

7 Trabalhos Relacionados

Até o momento atual, a maioria dos estudos experimentais no contexto do desenvolvimento de software orientado a aspectos se baseou em critérios subjetivos e investigações qualitativas [4, 5, 53, 54]. Por exemplo, Hannemann & Kiczales [5] comparam qualitativamente implementações em Java e implementações em AspectJ dos padrões de projeto da *GoF* [44], em termos de alguns critérios de avaliação pouco conhecidos, como transparência de composição e facilidade de (des)ligamento. Poucos trabalhos propuseram métricas de software para o desenvolvimento de software orientado a aspectos, como o trabalho de Lopes [33]. Ela definiu um conjunto de diferentes métricas para separação de *concerns*. De fato as métricas CDC, CDO e CDLOC propostas nessa dissertação (Seção 5.1) foram inspiradas pelo conjunto de métricas de Lopes. No entanto, as definições das métricas de Lopes são fortemente ligadas ao estudo empírico realizado por ela e adaptadas ao *concern* de distribuição em código Java. As métricas de separação de *concerns* do framework de avaliação proposto nesta dissertação generalizam as métricas de Lopes de forma a permitir que elas possam ser usadas para avaliar diferentes *concerns*.

Zhao propôs um conjunto de métricas para software orientado a aspectos, que são especialmente definidas para quantificar os fluxos de informação de um programa orientado a aspectos [31]. As métricas dele são baseadas em um modelo de dependência para software orientado a aspectos que consiste de um grupo de grafos de dependência. Cada grafo é usado para explicitamente representar várias relações de dependência em diferentes níveis de um programa orientado a aspectos. As métricas de Zhao podem ser vistas como complementares ao conjunto de métricas proposto nesta dissertação. No entanto, sua aplicação é trabalhosa e consome muito tempo, devido à necessidade de elaboração de todos os grafos. O uso de tais métricas requer que os engenheiros de software construam vários grafos de dependência para diferentes níveis de modularidade, tais como o grafo de dependência de método (MDG), o grafo de dependência de *advice*

(ADG), o grafo de dependência de *inter-type declaration* (IDG) e assim por diante. Conseqüentemente, tais métricas são complexas de entender e usar, e requerem a implementação de uma ferramenta de análise de dependência, que provavelmente é diferente de uma linguagem para outra. Além disso, as métricas de Zhao não são derivadas de métricas já existentes, testadas e validadas.

Portanto, as principais diferenças do trabalho apresentado nesta dissertação para os poucos trabalhos relacionados existentes são as seguintes:

- Os estudos experimentais apresentados aqui foram baseados em análises quantitativas, apoiadas pela aplicação de métricas de projeto e código;
- As métricas propostas medem atributos de software bem conhecidos da engenharia de software;
- O conjunto de métricas proposto se baseou em outras métricas já existentes, usadas e validadas.