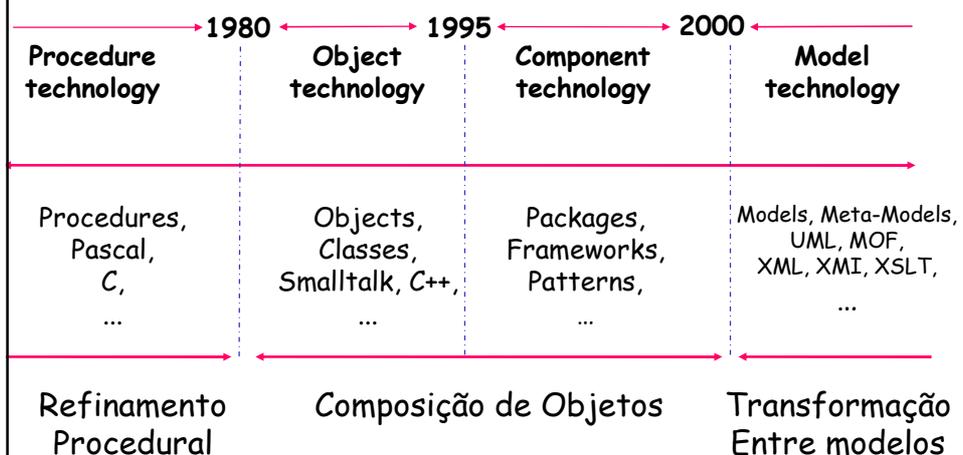
## MDA: Princípio Básico

Write Once, Run Anywhere → **Model Once, Generate Anywhere**



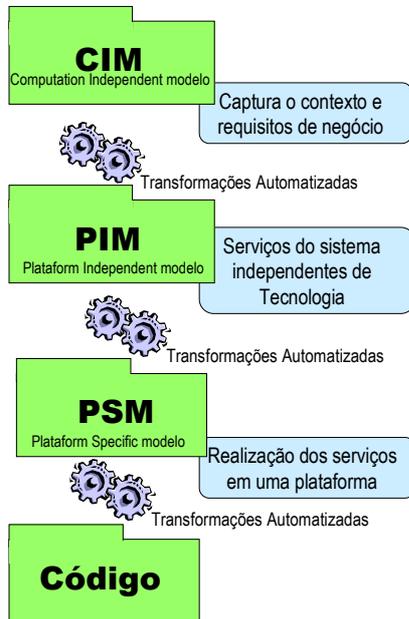
Minicurso 2: MDA: Uma abordagem Prática

## Dos Objetos Distribuídos para MDA



Minicurso 2: MDA: Uma abordagem Prática

## Níveis de Modelos

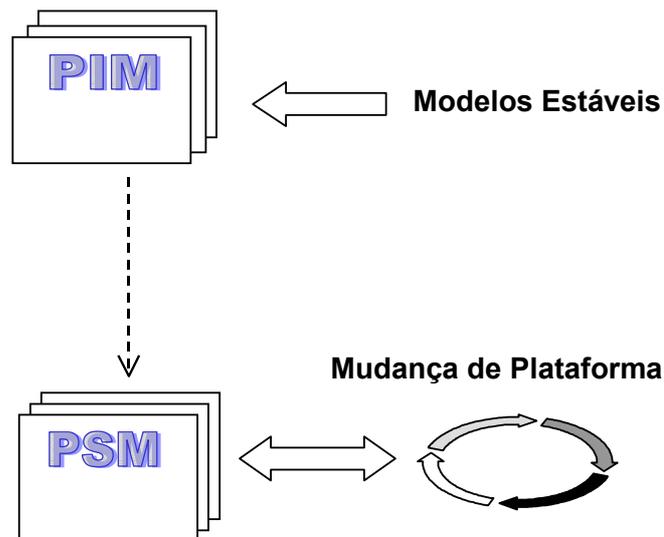


- Abordagem orientada a negócio para a construção de sistemas de software onde modelos são refinados a partir das necessidades do negócio até a sua realização em soluções de software

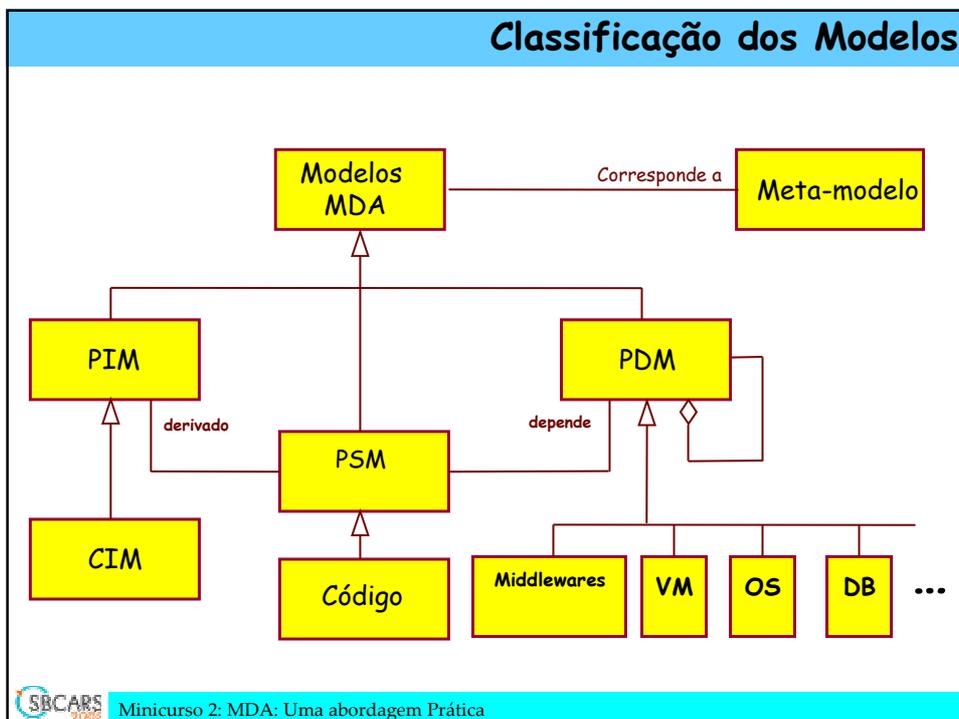
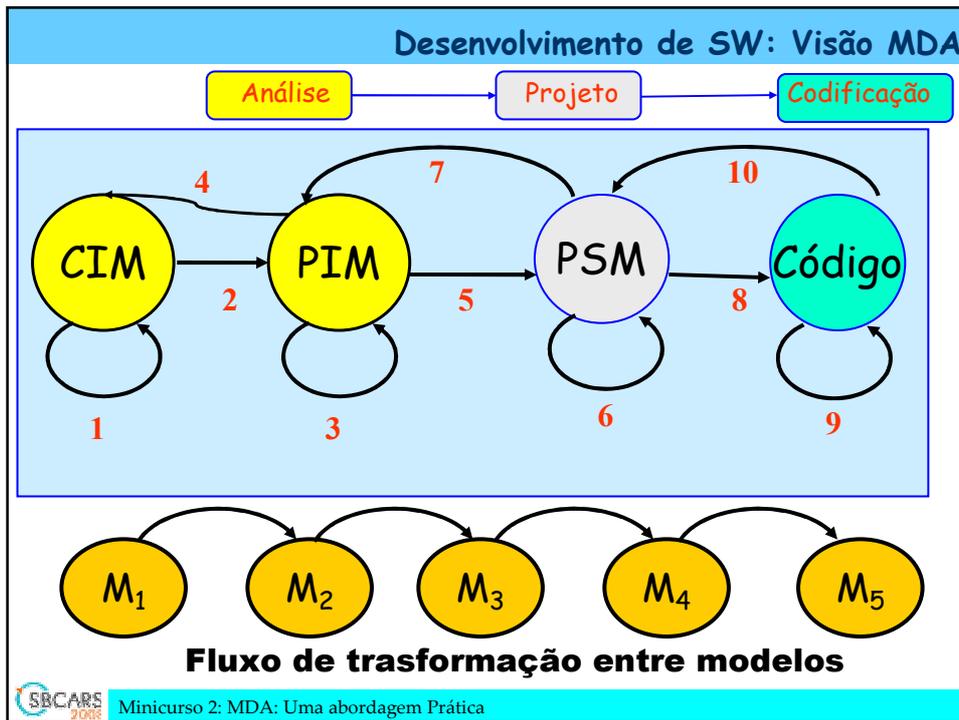


Minicurso 2: MDA: Uma abordagem Prática

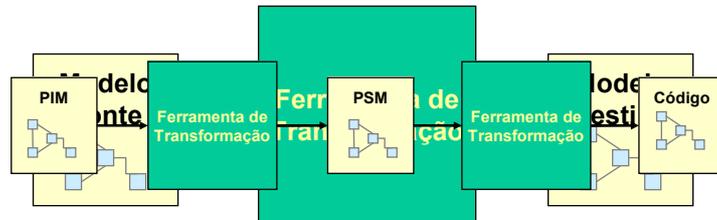
## Modelagem Independente de Plataforma



Minicurso 2: MDA: Uma abordagem Prática



## Framework MDA Básico

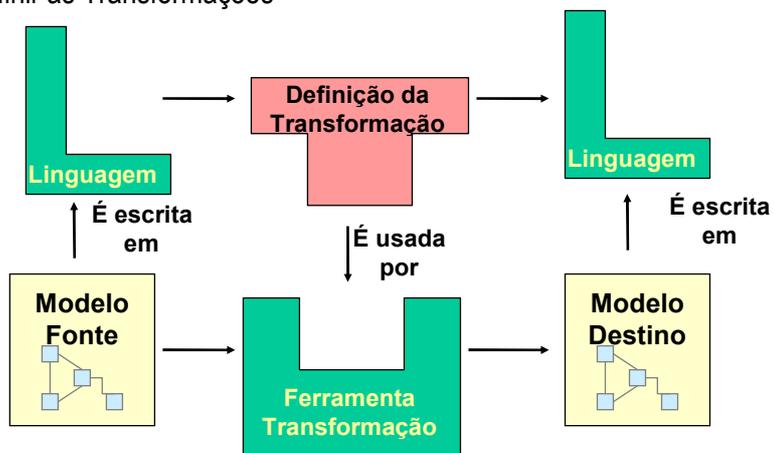


Minicurso 2: MDA: Uma abordagem Prática

## Framework MDA

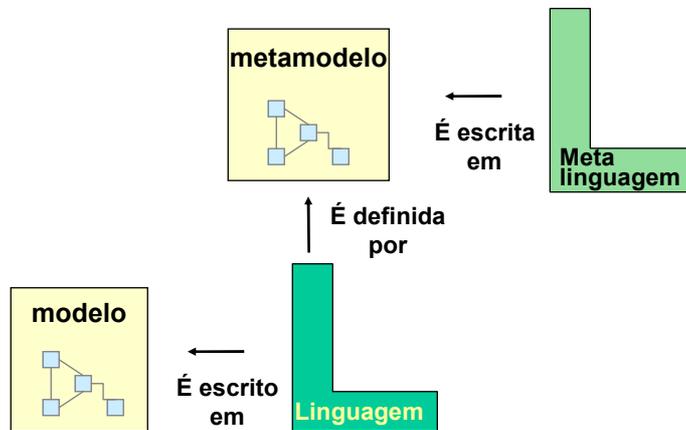
■ Para realizar essa funcionalidade é necessário:

- Definir as linguagens de modelagem
- Definir as Transformações



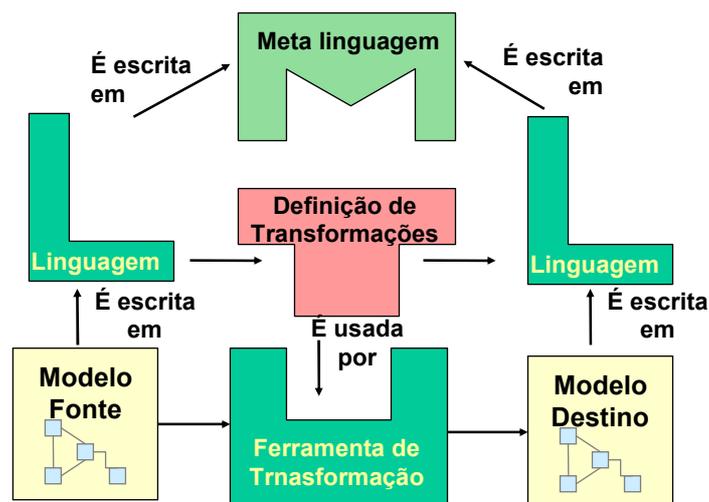
Minicurso 2: MDA: Uma abordagem Prática

## Definição das linguagens



Minicurso 2: MDA: Uma abordagem Prática

## Framework MDA



Minicurso 2: MDA: Uma abordagem Prática

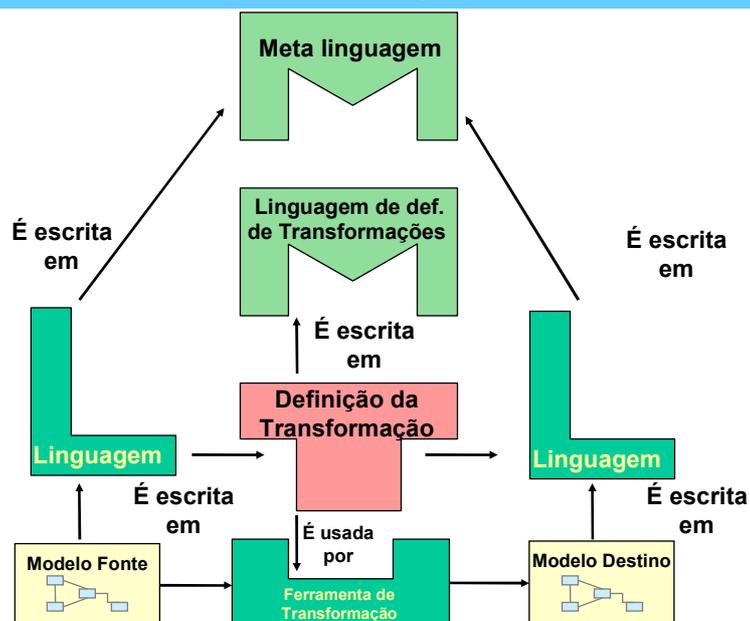
## Definição de Transformações

- **Uma definição de transformação mapeia:**
  - Elementos de um modelo (linguagem) fonte
- **Para**
  - Elementos de um modelo (linguagem) destino
- **Baseado nas respectivas definições de modelos (linguagens)**



Minicurso 2: MDA: Uma abordagem Prática

## Definição de Transformações



Minicurso 2: MDA: Uma abordagem Prática

## Desenvolvimento Dirigido a Modelos (1)

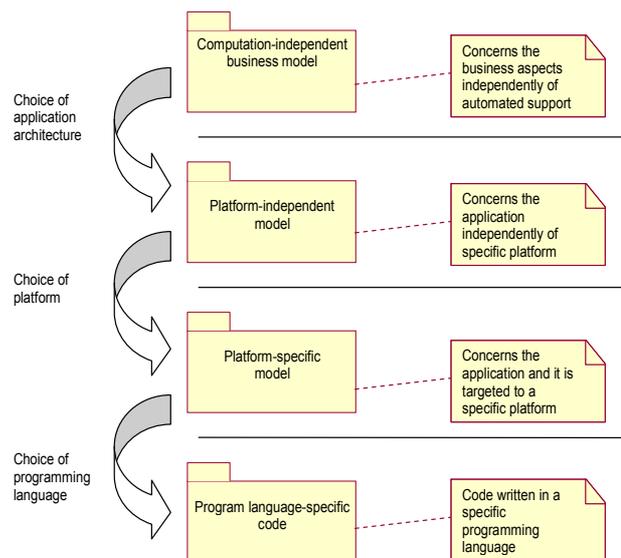
- MDA → Não define metodologia de desenvolvimento
- Necessidade:
  - Técnicas de modelagem independente de plataforma
  - Técnicas de modelagem dependente de plataforma
  - Técnicas de reuso de artefatos de modelo
  - Técnicas para definição de transformações
  - Etc...

### Model Driven Development

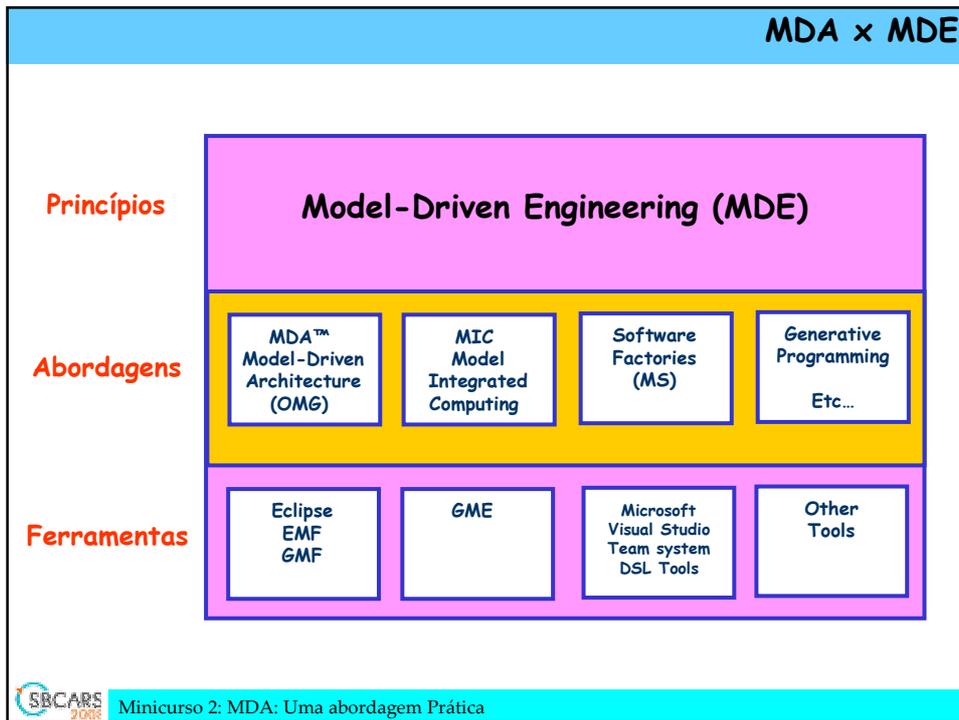


Minicurso 2: MDA: Uma abordagem Prática

## Desenvolvimento Dirigido a Modelos (2)



Minicurso 2: MDA: Uma abordagem Prática



  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

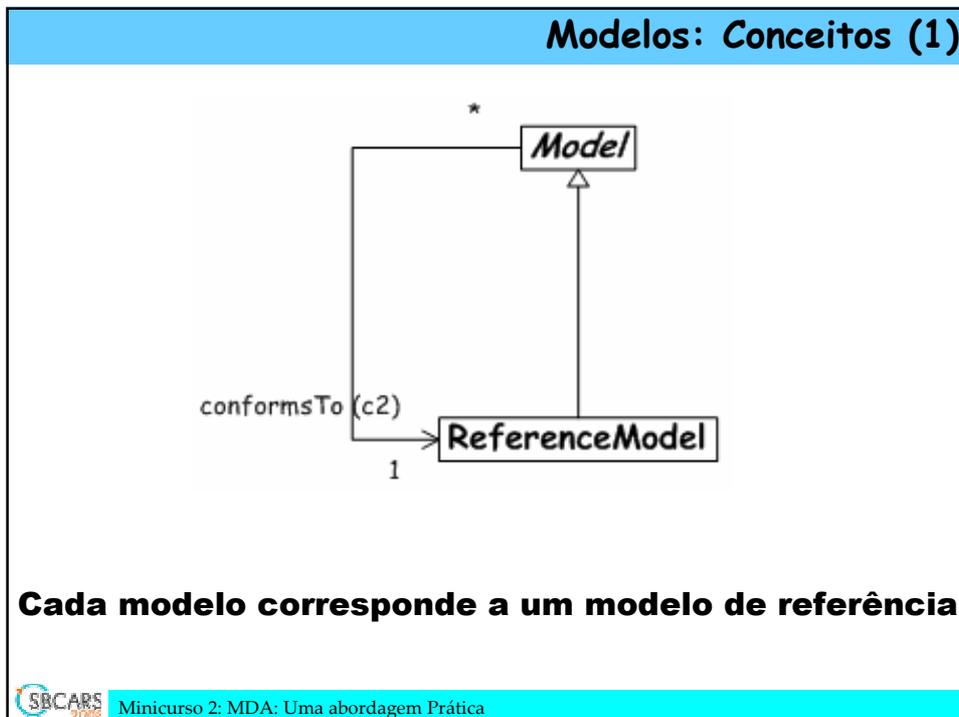
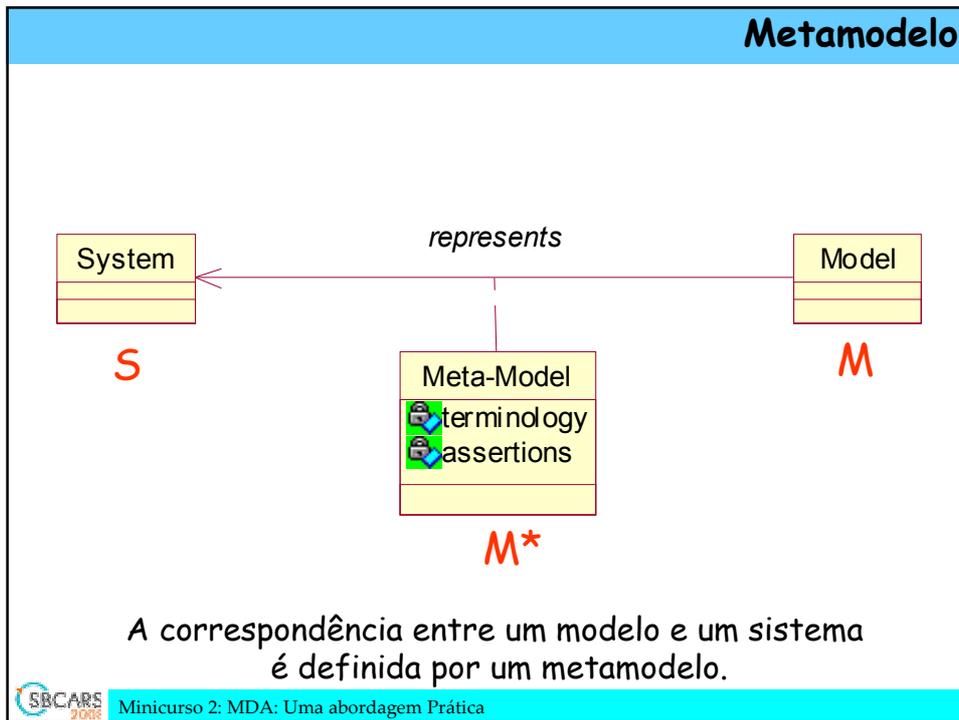
  
UFRRN



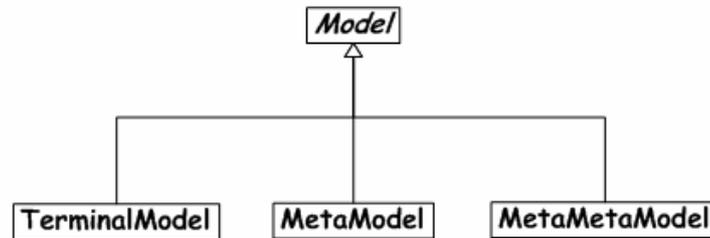
---

# Metamodelagem

Porto Alegre, Agosto de 2008



## Modelos: Conceitos (2)



**Existem diferentes tipos de modelos**



## Modelos: Definições

- **Definição 1.** Um meta-metamodelo é um modelo que é a sua própria referência (i.e. corresponde a ele mesmo).
- **Definição 2.** Um metamodelo é um modelo cuja referência é um meta-metamodelo.
- **Definição 3.** Um modelo terminal é um modelo cuja referência é um metamodelo.



## MOF: Meta-metamodelo MDA

- Meta-Object Facility (MOF) é a tecnologia adotada pela OMG para definição de metadados
- MOF 1.3 finalizada em Setembro de 1999 (OMG ad/99-09-05).
- MOF 1.4 finalizada (aprovada) em Dezembro de 2004. (ISO/IEC DIS 19502).
- O metamodelo MOF define a sintaxe abstrata de um metadado na representação MOF de um modelo
- O modelo MOF é autodescrito



## Meta Object Facility (MOF)

- Alinhamento com MDA
  - MOF 1.4 → MOF 2.0
- **Metadata management framework + metadata services**
- Permite o desenvolvimento e interoperabilidade de sistemas dirigidos a modelo e metadados
- MOF é a base das tecnologias associadas a **MDA**
  - UML, CWM, SPEM, XMI, UML profiles, JMI ...

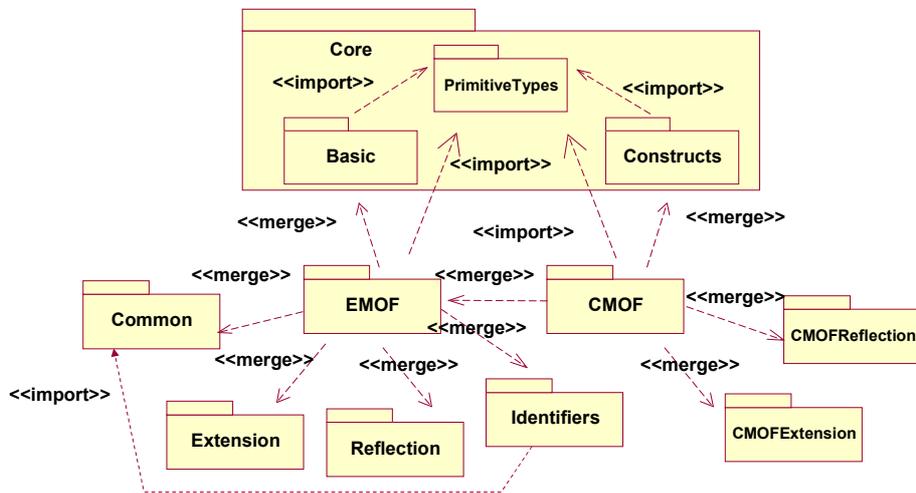


### ▪ Estrutura Modular

- **Modelo MOF 2.0**
  - *framework para metamodelagem, gerência e representação de metadados*
- **MOF 2.0 IDL + MOF 2.0 Java**
  - *mapeamentos MOF 2.0 → IDL and Java*
- **MOF 2.0 Query/View/Transformation**
  - *Framework para definir transformações entre modelos MOF*

- **MOF 2.0 integra e reúsa a infraestrutura UML 2.0 Core**
- **A infraestrutura UML 2.0 fornece uma notação para representar metamodelos MOF**
  - *Estrutura → diagrama de classes*
- **O próprio modelo MOF 2.0 é modular**

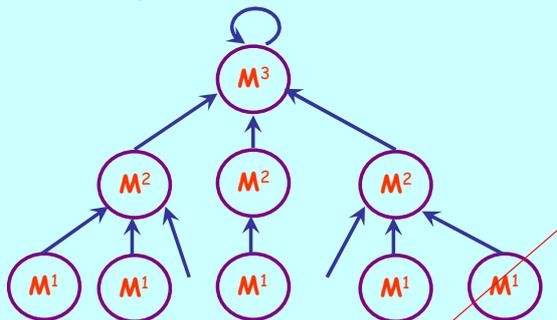
## Estrutura do MOF 2.0



Minicurso 2: MDA: Uma abordagem Prática

## Modelos, Metamodelos e Metametamodelos

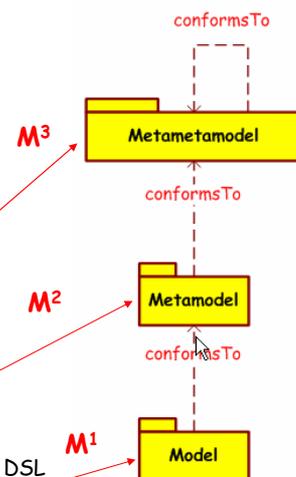
### Espaços $M^1$ , $M^2$ & $M^3$



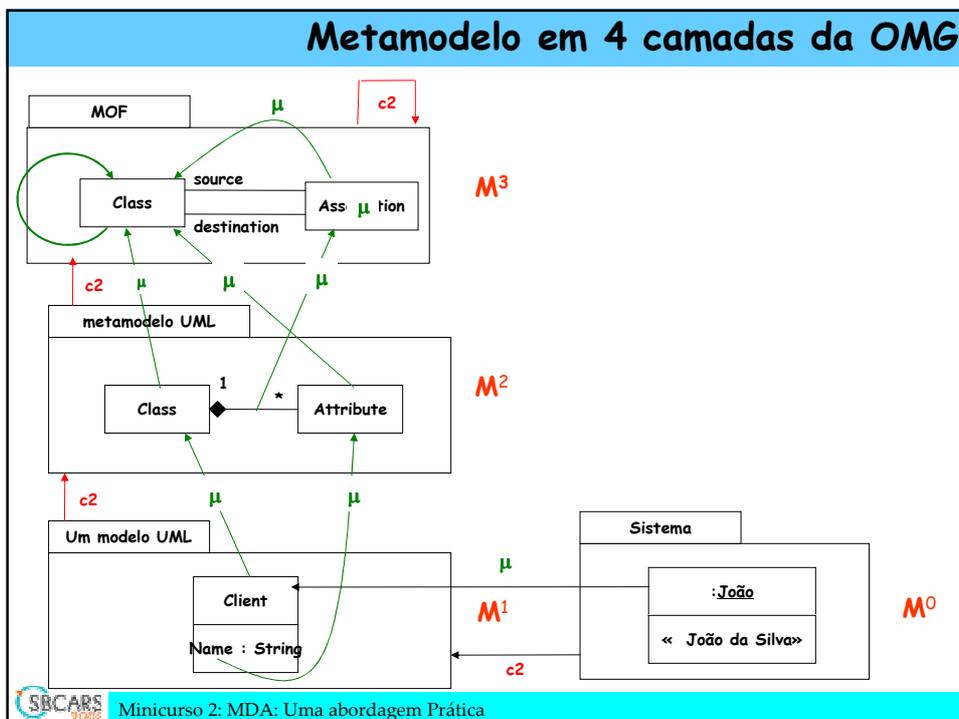
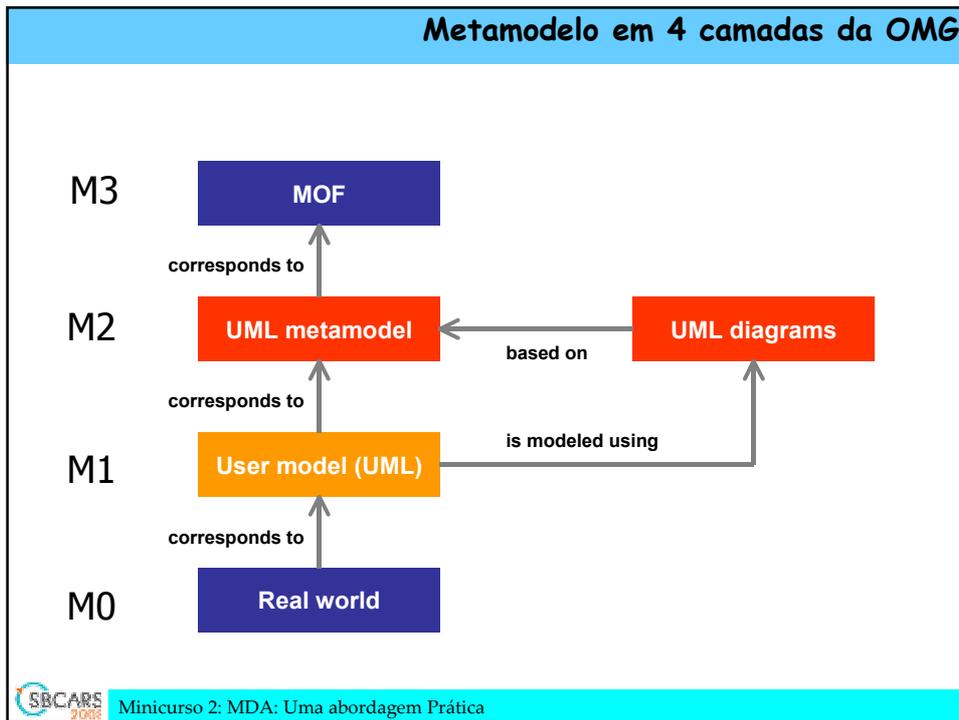
Université de NANTES

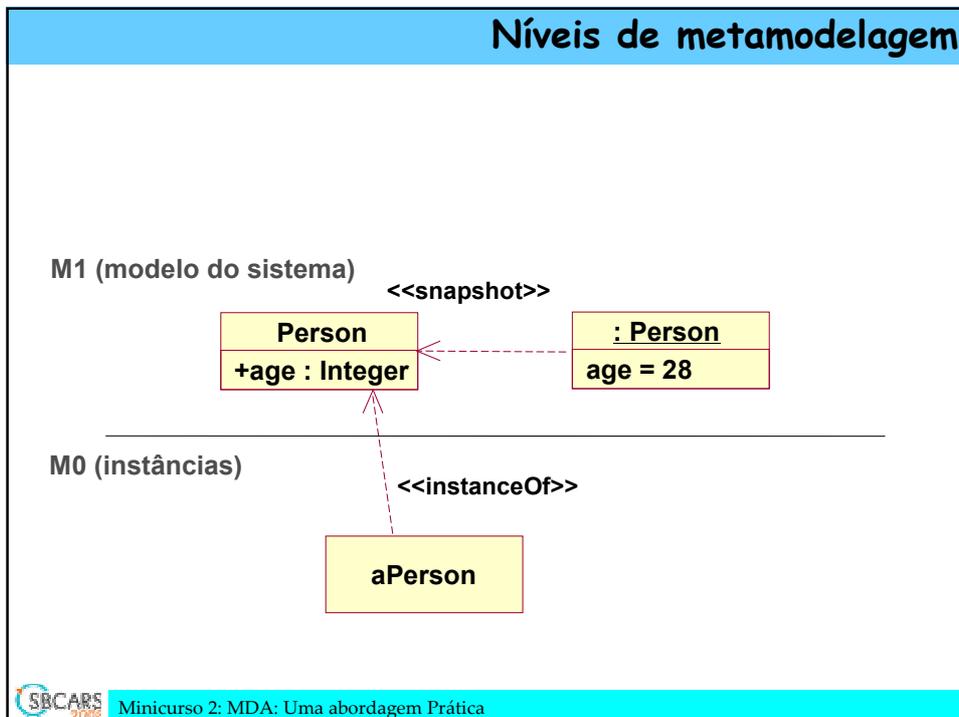
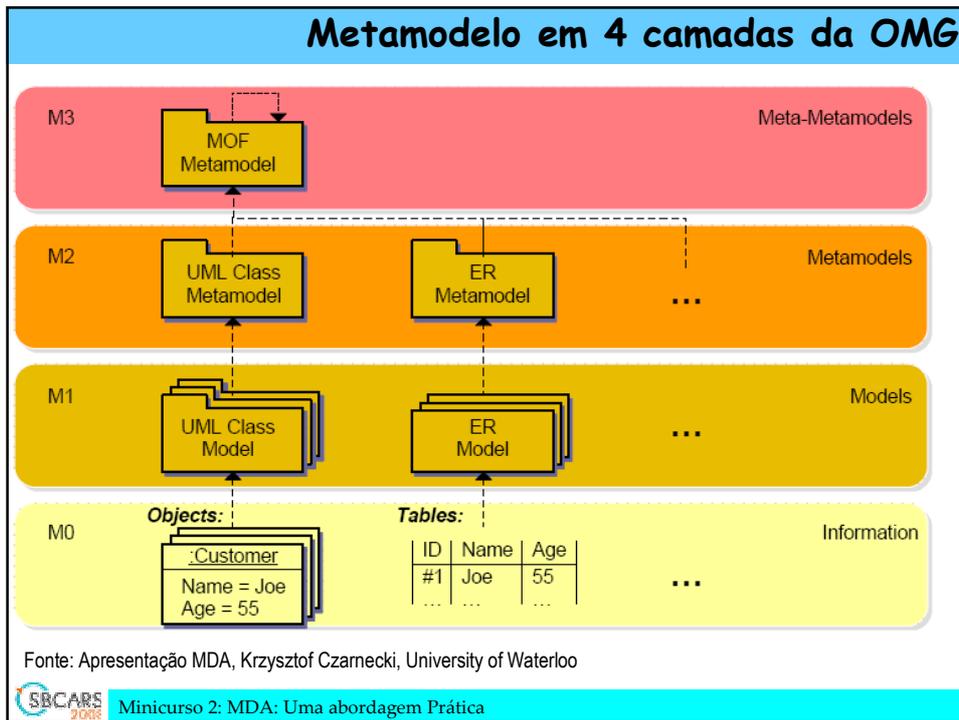
GRUPE SODIFRANCE

- Um único Meta-metamodelo (MOF)
- Metamodelos compatíveis com MOF → cada um definindo uma DSL
- Cada modelo é definido na linguagem única do seu metamodelo



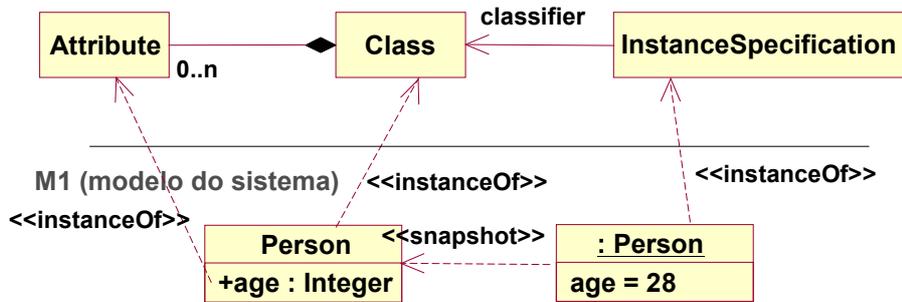
Minicurso 2: MDA: Uma abordagem Prática





## Níveis de metamodelagem

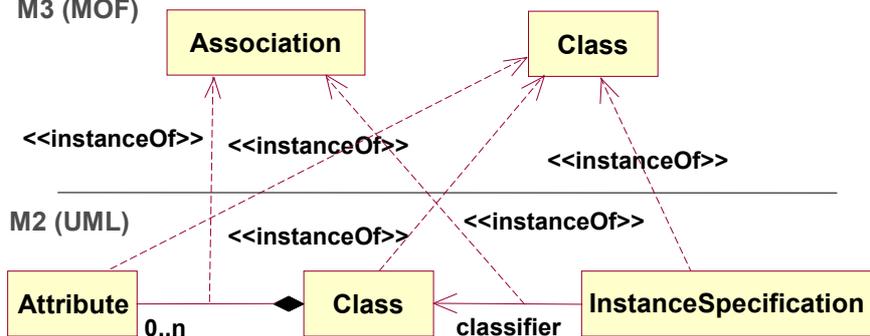
M2 (UML)



Minicurso 2: MDA: Uma abordagem Prática

## Níveis de metamodelagem

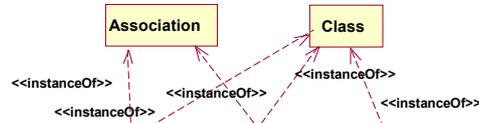
M3 (MOF)



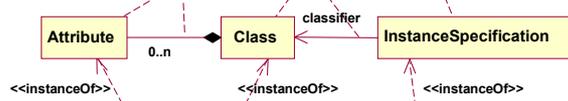
Minicurso 2: MDA: Uma abordagem Prática

## Níveis de metamodelagem

M3 (MOF)



M2 (UML)



M1 (Modelo do sistema)



M0 (Instâncias em tempo de execução)



Minicurso 2: MDA: Uma abordagem Prática

## Extensões de metamodelos

- Adaptação para terminologia de um domínio
- Adição de novas representações para os conceitos da linguagem
- Discernir entre especializações de conceitos
- Adicionar restrições ao uso da linguagem
- Adicionar informações para auxiliar nas transformações

**UML é uma linguagem genérica !**



**Criação de Linguagens específicas de domínio (DSLs)**



Minicurso 2: MDA: Uma abordagem Prática

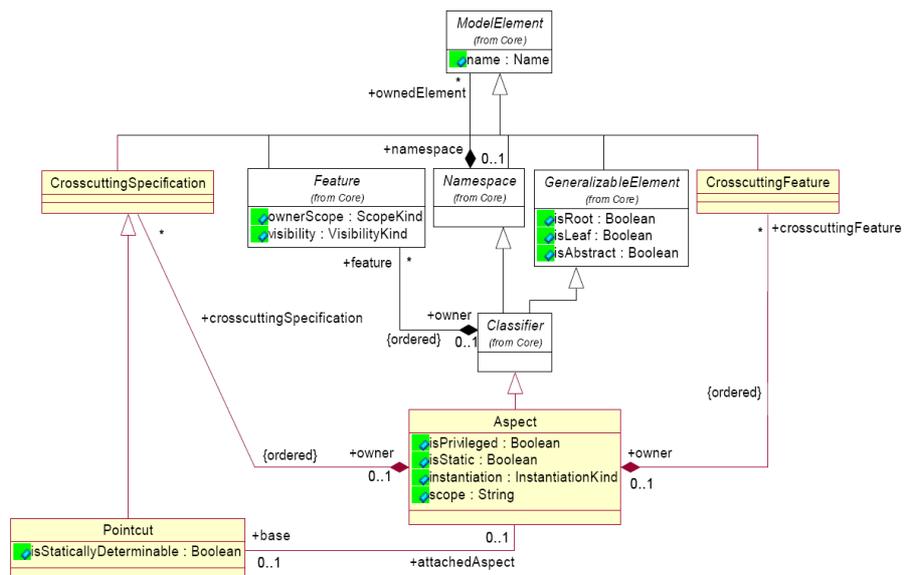
## Mecanismos de Extensão

- **Definição de uma nova linguagem M2**
  - Por exemplo: Linguagens não OO
- **Extensão da UML (mais usual e simples)**
  - Vantagem: Uso das ferramentas UML de visualização/edição
- **Mecanismos de Extensão**
  - Adição de elementos e restrições ao metamodelo UML
  - Uso de estereótipos (UML 1.x)
  - Uso de UML profiles (UML 2)



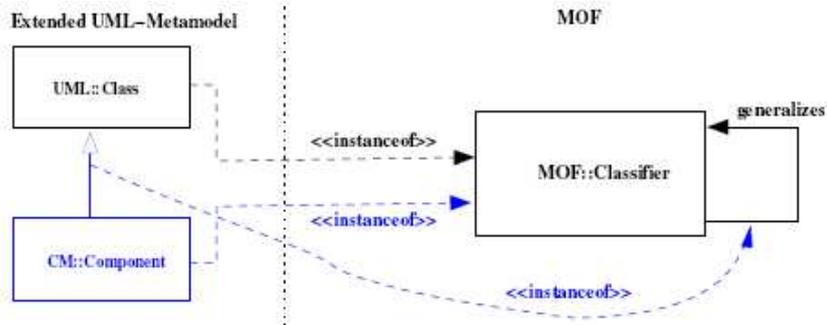
Minicurso 2: MDA: Uma abordagem Prática

## Exemplo Extensão do MOF: Aspectos

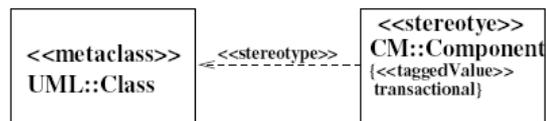


Minicurso 2: MDA: Uma abordagem Prática

## Exemplo extensão do metamodelo UML



## Exemplo extensão via estereótipo (UML 1.x)



- Mecanismo simples de extensão
- Tagged values não são tipados
- Não é possível definir novas metaassociações

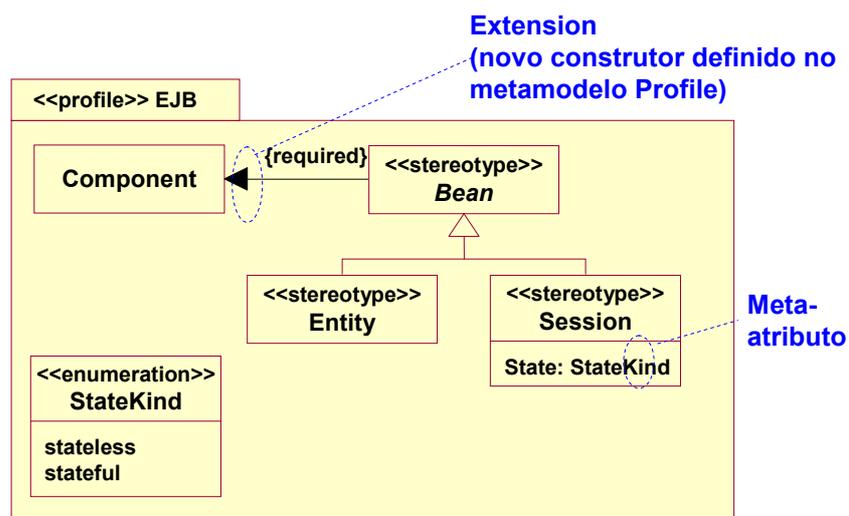
## Exemplo extensão via perfil (UML 2)

- **Dado um metamodelo de referência**
  - Um perfil estende o modelo através da definição de estereótipos das metaclasses do modelo
- **Um perfil é um caso especial de pacote (UML package)**
  - Podem ser adicionados dinamicamente a um modelo
- **Perfis tornam a UML uma família de linguagens**
  - Cada membro é definido através da aplicação de um ou mais perfis ao metamodelo UML base
- **Elementos:**
  - Estereótipos (stereotypes)
  - Valores Etiquetados (tagged values)
  - Restrições (constraints)
- **Perfis → restrições adicionais ao metamodelo**
- **São formalmente definidos pelo metamodelo UML**



Minicurso 2: MDA: Uma abordagem Prática

## Exemplo de Perfil: EJB



Minicurso 2: MDA: Uma abordagem Prática

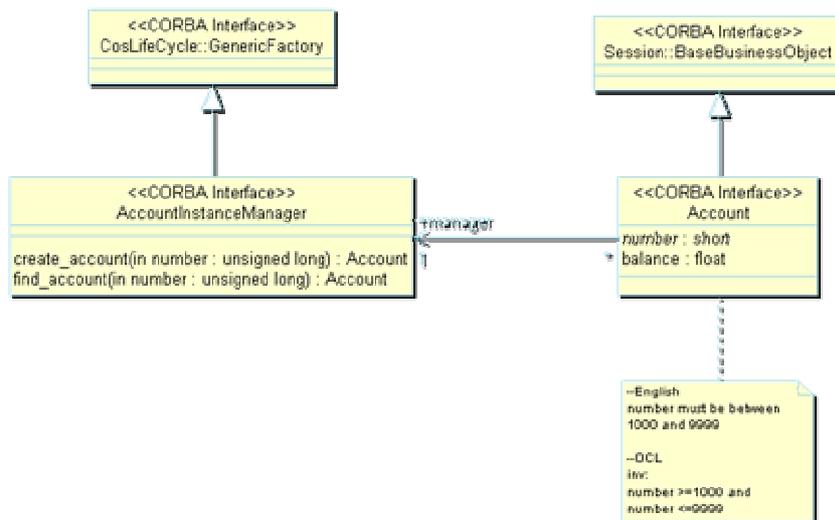
## Perfis: Exemplo na marcação de um PIM

```
--English
number must be between
1000 and 9999

--OCL
inv:
number >= 1000
and
number <= 9999
```

```
<<business entity>>
Account
<<UniqueId>> number : Integer
balance : Float
```

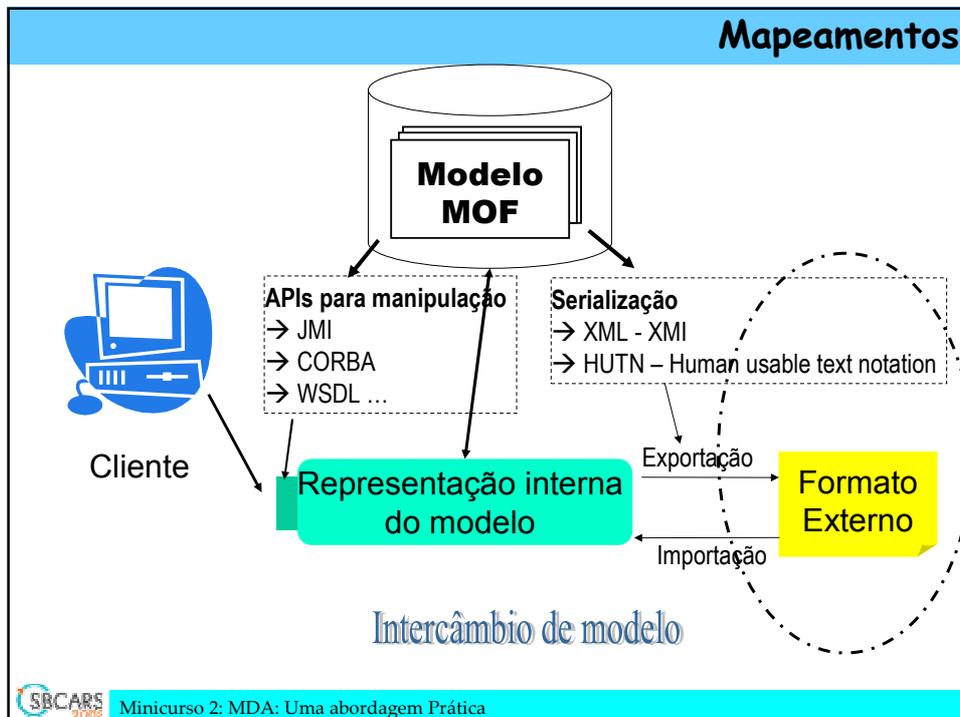
## Perfis: Exemplo na marcação de um PSM



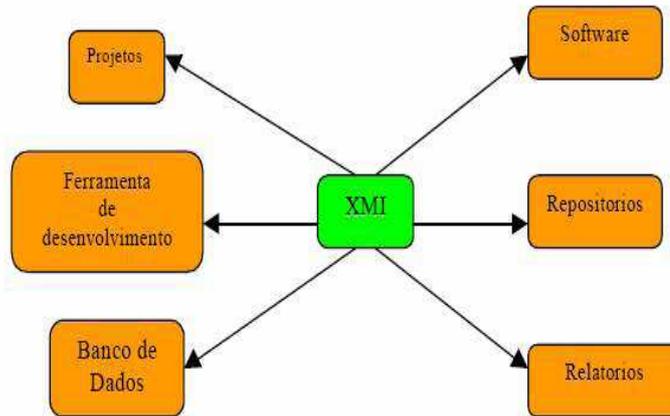
# Mapeamento de/para o MOF

**Modelo de Serialização**  
**APIs para manipulação**

Porto Alegre, Agosto de 2008

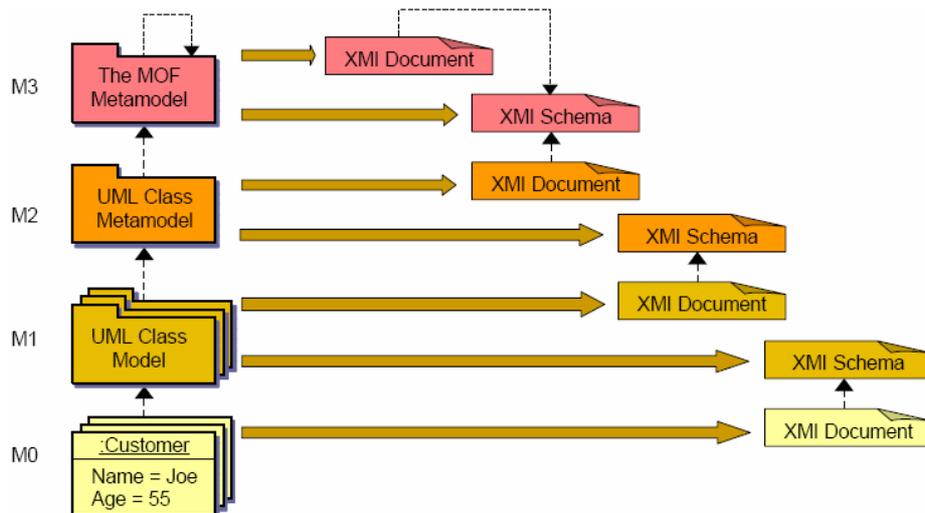


## XMI para intercâmbio de dados e Metadados



Minicurso 2: MDA: Uma abordagem Prática

## Serialização: Camadas Metamodelagem



Fonte: Apresentação MDA, Krzysztof Czarnecki, University of Waterloo



Minicurso 2: MDA: Uma abordagem Prática

# Transformação entre Modelos

**Conceitos**  
**QVT & ATL**  
**MOFScript**

Porto Alegre, Agosto de 2008

Uso

- Refinamento de modelos
- Refatoramento de modelos
- Engenharia reversa
- Geração de visões
- Geração de código
- Migração de tecnologias

## Transformação entre Modelos

- **Regras de transformação**
  - Descrição de como um ou mais elementos do modelo fonte pode ser transformado em um ou mais elementos do modelo destino
- **Definição de transformação**
  - Conjunto de regras que juntas descrevem como um modelo fonte é transformado em um modelo destino
- **Linguagem de transformação**
  - Linguagem na qual as definições de transformações são escritas



## TRANSFORMAÇÃO ENTRE MODELOS

- **Técnicas de transformação de modelos**
  - **Baseada em programação**
    - *Transformações de modelos são programas OO*
    - *Interfaces de manipulação (reflexiva e sob encomenda)*
  - **Baseada em templates**
    - *Transformações de modelos são templates*
    - *Escritos em uma linguagem dedicada*
    - *Parâmetros determinam o modelo destino*
  - **Baseada em modelos**
    - *Transformações são modelos de transformações*



## LINGUAGENS DE TRANSFORMAÇÃO

- Baseada em programação

- Java
- Flora

- Baseada em templates

- XSLT

- Baseada em modelos

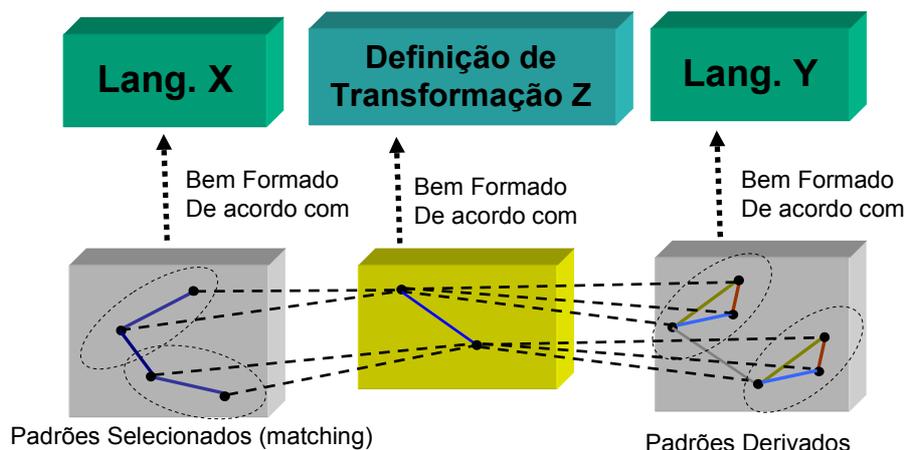
- ATL
- QVT

MDA



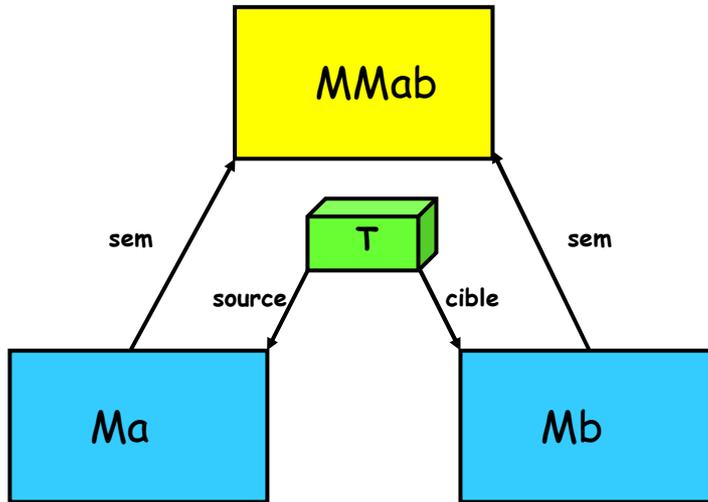
Minicurso 2: MDA: Uma abordagem Prática

## Transformações MDA

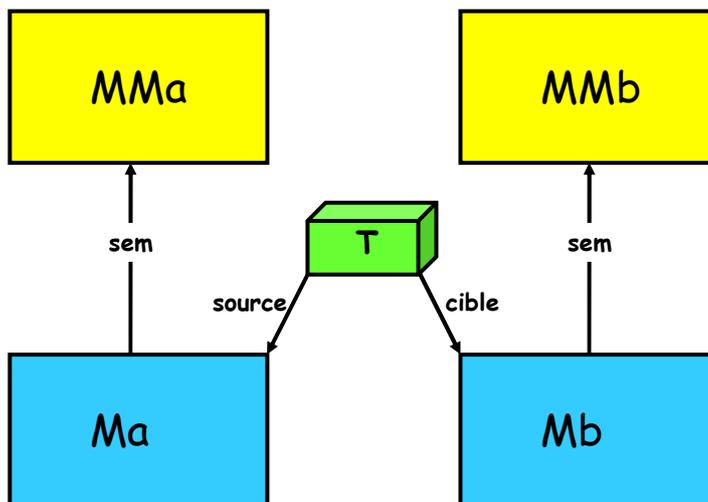


Minicurso 2: MDA: Uma abordagem Prática

## Transformações Endógenas



## Transformações Exógenas



# Linguagens Transformação MDA

Arquitetura QVT  
(Query/View/Transformation)

ATL

MOFScript

Porto Alegre, Agosto de 2008

## Introdução

- QVT = Query/Views/Transformations
- Linguagem padrão da OMG para expressar consultas, visões e transformações sobre modelos MOF
- OMG QVT Request for Proposals (QVT RFP, ad/02-04-10) enviada em 2002
- Status corrente (Junho, 2006): especificação final, OMG document ptc/05-11-01

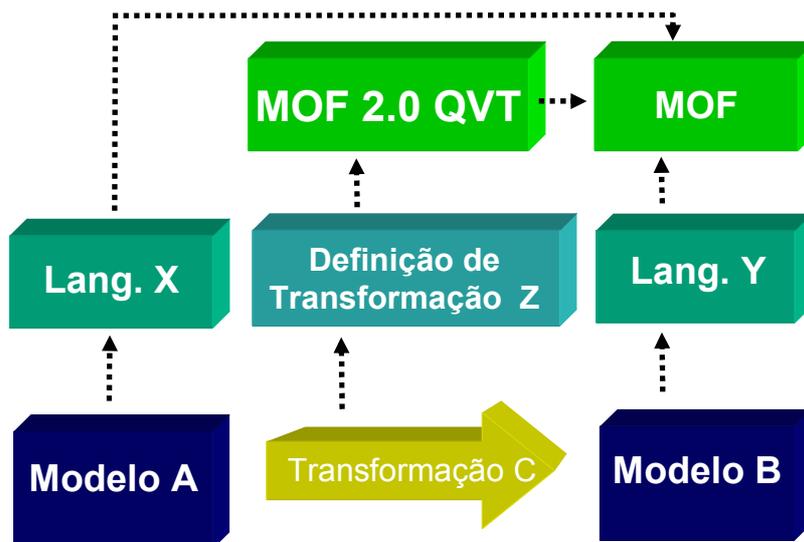
## Abordagem

- **Query:**
  - seleção de elementos específicos de um modelo já existente
- **View:**
  - projeção de um metamodelo a partir de consultas a um modelo já existente
- **Transformation:**
  - recebe um modelo como entrada e atualiza ou gera um novo modelo
- **QVT depende das especificações MOF 2.0 e OCL 2.0**



Minicurso 2: MDA: Uma abordagem Prática

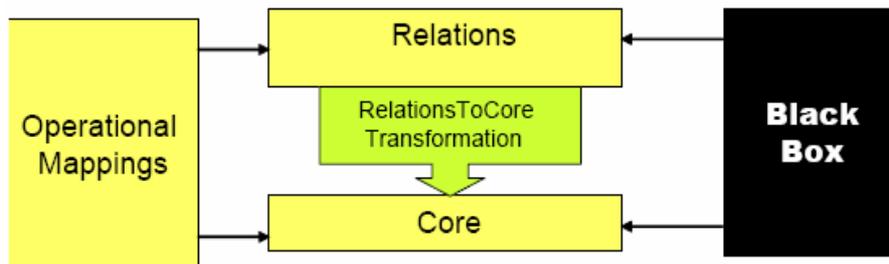
## QVT e MOF



Minicurso 2: MDA: Uma abordagem Prática

## Arquitetura do QVT

- O QVT possui uma arquitetura híbrida formada por uma parte declarativa (em 2 níveis) e outra imperativa



## Parte Declarativa

- A parte declarativa pode ser utilizada para fazer todo o processo de transformação
- A linguagem associada a essa parte descreve os relacionamentos entre as variáveis
- Para a execução da linguagem *Relations* é necessário o uso de um compilador ou interpretador
- Essa camada pode conter informações suficientes para transformações unidirecionais ou bidirecionais

## Parte Declarativa

- **Relations:**
  - Uma especificação declarativa dos relacionamentos entre os modelos MOF
  
- **Core:**
  - Linguagem mais simples que servem como referência semântica para as relations
  
- **A linguagem *Relations* e a linguagem *Core* possuem a mesma semântica, porém em níveis diferentes de abstração**



Minicurso 2: MDA: Uma abordagem Prática

## Parte Imperativa

- **Operational mappings:**
  - permite definir transformações usando uma abordagem imperativa e completa, como também que relações sejam implementadas por uma abordagem imperativa
  
- **Black Box:** permite que outras linguagens de transformação ou bibliotecas sejam agregadas ao QVT
  - XLST (*Extensible Stylesheet Language Transformations*)
  - XQuery
  - QVT <camada>\*, onde camada = *relations, core, operational mappings*



Minicurso 2: MDA: Uma abordagem Prática

# ATL

Porto Alegre, Agosto de 2008

## LINGUAGEM ATL

### ▪ **ATL (*Atlas Transformation Language*)**

- Linguagem baseada no MOF que possui uma sintaxe concreta para transformações modelo-modelo
- No domínio MDE, fornece aos programadores meios de produzir um número de modelos de origem para um conjunto de modelos de destino
- Ferramentas → plataforma Eclipse
- IDE ATL prevê uma série de ferramentas padrão para desenvolvimento que visam facilitar a concepção das transformações ATL

**▪ Visão Geral**

- Um programa ATL é composto por regras
- Fazem a correspondência entre os elementos do modelo de origem e os elemento do modelo de destino
- Linguagem de transformação de modelos híbrida: declarativa e imperativa.
- Transformação é escrita preferencialmente de forma declarativa e permite mapeamentos entre os elementos do modelo de origem para os elementos do modelo de destino.



# MOFScript

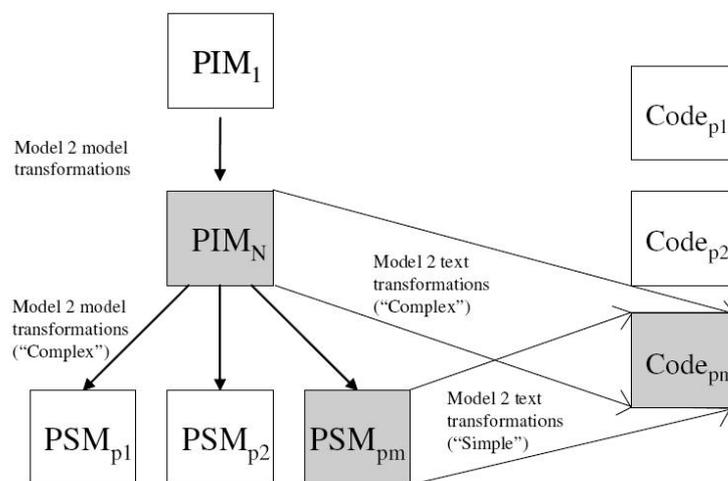
## MOFSCRIPT

- Linguagem de transformação modelo-texto
- Baseada no MOF
- É uma extensão do QVT
- Faz parte do processo de padronização da OMG
- Composta por Regras
- Operações chamadas de forma explícitas
  - Linguagem procedural
- Gera um arquivo texto a partir de um modelo de entrada
  - O modelo de entrada deve ser baseado no MOF



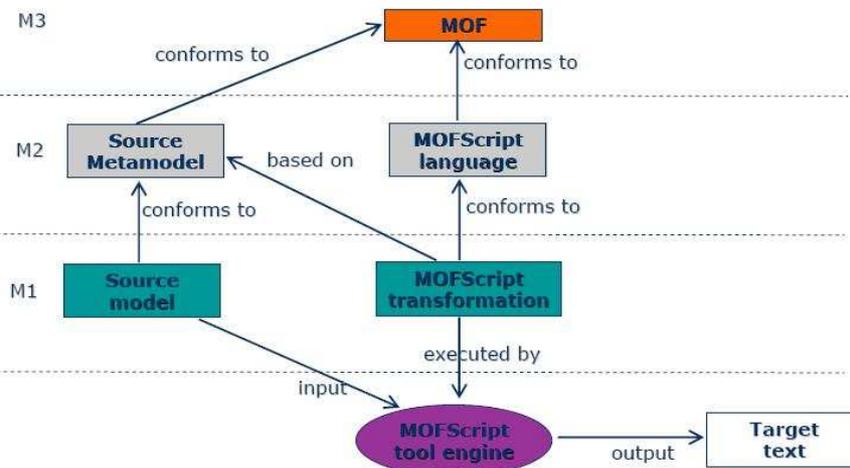
Minicurso 2: MDA: Uma abordagem Prática

## Transformação Modelo Texto



Minicurso 2: MDA: Uma abordagem Prática

## MOFSCRIPT



Minicurso 2: MDA: Uma abordagem Prática



# Processo de desenvolvimento baseado em MDA

Processo Baseado em MDA  
RUP e Metodologias Ágeis

Porto Alegre, Agosto de 2008

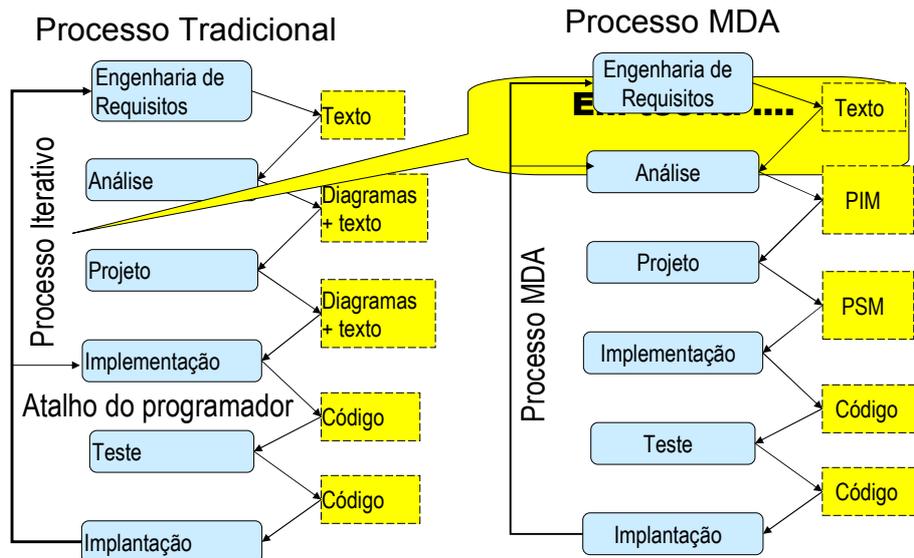
## Processo Baseado em MDA

- **MDA pode ser aplicado a qualquer processo**
  - Indicando quais modelos serão usados, como serão coordenados, e como serão produzidos
- **Mudanças:**
  - Análise, projeto e codificação
  - Modelos e transformações são artefatos relevantes



Minicurso 2: MDA: Uma abordagem Prática

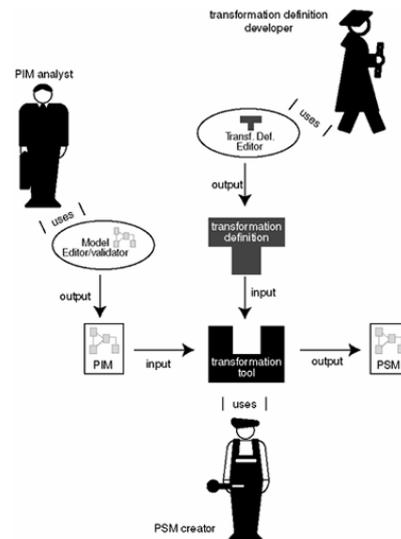
## Impacto da MDA no processo de desenvolvimento



Minicurso 2: MDA: Uma abordagem Prática

## Novos papéis

- **Analista PIM**
  - Necessidades do negócio
  - Modelo de negócio
- **Construtor PSM**
  - Detalhes da plataformas
  - Arquiteturas
- **Desenvolvedor de definição de transformações**
  - Escrita e compra



Minicurso 2: MDA: Uma abordagem Prática

## RUP e Metodologias Ágeis

- **RUP**
  - Já centrado na utilização de modelos - UML
- **Metodologias Ágeis**
  - “Codificação” no modelo
  - Código  $\Rightarrow$  alto nível de abstração
    - “Extreme Modeling”
  - Testes orientados a Modelo
- **Pode facilitar a comunicação com o cliente**
  - Linguagens de modelagem de domínio específico
- **Prototipação**
  - Simulações/ geração de código
- **DSL**
  - Facilitador na comunicação com o usuário



Minicurso 2: MDA: Uma abordagem Prática

# Ferramentas MDA

Características Desejáveis  
AndroMDA 4  
OpenArchitectureWare

Porto Alegre, Agosto de 2008

## Ferramentas

- Adaptive's Framework <http://www.adaptive.com/>
- France-Telecom Universalis <http://universalis.elibel.tm.fr/>
- Codagen Gen-it <http://www.codagen.com/>
- CodigoCodigoXpress <http://www.codigoxpress.com/>
- DSTC dMOF <http://www.dstc.edu.au/Products/CORBA/MOF/>
- Interactive Objects ArcStyler <http://www.io-software.com/>
- Kabira Business Accelerator <http://www.kabira.com/>
- Kennedy Carter iUML and iCCG <http://www.kc.com/>
- Metamatrix MetaBase <http://metamatrix.com/>
- NetBeans Meta Data Repository MDR <http://www.netbeans.org/>
- ONTOS ObjectSpark <http://www.objectspark.com/>
- ObjectRad Java Metadata Server <http://www.objectrad.com/>
- Objexion Software Netsilon <http://www.netsilon.com/>
- Project Technology BridgePoint/DesignPoint <http://www.projtech.com/>
- Secant Technologies ModelMethods <http://www.modelmethods.com/>
- Soft-Maint Scriptor & Semantor <http://www.sodifrance.fr/>
- Tata Research Development ADEX <http://www.tcs.com/>
- University of Berne MOOSE <http://www.iam.unibe.ch/>

Várias outras ...

## Características Desejáveis

### ▪ Modelagem e Metamodelagem

- Suporte UML
- Suporte Perfil UML
- Suporte para verificação com base em regras OCL
- Suporte para criação e edição de metamodelos MOF
- Controle da sintaxe concreta e comportamento de edição
  - Criação de DSLs
- Exportação/Importação XMI



Minicurso 2: MDA: Uma abordagem Prática

## Características Desejáveis

### ▪ Transformação

- M2M
- M2T
- Parametrização e customização de transformações
- Transformações definidas pelo desenvolvedor
- Transformações bi-direcionais
  - Engenharia reversa → C2M
- Transformação tanto automática quanto interativa
- Gerenciamento de *traces* e registro de transformações
- Suporte a codificação, teste e documentação do código gerado
- Sincronização entre M2M e M2C



Minicurso 2: MDA: Uma abordagem Prática

## Características Desejáveis

### ▪ Outras:

- Suporte a plataformas específicas
- DSLs, padrões e componentes suportados
- Interoperabilidade com outras ferramentas
- Suportem *Model Driven Testing*
- Suportem controle de versão e trabalho concorrente e distribuído



Minicurso 2: MDA: Uma abordagem Prática



# AndroMDA 4

Porto Alegre, Agosto de 2008

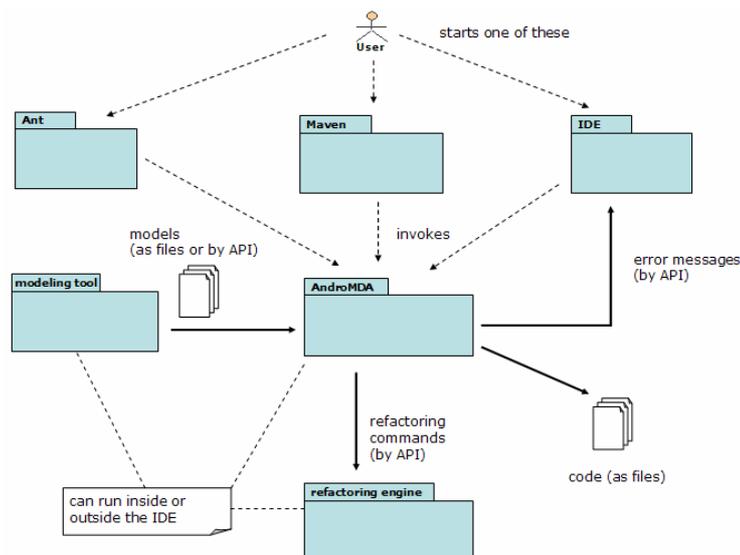
## AndroMDA 4

- **Open-source:** <http://www.andromda.org>
- **Alinhado as tecnologias MDA:**
  - Baseado no MOF
- **Supporta transformações**
  - M2M - ATL
  - M2T - MOFScript
- **Componentes:**
  - Cartuchos
    - *transformações ATL + transformações MOFScript + metamodelos*
  - Repositório de metamodelos (eclipse - ecore)
  - Workflows de transformações



Minicurso 2: MDA: Uma abordagem Prática

## AndroMDA



Minicurso 2: MDA: Uma abordagem Prática

Demonstração

AndroMDA 4



Minicurso 2: MDA: Uma abordagem Prática



# Model Driven Development (MDD) usando OpenArchitectureWare

Porto Alegre, Agosto de 2008

## Características

- Framework para suporte a MDD implementado em Java
- Suporta análise de modelos
- Dispõe de uma linguagem familiar para checagem e transformação de modelos
- Dispõe de um gerador de código a partir de um modelo
- Editor integrado ao Eclipse
- Transformações
  - M2M
  - M2T
  - T2M



Minicurso 2: MDA: Uma abordagem Prática

## Informações para instalação

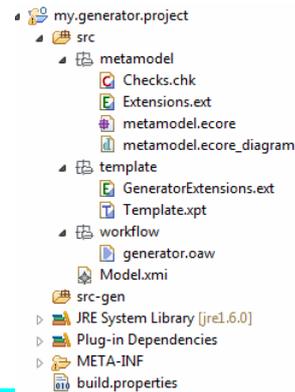
- Última versão liberada em 5 de maio de 2008 e está atualmente na versão 4.3  
Disponíveis em:
  - Para download: <http://www.eclipse.org/gmt/oaw/download/>
  - Para instalação via update do eclipse:  
<http://www.openarchitectureware.org/updatesite/milestone/site.xml>
- Dependências
  - Java 5/6
  - Eclipse 3.3 ou superior
  - EMF
  - UML2
  - GMF 2 (Ganymede/Europa releases)



Minicurso 2: MDA: Uma abordagem Prática

## Tipos de arquivos

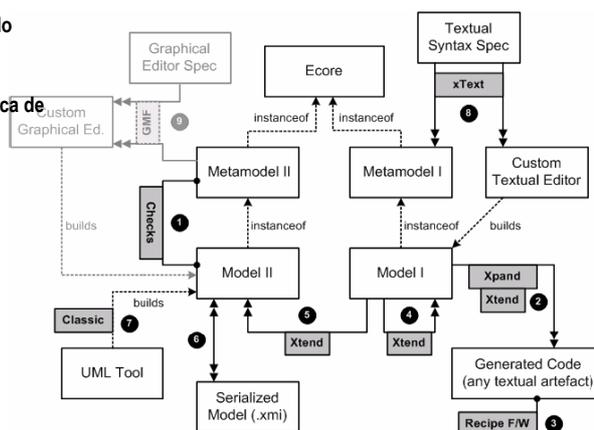
- Workflow (.oaw)
- Xpand para arquivos templates (.xpt)
- Xtend (.etx)
- Check (.chk)
- Recipes (.recipes)
- Xtext (txtxt)



Minicurso 2: MDA: Uma abordagem Prática

## Arquitetura do oAW

1. Verificação de modelo usando restrições
2. Geração de código
3. Integrando geração automática de código com escrito a mão
4. Modificação de modelo
5. Transformação Modelo para Modelo
6. Carga e armazenamento de modelos
7. Edição de modelos usando ferramentas UML
8. Edição de modelos usando editores textuais
9. Edição de modelos usando Editores GMF



Minicurso 2: MDA: Uma abordagem Prática

## Demonstração

Éberton Marinho (UFRN)  
ebertonsm@gmail.com



Minicurso 2: MDA: Uma abordagem Prática



Concluindo ...

Porto Alegre, Agosto de 2008

## Benefícios MDA

- **Preserva o investimento no “conhecimento”**
  - Conhecimento tácito é naturalmente explicitado
- **Velocidade de desenvolvimento**
  - Complexidade gerenciada pelos diferentes níveis de abstração
  - Automatização da geração de código
- **Qualidade do produto final**
- **Manutenção e documentação**
  - 100% rastreabilidade
  - Documentação não é “abandonada”



Minicurso 2: MDA: Uma abordagem Prática

## Situação Atual

- **Início da evolução ...**
  - Padrões em amadurecimento
  - Outros em desenvolvimento
- **Inexistência de ambientes completos**
  - Ferramentas implementam somente parte da abordagem
  - Suporte fraco para metamodelagem
- **Poucas implementações disponíveis de DSLs e mapeamentos para plataformas**
- **“Cultura” de modelagem não está bem estabelecida em grande parte das empresas**



Minicurso 2: MDA: Uma abordagem Prática

## Questões a serem avaliadas

- Qual nível de especialização é necessário?
- Os fornecedores seguirão os padrões?
- As linguagens de modelagens são expressivas o suficiente?
- Como realizar Gerência de configuração e Testes?
- .....



Minicurso 2: MDA: Uma abordagem Prática

## Quem são os usuários MDA?

<b>Construção e Extensão de meta-modelos</b>	 <b>Meta-modeladores</b>	<b>100</b>
<b>Construção de um repositório de transformações padrão</b>	 <b>Arquitetos</b>  <b>Especialistas em Plataforma</b>  <b>Eng de Qualidade</b>  <b>Especialistas em metodologias</b>	<b>50 000</b>
<b>Montagem, composição e customização do repositório</b>	 <b>Gerentes de Projeto</b>  <b>Eng. de qualidade</b>	<b>50 000</b>
<b>Uso do repositório</b>	 <b>Desenvolvedores</b>	<b>1 000 000</b>



Minicurso 2: MDA: Uma abordagem Prática

## Marcas da OMG

- Registradas:
  - MDA®
  - model Driven Architecture®
  - UML®
  - CORBA®
  - CORBA Academy®
  - XMI®
- Marcas:
  - CWM™
  - model Based Application Development™
  - MDD™
  - model Based Development™
  - model Based Management™
  - model Based Programming™
  - model Driven Application Development™
  - model Driven Development™
  - model Driven Programming™
  - model Driven Systems™
  - OMG Interface Definition Language (IDL)™
  - Unified Modeling Language™
  - <<UML>>™

[http://www.omg.org/legal/tm\\_list.htm](http://www.omg.org/legal/tm_list.htm)



Minicurso 2: MDA: Uma abordagem Prática

## Acrônimos

- **MDE** - model Driven Engineering
- **ME** - Model Engineering
- **MDA** - Model Driven Architecture
- **MDD** - Model Driven Development
- **MDSD** - Model Driven Software Development
- **MDSE** - Model Driven Software Engineering
- **MM** - Model Management
- **MDDE** - Model Driven Data Engineering
- **ADM** - Architecture Driven Modernization
- **MDRE** - Model Driven Reverse Engineering
- **DSL** - Domain Specific Language
- **DSM** - Domain Specific Modeling
- **MDE** termo genérico; ME ~ MDE
- **MDA™** and **MDD™** são marcas da **OMG**
  - MDD reservada para uso futuro da **OMG**
- **MDSD** e **MDSE** vêm sendo utilizadas qdo não se deseja restringir o tema as tecnologias, vocabulário e visão da **OMG**.
- **ADM** é um outro padrão que visa ser o oposto da **MDA**:
  - MDA → forward engineering
  - ADM → backward engineering.
    - ADM ~ MDRE
- **MM** ~ **MDDE**
- **DSM** tem sido utilizada como uma “marca” da **Microsoft** mas vem sendo usada também para descrever soluções para desenvolvimento de sistemas da comunidade acadêmica



Minicurso 2: MDA: Uma abordagem Prática

## Fontes de Consulta

- **Esta apresentação se baseia, principalmente, no material descrito a seguir:**
  - Introduction to model Engineering, Jean Bézivin, ATLAS Group (INRIA & LINA), Nantes
    - <http://www.sciences.univ-nantes.fr/lina/atl/>
  - Tutorial Model Driven Development, OOPSLA'03, Krzysztof Czarnecki & Peter Graff



## Referências

- **Especificações**
  - modelo Driven Architecture specification, OMG Architecture board, July 2001 ([www.omg.org/mda](http://www.omg.org/mda))
- **Livros**
  - MDA Explained: The modelo Driven Architecture™: Practice and Promise. Anneke Kleppe, Jos Warner, Wim Bast
  - D. Frankel, modelo Driven Architecture™. Applying MDA™ to Enterprise Computing. Indianapolis, IN: Wiley, 2003.
  - S. Mellor et al, MDA Distilled. Principles of Model-Driven Architecture. Boston, MA: Addison-Wesley, 2004.
  - A. Kleppe et al, MDA Explained. The modelo Driven Architecture™: Practice and Promise. Boston, MA: Addison-Wesley, 2003.
  - J. Miller et al, MDA Guide Version 1.0. Needham, MA: OMG, 2003.
  - R. Hubert, Convergent Architecture: Building Model-Driven J2EE Systems with UML. New York, NY: 2002.
- **Conferência:**
  - MoDELS - Conference on Model Driven Engineering Languages and Systems
    - <http://www.modelsconference.org/>
- **Artigos**
  - Examining the modelo Driven Architecture (MDA), Scott W. Ambler.
  - Model-Driven Architecture Tutorial, Krzysztof Czarnecki and Petter Graff. OOPSLA 2003
  - modelo Driven Engenineering, Stuart Kent
  - Model-Based Testing in Practice, S. R. Dalal et al.
  - Towards an MDA-Oriented Methodology, Marie-Pierre Gervais



## Referências

### Artigos

- Colin Atkinson and Thomas Kuhn. Model-Driven Development: A Metamodeling Foundation. IEEE Software, 20(5):46–51, September/October 2003.
- Mellor, S.J. and Clark, A.N. and Futagami, T. Model-driven development – Guest editor's introduction. IEEE Software, 20(5):14– 18, Sept.-Oct. 2003.
- Lutz Bichler. A flexible code generator for MOF-based modeling languages. In 2nd OOPSLA Workshop on Generative Techniques in the context of Model-Driven Architecture, 2003.
- Bran Selic. The Pragmatics of Model-Driven Development. IEEE Software, 20(5):46–51, September/October 2003.
- Aditya Agrawal and Tihamer Levendovszky and Jon Sprinkle and Feng Shi and Gabor Karsai. Generative Programming via Graph Transformations in the Model-Driven Architecture. In Workshop on Generative Techniques in the Context of Model-Driven Architecture, 2002.
- Krzysztof Czamecki and Simon Helsen. Classification of Model transformation approaches. In 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model-Driven Architecture, 2003.
- Daniel Varró and András Pataricza. UML Action Semantics for Model Transformation Systems. Periodica Politecnica, 2003.
- T. Gardner and C. Grin and J. Koehler and R. Hauser. A review of OMG MOF 2.0 Query / Views / Transformations Submissions and Recommendations towards the final Standard. In MetaModelling for MDA Workshop, 2003.
- Torben Weis and Andreas Ulbrich and Kurt Geihs. Model Metamorphosis. IEEE Software, 20(5):46–51, September/October 2003.



Minicurso 2: MDA: Uma abordagem Prática

UFRN  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

DIMAP

VENHA FAZER O SEU MESTRADO OU DOUTORADO NA UFRN !  
<http://www.dimap.ufrn.br>  
<http://www.dimap.ufrn.br/~paulo.pires>

Minicurso 2: MDA: Uma abordagem Prática