

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO



Model-Driven Architecture

Guilherme Potenciano
Ricardo Cacheta Waldemarin

SSC5944 - ARQUITETURA DE SOFTWARE

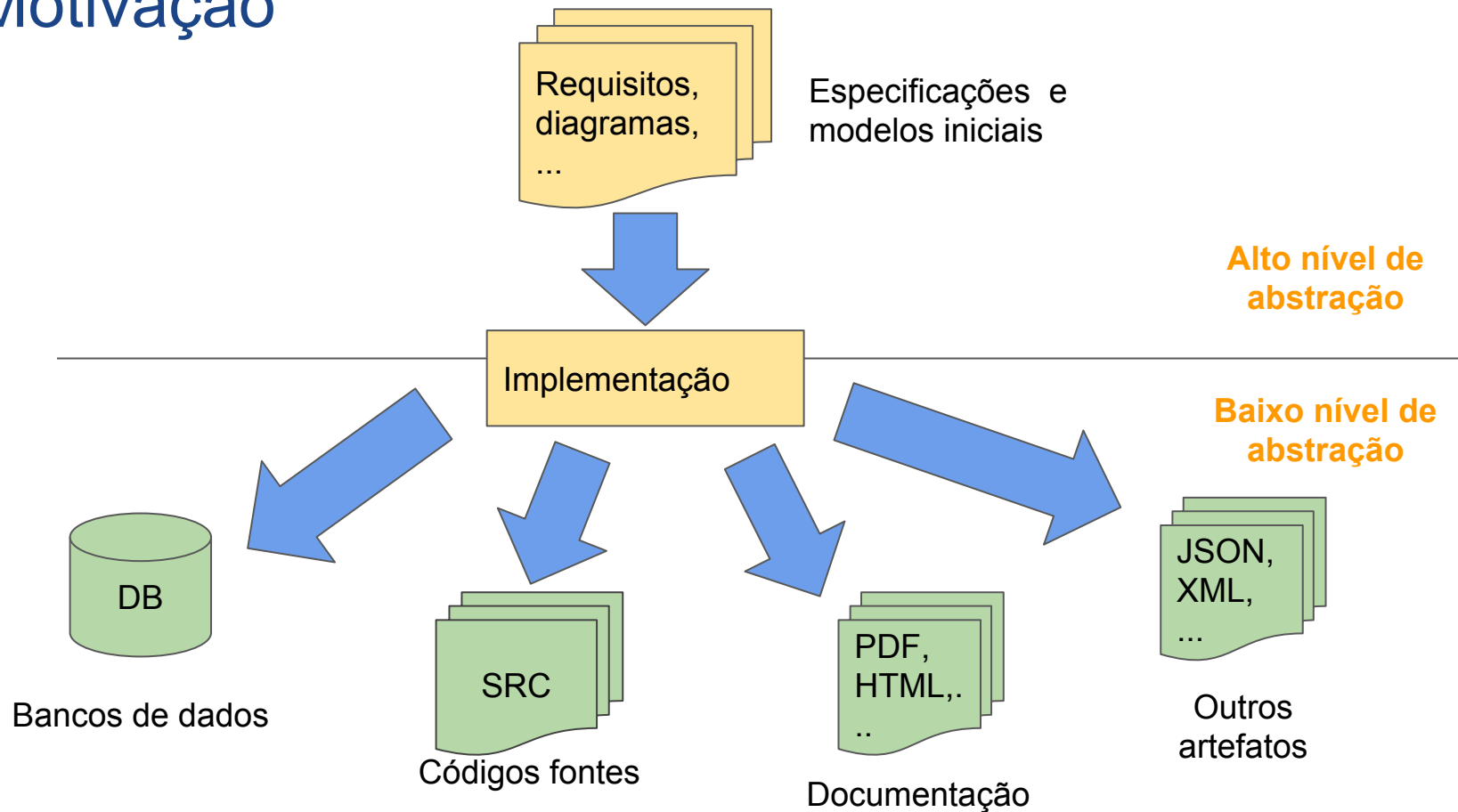
“(...) it might be worth-while to point out that the purpose of abstracting in not to be vague, but to create a new semantic level in which one can be absolutely precise.”

DIJKSTRA, 1972

Motivação

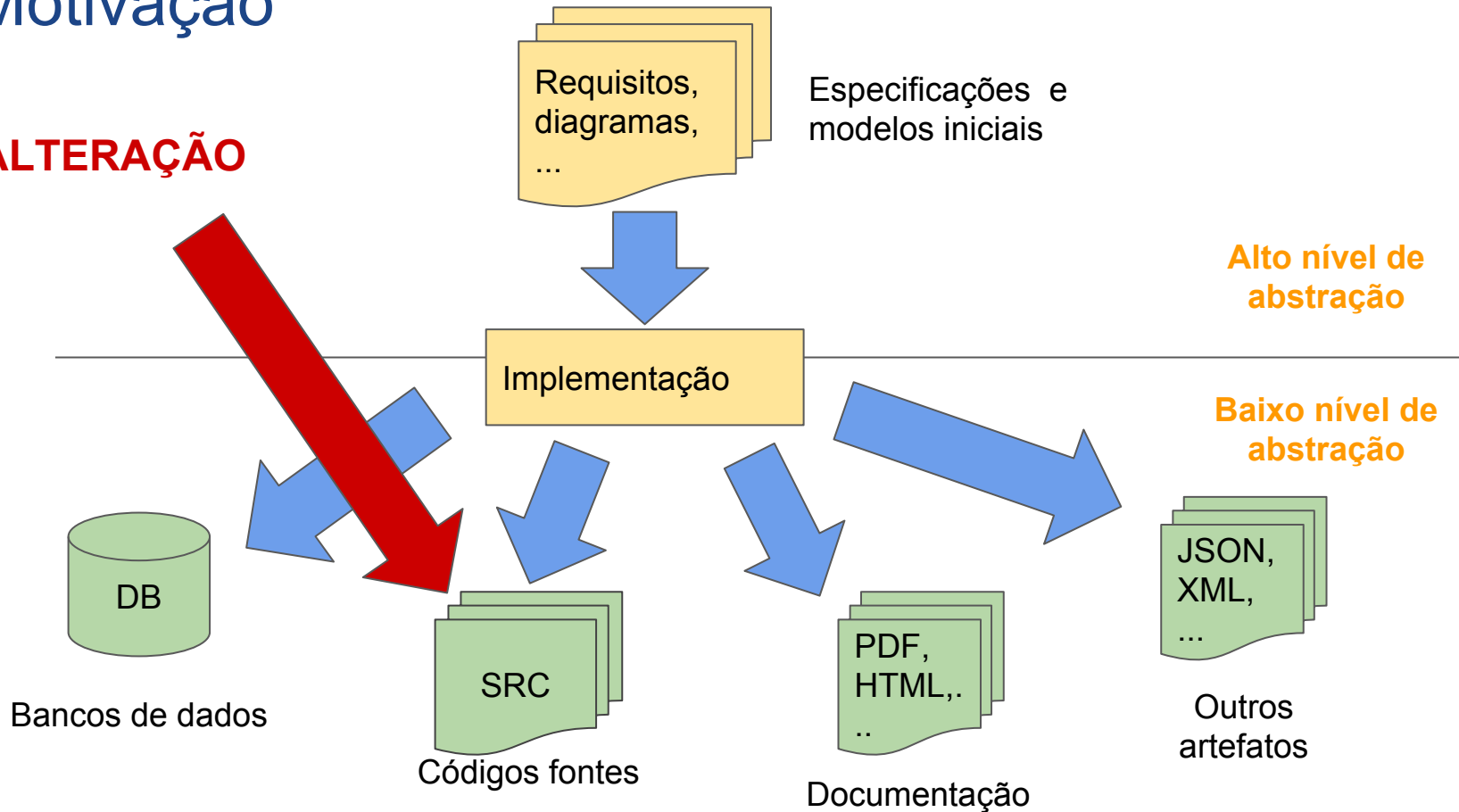
- Existem dois fatores importantes quando se consideram os problemas de documentação de um sistema:
 - em geral existe pouco interesse no seu desenvolvimento;
 - a cada manutenção feita no sistema deveria corresponder uma mudança na documentação o que dificilmente é feito.
- Uma solução seria o uso de uma ferramenta que promovesse a documentação automática, a partir do código fonte (como existe na plataforma Java), o que resolveria o problema de documentação em baixo nível. No caso de sistemas complexos a documentação a níveis mais altos é essencial, permanecendo, neste caso, os problemas de documentação.

Motivação



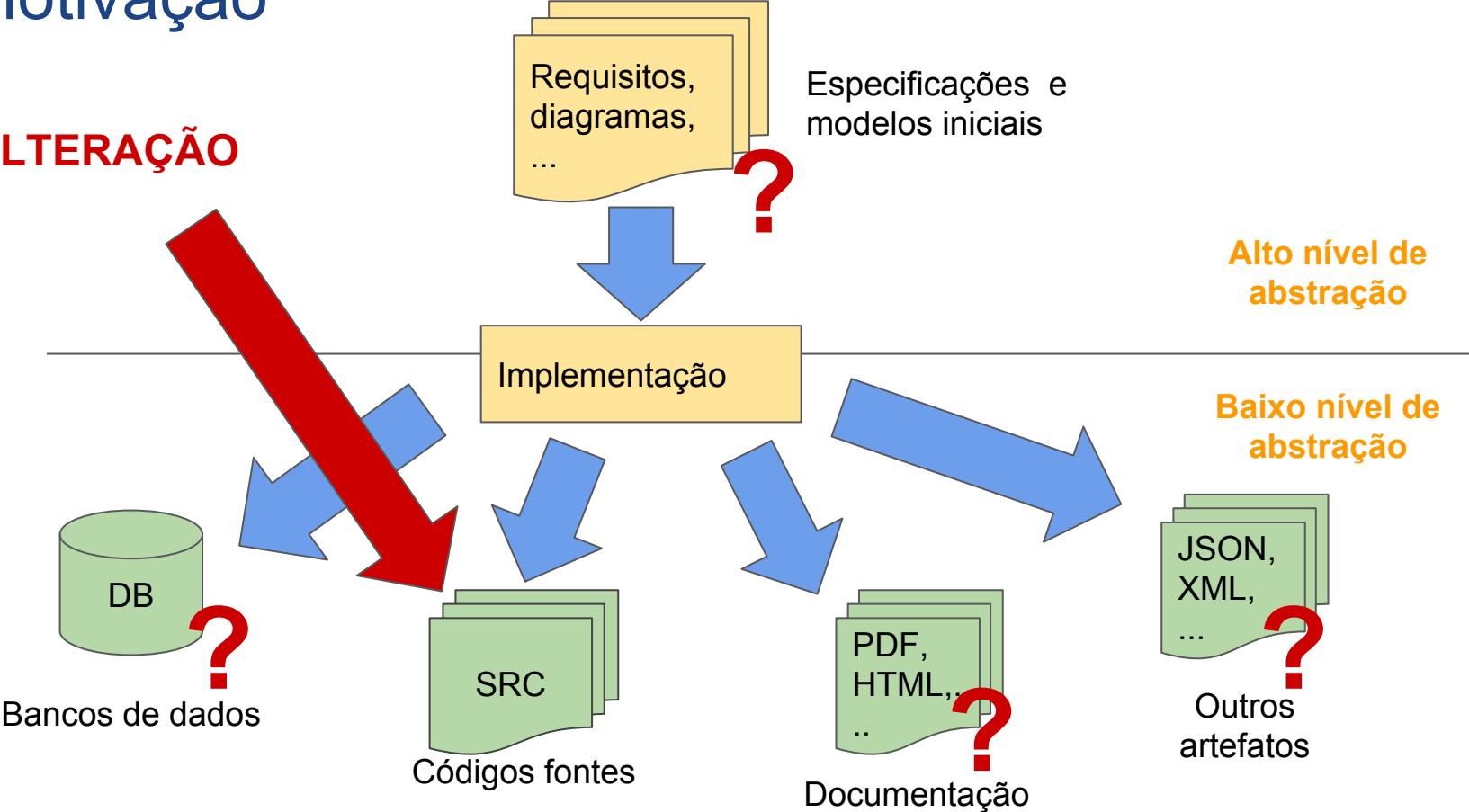
Motivação

ALTERAÇÃO



Motivação

ALTERAÇÃO



Motivação

Uma solução a esse problema seria que toda alteração fosse feita primeiramente nos modelos e especificações de mais alto nível.

Dado uma representação e suporte à traceabilidade adequados, as alterações nas especificações de mais alto nível poderiam ser propagadas, então, para os artefatos de baixo nível.

Motivação

Uma solução ainda melhor seria que todos os artefatos relacionados ao sistema pudessem ser gerados automaticamente a partir dos modelos iniciais.

Model-driven Development

“Model-driven development is simply the notion that we can construct a model of a system that we can then transform into the real thing.”

MELLOR; CLARK; FUTAGAMI, 2003

Modelos e metamodelos

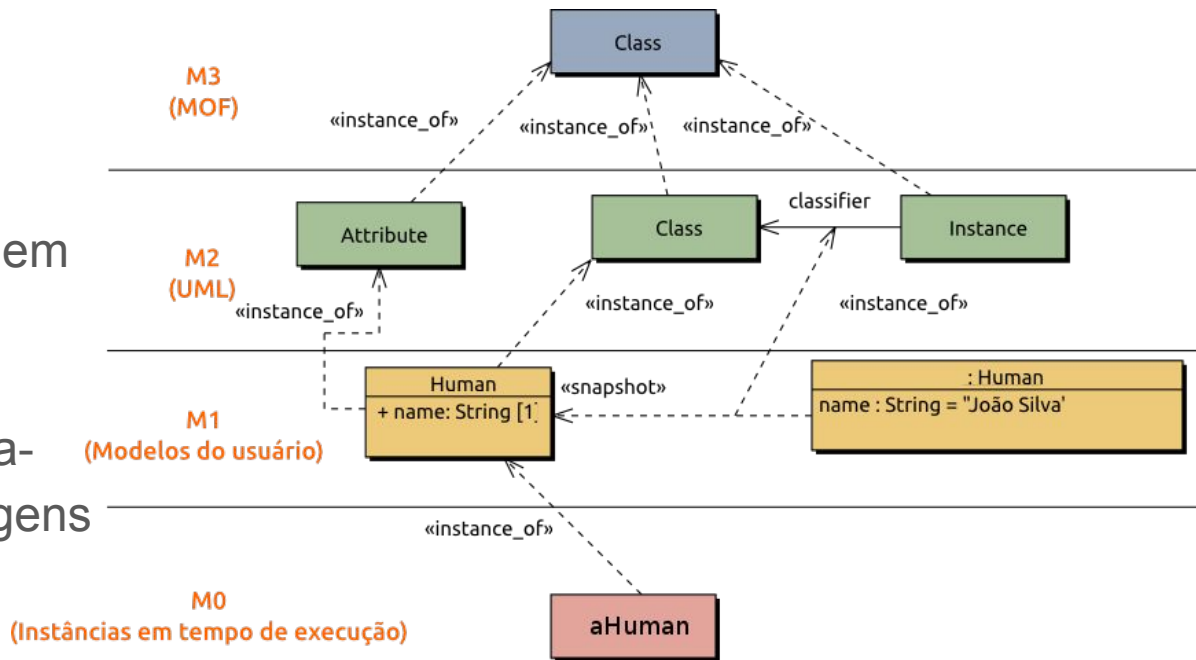
- Modelos
 - Declarações formais ou semi-formais sobre um sistema qualquer
 - Prescritivos ou descritivos
- Características chaves de um modelo
 - Abstração
 - Correção
 - Preditividade
 - Baixo-custo
- Um modelo que especifica como modelos podem ser criados é chamado metamodelo.

Modelos e metamodelos - Metamodelagem



Arquitetura de metamodelagem da UML

MOF é utilizado como meta-metamodelo de várias linguagens da OMG



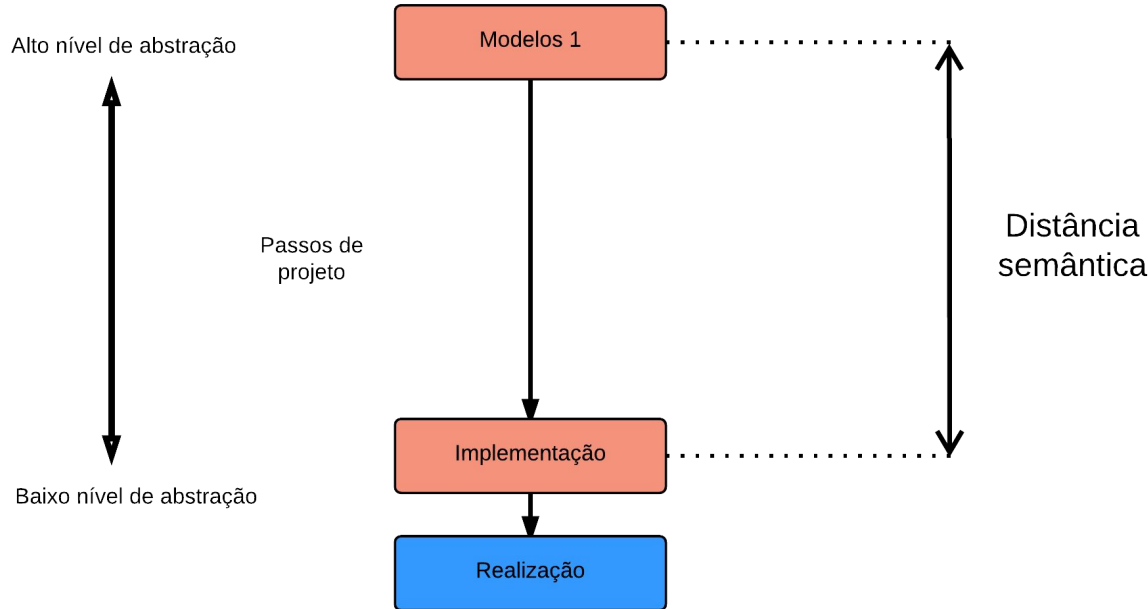
Model-driven Development

- Visa o refinamento e a transformação sistemática de modelos de alto nível de abstração em modelos mais concretos até a obtenção final de uma implementação de um sistema computacional em uma dada linguagem de programação
- Altera o foco de desenvolvimento de um sistema computacional, normalmente posicionado sobre a atividade de programação, para a atividade de modelagem

Model-driven Development

- Permite criação de modelos independentes de uma dada tecnologia de implementação e mais próximos do domínio de aplicação
 - Facilita a compreensão, especificação e manutenção do sistema
 - Tais modelos são menos suscetíveis a mudanças tecnológicas
 - Permite o reuso desses modelos no desenvolvimento para diferentes tecnologias de implementação
- Suporte adequado pode ser provido de modo a transformar os modelos criados em código de aplicação

MDD - Refinamentos Sucessivos



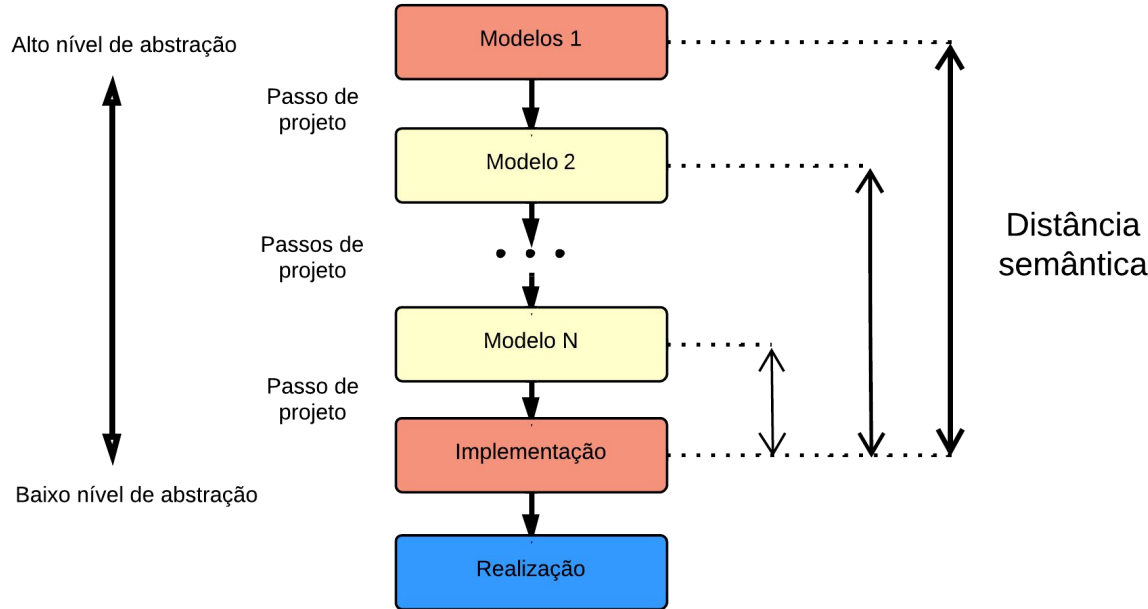
Conceito importante para o MDD.

Os modelos iniciais da especificação de um sistema são mais próximos do domínio de aplicação e muito abstratos.

Isso se reflete em um grande gap semântico entre tais modelos e implementação final do sistema.

Muitas decisões de implementação ainda precisam ser tomadas.

MDD - Refinamentos Sucessivos

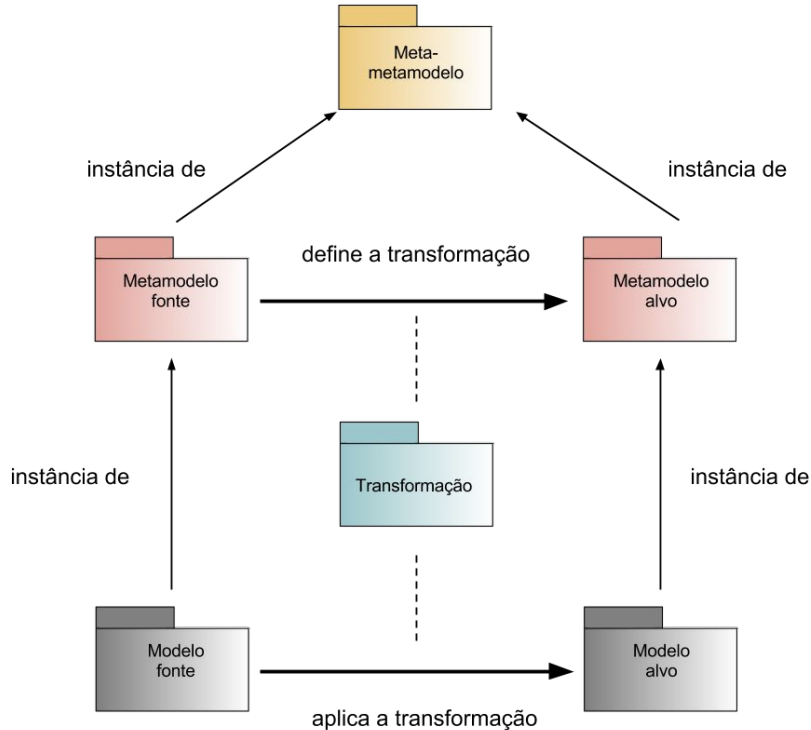


Refinamentos sucessivos podem ser utilizados para introduzir detalhes e reduzir a distância semântica gradativamente.

Permitem documentar as decisões de implementação de forma sistemática e rastreável.

Eventualmente pode-se conseguir um modelo tão próximo da linguagem de implementação que possa ser transformado em código automaticamente.

MDD - Transformações de modelos

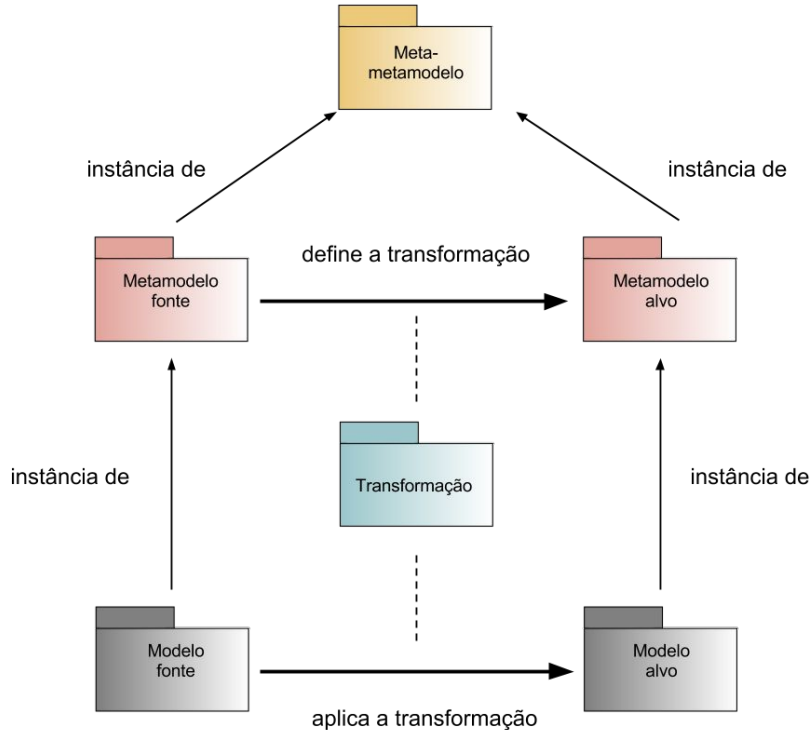


O suporte completo ao MDD só pode ser alcançado se houver suporte a transformações sucessivas que permitam obter uma implementação do sistema em uma dada linguagem de programação a partir dos modelos iniciais.

Nesse sentido, uma transformação de modelos pode ser vista como um mapeamento entre elementos de um ou mais modelos fontes para elementos de um ou mais modelos alvos.

Esse mapeamento pode ser definido formalmente em relação aos metamodelos envolvidos, possibilitando a execução automática da transformação para quaisquer modelos.

MDD - Transformações de modelos



A definição de transformações é facilitada se os metamodelos envolvidos são instâncias de um mesmo meta-metamodelo.

Qualquer artefato do desenvolvimento de um sistema pode ser visto como um modelo.

A estrutura desses artefatos seria o metamodelo dos mesmos.

Num contexto de MDD, é desejável que todos esses artefatos sejam, de alguma forma, obtidos a partir de transformações automáticas dos modelos desenvolvidos.

Assim, alterações nos modelos poderiam ser propagadas de forma mais rápida e confiável para todos os artefatos.

“OK, I got it! But what’s MDA?”

YOU, 2016

Model-driven Architecture

Abordagem de desenvolvimento orientado a modelos

Proposta pela OMG para o desenvolvimento de sistemas computacionais

Problema inicial: proliferação de middlewares em aplicações enterprise

- É difícil a uma empresa padronizar todos os seus negócios em apenas um middleware
- Por vezes, É NECESSÁRIA a integração com os diferentes middlewares
- Todas essas integrações consomem tempo e recursos

Model-driven Architecture

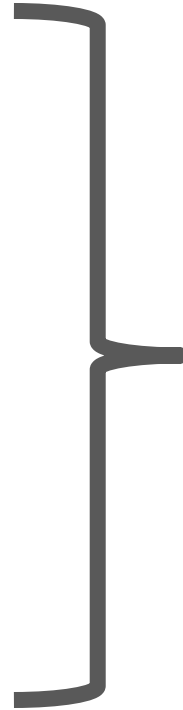
- **Idéia geral:** Separar a especificação operacional do sistema dos detalhes sobre a forma que o sistema usa as capacidades de uma dada plataforma
- **Como?**
 - Especificar sistemas de forma independente da plataformas
 - Especificar plataformas
 - Escolher uma plataforma em particular para um sistema
 - Transformar a especificação do sistema em uma especificação adequada à plataforma escolhida

Model-driven Architecture

Modelos independentes de computação
Computation Independent Models (CIMs)

Modelos independentes de plataformas
Platform Independent Models (PIMs)

Modelos específicos de plataformas
Platform Specific Models (PSMs)



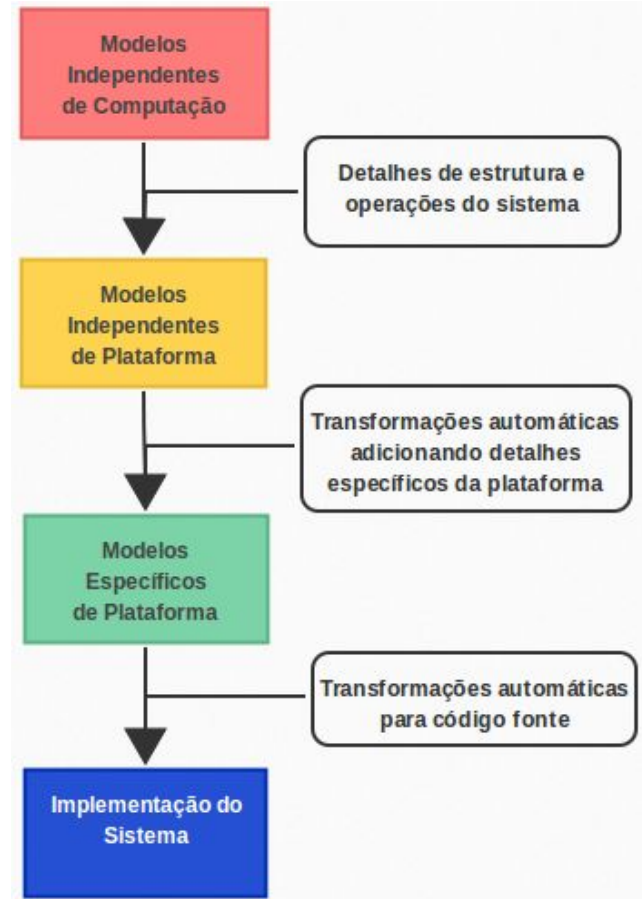
**Principais
pontos de
vista do MDA**

Model-driven Architecture

Modelos independentes de computação
Computation Independent Models (CIMs)

Modelos independentes de plataformas
Platform Independent Models (PIMs)

Modelos específicos de plataformas
Platform Specific Models (PSMs)



Model-driven Architecture - CIM

Computation Independent Models

- Descreve requisitos do sistema e o ambiente de uso
 - Às vezes chamado de *domain model* ou *business model*
- Independe de como o sistema é implementado
- Apresenta o que se espera que o sistema faça
- No contexto MDA, cada requisito de um CIM deve ser rastreável para elementos do PIM e do PSM que o implementam, e vice-versa

Model-driven Architecture - PIM

Platform Independent Models

- Descreve o sistema, criando especificações para os serviços necessários sem os detalhes técnicos da plataforma
 - Considera a plataforma de maneira abstrata
- O PIM mantém todo o seu valor ao longo dos anos, exigindo mudanças apenas quando tem alterações de regras de negócio
- Mapeável para diferentes plataformas

Model-driven Architecture - PSM

Platform Specific Models

- É derivado dos PIMs por meio de transformações e adições de detalhes específicos de uma dada plataforma
- Um PSM pode prover diferentes níveis de detalhamento dependendo do seu propósito.
 - Se provê todas as informações necessárias para a construção do sistema e colocá-lo em operação é considerado um PSM de implementação,
 - Caso tenha poucos detalhes, pode continuar a ser refinado e no futuro servir como um PSM de implementação

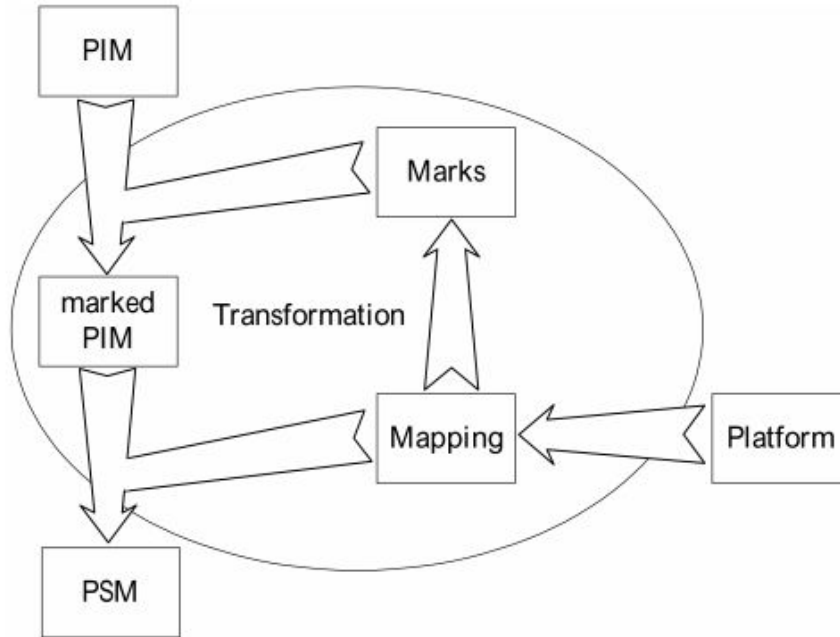
Model-driven Architecture - Platform Models

- Provêm um conjunto de conceitos representando os diferentes componentes de uma plataforma e os serviços providos por ela.
- Também provê conceitos representando os diferentes tipos de elementos a serem usados para especificar o uso da plataforma por uma aplicação
 - Esses elementos são usados para a construção dos PSMs.
- Genericamente, um **platform model** pode ser considerado como uma **especificação de um estilo arquitetural**

Model-driven Architecture - Mappings & Markings

- Mapeamentos entre os elementos das linguagens de representação dos PIMs e PSMs são utilizados para especificar a transformação
- Tipos de mapeamento
 - **Model Type Mappings:** Atributos e padrões dos elementos do PIM são usados para a decisão de como criar o elemento adequado no PSM
 - **Metamodel Type Mappings:** Quando PIM e PSM possuem mesmo meta-metamodelo (MOF), é possível criar regras e algoritmos envolvendo todas as instâncias de um dado tipo de elemento
 - **Model Instance Mappings:** Elementos dos PIMs podem ser marcados para ser transformados de uma forma específica. Essa marcação não deve fazer parte do PIM, devendo ser adicionada de uma forma transparente. Por exemplo, pela adição de estereótipos UML.

MDA - Model Transformation Approaches



- Uma plataforma em particular é escolhida.
- O mapeamento da plataforma está disponível ou é criado.
- Esse mapeamento inclui um conjunto de marcas.
- As marcas são usadas para marcar os elementos do modelo e orientar a transformação do modelo.
- O PIM marcado é ainda transformado, utilizando o mapeamento para produzir o PSM

Figure 3-1 Marking a Model

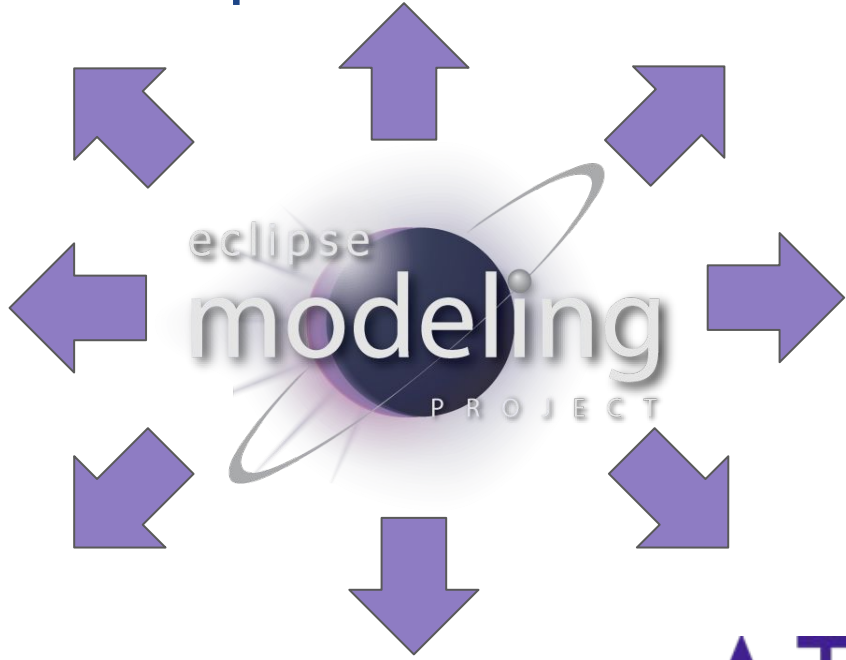
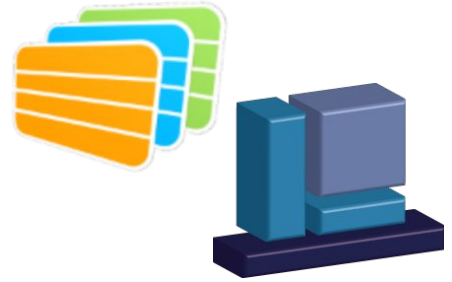
Suporte ao MDD e MDA

Ferramentas de suporte são necessário para promover e possibilitar o maior uso dessas tecnologias

Há diversas ferramentas que provêm suporte ao MDD e ao MDA

Dois concorrentes de peso são o Eclipse e o Enterprise Architect

Suporte a MDD - Eclipse



Suporte a MDD - Modelagem Conceitual

Framework de modelagem conceitual e criação de metamodelos

Provê mecanismos para validação dos modelos e transformação dos metamodelos para classes Java

Espinha dorsal de integração do EMP

MDT

Provê metamodelos, frameworks e editores para modelos baseados em padrões da industria de software

UML, OCL, BPMN, QVT...



Suporte a MDD - Desenvolvimento de UI

Framework para desenvolvimento MDD de editores gráficos para a edição de modelos EMF

Modelos descrevendo o editor criado são compilados para classes Java (semi-)automaticamente



Permite criar pontos de vistas para modelos EMF

Diagramas, tabelas, matrizes e árvores hierárquicas

Declarativo e independente de geração de código

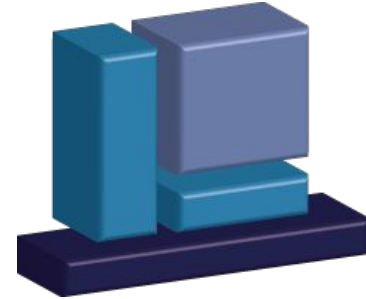
**Graphical
Modeling
Framework
(GMF)**

Suporte a MDD e MDA - Desenvolvimento de UI

EMF Client Platform

Framework para desenvolvimento de aplicações clientes baseadas em EMFs

Componentes podem ser usados stand-alone ou integrados na aplicação



EMF FORMS

Subcomponente do EMF Client Platform

MDD para form-based UIs

Renderers para Swing, SWT, JavaFX or Web



Suporte a MDD - Desenvolvimento de Representação Textual

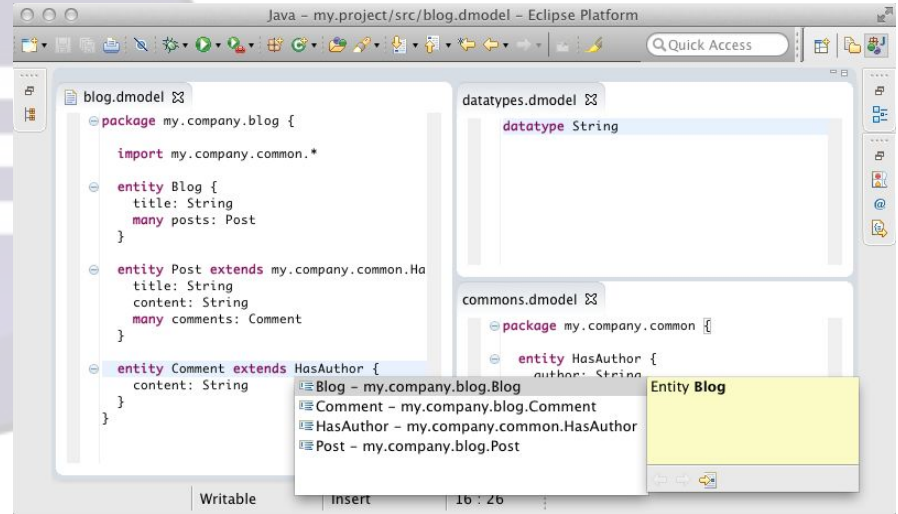
Framework para desenvolvimento de DSLs textuais para modelos EMF

Gramáticas são automaticamente implementadas como editores Eclipse/IntelliJ/Web que provém:

- Validação sintática
- Autocompleting
- Parsers e serializers

Transformação de modelos para texto/código fonte

Xtext



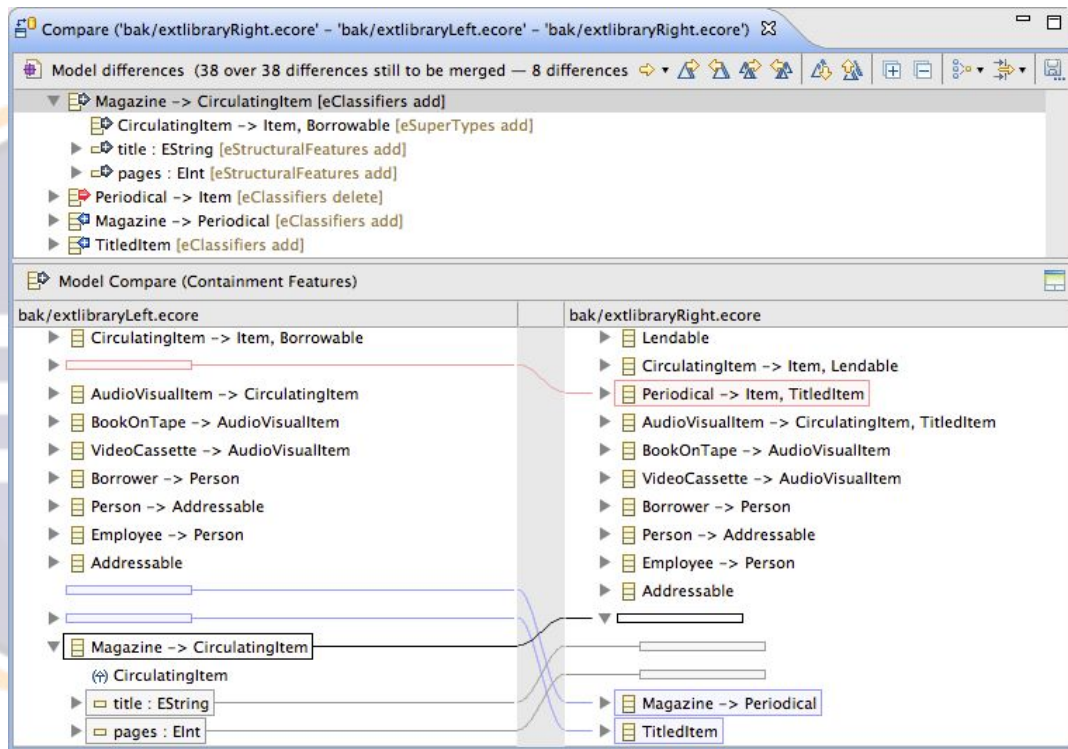
Suporte a MDD - Colaboração



Comparação e merging entre modelos

Extensível

Apresenta suporte a Git e outros sistemas de versionamento o nível dos modelos



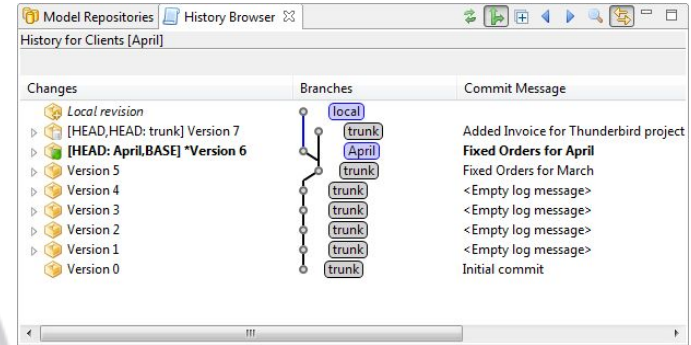
Suporte a MDD - Colaboração

EMFStore

Repositório versionado para artefatos EMF

Entende o artefato ao nível do metamodelo,
permitindo comparações e merging
semânticos

Extensível



Suporte a MDD - Transformações



Permite definir transformações entre metamodelos baseados em Ecore

Expressões declarativas e imperativas

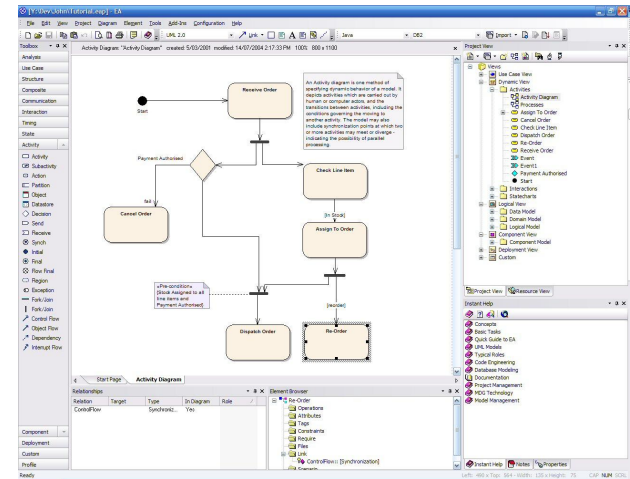
Linguagem OCL-like

Permite descrever high-order transformations

```
rule rule_name {
  from
    in_var : in_type [in model_name]? [(
      condition
    )]?
  [using {
    var1 : var_type1 = init_exp1;
    ...
    varn : var_typen = init_expn;
  }]?
  to
    out_var1 : out_type1 [in model_name]? (
      bindings1
    ),
    out_var2 : distinct out_type2 foreach(e in collection)(
      bindings2
    ),
    ...
    out_varn : out_typen [in model_name]? (
      bindingsn
    )
  [do {
    statements
  }]?
}
```

Suporte a MDD e MDA

- Ferramenta de modelagem e projeto
- Suporte a varias linguagens, metamodelos e metodologias
 - SysML
 - BPMN
 - UML 2.5
 - MDA
- Provê traçabilidade entre os diversos artefatos
- Software comercial



Conclusões

- MDD move o foco de desenvolvimento para os modelos mais próximos do domínio da aplicação
 - Permite uma melhor abstração e compreensão do sistema a ser implementado
 - Busca obter os artefatos implementando e documentando o sistema modelado por meio do refinamento sucessivo e da transformação dos modelos iniciais.
 - Provê maior confiabilidade na propagação de alterações e ciclos de desenvolvimento mais rápidos.
- MDA que visa a separação dos detalhes independentes de uma plataforma de implementação daqueles dependentes de uma dada plataforma
 - Permite o reuso e integração mais fácil dos modelos desenvolvidos
 - Facilita o desenvolvimento para diferentes plataformas.
- O suporte a essas abordagens é provido por diferentes ferramentas.

“This rejection of modeling for software is particularly ironic when you consider that software is the engineering medium best positioned to benefit from it.”

SELIC, 2003

Bibliografia

DIJKSTRA, E. W. EWD 340: The humble programmer was. Commun. ACM 15, v. 15, 1972.

MELLOR, S. J. et al. Model-Driven Architecture. In: BRUEL, J.-M.; BELLAHSENE, Z. (Eds.). . Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 290–297.

MELLOR, S. J.; CLARK, A. N.; FUTAGAMI, T. Model Driven Development. IEEE Software, v. 20, n. 5, 2003.

TRUYEN, F.; Implementing Model Driven Architecture using Enterprise Architect. Disponível em <http://www.sparxsystems.com.au/downloads/whitepapers/EA4MDA_White_Paper_Features.pdf>

OBJECT MANAGEMENT GROUP. Model Driven Architecture. . [S.l: s.n.]. , 2000

OBJECT MANAGEMENT GROUP. MDA Guide Version 1.0.1. . [S.l: s.n.]. Disponível em: <<http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf>>. , 2003

Muito obrigado pela atenção!