

Escola Regional de Informática

SBC Mato Grosso

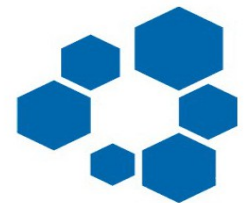
De 14 a 17 de Outubro de 2014

Desenvolvimento Dirigido por Modelos:
Conceitos, Aplicações, e Perspectivas

Prof. Valdemar Neto
INF-UFG



INSTITUTO DE
INFORMÁTICA
UFG



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS



Agenda

- Introdução
- Conceitos
- Ferramentas
- Aplicações
- Perspectivas



Engenharia de Software Convencional

Uma Sucessão de transformações entre
modelos



Fonte: http://3.bp.blogspot.com/-FOot2_frID0/URRVaSUWmml/AAAAAAAAAIA/m1oIFpitBqI/s1600/charmmander.jpg



Engenharia de Software Convencional

Pressões de tempo e orçamento



Fonte: http://caputconsultoria.com.br/site_temp/wp-content/uploads/2013/10/sobreviver-pressao-trabalho.jpg



Engenharia de Software Convencional

Modelos e Documentação marginalizada



Fonte: <http://gartic.uol.com.br/imgs/mural/ls/lslf/1228800638.png>

Fonte: <http://imagensengracadas.com.br/wp-content/uploads/2012/04/CAO-ABANDONADO.jpg>



Engenharia de Software Convencional

Alterações diretamente no código



Adaptado de: <http://2.bp.blogspot.com/-Y22of21RsII/UJqXkz2DpgI/AAAAAAAAADNI/Wsnp2ajl5vo/s1600/BEBE+MALEFICO+-+C%C3%B3pia.png>

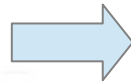


Engenharia de Software Convencional

Resultado?



CHARMANDER



CHARMELEON

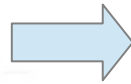


Engenharia de Software Convencional

Resultado?



CHARMANDER



CHARMELEON



MODELOS NÃO REFLETEM O PRODUTO ENTREGUE



Década de 1990

Geração Automática de Código



Década de 1990



- Engenharia de Software Embrionária;
- Não havia foco na documentação;
- Geração de sistemas legados;



Fontes:

http://3.bp.blogspot.com/-4HNPuam0qzs/U_3t_7d3Jzl/AAAAAAAAEPM/XjOFCRwT7Mw/s1600/27.PNG

http://careers.unc.edu/sites/careers.unc.edu/files/imagecache/content_image/images/Carl%20Up.jpg



Início da Década dos anos 2000

- 2001/2003 – Especificação da Model-Driven Architecture (MDA);
- Um padrão OMG;
- Uma tendência antiga;
- Mais que apenas geração automática de código;
- O início de um modelo prescritivo de processo de desenvolvimento inteiro;

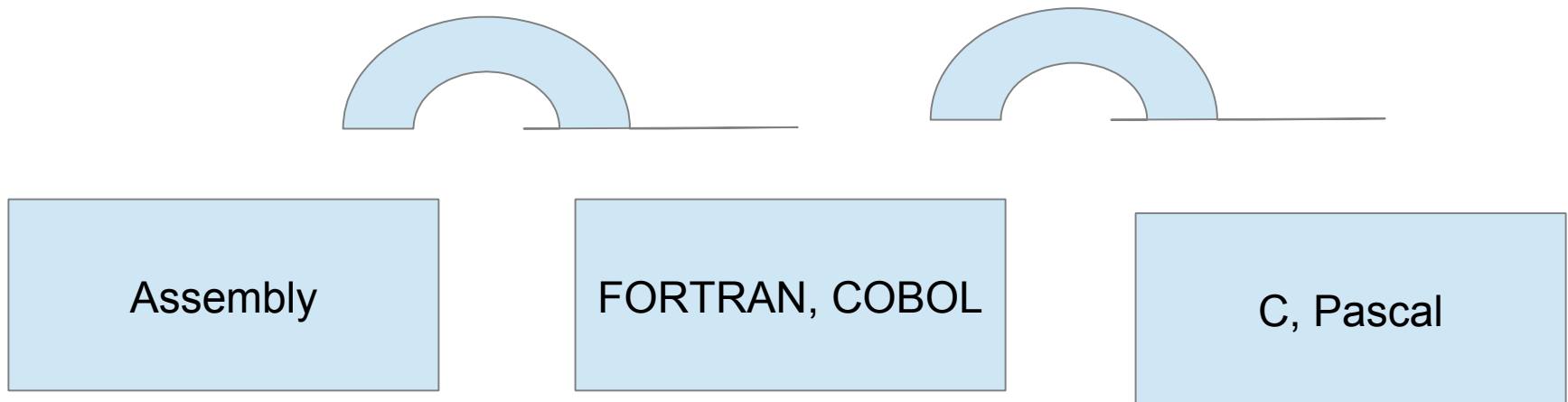


A OMG



A MDA

- Um novo salto no nível de abstração



A MDA

- Um novo salto no nível de abstração
(continuação)



C, Pascal

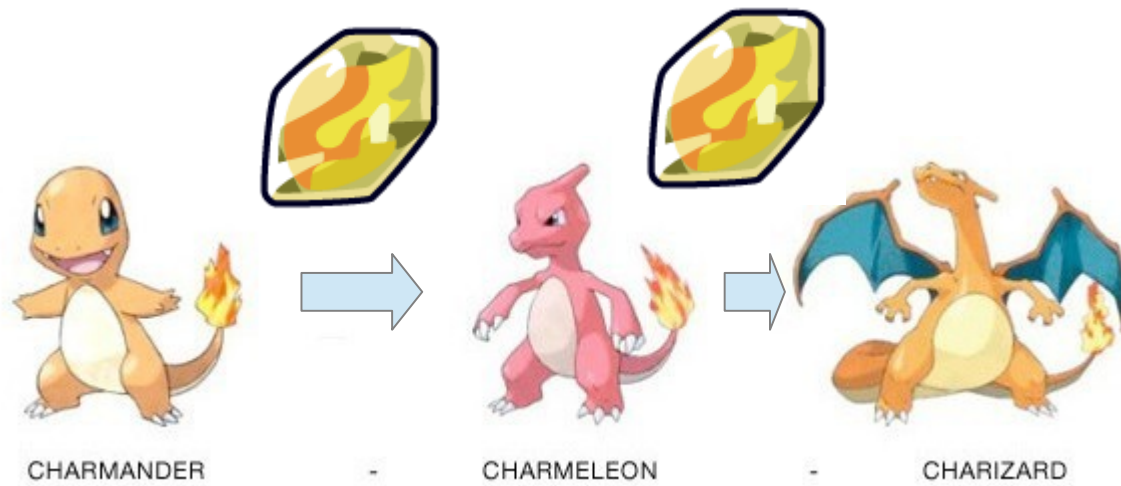
C#, Ruby, Java

Modelos



A MDA

- Produto gerado por transformações automáticas de modelos usando transformadores;



Fonte: http://3.bp.blogspot.com/-FOot2_friD0/URRVaSUWMml/AAAAAAAAAIA/m1oIFpitBqI/s1600/charmander.jpg

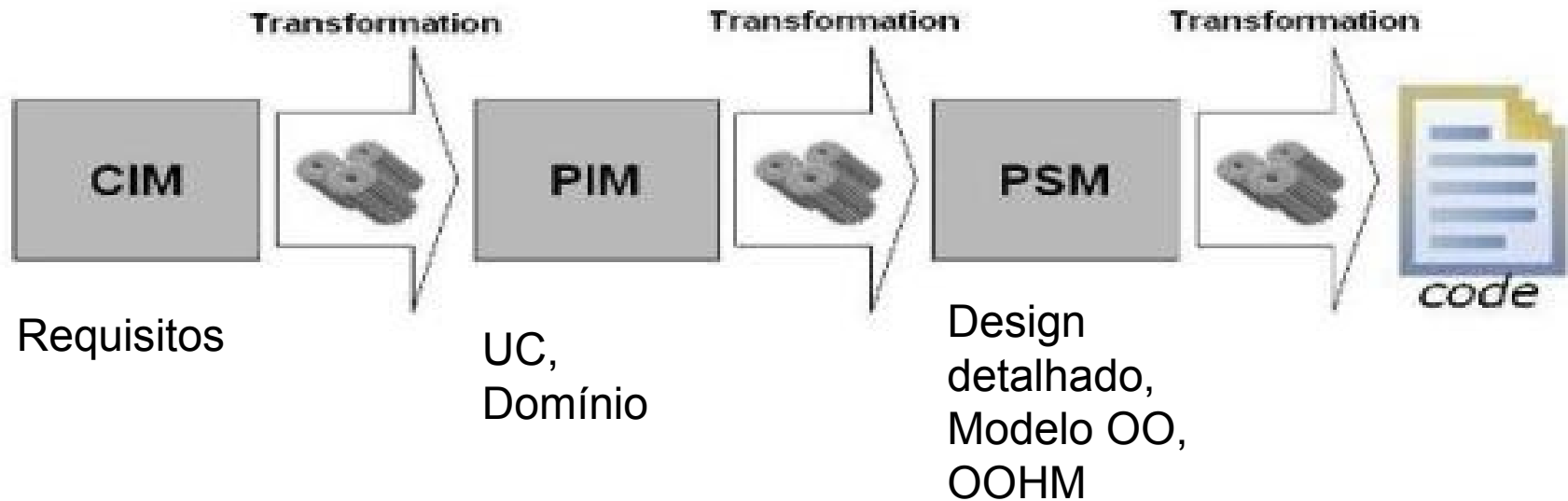


A MDA

- Princípio da Plugabilidade – Cartuchos;



MDA - Conceitos



Fonte: <http://modeldrivenarchitecture.wordpress.com/2008/08/31/introduction3-the-mda-framework/>



MDA - Conceitos

- Metamodelos



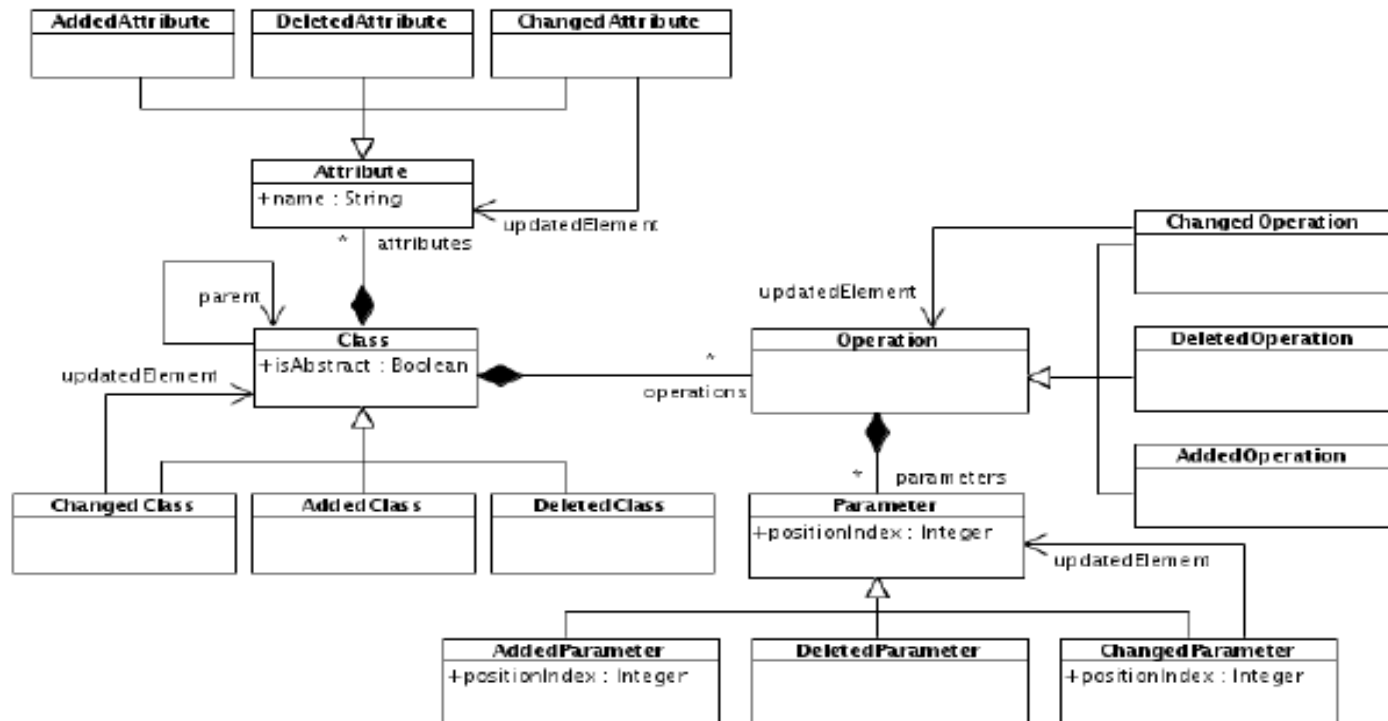
Copyright ©1999 Mauricio de Sousa Produções Ltda. Todos os direitos reservados.

Fonte: <http://www.divulgavale.com.br/wp-content/uploads/2012/09/turma-da-monica-127.gif>



MDA - Conceitos

- Metamodelos



Fonte: http://www.jot.fm/issues/issue_2007_10/paper9/images/figure5.gif



MDA - Conceitos

- Metamodelo
 - Restringe a sintaxe da escrita do modelo;
 - Mostra como construir modelos;
 - Cria insumo para escrita de uma transformação porque:

“uma transformação de modelos é um mapeamento entre um conceito do metamodelo de origem no metamodelo de destino”



MDA - Conceitos

- Transformação

Metaclasse “Classe UML”



Metaclasse “Classe Java”

É um conceito
Derivado de

É um conceito
Derivado de



Modelo

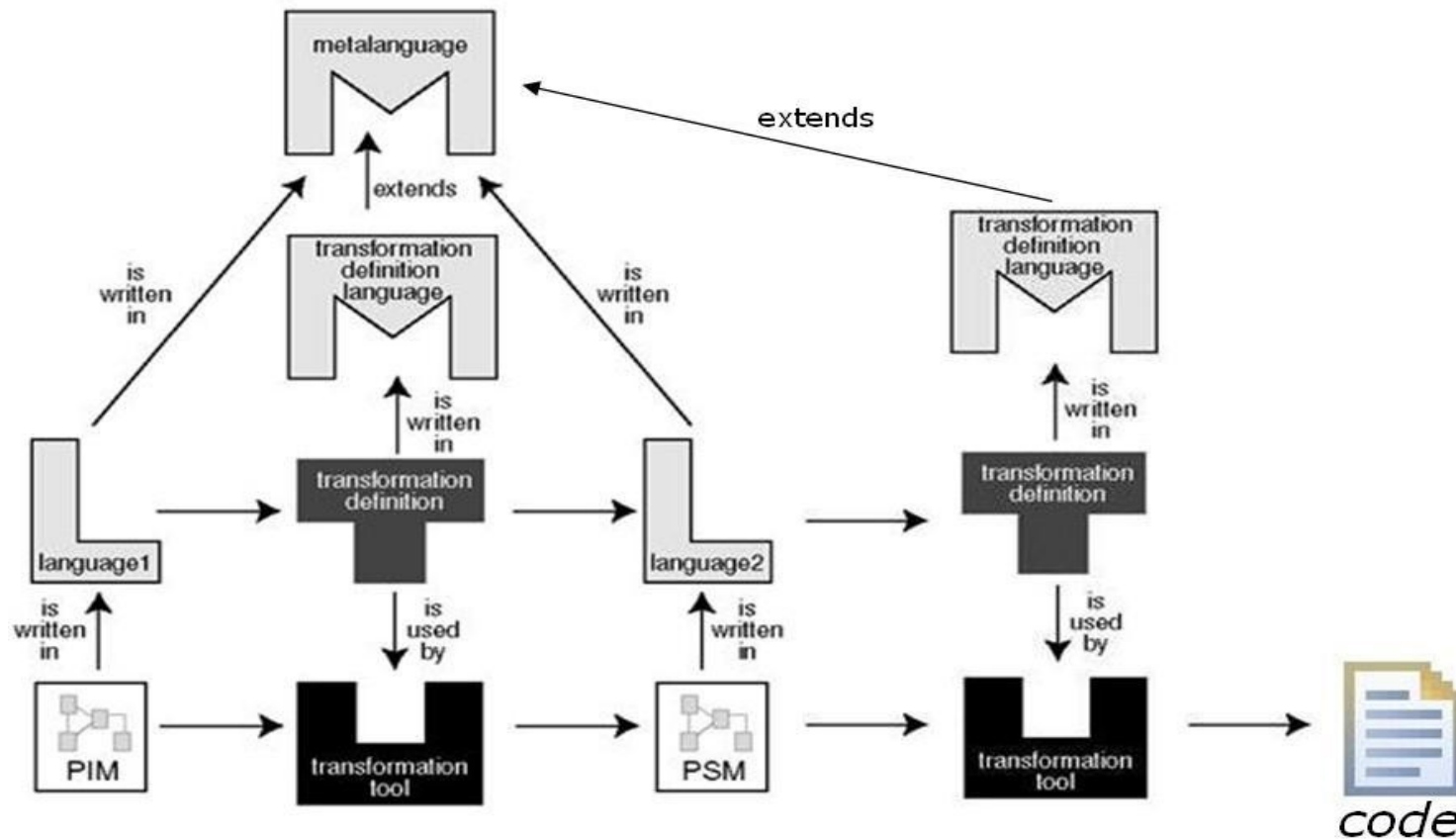
```
public class Pessoa
{
    private string    Nome;
    private int      Idade;

    public Pessoa(string nome, int idade) {
        this.Nome = nome;
        this.Idade = idade;
    }

    public void MostraInfo() {
        Console.WriteLine("{0}, tem {1} anos",
            Nome, Idade);
    }
}
```



MDA - Conceitos



Fonte: <http://modeldrivenarchitecture.files.wordpress.com/2008/08/mda-framework1.jpg>



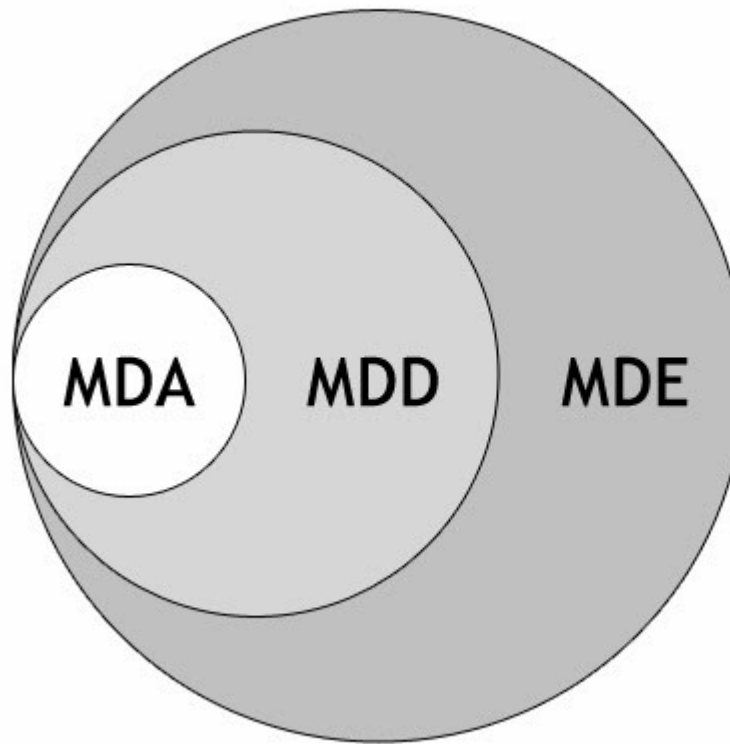
MDA - Conceitos

- Outros Conceitos
 - Model-Driven Development (MDD);
 - Model-Driven Engineering (MDE);
 - Model-Based Engineering (MBE);
 - Model-Driven * (MD-*)



MDA - Conceitos

- Outros Conceitos

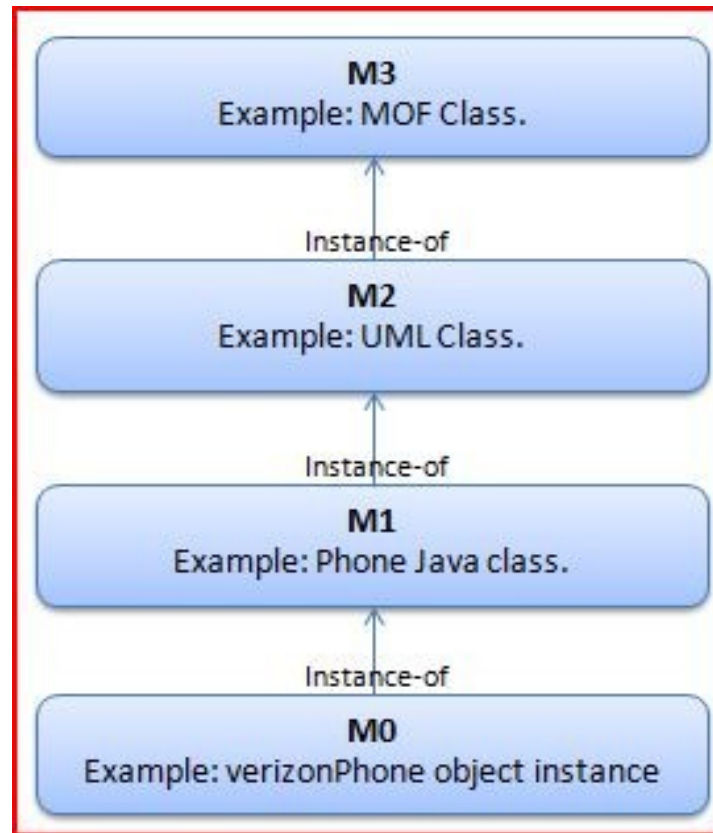


Fonte: http://4.bp.blogspot.com/-7VukvCWFs5I/UtyRcFFn_XI/AAAAAAAAACHk/Zxe2-RcKnqU/s1600/md.jpg



MDA - Conceitos

- MOF



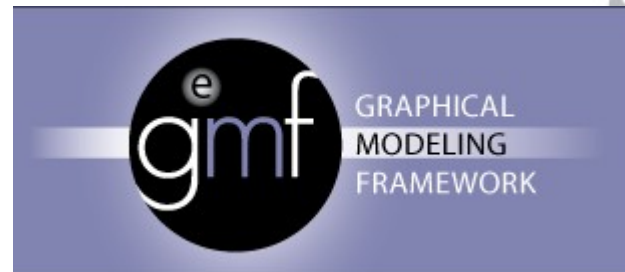
MDA - Conceitos

- Em geral, metamodelos são consideradas DSL (Domain-Specific Language);
- MDA requer todas as características mencionadas anteriormente;
- Modelos são compilados;



MDD - Tecnologias

- A maior: Eclipse Modeling Framework (EMF)



MDD - Tecnologias

- EMF disponibiliza para várias finalidades.
- Considerando transformações:
 - M2M – Model-To-Model;
 - M2T – Model-To-Text;



MDD - Tecnologias

M2M:

- ATL;
- QVT;
- Xpand;

M2T:

- Acceleo;
- Xtext;



Exemplos

M2M: ER para OO, por exemplo

- Exemplo em ATL;
- Linguagem Declarativa;

```
rule Attribute2Attribute {  
  from  
    q : sWML!Attribute  
  
  to  
    r : MVC!Attribute (  
      name <- q.name,  
      multiplicity <- 1,  
      type <- q.type.toString()  
    )  
}
```



Exemplos

- M2T: OO para Java, por exemplo
 - Exemplo em Acceleo;



Exemplos

```
[template public gerabean(e : Class){s : Integer = e.atts->size();}]

[comment GERA BEAN/]
[file ('src//com//br//gabarito//model//'.concat(e.name.replace('_main',
'').concat('.java')), false, 'UTF-8') ]
[comment package com.br.gabarito.model;/]
package com.br.gabarito.model;

public class [e.name.replace('_main', '')/] {

    [for (Sequence{1..s}) ]
    private [e.atts->at(i).type/] [e.atts->at(i).name.replace('_main', '')/];
    [/for]

    [for (Sequence{1..s}) ]
    public [e.atts->at(i).type/] get[e.atts->at(i).name.replace('_main',
'').toUpperFirst()/](){
        return this.[e.atts->at(i).name.replace('_main', '').toLowerFirst()/];
    }

    public void set[e.atts->at(i).name.replace('_main', '').toUpperFirst()/]([e.atts-
>at(i).type/] [e.atts->at(i).name.replace('_main', '')/]){
        this.[e.atts->at(i).name.replace('_main', '')/] = [e.atts-
>at(i).name.replace('_main', '')/];
    }
    [/for]

    [comment toString nao sera gerado /]
}

[/file]
```



Vantagens em usar MD-*

- Automação;
- Agilidade;
- Manutenibilidade;
- Sincronia entre modelo e código;
- Qualidade;
- Reúso;
- Rastreabilidade;
- Produtividade;
- Portabilidade;



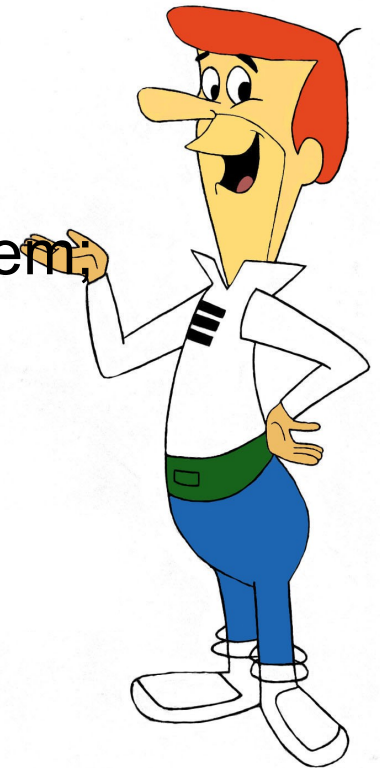
Desvantagens em usar MD-*

- Especificidades são difíceis de modelar;
- Coevolução;
- Metamodelos também evoluem (são linguagens);
 - Como manter modelos antigos e outros artefatos ainda consistentes – questão clássica de evolução de linguagens de programação;
- Produtividade
 - Tempo é gasto em aprendizado e implementação de transformadores/regras de transformação



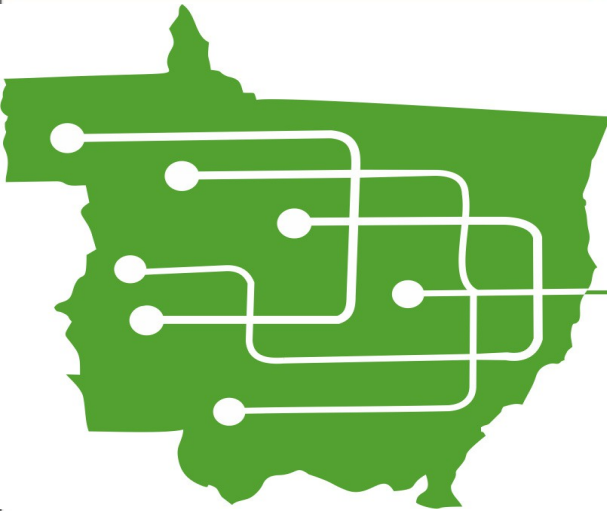
Perspectivas e Futuro

- Valorização progressiva dos modelos;
- Transformadores mais robustos;
- Foco em modelagem;
- Um “shifting” de programação para modelagem;
- Mudanças nos currículos universitários;
- Mudanças na cultura organizacional;



Perguntas





Escola Regional de Informática

SBC Mato Grosso

De 14 a 17 de Outubro de 2014



Muito obrigado, terrestres!!

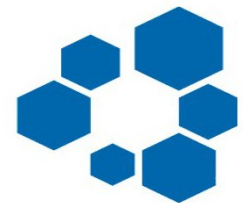
Desenvolvimento Dirigido por Modelos:
Conceitos, Aplicações, e Perspectivas

Prof. Valdemar Neto

valdemarneto@inf.ufg.br



INSTITUTO DE
INFORMÁTICA
UFG



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

