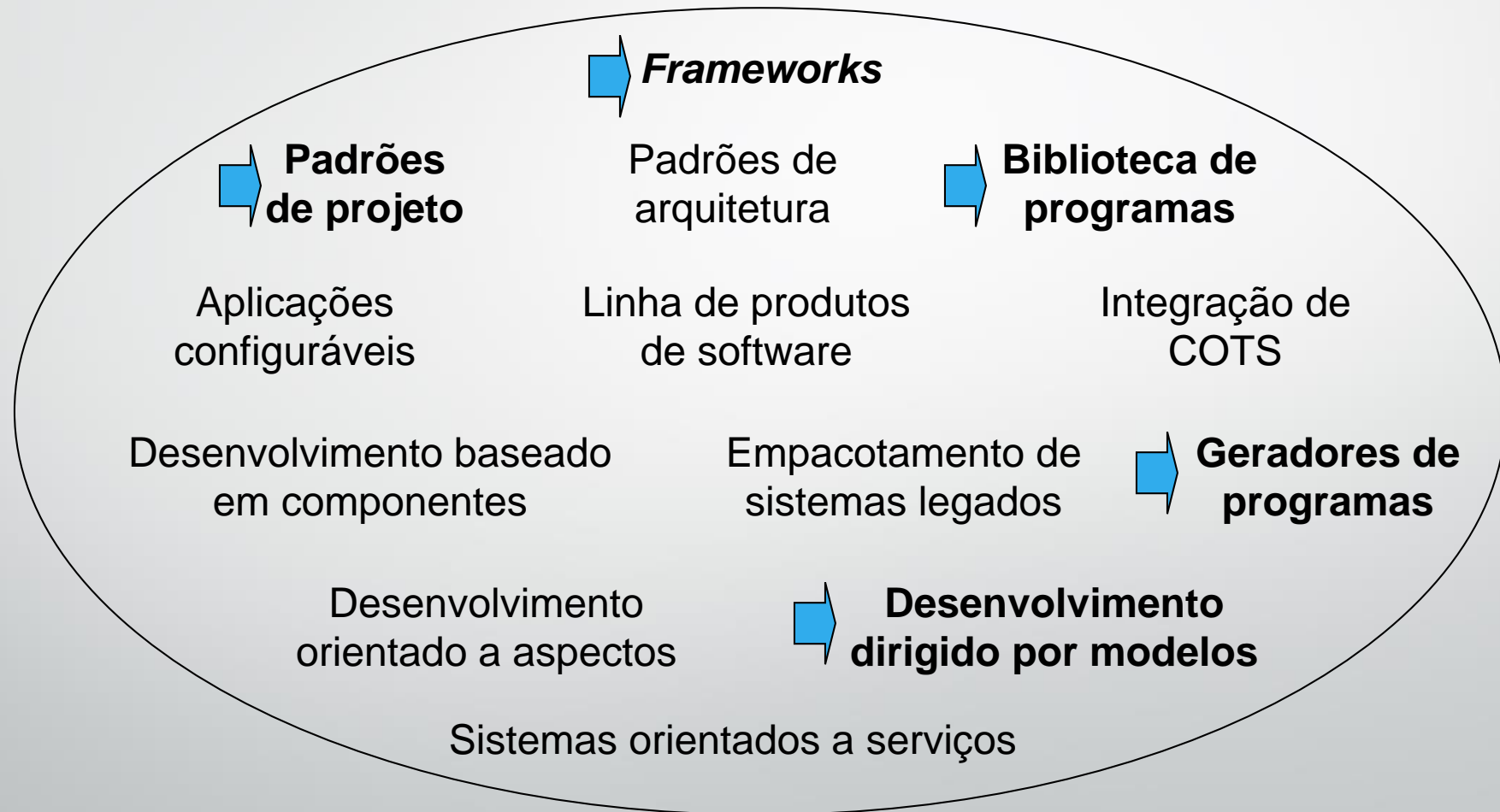


Técnicas para Reutilização de Software

Prof. Eduardo Figueiredo

Estagiário: Johnatan Oliveira

Panorama de Reutilização





Bibliotecas de Software

Bibliotecas de Software

Disponibiliza funcionalidades comuns a diferentes tipos de sistemas

- Converter informação entre formatos conhecidos (e.g., string para inteiro)
- Acesso a recursos, arquivos, BD, etc.
- Tipos abstratos de dados: fila, pilha, lista...

Biblioteca em Java

```
public class Dado {  
  
    public Dado(){ }  
  
    public int jogarDado() {  
        Random r = new Random();  
        int valor = r.nextInt(6)+1;  
        return valor;  
    }  
}
```




Framework

Framework

É uma estrutura genérica que deve ser estendida para criar uma aplicação específica.

Um conjunto integrado de artefatos de software (como classes, objetos e componentes) que colaboram para fornecer uma arquitetura reusável



O principal problema com frameworks é a sua complexidade e o tempo que leva para aprender a utiliza-los.

Exemplos: PrimeFaces (Java), JSF (Java), JQuery (JS), Junit(Java), Maven (...)

Tipos de Framework

- **Frameworks de infra-estrutura:** simplificam o desenvolvimento da infraestrutura portátil e eficiente do sistema
- **Frameworks de integração:** geralmente para integrar aplicações e componentes distribuídos
- **Frameworks de aplicações:** apoiam o desenvolvimento de um tipo de aplicações

Extensão de Frameworks

- Frameworks são entidades grandes que devem ser estendidas para reutilização
- **Exemplos de extensão**
 - Adição de classes concretas que implementam métodos abstratos
 - Adição (sobrescrita) de métodos que implementam comportamento padrão
 - Adição de arquivos de configuração (XML)



Padrão de Projeto

Padrão de Projeto

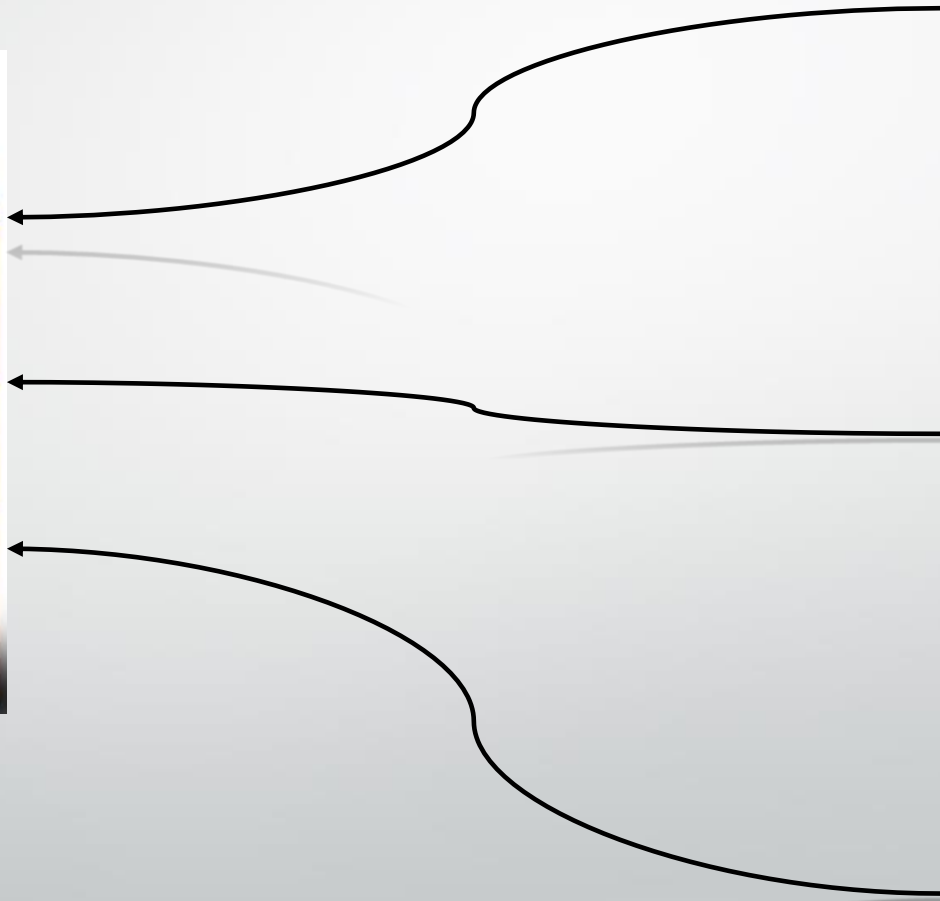
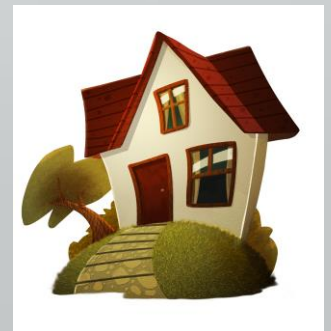
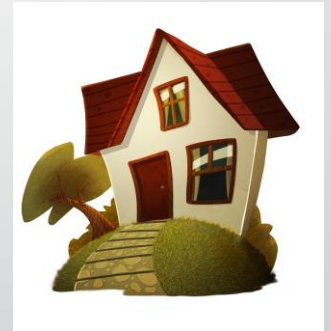
- **Um padrão de projeto** é uma forma de reutilizar o conhecimento abstrato sobre um problema e sua solução
- **Documenta boas soluções para problemas recorrentes:** reutilização de conhecimento anterior documentados em boas práticas
- A descrição deve ser suficientemente abstrata de modo que possa ser reutilizada em diferentes casos

Elementos essenciais dos padrões

- **Nome:** Um identificador significativo para o padrão
- **Descrição do problema:** reutilização de conhecimento anterior documentados em boas práticas
- **Descrição da solução:** é um template para uma solução que pode ser instanciada de diferentes maneiras
- **Consequências:** os resultados e compromissos de aplicação do padrão

Exemplo Correios

Caixa postal+ Assinatura = Padrão Observer



Exemplo Correios (SOLUÇÃO)

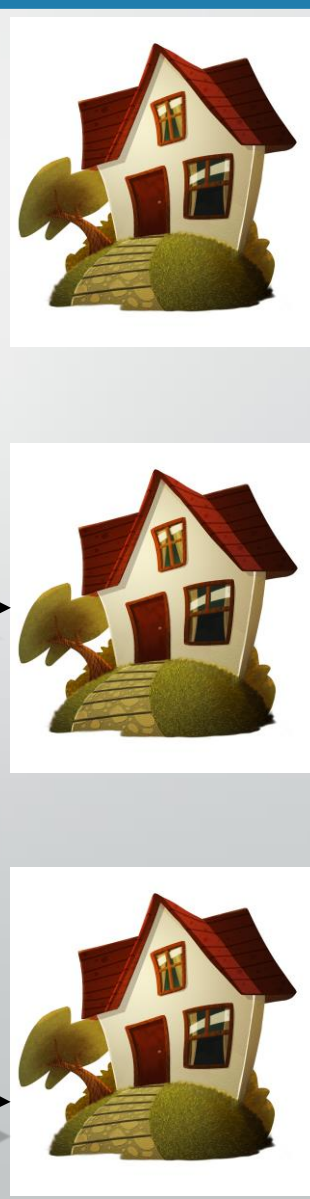
➤ Padrão Observer



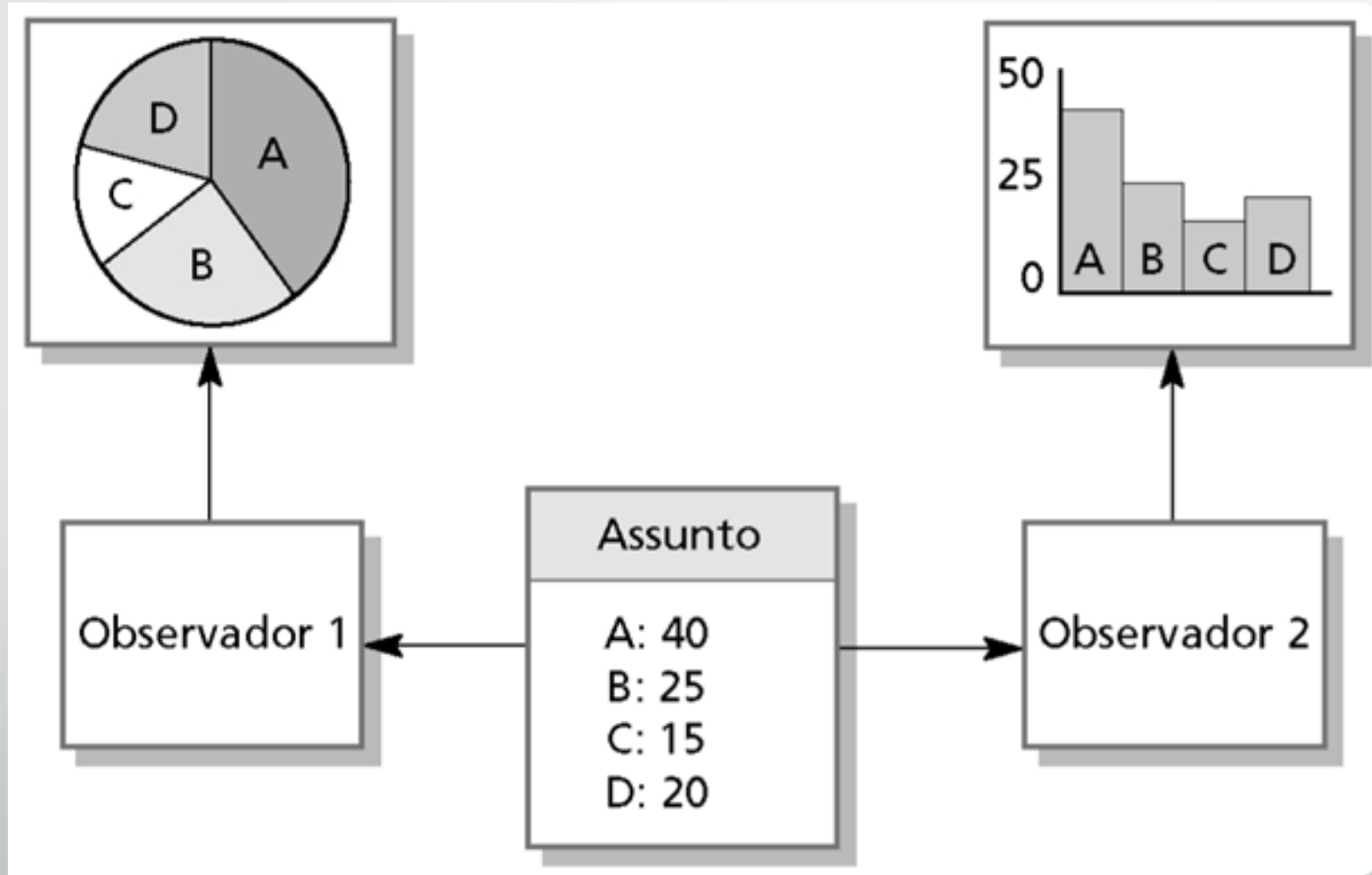
➤ Assunto



➤ OBSERVES

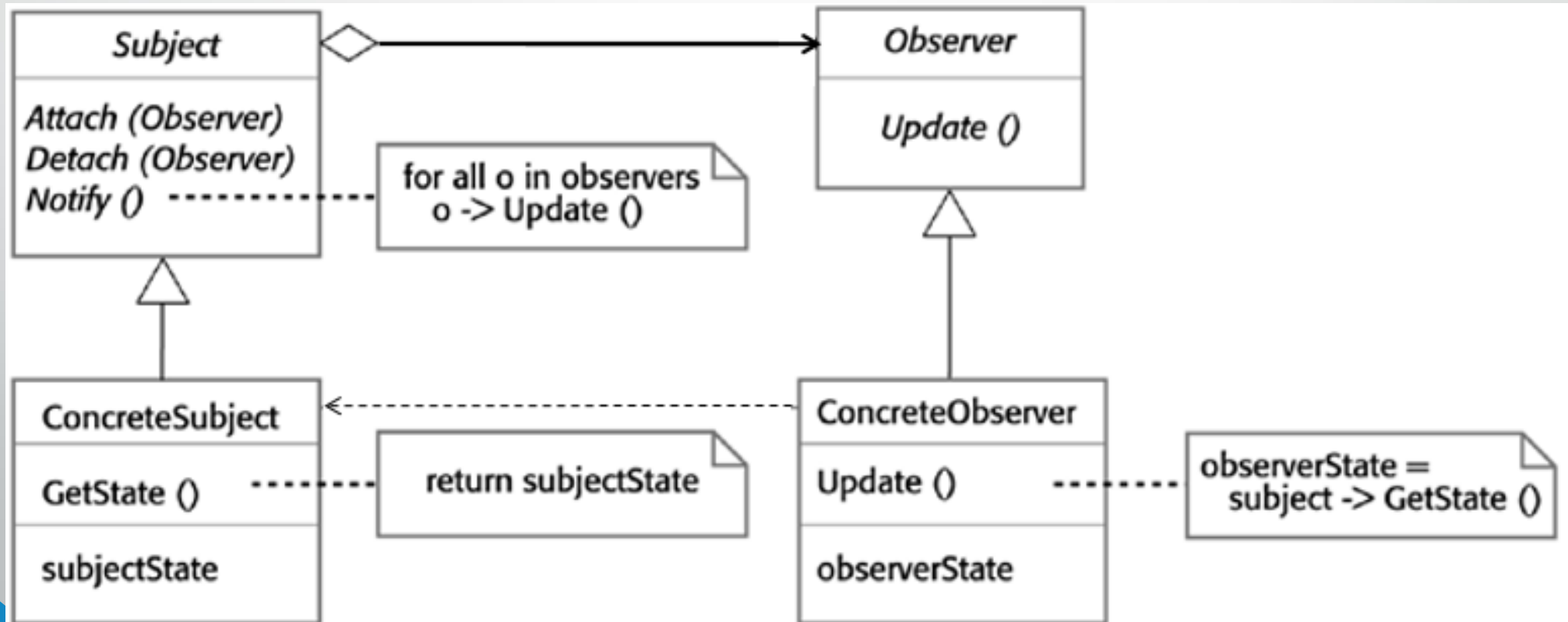


Exemplo de Problema



Padrão Observer

- **Nome:** Observer
- **Descrição do problema:** separa o objeto de suas formas de apresentação
- **Consequência:** atualização da apresentação
- **Descrição da solução:**





Reutilização de Modelos e Geração de Código

Motivação

➤ **A reutilização no nível de código é geralmente difícil :**

Envolve vários detalhes específicos da solução ou tecnologia adotada

Proposta de Solução

- **Elevar o nível de abstração**

 - Reutilização em nível de modelos

 - Reutilização de código (Reutilização de modelos)

- **Uso de geradores:** o programa (código) é automaticamente gerado

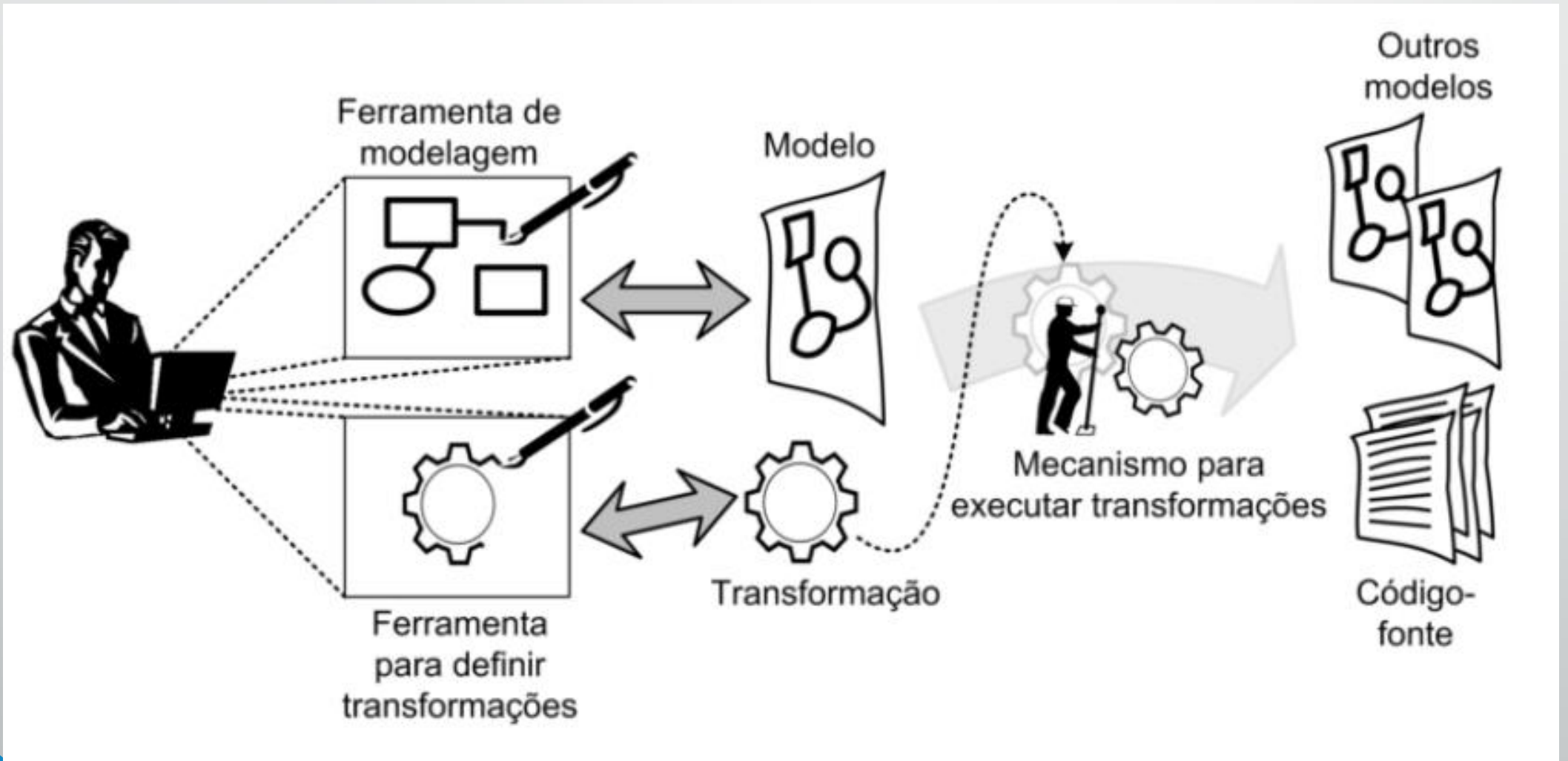
Modelos x Código

- Modelos têm vida útil mais longa
- Modelos facilitam a comunicação entre desenvolvedores (e clientes)
- Modelos são geralmente produzidos, mesmo que não se use um abordagem de geração de código

MDD

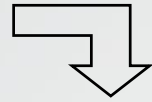
- Desenvolvimento Dirigido por Modelos
- Propõe que o desenvolvimento, reuso, manutenção e evolução sejam feitos no nível de modelagem
- Reutilização de modelos ainda é uma técnica pouco adotada: Se limitam a alguns domínios específicos ou em centros de pesquisa

Abordagem MDD

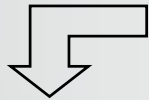


Abordagem MDD

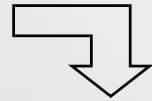
Modelos
Executáveis



Compilador
de Modelos



Código de
Alto Nível



Compilador
de Código

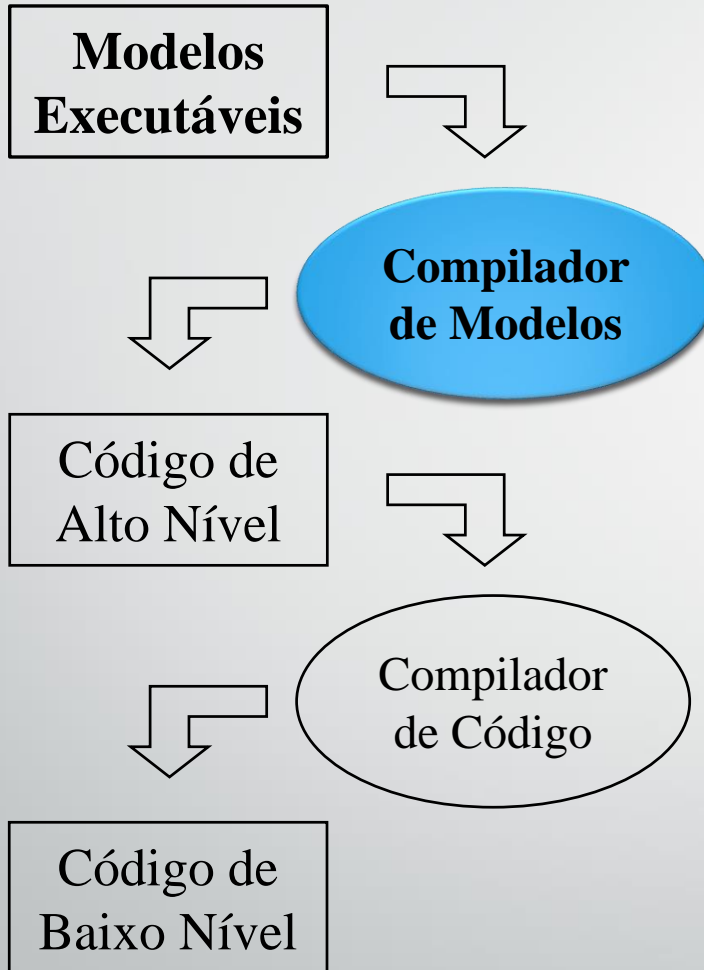
Código de
Baixo Nível

Hoje

➤ **Os modelos são independentes de software**

Assim como, código de alto nível é independente de hardware

Abordagem MDD



➤ **Modelos podem ser compilados para várias linguagens de programação**

Modelos podem ser parcialmente ou totalmente reusados em diferentes contextos

O Processo MDD

- **Selecionar um modelo existente**
- **“Recortar” as partes do modelo que interessam:** pode ser necessário criar novos modelos ou adaptar os modelos existentes
- **Integrar as partes selecionadas ao modelo do sistema**
- **Selecionar uma tecnologia de implementação**
- **Descrever (ou reusar) o mapeamento dos modelos para a implementação**
- **Gerar o sistema**

Potenciais Problemas do MDD

- **Falta de suporte de ferramentas e ambientes de desenvolvimento**
- **Modelos são vistos como extras:** código seria o principal
- **Desenvolvedores são resistentes:** não gostam de “brincar” com figuras . Os programadores temem por seus empregos

Bibliografia

Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.

Cap. 16 Reuso de Software
(Seções 16.1 e 16.2)

Seção 5.5 Engenharia Dirigida
por Modelos

