Modularizing Design Patterns with Aspects: A Quantitative Study

Autores: Alessandro Garcia, Cláudio Sant'Anna, Eduardo Figueiredo, Uirá Kulesza, Carlos Lucena, Arndt von Staa

Proceedings of the 4th International Conference on Aspect Oriented Software Development (AOSD), pp. 3-14, 2005.

Apresentadora: Juliana Padilha

Motivação

- Padrões de projeto são importantes e úteis para o desenvolvimento de software
  - alguns padrões de projeto possuem interesses transversais
  - é importante averiguar se a AOP apoia a melhoria da modularização dos interesses

Este Artigo

- Complementa estudo de Hannemann e Kiczales
- Realiza avaliações quantitativas das implementações Java e AspectJ para os 23 padrões GoF
- É um estudo baseado em atributos de engenharia de software
  - separação de interesses
  - acoplamento
  - coesão
  - tamanho

Estudo de Hannemann e Kiczales

- Alguns padrões de projeto apresentam interesses transversais
- Desenvolveram e compararam as implementações Java e AspectJ nos 23 padrões de projeto GoF
- Para cada um dos padrões
  - desenvolveram um exemplo representativo – usa o padrão
  - e implementaram exemplos em Java e AspectJ

Métricas Utilizadas

- Foi selecionado um conjunto de métricas
  - separação de interesses (**Métricas de Interesse**)
  - acoplamento, coesão e tamanho (**Métricas Tradicionais**)
- Para avaliar as implementações dos padrões de Hannemann e Kiczales (HK)

Métricas de Interesse

Separação de Interesses	- Espalhamento de Interesses em Componentes (CDC) - Espalhamento de Interesses em Operações (CDO) - Espalhamento de Interesses em LOC (CDLOC)

Métricas Tradicionais

Acoplamento	- Acoplamento entre Componentes (CBC) - Profundidade em Árvore de Herança (DIT)
Coesão	- Perda de Coesão em Operações (LCOO)
Tamanho	- Linhas de Código (LOC) - Número de Atributos (NOA) - Operações Ponderadas por Componentes (WOC)

Procedimentos de Avaliação

- Comparou-se as duas implementações dos padrões (OO e AO)
  - Objetivo ➡ averiguar se ambas as versões (OO e AO) têm as mesmas funcionalidades
- Pequenas modificações foram realizadas no código dos padrões

Procedimentos de Avaliação

- Exemplo das modificações
  - Adicionar ou Remover uma funcionalidade
    - para avaliar a relevância da implementação do padrão
  - Garantir que ambas as versões (OO e AO) utilizaram os mesmos idiomas de programação

Procedimentos de Avaliação

- Por que a necessidade das modificações?
  - implementações HK englobam algumas regras por classes
- Ex:
  - Padrão *Mediator* → REGRAS → *Mediator* *Colleague*

Procedimentos de Avaliação

- Investigação da estrutura transversal dos padrões
  - adicionou-se mais classes participantes
- Tabela apresentada a seguir
  - as funções de cada modelo estudado
  - e as classes participantes introduzidas para cada exemplo de implementação padrão

Procedimentos de Avaliação

Design Pattern	Introduced Changes
Abstract Factory	4 Factories
Adapter	4 Adaptee Methods
Bridge	2 Abstractions and 2 Implementors
Builder	4 Builders
Chain of Responsibility (CoR)	4 Handlers
Command	4 Commands and 2 Invokers
Composite	2 Composites and 2 Leafs
Decorator	4 Decorators
Façade	No Change
Factory Method	4 Creators
Flyweight	4 Flyweights
Interpreter	4 Expressions
Iterator	2 Iterators and 2 Aggregates
Mediator	4 Mediators and 4 Colleagues
Memento	2 Mementos and 2 Originators
Observer	4 Observers and 4 Subjects
Prototype	4 Prototypes
Proxy	4 Proxies and 2 Real Subjects
Singleton	4 Singletons and 4 subclasses
State	4 States
Strategy	4 Strategies and 4 Contexts
Template Method	4 Concrete Classes
Visitor	4 Elements and 2 Visitors

Processo de Medição

- Métricas tradicionais (LOC, CBC, NOA, WOC e DIT) foram coletadas de forma automatizada – ferramenta Together
- Métricas de interesse (CDC, CDO e CDLOC) foram medidas através da contagem manual - sombreamento do código

Resultados

- Apresenta-se os resultados do processo das medições
- Resultados das medições nas aplicações da mudança selecionada (***before*** e ***after***)

Resultados

- A análise foi dividida em duas partes
  - Medição do código (AO) e (OO) em termos de separação de interesses
    - ➡ **Métricas de Interesse**
  - Medição em termos de acoplamento, coesão e tamanho
    - ➡ **Métricas Tradicionais**

Separação de Interesses

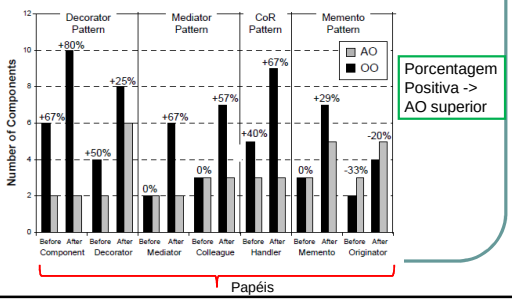
- Dados coletados através das
  - métricas de interesse
- Os padrões investigados foram classificados em três grupos

Separação de Interesses

- **Grupo 1**
- Envolveu 14 padrões:
  - Decorator
  - Adapter
  - Prototype
  - Visitor
  - Proxy
  - Singleton
  - Mediator
  - Composite
  - Observer
  - Command
  - Iterator
  - CoR – Chain of Responsability
  - Strategy
  - Memento

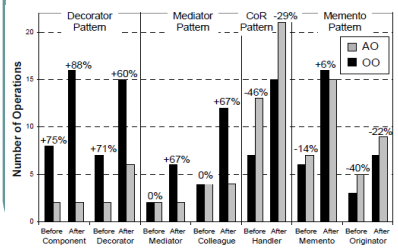
Grupo 1:Resultado Bom para AO

Espalhamento de Interesses em Componentes (CDC)



Grupo 1:Resultado Bom para AO

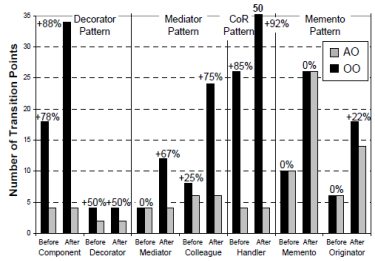
Espalhamento de Interesses em Operações (CDO)



Porcentagem Positiva -> AO superior

Grupo 1:Resultado Bom para AO

Espalhamento de Interesses em LOC (CDLOC)



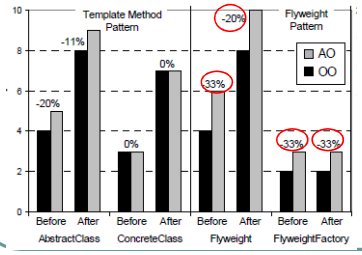
Porcentagem Positiva -> AO superior

Separação de Interesses

- **Grupo 2**
- Envolveu 6 padrões:
  - Template Method
  - Abstract Factory
  - Factory Method
  - Bridge
  - Builder
  - Flyweight

Grupo 2: Resultado Bom para OO

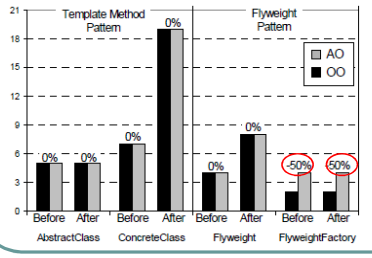
- Espalhamento de Interesses em Componentes (CDC)



Porcentagem Negativa -> OO superior

Grupo 2: Resultado Bom para OO

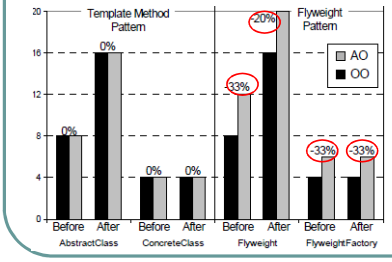
- Espalhamento de Interesses em Operações (CDO)



Porcentagem Negativa -> OO superior

Grupo 2: Resultado Bom para OO

- Espalhamento de Interesses em LOC (CDLOC)



Porcentagem Negativa -> OO superior

Separação de Interesses

- **Grupo 3**
- Envolveu 3 padrões:
  - Façade
  - Interpreter
  - State

Grupo 3: Nenhum Efeito

- Este grupo incluiu 3 padrões de projeto: *Façade, Interpreter e State*
- Os resultados foram muito similares para as implementações AO e OO
- O uso de aspectos não impactou os resultados

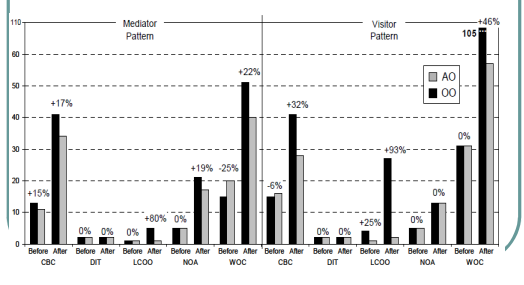
Acoplamento, Coesão e Tamanho

- Dados coletados através das
  - métricas tradicionais
- Os resultados obtidos foram divididos em 5 grupos

Acoplamento, Coesão e Tamanho

- **Grupo 1**
- Envolveu 5 padrões:
  - Composite
  - Observer
  - Adapter
  - Mediator
  - Visitor

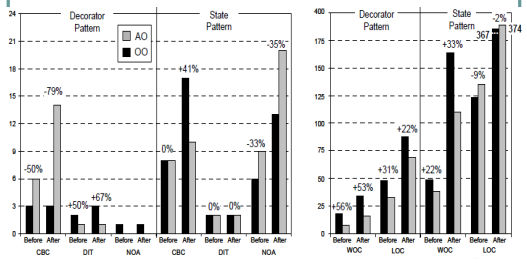
Grupo 1: Bons Resultados para AO



Acoplamento, Coesão e Tamanho

- **Grupo 2**
- Envolveu 4 padrões:
  - Decorator
  - Proxy
  - Singleton
  - State

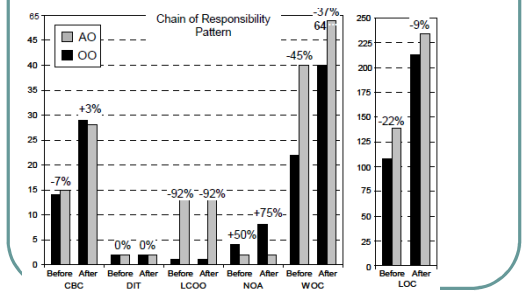
Grupo 2: Bons Resultados para AO em Muitas Métricas



Acoplamento, Coesão e Tamanho

- **Grupo 3**
- Envolveu 4 padrões:
  - CoR- Chain of Responsability
  - Command
  - Prototype
  - Strategy

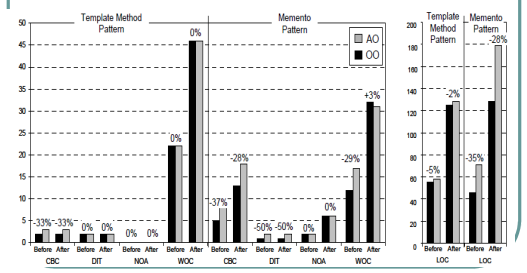
Grupo 3: Bons Resultados em OO em Muitas Métricas



Acoplamento, Coesão e Tamanho

- **Grupo 4**
- Envolveu 8 padrões:
  - Template Method
  - Abstract Factory
  - Bridge
  - Interpreter
  - Factory Method
  - Builder
  - Memento
  - Flyweight

Grupo 4: Bons Resultados para OO



Acoplamento, Coesão e Tamanho

- **Grupo 5**
- Envolveu 2 padrões:
  - Iterator
  - Façade

Grupo 5: Nenhum Efeito

- *Façade* – implementação é a mesma em AO e OO
- *Iterator* – implementação em AO produziu melhorias insignificates

Análise quanto ao Reuso

- O reuso foi observado em apenas quatro padrões:
  - Mediator
  - Observer
  - Composite
  - Visitor
- Eles também foram qualificados como reutilizáveis no estudo HK, pois tem várias características em comum, dentre elas
  - melhor separação de interesses
  - baixo acoplamento - CBC - e alta coesão - LCOO

Limitações do Artigo

- O escopo do artigo é limitado a:
  - aos padrões selecionados para o estudo comparativo
  - as implementações específicas do livro GoF e o estudo HK
  - aos idiomas de programação Java e AspectJ

Conclusão

- Estudo quantitativo
  - comparou as implementações AO e OO nos padrões GoF
- Resultados indicaram que
  - as implementações em AO melhoram a separação de interesses
- Alguns padrões resultaram no aumento de
  - componentes acoplados
  - operações mais complexas
  - mais linhas de código nas soluções AO.

Conclusão

- Soluções AO melhoram a separação de interesses relacionados ao padrão
  - apesar de apenas 4 implementações AO exibirem reutilização significativa
- Este foi o primeiro estudo exploratório
  - Os experimentos não foram controlados
  - Não tem análise estatística

OBRIGADA!!!