

# 1 Introdução

## 1.1. Desenvolvimento de Software

O desenvolvimento de software tradicional não tem como foco a adaptação de possíveis mudanças provenientes da evolução dos requisitos, aumentando assim o custo das futuras modificações. Devido a essa necessidade, algumas metodologias de desenvolvimento de software, passaram a ter como foco principal o suporte a essas inevitáveis mudanças.

A idéia principal é trabalhar com um nível de abstração mais alto, sem o comprometimento com tecnologias específicas de implementação, onde somente os aspectos do domínio de aplicação estejam mapeados. Assim sendo, caso haja uma mudança de tecnologia, o mapeamento não será afetado.

Posteriormente esse mapeamento será o ponto de partida para o desenvolvimento do sistema, podendo ser adaptado a ele diferentes tipos de tecnologias e caso uma delas necessite ser alterada, não haverá necessidade de alterações em larga escala.

## 1.2. Generative Programming

Generative Programming [Czarnecki04] é um paradigma de desenvolvimento de software baseado na modelagem de famílias de sistemas de software tal que, dada uma especificação particular de requisitos, um produto final ou intermediário altamente adaptado e otimizado, pode ser automaticamente manufaturado sob demanda a partir de componentes de implementação elementares e reutilizáveis através de um conhecimento de configuração [Czarnecki04].

Seu foco está na construção de modelos de alto nível que serão a base para a geração de sistemas concretos baseados nesses modelos. Os componentes implementados e o conhecimento de configuração estão armazenados e são sempre atualizados, para que possam ser posteriormente reutilizados. Com isso os custos vão sendo reduzidos a cada geração. A figura 1, mostra alguns exemplos de Generative Programming.

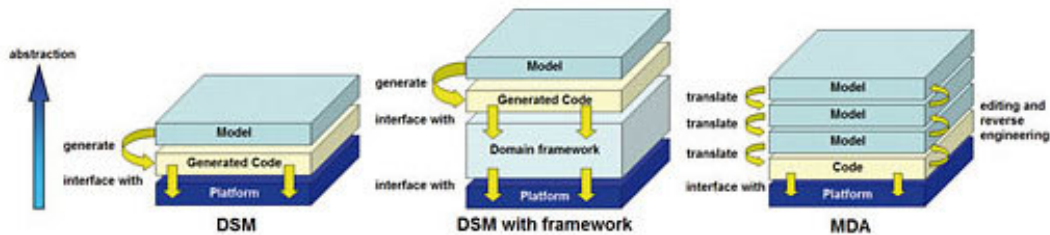


Figura 1 - Exemplos de Generative Programming [Iseger05]

### 1.3.

#### Desenvolvimento de Software Utilizando MDA (Model Driven Architecture)

O Model Driven Architecture (MDA) [MDA] tem como objetivo principal separar o negócio e a lógica da aplicação da parte tecnológica. Assim, seus proponentes esperam que mudanças no software básico e no hardware não façam a aplicação empresarial tornar-se obsoleta antes do tempo. Mais importante, através do desacoplamento, o uso do MDA pode resultar em melhores projetos para aplicações dando uma vida útil mais longa e portabilidade a outras plataformas.

As principais etapas do MDA são: a criação de um modelo independentemente de plataforma (PIM), a “configuração” do modelo através de marcações e o novo modelo gerado baseado em uma plataforma escolhida pelo usuário (PSM). A figura 2 ilustra os passos acima.

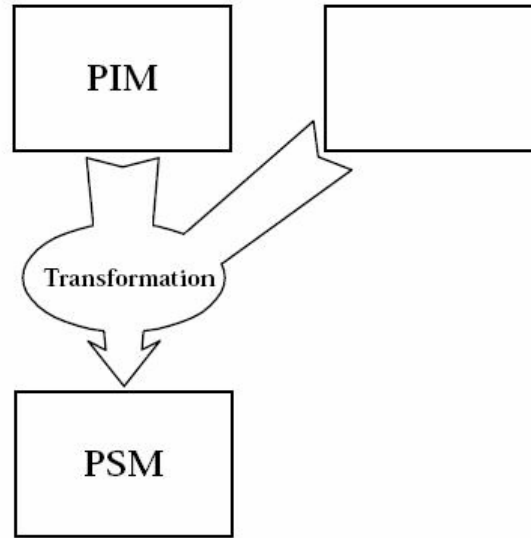


Figura 2 - Visão geral do Model Driven Architecture [MDA]

O processo proposto nesta dissertação é baseado no Model Driven Architecture, logo todo o desenvolvimento ocorre de acordo com as etapas do MDA, podendo haver algumas mudanças provenientes de alguma funcionalidade a mais.

#### 1.4.

#### **O Objetivo da Dissertação e o Resumo da Solução Proposta**

O objetivo da dissertação é a criação de um processo que auxilie o desenvolvimento de aplicações utilizando MDA, para que mudanças de tecnologias não acarretem no reprojeto dos sistemas. Será proposto algumas etapas e uma ferramenta de apoio, para que seja possível adaptar essas mudanças aos projetos.

O foco do processo está no desenvolvedor criar um modelo independente de plataforma (PIM) e através das etapas do processo e da ferramenta de apoio, configurar esse modelo para que seja possível adaptar ao mesmo diversas tecnologias, resultando em um ou mais modelos específicos para uma plataforma (PSM).

Existe uma base de conhecimento que acompanha todo o processo fornecendo informações ao desenvolvedor. Caso alguma informação necessite

ser adicionada ou alterada, ela poderá ser visualizada nas próximas utilizações do processo.

### **1.5.**

#### **Lista das Contribuições da Dissertação**

- Criação de um processo para transformações de modelos, utilizando a abordagem MDA, o conceito de base de conhecimento e a linguagem RDL para descrever as regras de transformação.
- O ponto chave do processo é apoiar nas transformações de modelos PIM (modelo independente de plataforma) em modelos PSM (modelo específico para uma plataforma), ou seja, adaptar tecnologias ao PIM para gerar modelos PSM de acordo com cada necessidade específica.
- Utilização do conceito de base de conhecimento, com o intuito de manter atualizada as informações necessárias para a transformação dos modelos. Conforme o processo é utilizado e configurado, mais informações a base conterà.
- Utilização da linguagem RDL para armazenar as regras de transformações. Essa linguagem se encaixa muito bem nessa função, devido ao fato de possuir regras bem definidas e que estão diretamente ligadas a transformações em modelos.
- Extensão da linguagem RDL. Foi necessário estender a linguagem RDL para poder tratar algumas alterações nos modelos que a mesma não tratava.
- Criação de uma ferramenta de apoio de todo o processo de transformação, desde a manutenção da base de conhecimento até a geração do modelo final.

## **1.6. Estrutura da Dissertação**

- O capítulo 02 apresenta alguns trabalhos relacionados à dissertação.
- O capítulo 03 explica de forma resumida as tecnologias envolvidas no processo.
- O capítulo 04 descreve passo a passo o processo.
- O capítulo 05 mostra a ferramenta de apoio ao processo.
- O capítulo 06 mostra dois casos de estudo, onde é possível verificar se o resultado do processo é compatível com o resultado de cada caso de estudo.
- O capítulo 07 é composto pelas considerações finais da dissertação.
- O capítulo 08 é formado pelas bibliografias da dissertação.