

## 5 Avaliação Quantitativa dos Estudos de Caso

Neste capítulo, apresentamos os resultados de medição que nos permitiram avaliar quantitativamente o reuso de sensibilidade ao contexto obtido através do uso de ESSMA (Seção 5.1). Como a tarefa de reengenharia de VL e WMS (Seção 2.2) nos levou à abstração do *framework* CAAF (Capítulo 4), os resultados da medição também nos possibilitaram a *confirmação* dos resultados da avaliação qualitativa especificados na Seção 4.3. Os resultados encontrados foram obtidos a partir da comparação das duas versões de VL e WMS (Seção 5.2) e, para esta comparação, usamos um conjunto de métricas que nos permitiu quantificar atributos de software, como acoplamento, tamanho e coesão (Seção 5.1).

### 5.1. Procedimentos e Métricas

O objetivo central do estudo foi observar o comportamento de atributos bem conhecidos de engenharia de software, após a reengenharia de projeto e código orientado a objetos (OO) de ASCs usando ESSMA. O estudo foi organizado de acordo com os seguintes passos: (1) seleção das métricas de software; (2) execução de procedimentos de avaliação; (3) coleta dos valores das métricas; e (4) análise dos valores obtidos. A execução de procedimentos de avaliação consistiu basicamente no alinhamento do código das versões de VL e WMS, de forma a garantir que ambas as versões estavam utilizando um mesmo padrão de codificação. A segunda versão de cada ASC foi obtida a partir da reengenharia da versão OO. Isso garantiu que ambas as versões estavam implementando as mesmas funcionalidades.

O conjunto de métricas utilizado no estudo quantitativo foi definido com base em métricas OO existentes (Chidamber & Kemerer, 1994; Fenton & Pfleeger, 1997) que permitem a medição de atributos estruturais internos com impacto direto na modularidade das aplicações em geral e, conseqüentemente, em

seu nível de reuso. A Tabela 1 apresenta a definição de cada uma das métricas utilizadas e sua associação com diferentes atributos de modularidade (acoplamento, coesão e tamanho). Tais métricas também têm sido aplicadas a vários outros estudos do Laboratório de Engenharia de Software da PUC-Rio (Sant'Anna et al, 2003; Garcia, 2004; Filho et al, 2004; Garcia et al, 2006; Cacho et al, 2006a; Cacho et al, 2006c; Kulesza et al, 2006). Em nosso estudo, os valores das métricas foram calculados automaticamente usando a ferramenta Together (TOGETHER TECHNOLOGIES).

**Tabela 1.** Métricas para Avaliação dos Estudos e do Framework

Atributo	Métrica	Descrição
Tamanho	Vocabulary Size (VS)	Calcula o número de componentes (classes e interfaces) existentes no sistema.
	Lines of Code (LOC)	Determina o número de linhas de código de um dado componente do sistema. Comentários não devem ser considerados na contagem. O cálculo dessa métrica exige a adoção de convenções de codificação em todos os componentes do sistema.
	Number of Attributes (NOA)	Conta o número de atributos em uma determinada classe do sistema.
	Weighted Operations per Component (WOC)	Determina a complexidade de um componente em função das suas operações. Ela é calculada baseada na quantidade de parâmetros existentes em todos os métodos de um dado componente.
Acoplamento	Coupling between Components (CBC)	Calcula o grau de acoplamento de um dado componente (classes e interfaces) em relação aos demais componentes do sistema. São contados atributos, parâmetros, tipos de retorno, declaração de throws, variáveis locais, etc.
	Depth of Inheritance Tree (DIT)	Mede a profundidade de uma dada subclasse para a classe raiz da sua árvore de herança.
Coesão	Lack of Cohesion in Operations (LCOO)	Mostra a proximidade de relação entre seus componentes internos. Ela é calculada baseada na manipulação dos atributos de uma classe por seus métodos.

## 5.2. Resultados Encontrados

As Tabelas 2 e 3 apresentam os valores absolutos obtidos após a medição das duas versões de VL e WMS e os valores percentuais que representam a diferença entre estes resultados para ambas as versões de cada aplicação.

**Tabela 2.** Valores de Acoplamento, Coesão e Tamanho em VL

Virtual Lines	Acoplamento		Coesão	Tamanho			
	CBC	DIT	LCOO	VS	LOC	NOA	WOC
OO	127	23	83	15	676	32	68
ESSMA	132	24	69	18	720	30	58
Diferença	3,94%	4,35%	-16,87%	20%	6,51%	-6,25%	-14,71%

**Tabela 3.** Valores de Acoplamento, Coesão e Tamanho em WMS

Wireless Marketing Service	Acoplamento		Coesão	Tamanho			
	CBC	DIT	LCOO	VS	LOC	NOA	WOC
OO	208	44	129	25	3091	176	157
ESSMA	236	49	127	25	2632	155	145
Diferença	13,46%	11,36%	-1,55%	0%	-14,85%	-11,93%	-7,64%

Primeiramente, a diferença no valor da métrica VS, usada para medir o número de componentes nas ASCs OO e usando ESSMA, foi significativa apenas no estudo VL (20%); no WMS, o “tamanho do vocabulário” permaneceu constante (0%). O fato de usarmos as abstrações de agentes e comportamentos provoca um desmembramento das funcionalidades de VL, que na versão original estão encapsuladas em uma *única* classe. Em outras palavras: para cada funcionalidade de uma classe que representa um agente, um tipo de comportamento específico foi incorporado à versão de VL usando ESSMA e, sendo assim, a métrica VS necessariamente deveria aumentar. Ao contrário, as características da versão OO de WMS foram projetadas de modo muito mais próximo daquele que resultou quando da aplicação de ESSMA (compare as Figuras 12 e 13 com as Figuras 29 e 30) em contraste com o que ocorreu entre as duas versões de VL (compare as Figuras 6 e 7 com as Figuras 15 e 16).

Houve um pequeno aumento nos valores da métrica DIT para ambas as versões usando ESSMA: 4,35% para VL e 11,36% para WMS. Este aumento foi ocasionado pela adição de alguns níveis de profundidade na árvore de herança das ASCs, *necessária* pelo uso das classes do *framework* CAAF. Os valores obtidos para a métrica de acoplamento (CBC) também foram superiores para ambas as versões dos casos usando ESSMA. Entretanto, entendemos que as diferenças

obtidas (3,94% e 13,46%) podem ser minimizadas se percebermos que as versões usando ESSMA oferecem a vantagem de permitir a inserção e/ou remoção fáceis de classes associadas às funcionalidades de uma ASC e que isto invariavelmente leva a um aumento do nível de reuso das aplicações (como queremos). Por outro lado, as versões OO requerem mudanças invasivas em suas classes internas, que dificultam a evolução do projeto. Assim, levando em consideração a natureza de cada solução, a versão OO apresenta um alto nível de acoplamento, embora com valor menor para CBC quando comparado à versão usando ESSMA.

Agora passamos a analisar em geral todas as outras métricas, que absolutamente refletiram a superioridade do paradigma ESSMA para o desenvolvimento de ASCs. De fato, nas versões dos casos usando ESSMA obtivemos uma diminuição do número de atributos (NOA) tanto em WMS (-11,93%) como em VL (-6,25%) e, em consequência, houve também um aumento de coesão (LCOO) e diminuição de complexidade (WOC) nestas versões. Ao contrário, o aumento de linhas de código (LOC) em VL é facilmente justificável pelo acréscimo *necessário* das classes referentes a agentes e comportamentos aliado a uma distância maior do projeto original quando usamos ESSMA (como explicamos no parágrafo relativo à métrica VS).

Os resultados do estudo quantitativo demonstraram que, de modo geral, houve uma melhora significativa nos valores das métricas obtidas para ambas as versões de ASCs usando ESSMA. Tais resultados eram por nós esperados, visto que nossa proposta para um aumento de reuso em ASCs (Capítulo 4) permitiu a integração e composição de duas outras soluções comumente utilizadas para o desenvolvimento de ASCs *publish-subscribe* usando agentes: JADE e MoCA.