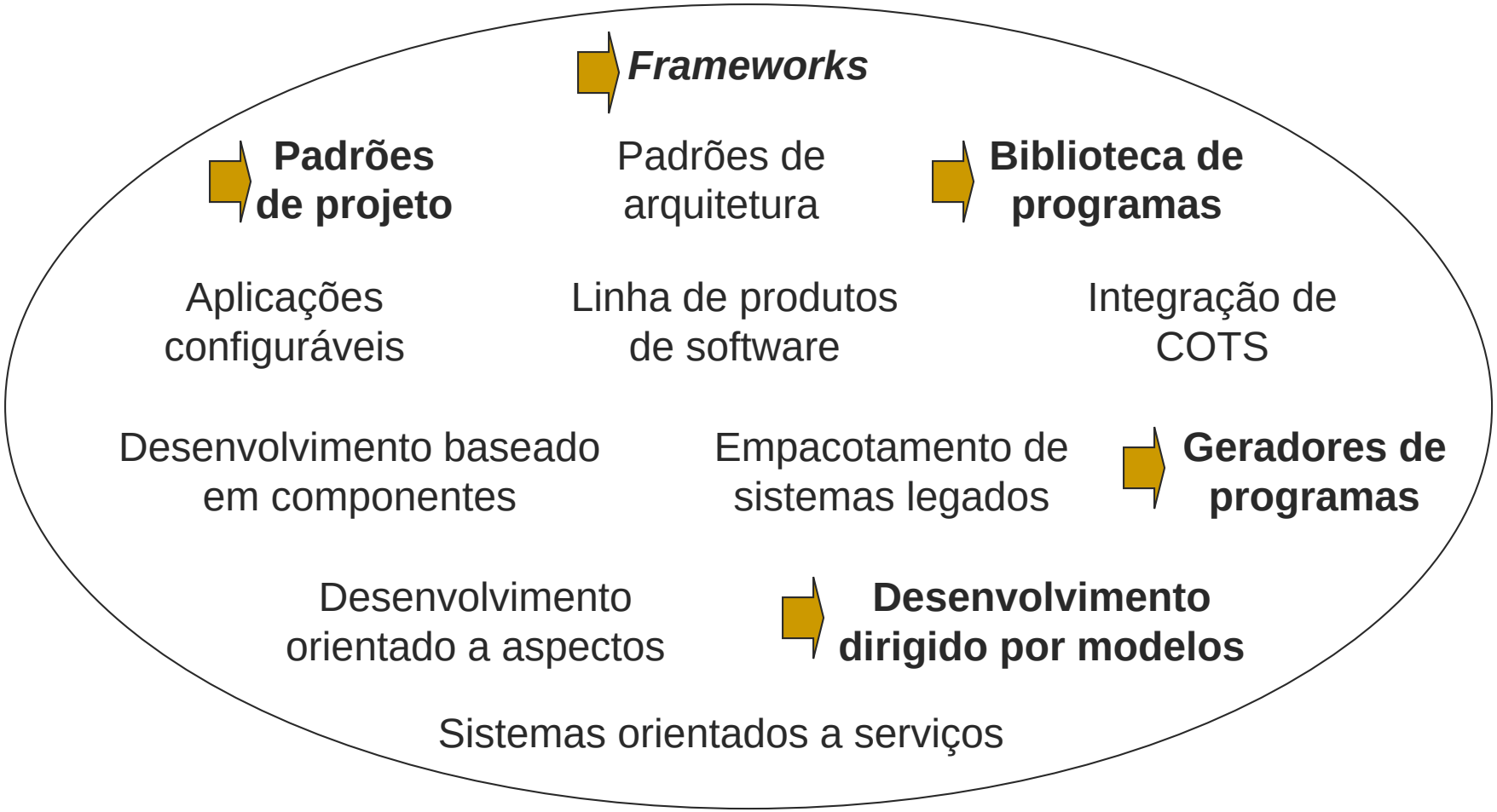
Técnicas para Reutilização de Software

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Panorama de Reutilização



Biblioteca e *Framework*

Bibliotecas de Software

- Bibliotecas implementam serviços que podem ser usados por programas
  - É uma forma comum de reutilização
- 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...

Uso de Biblioteca em Java

```
import java.util.Vector;

public class Customer {

   String name;
   Vector phoneNumbers = new Vector();

   void removePhoneNumber(String c){
     phoneNumbers.removeElement(c);
   }

   void addPhoneNumber(String c){
     phoneNumbers.addElement(c);
   }

   ...
}
```

Framework

- *Frameworks* são aplicações incompletas
  - São formados por interfaces, classes abstratas e classes concretas (OO)
- O conjunto de classes e interfaces formam uma estrutura genérica
- Um sistema é implementado pela adição de componentes para preencher lacunas
  - Por exemplo, pela implementação das classes abstratas do *framework*

Tipos de Frameworks

- Frameworks de infra-estrutura
  - Apóiam a criação de infra-estruturas de sistemas, tais como comunicações, interfaces de usuário e compiladores
- Frameworks de integração
  - Apóiam a comunicação e a troca de informações de componentes
- Frameworks de aplicações
  - Apóiam o desenvolvimento de um tipo de aplicações (e.g., aplicações Web)

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)

Principal Problema

- *Framework* é normalmente uma entidade grande e complexa
  - Pode levar um longo tempo para entendê-lo e usá-lo efetivamente
  - O desenvolvedor pode querer apenas uma funcionalidade simples

Padrão de Projeto

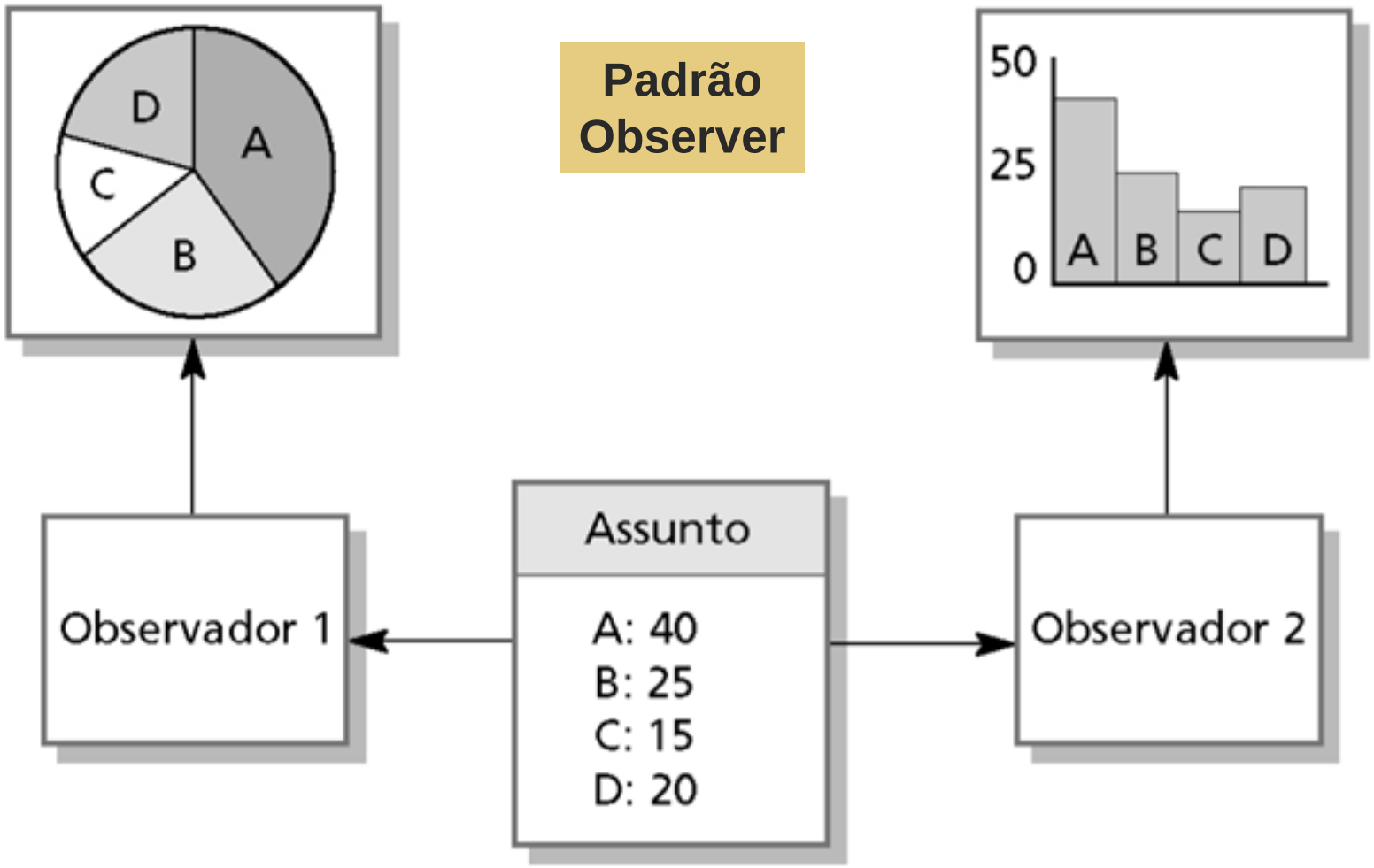
Padrões de Projeto

- Um padrão é uma descrição do problema e a essência da sua solução
- Documenta boas soluções para problemas recorrentes
  - Permite a reutilização de conhecimento anterior documentados em boas práticas
- Deve ser suficientemente abstrato para ser reusado em aplicações diferentes

Elementos de um Padrão

- Nome
  - Um identificador significativo para o padrão
- Descrição do problema
- Descrição da solução
  - Um *template* de solução que pode ser instanciado em maneiras diferentes
- Consequências
  - Os resultados e compromissos de aplicação do padrão

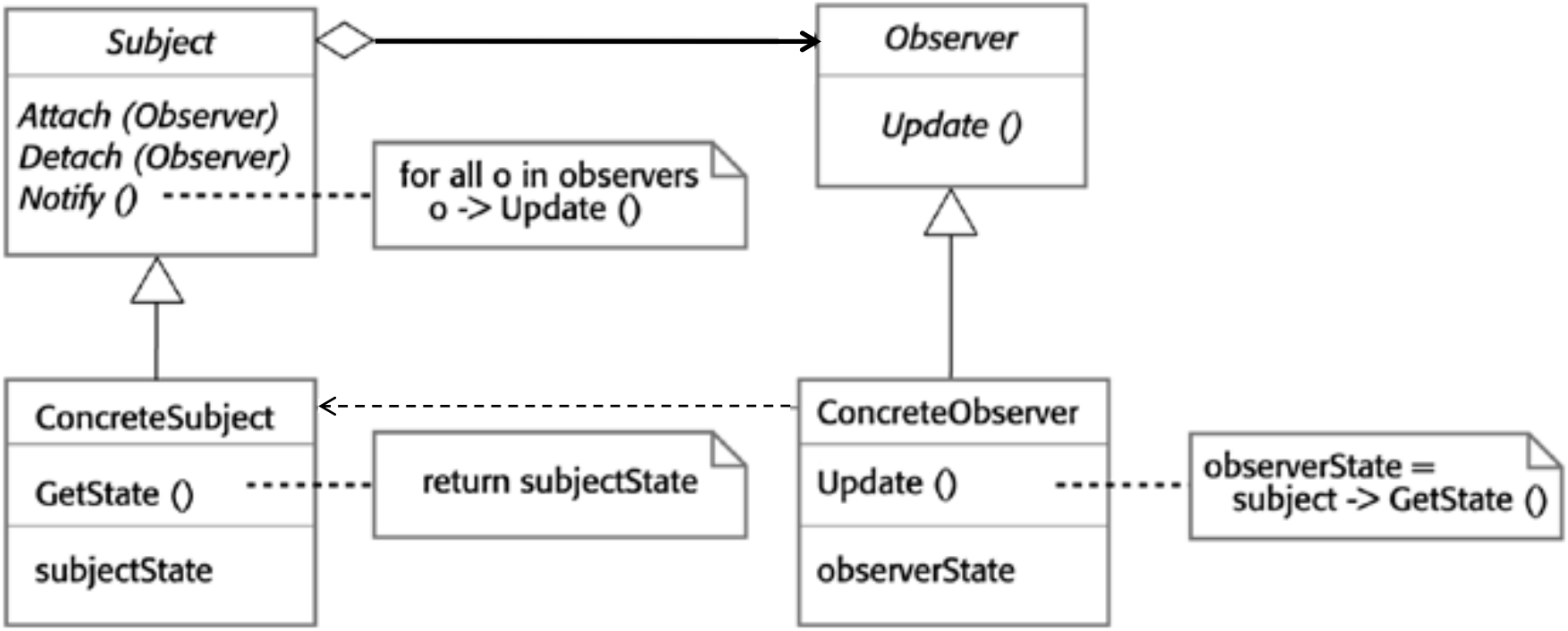
Exemplo de Problema



Padrão Obsever

- Nome
  - Observer
- Descrição do problema
  - Separa o objeto de suas formas de apresentação
- Descrição da solução (próximo slide)
- Consequências
  - Otimizações para melhorar a atualização da apresentação

Solução do Observer



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
- Pelo uso de geradores, o programa (código) é automaticamente gerado

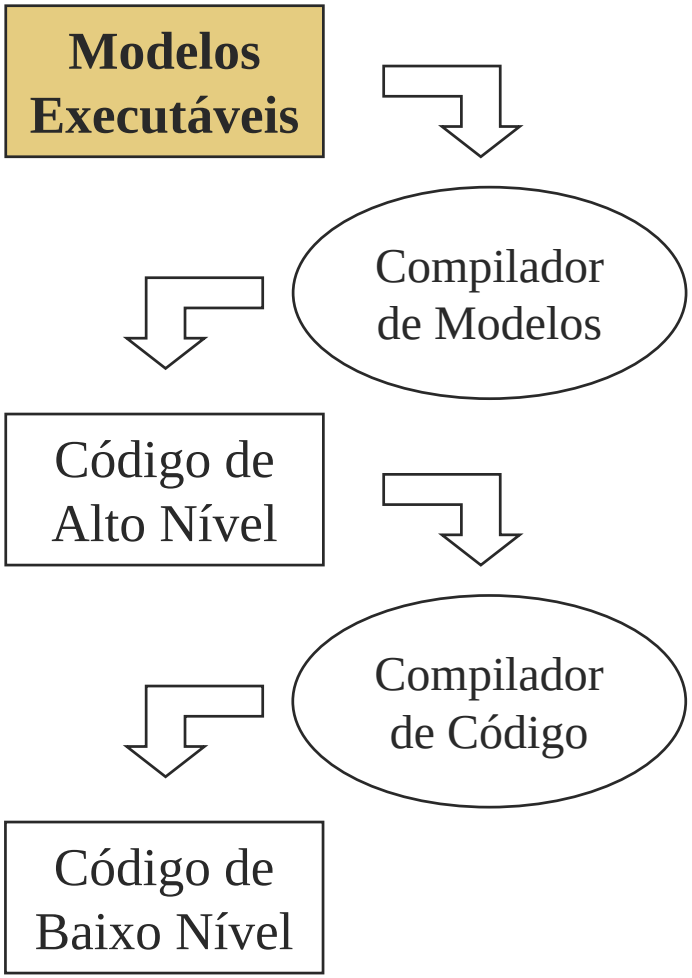
Modelos x Código

- 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

A Proposta MDD

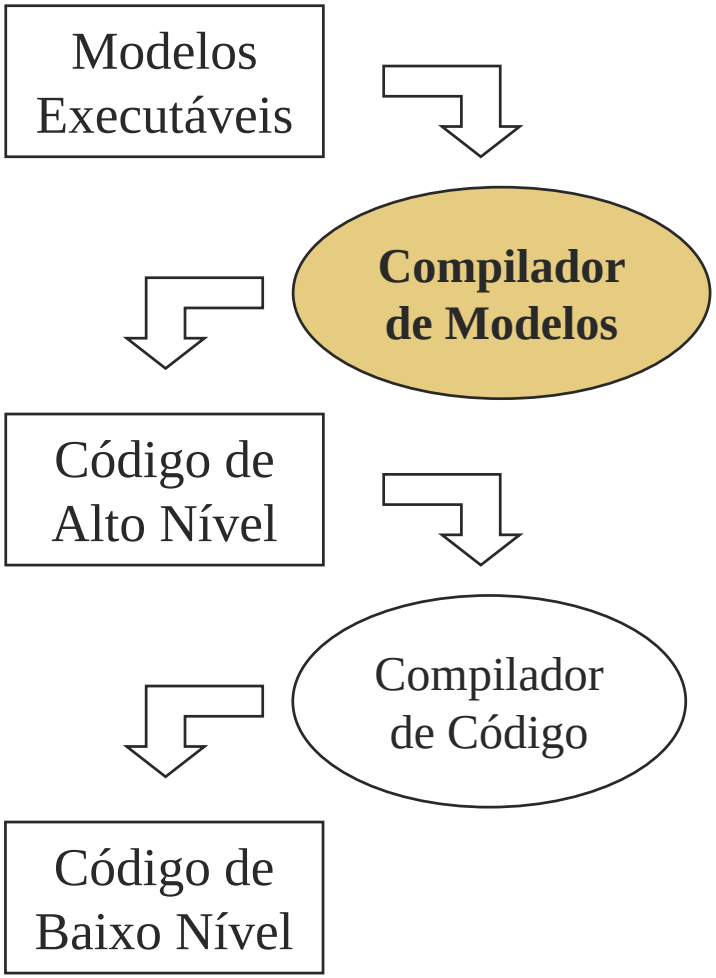
- Desenvolvimento Dirigido por Modelos
- Propõe que o desenvolvimento, reuso, manutenção e evolução sejam feitos no nível de modelagem
- Reutilização de modelos ainda é uma técnica pouco adotada
  - Se limitam a alguns domínios específicos ou em centros de pesquisa

Abordagem MDD



- 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

- Imaturidade do desenvolvimento dirigido por modelos
- 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
  - Temem por seus empregos como programadores

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