

MDD - Desenvolvimento de Software Dirigido a Modelos

Slide 1/21

Introdução

- Modelar é uma atividade essencial da Engenharia de Software
- Na década de 1970 os métodos formais difundiram o desenvolvimento de software a partir de modelos formais matemáticos
- Em 2001 o OMG especificou a MDA, definindo assim uma nova instância do processo MDD
- Principal idéia em MDD: transformação de modelos de maiores níveis de abstração em modelos mais concretos, até se obter o código do sistema
 - Desenvolvimento inicial e modificações futuras devem ser efetuados apenas no modelo mais abstrato

Slide 2/21

Introdução

- Principais argumentos para a utilização de MDD:
 - Interoperabilidade
 - Maior produtividade
 - Portabilidade
 - Redução de custo e esforço de desenvolvimento
 - Mais facilidade na evolução do software
 - Ou seja, Maior qualidade do produto
- Não há um consenso na comunidade acadêmica sobre qual modelo de maior nível de abstração é mais adequado para se modelar um sistema

Slide 3/21

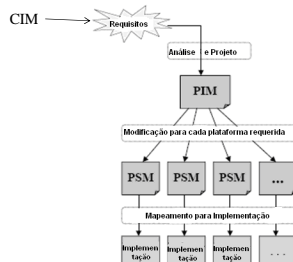
Arquitetura Dirigida a Modelos

- Conceitos Básicos:
 - Modelo: Representação funcional, estrutural e comportamental
 - Dirigido a Modelos: Provê meios para direcionar o entendimento, projeto, construção, e modificação de um sistema.
 - Arquitetura: Especificação de partes e conectores de um sistema, além das regras de interação dessas partes usando os conectores.
 - Plataforma: Conjunto de subsistemas e tecnologias que provê um conjunto coerente de funcionalidade
 - Pontos de Vista (Modelos MDA):
 - ✓ Modelo Independente de Computação (CIM)
 - ✓ Modelo Independente de Plataforma (PIM)
 - ✓ Modelo Específico de Plataforma (PSM)
 - Transformações e Mapeamentos

Slide 4/21

Arquitetura Dirigida a Modelos

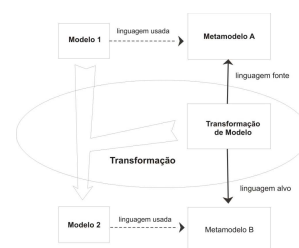
- Transformações em MDA



Slide 5/21

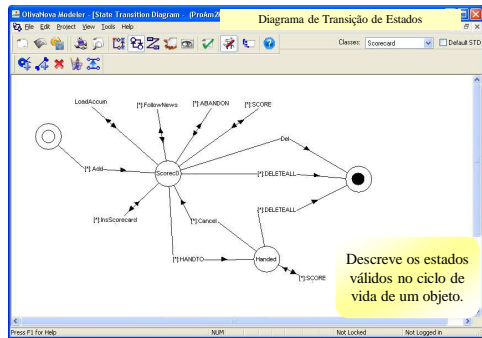
Arquitetura Dirigida a Modelos

- Transformações por meio de Metamodelos



Slide 6/21

INTEGRANOVA: MODELO DINÂMICO

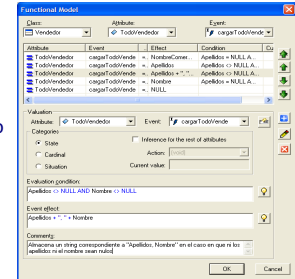


Slide 13/21

INTEGRANOVA: MODELO FUNCIONAL

Especifica o relacionamento entre o estático e o dinâmico

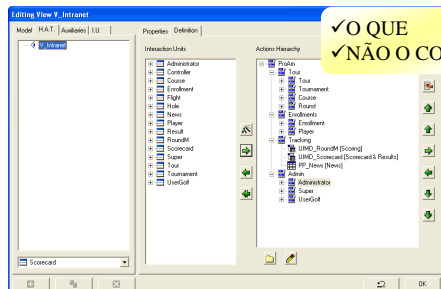
- Descreve a semântica relacionada com as mudanças de estado
- Descreve como a execução de eventos altera os valores dos atributos



Slide 14/21

INTEGRANOVA: MODELO DE APRESENTAÇÃO

Especificação Abstrata da Interface do Usuário



Slide 15/21

Outras Ferramentas MDD

- **AndroMDA**
 - É uma ferramenta MDA OpenSource (www.andromda.org/)
 - Desenvolvida sobre o Eclipse
 - Possibilita a definição de transformação de modelos PIM para PSM para depois permitir a geração de código fonte
 - É possível gerar componentes para várias linguagens (Java, .Net, HTML, PHP, C++, etc)
 - Não provê recursos de definição abstrata de interface do usuário
 - Trata questões de rastreabilidade e validação de modelos de forma limitada

Slide 16/21

Outras Ferramentas MDD

- **MOSKitt (Modeling Software Kit - Kit de Modelagem de Software)**
 - Construída sobre o Eclipse
 - Permite a edição gráfica de modelos
 - Suporta persistência
 - Suporta trabalho em time e versão de modelos
 - Suporta Transformação de Modelo, Rastreabilidade e Sincronização

Slide 17/21

Lições Aprendidas na Adoção de Soluções MDA

- Examinar os modelos atualmente usados na empresa
- Identificar as transformações candidatas para a automação
- Especificar os requisitos dessas transformações
- Desenvolver o código da transformação
- Esboçar documentos de uso, empacotar e distribuir

Slide 18/21

Problemas e Desafios dos Processos MDD

- Elevação dos níveis de abstração dos modelos
- Melhor precisão semântica dos modelos em relação às características estáticas, dinâmicas e de apresentação (interação-usuário) dos sistemas
- Melhores mapeamentos entre os modelos
- Melhor transformação automática de modelos (automação)
- Melhor suporte à Validação de Modelos
- Melhor integração com as plataformas específicas (PSM)
- Melhor e maior porcentagem de código fonte gerado
- Melhor suporte à rastreabilidade
- Melhor suporte à engenharia reversa
- Melhor Suporte a Testes Dirigidos a Modelos
- Suporte a requisitos não-funcionais

Slide 19/21

Considerações Finais

- MDD, conforme a arquitetura MDA, foi concebido para alcançar os seguintes benefícios:
 - Interoperabilidade
 - Produtividade
 - Portabilidade
 - Redução de custo e esforço

Slide 20/21

Considerações Finais

- Outros processos de desenvolvimento de software também têm essas metas. No entanto, em MDD já está explícita a ideia de automatização
- Existem muitas ferramentas que suportam o MDD, inclusive gratuitas
 - Grande maioria utiliza a UML como linguagem para modelagem
- O Processo MDD ainda está na sua infância
 - Nem as linguagens (modelos) e nem as ferramentas se desenvolveram o suficiente para concretizar todas as suas promessas feitas.
 - Nível de maturidade insuficiente para se ter um produto final de software com qualidade plena e com 100% do código gerado automaticamente a partir dos modelos

Slide 21/21