

Em Direção ao Desenvolvimento de Programas Adaptativos Utilizando MDE

Sergio Roberto de Mello Canovas
Carlos Eduardo Cugnasca
WTA 2015

Agenda

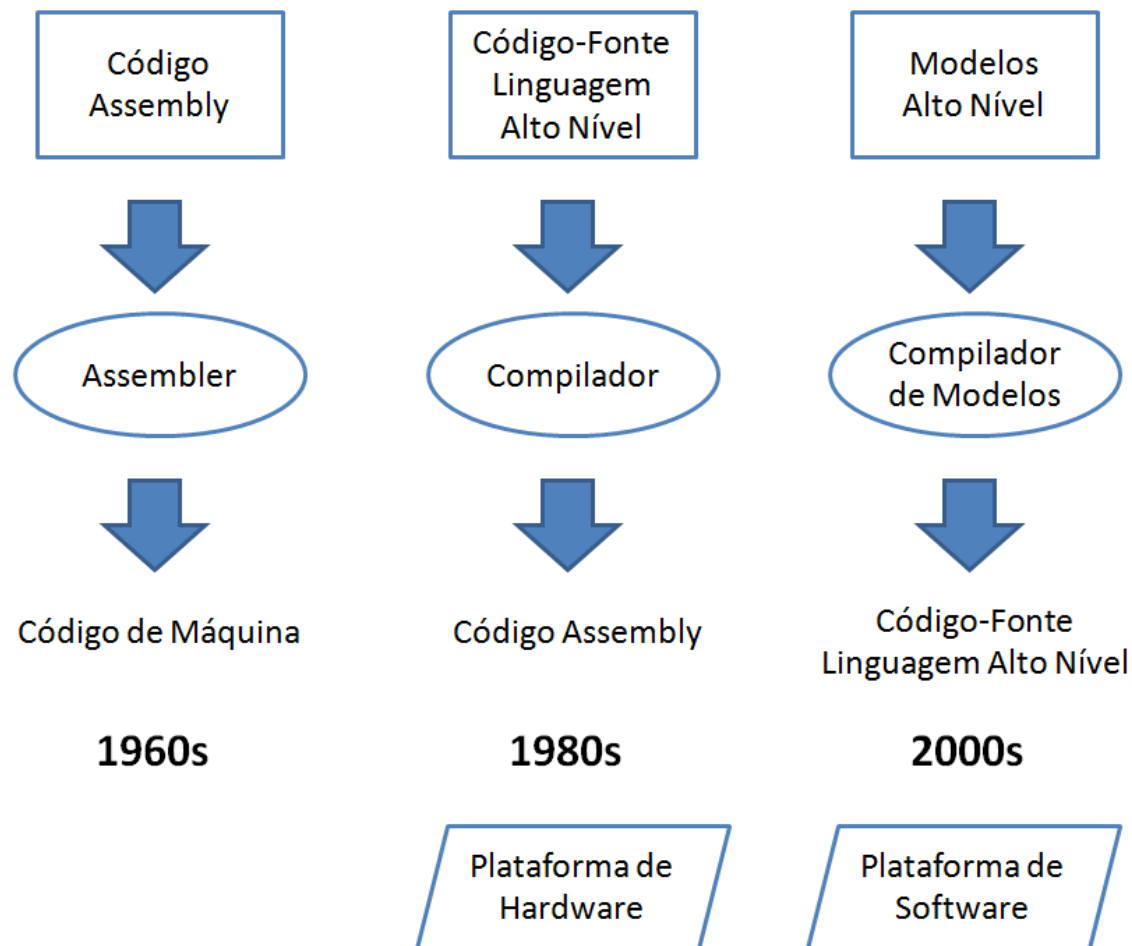
- ▶ Introdução
 - Motivação;
 - MDE;
 - Programas Adaptativos.
- ▶ SBMM;
- ▶ Metamodelo para Programas Adaptativos;
- ▶ Ferramenta CASE para Programas Adaptativos;
- ▶ Resultados e Conclusão.

Introdução

Motivação

- ▶ **Motivação 1: Aumento do nível de abstração.**
- ▶ Por quê? Projetos de software cada vez mais complexos requerem maior eficiência e menor prazo de desenvolvimento e manutenção. Isso reflete em custos menores para as organizações.

Motivação



Motivação

- ▶ **Motivação 2: Compatibilidade com múltiplas plataformas.**
- ▶ **Por quê? Cenário atual sem precedentes.**

Motivação

A Evolução do Cenário “Client”

1999



Windows

2005



Windows & Web

Fonte: Embarcadero (2013)

Motivação

2013: A Revolução dos Dispositivos “Client”

1 Billion



Windows

65 Million



Mac

1 Billion



Mobile

2 Billion



Web

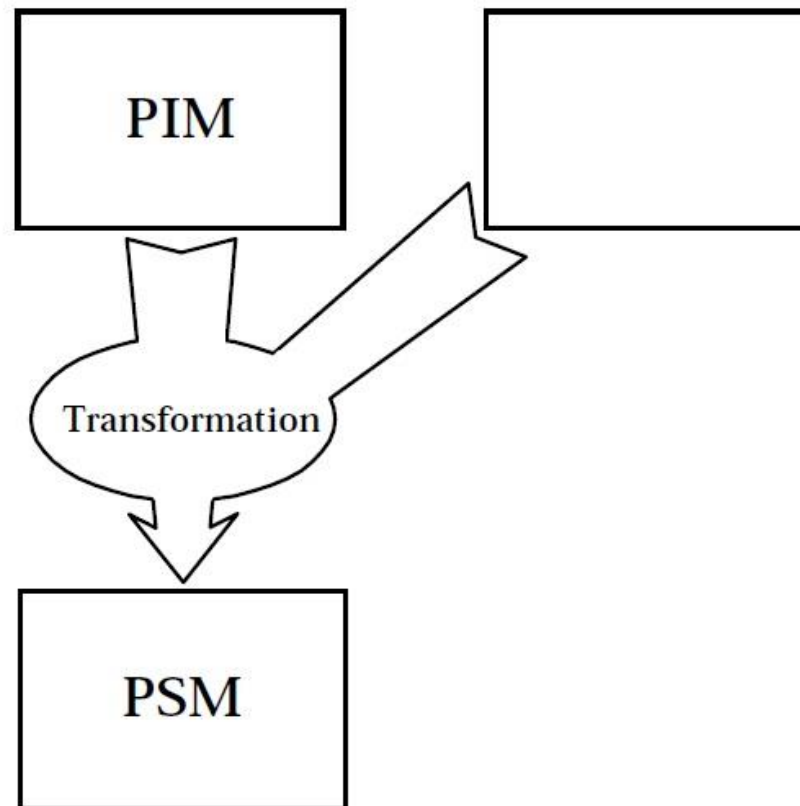
Vivemos um cenário **Multi-Device** sem precedentes...

Fonte: Embarcadero (2013)

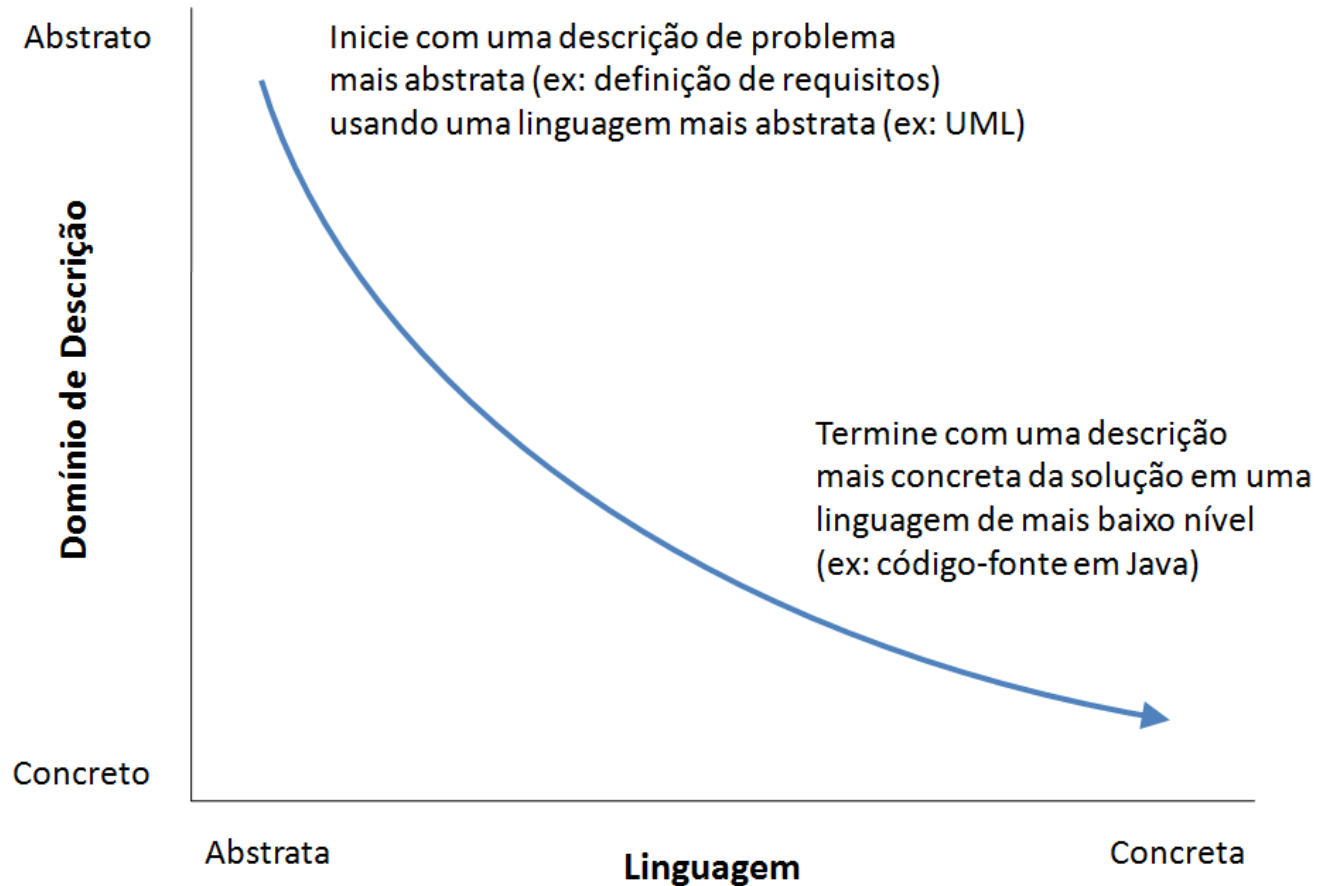
MDE

- ▶ **Model Driven Architecture (MDA):** marca da OMG, referencia tecnologias da OMG: MOF, UML, etc.;
- ▶ **Model Driven Engineering (MDE):** conceito geral.
- ▶ **Definição:** Um modelo de sistema de software é um modelo que descreve ou especifica o sistema, podendo considerar aspectos de seu ambiente. É frequentemente apresentado como uma combinação de desenhos e texto. O texto pode estar expresso em uma linguagem de modelagem ou em linguagem natural (OMG, 2003).

MDE



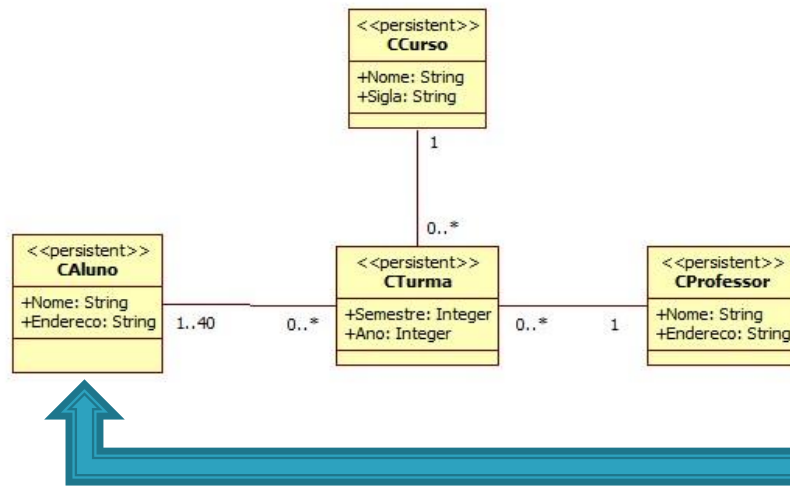
MDE



Mellor et al. (2004)

MDE

► Transformação Manual de Modelos

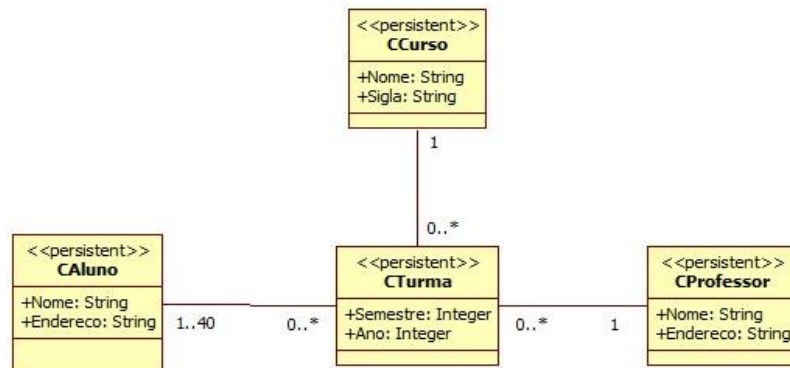


```
<html>
<head>
  <title>Cadastro do Aluno</title>
</head>
<body>
  <h1>Cadastro do Aluno</h1>
  <br /><br />
  <form name="CadastroAluno">
    <table border="0">
      <tr>
        <td>Nome:</td>
        <td><input type="text" /></td>
      </tr>
      <tr>
        <td>Endereco:</td>
        <td><input type="text" /></td>
      </tr>
    </table>
    <br />
    <input type="submit" />
  </form>
</body>
</html>
```

Problema: Sincronização de modelos executada de forma manual é suscetível a erros e pouco eficiente.

MDE

► Transformação Automática de Modelos



```
<html>
<head>
  <title>Cadastro do Aluno</title>
</head>
<body>
  <h1>Cadastro do Aluno</h1>
  <br /><br />
  <form name="CadastroAluno">
    <table border="0">
      <tr>
        <td>Nome:</td>
        <td><input type="text" /></td>
      </tr>
      <tr>
        <td>Endereco:</td>
        <td><input type="text" /></td>
      </tr>
    </table>
    <br />
    <input type="submit" />
  </form>
</body>
</html>
```

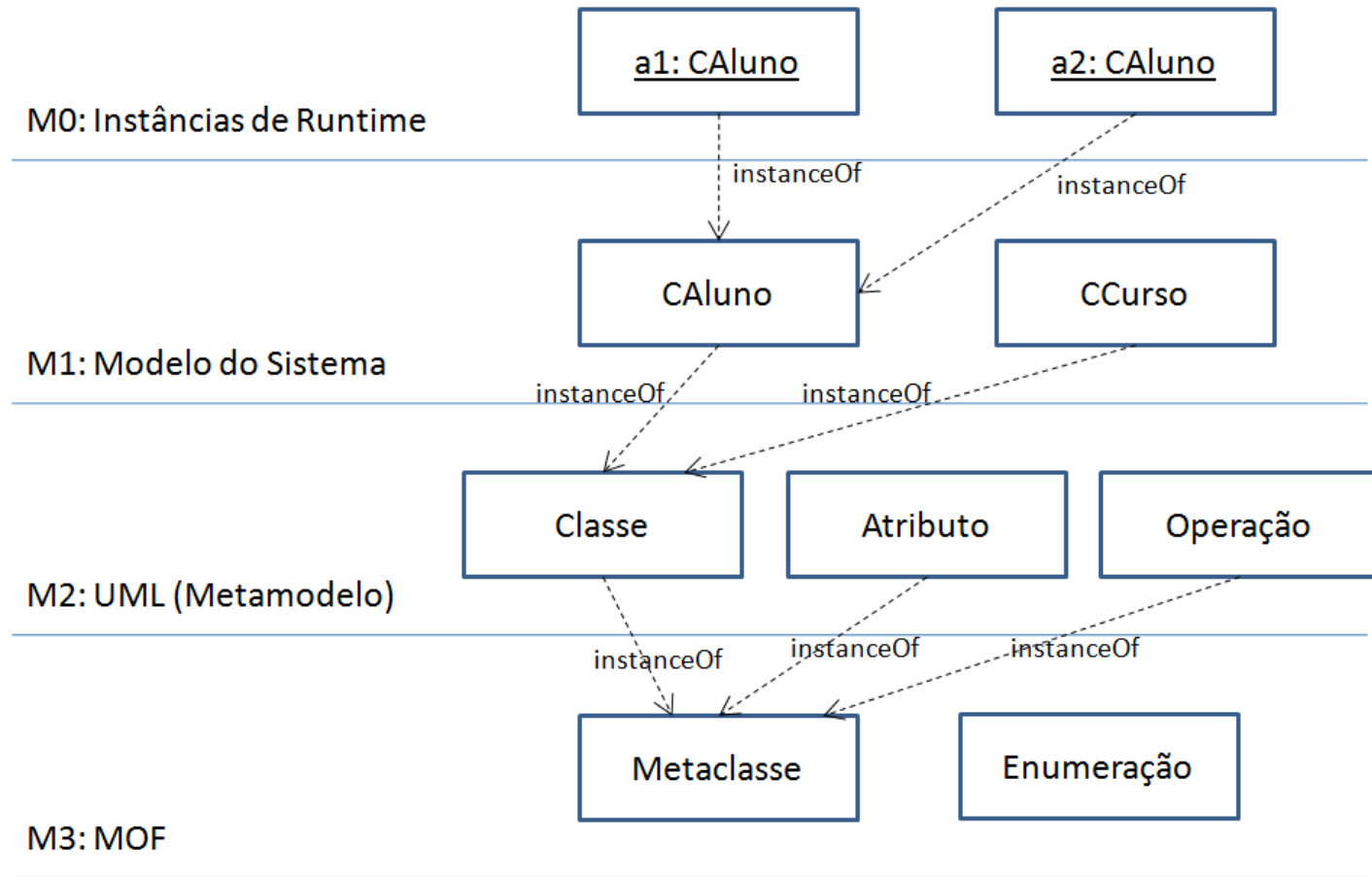
Função de mapeamento

Solução: Sincronização de modelos ocorre de forma automática, rápida e não está sujeita a erros do programador.

MDE

- ▶ Independentemente da abordagem de definição de funções de mapeamento, **modelos precisam estar escritos de forma que possam ser lidos e decodificados por máquina (*machine readable*)**;
- ▶ **Modelos devem ser escritos de acordo com um metamodelo**, que captura formalmente o que é possível existir no modelo e sua sintaxe (mesmo que gráfica);

MDE



MDE

- ▶ Meta Object Facility (MOF) é a tecnologia da OMG para definir metamodelos;
- ▶ Definido por conjunto de 7 especificações da OMG;
- ▶ Exemplo: metamodelos para a UML

MDE

- ▶ MOF importa definições da UML para ser definido;
- ▶ Definição do MOF *standalone* utiliza ele mesmo para se auto-definir;
- ▶ Divisão em EMOF e CMOF. EMOF surgiu para estimular o surgimento de ferramentas de metamodelagem;

Programas Adaptativos

- ▶ Dispositivo adaptativo: dispositivo não-adaptativo subjacente mais um mecanismo formado por funções adaptativas capaz de alterar o conjunto de regras que define o comportamento do dispositivo.
- ▶ Programa adaptativo: dispositivo adaptativo onde o dispositivo não-adaptativo subjacente é um programa de computador estático (Silva, 2010).

Programas Adaptativos

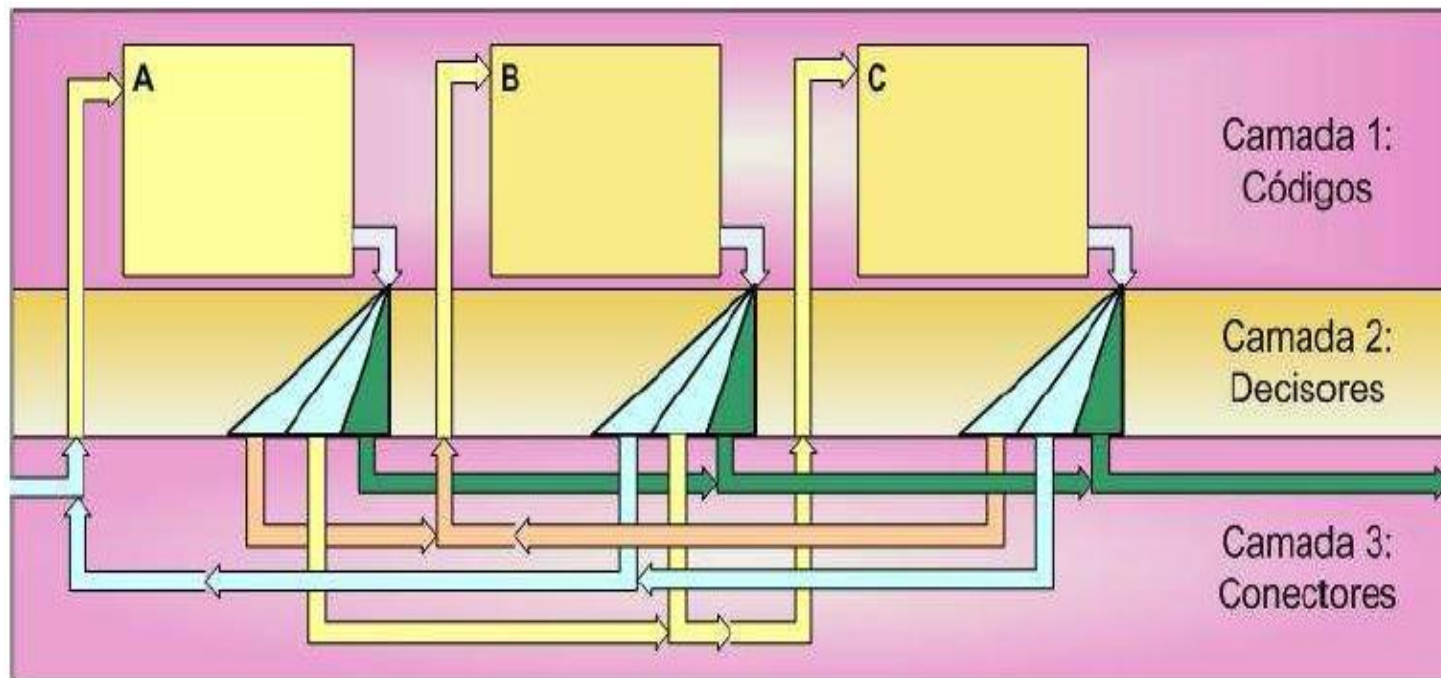
- ▶ As ações adaptativas podem inserir ou remover linhas de código, antes ou depois de processar um estímulo;
- ▶ BADAL (*Basic Adaptive Language*) é uma linguagem de programação adaptativa de alto nível proposta por Silva (2010). Provê instruções explícitas para alteração do código-fonte em tempo de execução.

Programas Adaptativos

- ▶ Também é apresentada uma representação gráfica para programas adaptativos descrita em linguagem natural;
- ▶ Dispositivo não-adaptativo subjacente: programa estático;
- ▶ Camada 1: Blocos de código;
- ▶ Camada 2: Decisores;
- ▶ Camada 3: Conectores.

Programas Adaptativos

- ▶ Programa estático:



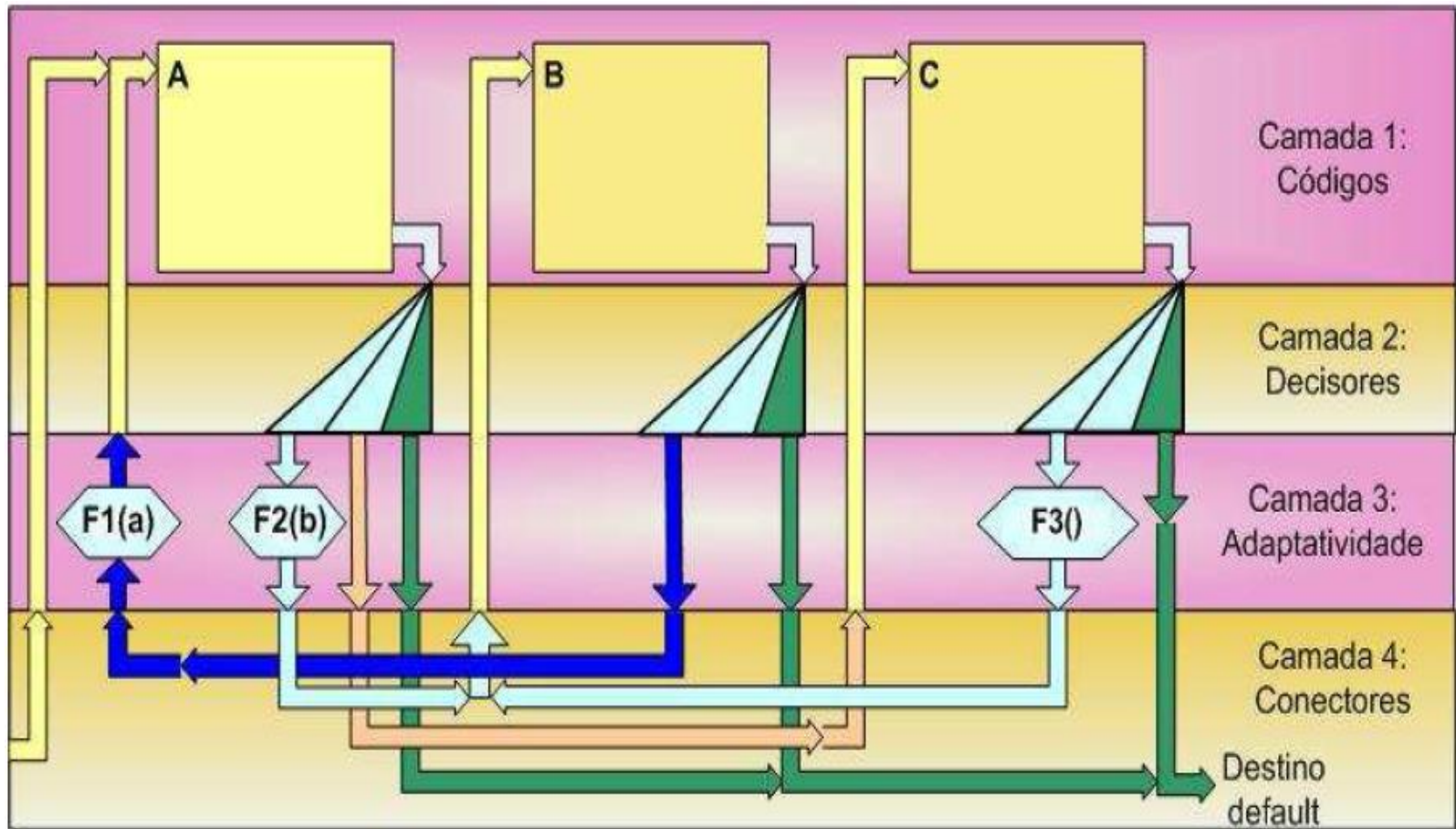
Programas Adaptativos

- ▶ Para assegurar a coerência estrutural dos programas assim construídos, é preciso que o programador projete adequadamente as conexões e decisores, e que o compilador faça as validações necessárias;
- ▶ Uma entrada de bloco básico pode receber mais de uma conexão. Por outro lado, cada valor de saída de um bloco básico deve estar associado a uma única conexão;

Programas Adaptativos

- ▶ A camada adaptativa é introduzida entre a camada de decisores e de conectores. Ela é responsável pela capacidade de alteração do programa em tempo de execução.
- ▶ A declaração de uma função adaptativa resume-se a indicar as ações de modificação do programa adaptativo, a serem efetuadas em tempo de execução nas ocasiões em que a função for ativada.

Programas Adaptativos



Programas Adaptativos

- ▶ Cada instância dessa representação gráfica é um modelo, não necessariamente completo, para um programa adaptativo, assim como um diagrama de classes UML é um modelo, também não necessariamente completo, para um programa orientado a objetos.

Programas Adaptativos

- ▶ Um metamodelo definido formalmente determina os modelos possíveis de programas adaptativos;
- ▶ Além disso, o metamodelo abre caminho para descrever funções de mapeamento e utilizar MDE para o desenvolvimento de programas adaptativos.

SBMM

SBMM

- ▶ SBMM é um formalismo para descrever metamodelos (camada M3).
- ▶ $MM = (n, C, \Gamma, E, R)$
 - n é o nome do metamodelo;
 - C é um conjunto de metaclasses;
 - $\Gamma \subset C \times C$ é uma relação de generalização;
 - E é um conjunto de enumerações.
 - R é um conjunto de restrições.

SBMM

- ▶ $C = \{c_1, c_2, \dots c_n\}$
- ▶ $c_i = (w_i, P_i)$ onde:
 - w_i é o nome da metaclasses;
 - P_i é o conjunto de propriedades da metaclasses.
- ▶ $P_i = \{p_{i1}, p_{i2}, \dots p_{im}\}$

SBMM

- ▶ $p_{ij} = (v_{ij}, t_{ij}, m_{ij})$ onde:
 - v_{ij} é o nome da propriedade;
 - $t_{ij} \in C \cup E$ é o tipo alvo;
 - $m_{ij} \in \mathbf{N} \times (\mathbf{N}_+ \cup \{*\})$. Exemplo: $(0,*)$ ou $0..*$
- ▶ $E = \{e_1, e_2, \dots e_m\}$
- ▶ $e_i = (u_i, L_i)$ onde:
 - u_i é o nome da enumeração;
 - L_i é o conjunto de valores permitidos;

SBMM

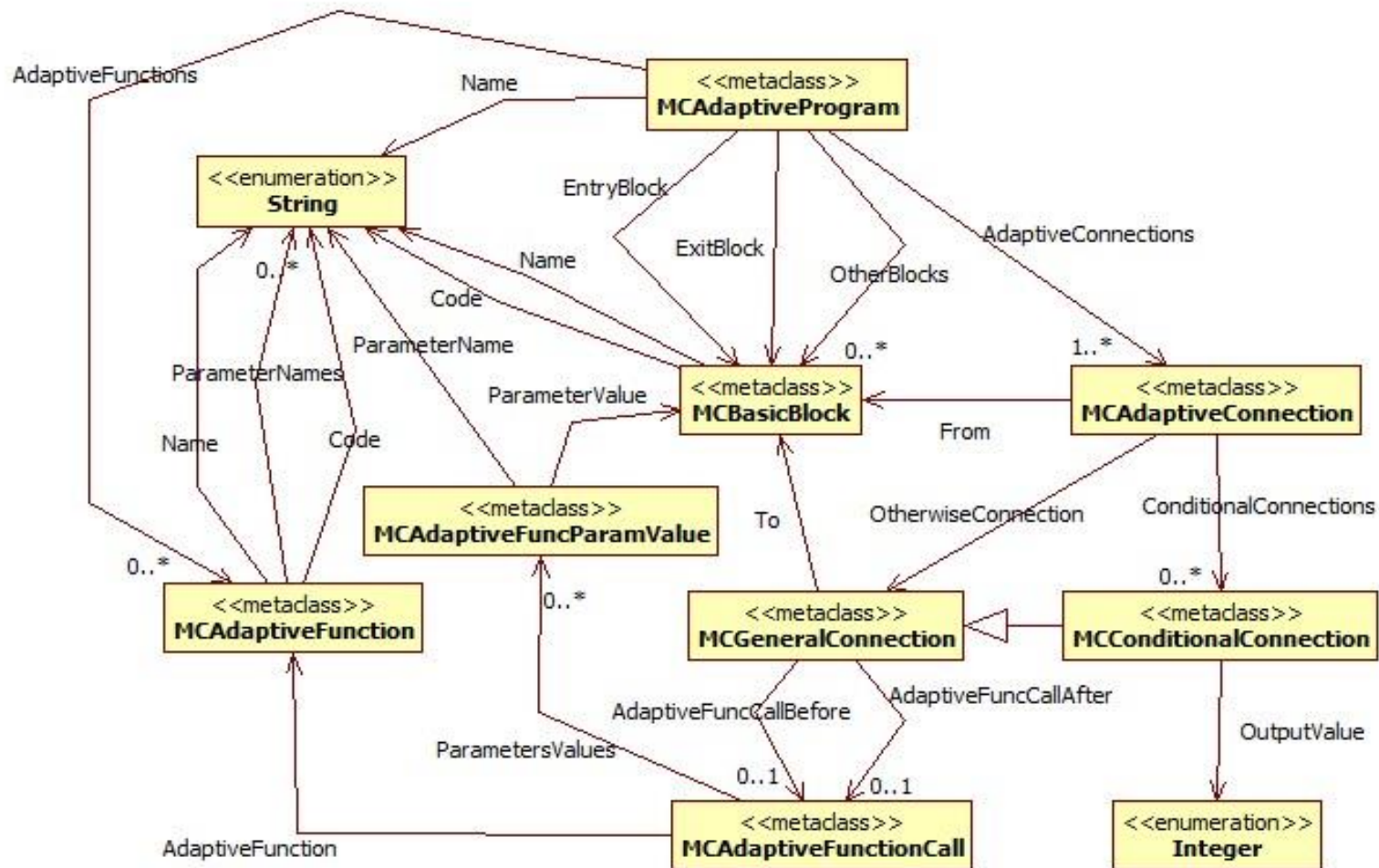
- ▶ $R = \{r_1, r_2, \dots r_n\}$ onde:
- ▶ r_i é uma sentença em lógica de primeira ordem que representa uma restrição no metamodelo.

SBMM

- ▶ Metaclasses e enumerações são predicados unários a serem aplicados sobre instâncias de metaclasses, ou seja, sobre elementos do modelo;
- ▶ Propriedades são funções a serem aplicadas sobre instâncias de metaclasses. Se a propriedade possui multiplicidade maior que 1, a função retorna um conjunto cujos elementos são os valores.

Metamodelo para Programas Adaptativos

Metamodelo para Programas Adaptativos



Metamodelo para Programas Adaptativos

- ▶ Metamodelo descrito em SBMM:
 - 8 metaclasses;
 - 1 generalização;
 - 2 enumerações; e
 - 13 restrições.

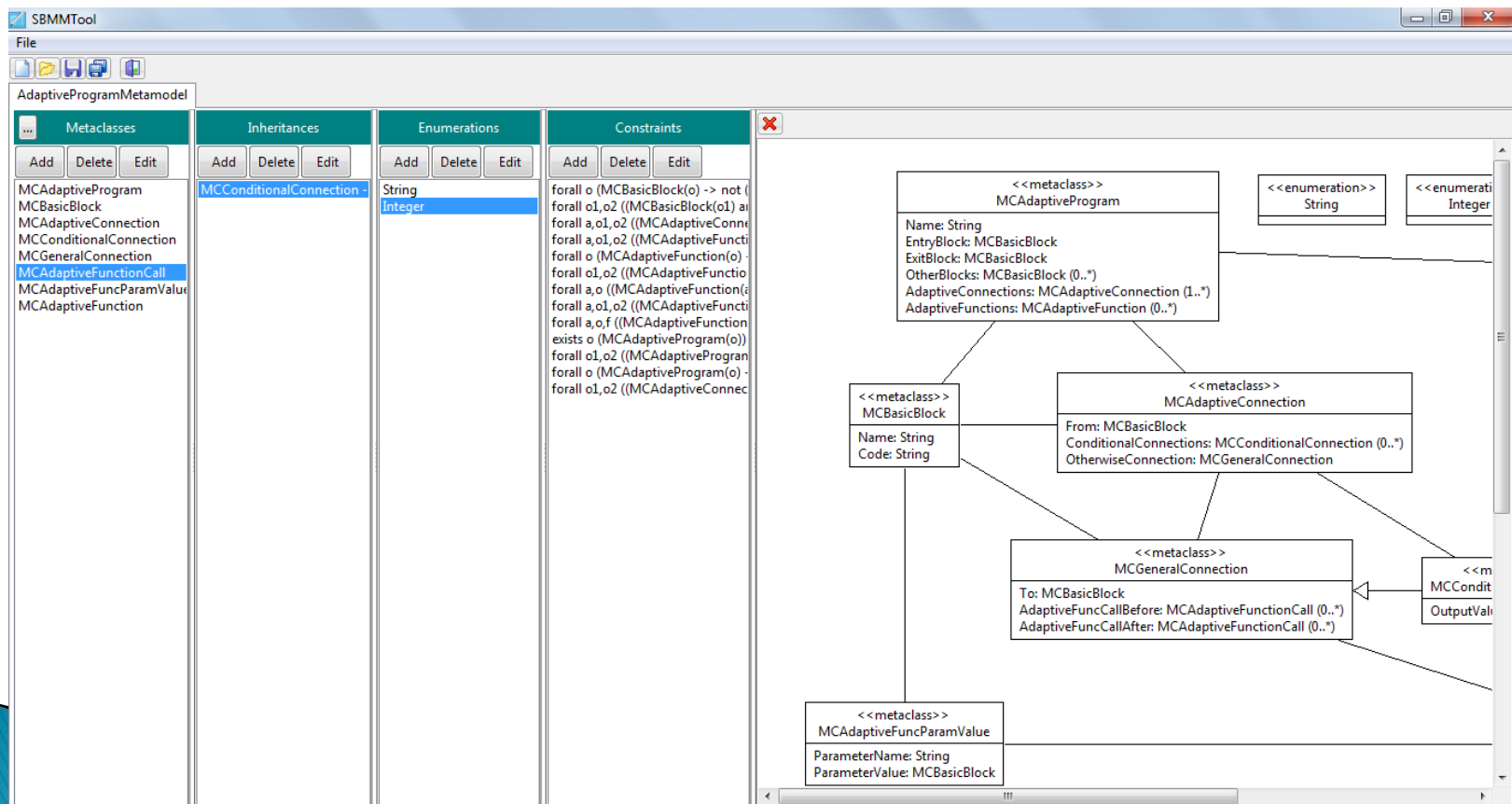
Metamodelo para Programas Adaptativos

- ▶ Possibilita a criação de modelos de programas adaptativos, que são instâncias do metamodelo (ferramentas CASE e meta-CASE para programas adaptativos);
- ▶ Possibilita escrever funções de mapeamento para conversão dos modelos em código-fonte de programa adaptativo;
- ▶ Abre caminho para a MDE, trazendo seus benefícios para a programação adaptativa.

Ferramenta CASE para Programas Adaptativos

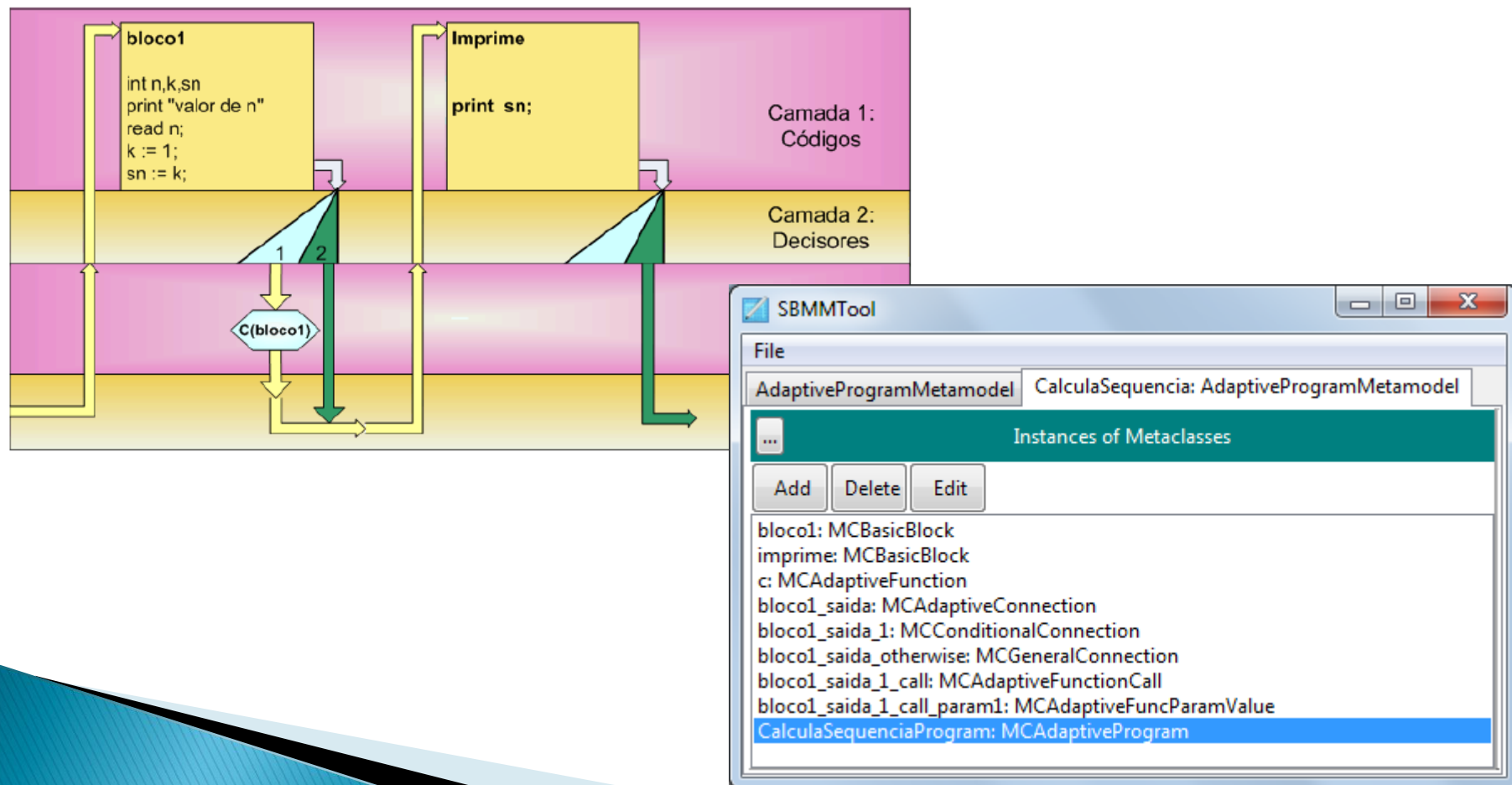
Ferramenta CASE para Programas Adaptativos

- ▶ Metamodelo inserido na ferramenta SBMMTool:



Ferramenta CASE para Programas Adaptativos

- ▶ Modelo do aplicativo exemplo (Silva, 2010):



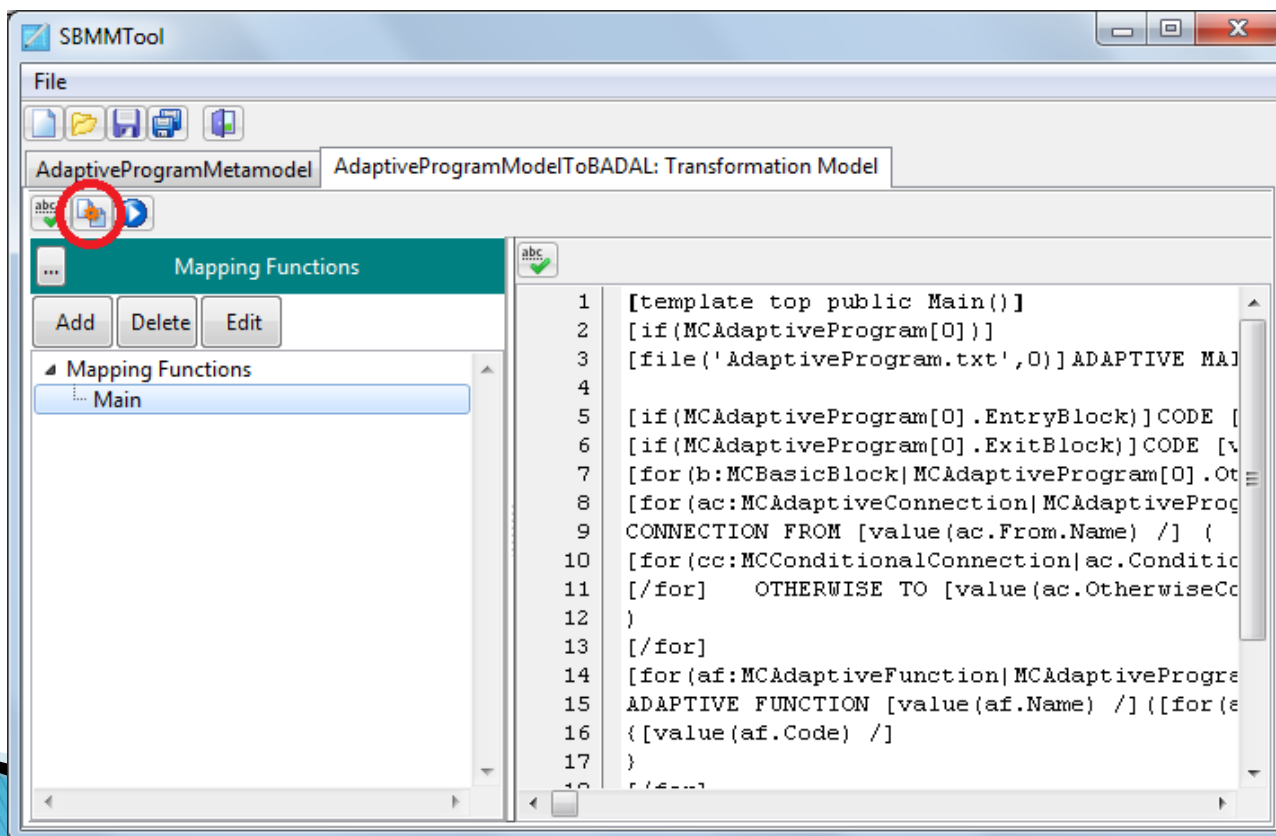
Ferramenta CASE para Programas Adaptativos

- ▶ Função de mapeamento para linguagem BADAL descrita em MOFM2T:

```
[template top public Main()]  
[file(MCAdaptiveProgram[0].Name+'.txt',0)]ADAPTIVE MAIN \[NAME =  
    [value(MCAdaptiveProgram[0].Name) /], ENTRY =  
    [value(MCAdaptiveProgram[0].EntryBlock.Name) /], EXIT =  
    [value(MCAdaptiveProgram[0].ExitBlock.Name) /] ] IS  
  
[for(b:MCBasicBlock|MCAdaptiveProgram[0].EntryBlock)]CODE [value(b.Name) /]:  
    <[value(b.Code) /]>[/for]  
  
[...]  
END MAIN.  
[/file]  
[/template]
```

Ferramenta CASE para Programas Adaptativos

- ▶ Transformação automática de modelo em código:



Resultados e Conclusão

Resultados e Conclusão

- ▶ Ferramenta meta-CASE com metamodelo carregado: ferramenta CASE;
- ▶ Modelagem de programas adaptativos;
- ▶ Geração automática de código BADAL, mesmo que parcial;

Resultados e Conclusão

- ▶ Possibilidades futuras:
 - Editor gráfico de modelos com notação específica;
 - Extensão do metamodelo proposto para considerar operações adaptativas sem requerer inserção de código posterior pelo programador.
- ▶ Aplicação da abordagem MDE, em um nível inicial, para desenvolvimento de programas adaptativos.

Fim