

Desenvolvimento de Software Orientado a Aspectos

Eduardo Figueiredo

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

Interesses transversais

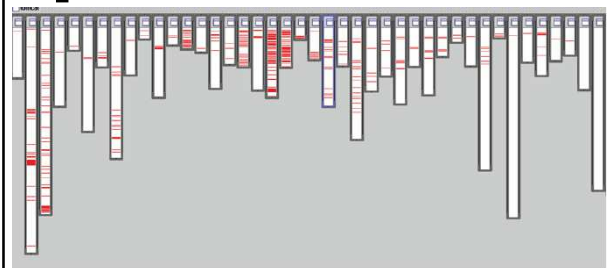
- São interesses cuja implementação atravessa uma série de componentes de programa
- Interesses transversais resultam em dois problemas no código
 - Espalhamento (*scattering*)
 - Entrelaçamento (*tangling*)
- Causam problemas quando mudanças devem ser feitas em um interesse

Exemplo: Logging

```
public class Conta {
    private String numero;
    private double saldo;
    ...
    public void debitar (double valor) {
        if (this.getSaldo() >= valor) {
            this.setSaldo(this.getSaldo()-valor);
            System.out.println("ocorreu um debito!");
        }...
    }...
}
```

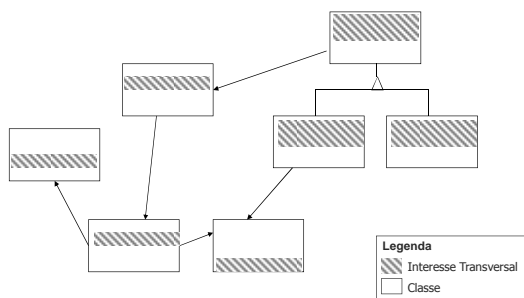
O código em vermelho implementa *Logging* e se espalha por várias partes do sistema

Logging no Tomcat

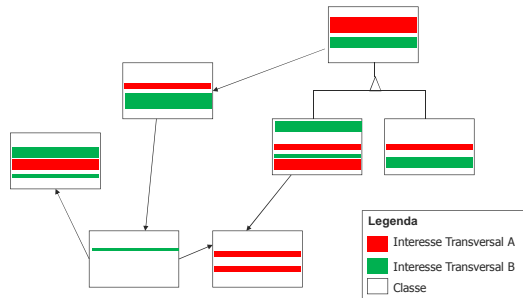


O código em vermelho implementa *Logging* e se espalha por várias partes do sistema

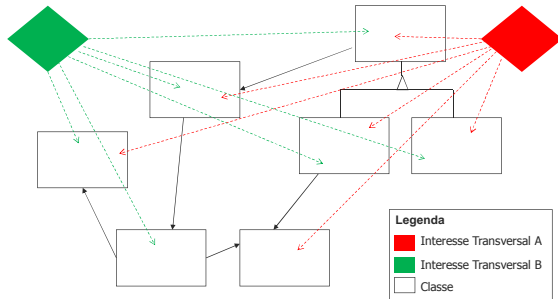
Espalhamento



Entrelaçamento



[Solução com Aspectos]



[Aspectos...]

- Desenvolvimento de software orientado a aspectos (DSOA)
 - Abordagem para desenvolvimento baseada em uma nova abstração: aspecto
- DSOA é usado em conjunto com outras abordagens
 - Geralmente com orientação a objetos (OO)

[Definições de DSOA]

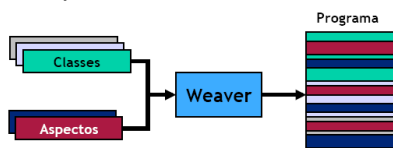
- Um aspecto é uma abstração que implementa um interesse transversal
 - Contém informações de onde deve ser incluído código do interesse
- Um ponto de junção é o local no programa onde pode ser incluído código do interesse
- Um ponto de corte é um conjunto de pontos de junção

[Definições de DSOA]

- Adendo é o código que implementa o comportamento do interesse
 - Semelhante a um método em classes
- Declarações inter-tipo são métodos, atributos e relacionamentos que implementam o interesse
 - Elementos de associação estática com as classes

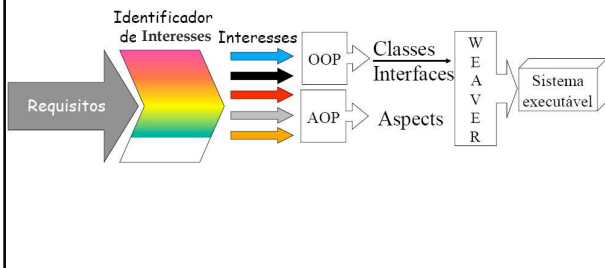
[DSOA]

- Complementa a orientação a objetos
 - Não existe programa que só possuem aspectos
- Classes e aspectos são “costurados” juntos pelo *Weaver*

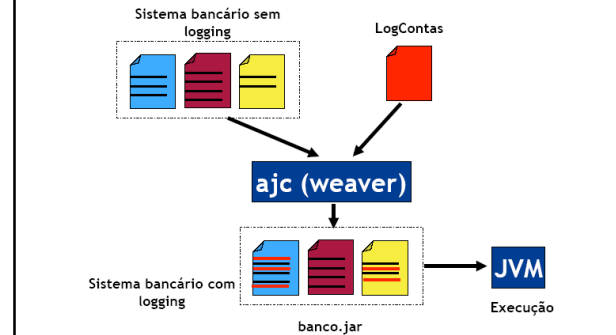


[O Processo DSOA]

[O processo de DSOA]



[Exemplo em AspectJ]



[Vantagens do DSOA]

- Separação de interesses transversais
 - Melhor modularidade, diminui a complexidade dos componentes, mais fácil o reuso, extensão, etc.
- Inconsciência e produtividade
 - As classes não conhecem os aspectos
 - Interesses podem ser implementados em paralelo
- Configuração (plugabilidade)

[Desvantagens de DSOA]

- Novo paradigma
 - Requer aprendizado e treinamento (nova forma de pensar)
- Fragmentação de código
 - Muitos componentes pequenos
- Ausência de suporte ferramental
 - Poucas ferramentas e poucos recursos
- Linguagens ainda pouco maduras

[Bibliografia]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Capítulo 21