

## A Unified Framework for Cohesion Measurement in Object-Oriented

- Daniel Edilson Souza

## Visão geral

\* A importância cada vez maior de serem colocadas medições em softwares tem levado a um aumento da quantidade de pesquisas de desenvolvimento de medidas de software.

\* é difícil determinar como estas medidas se relacionam umas com as outras e para aplicação elas podem ser usadas.

\* uma terminologia padronizada e formalismo para expressar medidas.

\* revisão das abordagens existentes para medir coesão em sistemas orientados a objetos. Posteriormente, uma estrutura unificada.

\* revisão do trabalho de validação empírica sobre as medidas de coesão existentes é fornecida.

## Visão geral

\* Contribuir para uma maior compreensão do GQM: é fornecido um mecanismo para comparação medidas e sua potencial utilização.

\* Análise de questões importantes: (i) muitas medidas não são definidas de uma forma totalmente operacional, (ii) relativamente algumas delas são baseados em modelos empíricos explícitos tal como recomendado pela teoria de medição.

## Terminologia e formalismo

\*No passado, a pesquisa dentro da área de medição de software sofreu com a falta de um formalismo padronizado, e para definir medidas de maneira que não seja necessária interpretação adicional.

## Introdução

- Indústria de desenvolvimento de software começou a dar muito mais ênfase na qualidade de software, em particular para avaliar e prever a qualidade de software, como a coesão.
- Em software orientado a objetos, classes substituem módulos, com métodos e atributos como seus elementos. Neste contexto, a coesão é o grau em que os métodos e os atributos a uma classe pertencem.

## Introdução

\*Por exemplo, porque não existe uma terminologia padrão para expressar o formalismo de medidas, ou seja, há alguma ambiguidade em suas definições.

\*Como resultado, é difícil compreender como as diferentes medidas de coesão referem-se uma a outra e como estas medidas diferentes podem ser utilizadas de uma forma complementar.

## Introdução

\*Estrutura utilizada (i) para facilitar a comparação das medidas de coesão existentes, (ii) facilitar a avaliação e validação empírica de medidas existentes de coesão, e (iii) apoiar a definição de novas medidas de coesão e a seleção das já existentes com base em um objetivo particular da medição.

## Motivação

\* Como nenhuma terminologia padrão ou formalismo existe, muitas medidas são expressas de forma ambígua o que limita a sua utilização ou se relacionam entre si.

## Motivação

\* Uma revisão das medidas existentes(15) foi realizada, e estas medidas têm que ser classificadas de acordo com a estrutura unificada. Este framework será, então, um mecanismo para comparação de medidas e os seu uso potencial, integrar as medidas existentes que examinam os mesmos conceitos de uma forma diferente forma.

\* Facilitar a avaliação e validação empírica das medidas de coesão, garantindo que hipóteses específicas sejam um link de medidas de coesão para atributos de qualidade externas.

## Terminologia e formalismo

\*No passado, a pesquisa dentro da área de medição de software sofreu com a falta de um formalismo padronizado, e para definir medidas de maneira que não seja necessária interpretação adicional.

## Terminologia e formalismo

- **Métodos**
- Uma classe possui um conjunto de métodos declarados e implementados onde:

where  $M(c) = M_D(c) \cup M_I(c)$  and  $M_D(c) \cap M_I(c) = \emptyset$ .

## **Métodos**

- Métodos declarados e implementados
- Métodos herdados, sobrescritos e novos
- Métodos públicos e não-públicos
- Parâmetros
- Métodos podem ser invocados staticamente, direta ou indiretamente.

### **Atributos**

- Classes possuem atributos que podem herdadas ou recém-definidos.
- Referência a atributo

### **Tipos**

- \* Atributos e parâmetros podem possuir tipos que afetam a coesão da classe, podem ser eles tipos nativos da LP ou user-types.

### **Levantamento de Abordagens de Medição e Medidas de Coesão**

- Foi realizada uma revisão crítica das abordagens existentes e medidas de coesão em sistemas orientados a objeto. Eder et al. (1994) propôs um framework destinado a fornecer critérios qualitativos para a coesão e Henderson-Sellers (1996) uma medida quantitativa.

### **Framework de Eder et al. (1994)**

- Eder et al. (1994) propôs um framework destinado a fornecer critérios qualitativos para medir coesão em sistemas O.O.
- Foram adaptadas estruturas existentes para a coesão do paradigma processual.
- Foram discutidos três tipos de coesão em um sistema orientado a objetos: coesão de classe, método e herança.

### **Framework de Eder et al. (1994)**

- Coesão de método: Eder et al. Aplica a definição clássica de Myers de coesão (Myers, 1978) para métodos. Elementos de um método são suas declarações, variáveis locais e atributos de classe.

### **Framework de Eder et al. (1994)**

- Coesão de classe aborda as relações entre os elementos de um classe, métodos não-herdados e atributos não-herdados.
- Adaptado conceito de coesão de Embley and Woodfield (1987) para sistemas orientados a objeto.

### Framework de Eder et al. (1994)

- Herança aborda a coesão de herança dos relacionamentos entre os elementos de uma classe e também os métodos e atributos herdados, e suas especializações pelos filhos.

### Framework de Eder et al. (1994)

A análise de coesão requer um bom conhecimento do domínio do sistema, ela pode ser subjetiva e não pode ser automatizada.

### Framework por Henderson-Sellers (1996)

- Henderson-Sellers propõem a definição de uma medida de coesão tendo as seguintes propriedades:
  - A medida dá 0, se cada método da classe referencia cada atributo da classe (a situação é chamada de "coesão perfeita" por Henderson-Sellers).
  - A medida produz 1, se cada método da classe referencia apenas um único atributo.
  - Valores entre 0 e 1 são para ser interpretados como percentagens de valor ideal.

### Fórmula de coesão

- A - conjunto de atributos acessados pelo método.
- M - Métodos da classes.
- AR - Argumentos dos métodos.

$$LCOMS(c) = \frac{|M_T(c)| - \frac{1}{|A_T(c)|} \sum_{a \in A_T(c)} \{ |m| \mid m \in M_T(c) \wedge a \in AR(m) \}}{|M_T(c)| - 1}$$

### Conclusão

\* A partir da discussão acima, podemos ver que existe uma variedade de decisões a serem tomadas durante a definição de uma medida de coesão. É importante que as decisões sejam baseadas na aplicação pretendida da medida se a medida será útil. Quando não há decisão para um aspecto em particular, todas as alternativas devem ser investigadas empiricamente.

\* A segunda observação é que por que os diferentes aspectos de coesão são amplamente independentes um do outro, um grande número das medidas de coesão pode ser definido.

Name	Definition	Number of attributes	Number of methods	Number of arguments	Number of methods	Number of arguments	Number of methods	Number of arguments	Number of methods	Number of arguments	Source
LCOM1 (lack of cohesion in methods)	$LCOM1(c) = \frac{1}{ M_T(c) } \sum_{m \in M_T(c)} \frac{ A(m) }{ A_T(c) }$	no	obj	ratio	HELD	Emp	no	no	Henderson-Sellers (1996)		
LCOM2	$LCOM2(c) = \frac{1}{ M_T(c) } \sum_{m \in M_T(c)} \frac{ A(m) }{ A_T(c) }$	no	obj	obj	HELD	Emp	no	no	Chidambare and Ramesh (1994)		
LCOM3	$LCOM3(c) = \frac{1}{ M_T(c) } \sum_{m \in M_T(c)} \frac{ A(m) }{ A_T(c) }$	no	obj	obj	HELD	Emp	no	no	Chidambare and Ramesh (1994)		
LCOM4	$LCOM4(c) = \frac{1}{ M_T(c) } \sum_{m \in M_T(c)} \frac{ A(m) }{ A_T(c) }$	no	obj	obj	HELD	Emp	no	no	Chidambare and Ramesh (1994)		
LCOM5	$LCOM5(c) = \frac{ M_T(c)  - \frac{1}{ A_T(c) } \sum_{a \in A_T(c)} \{  m  \mid m \in M_T(c) \wedge a \in AR(m) \}}{ M_T(c)  - 1}$	no	obj	ratio	HELD	Emp	no	no	Henderson-Sellers (1996)		

### A Unified Framework for Cohesion Measurement

- O framework é definido com base nas questões identificadas pela comparação das diferentes abordagens para medir a coesão e discussão das medidas existentes.
- Apoiar a comparação e seleção das medidas de coesão existentes para GQM.

### *Critérios do Framework*

- Determinam um aspecto básico da medida resultante. Para cada critério, temos que escolher uma ou mais das opções disponíveis. Estes critérios são necessários para considerar quando se especificar a medida de coesão.
- *Tipos de conexão*
- *Domínio das medidas*
- *Conexões diretas e indiretas*
- *Herança*

### *Tipos de conexão*

- O que torna a classe coesa (direta, indireta, estática...).
- A conexão com a classe é o link entre elementos de uma classe (atributos, métodos, ou declarações de dados).

### *Domínio das medidas*

- O domínio da medida especifica os objetos a serem medidos (métodos, classes, etc.).
- Grande maioria de medidas é de classe.

### *Conexões diretas ou indiretas*

- Decidir se conta apenas conexões diretas ou também conexões indiretas.

### *Herança*

- Dois aspectos devem ser considerados em relação à herança:
  - 1.a) *Excluir atributos herdados e métodos de análise.*
  - 1.b) *Incluir atributos herdados e métodos na análise.*

### Polimorfismo

- Classe pode conter invocação de métodos implementados polimorficamente na classe ancestral.
- Contar polimorfismo.
- Não contar polimorfismo.

### Métodos de acesso e construtores

- métodos de acesso e construtores podem artificialmente aumentar ou diminuir os valores para medidas de coesão.
- a) Métodos de acesso: Getter's n' setter's . Solução consiste em contar a invocação de um método de acesso como referência para o atributo.
- b) Construtores: Construtores normalmente referenciar todos os atributos. Isto aumenta artificialmente a coesão da classe, uma vez que gera muitos pares de métodos que utilizam um atributo em comum.

### Construção e Normalização da Medida da Coesão

- Para a definição da medida, precisamos de dois números: o número de conexões reais de interesse, também o número máximo de conexões possíveis de interesse,

(Número de conexões atuais de interesse)

(Máximo número possível de conexões de interesse)

obtemos uma medida normalizada se esta está num intervalo entre 0 e 1.

### Resumo

- Medidas existentes foram classificadas de acordo com as opções disponíveis para cada critério do framework. Esta classificação permite tais medidas serem comparadas e avaliadas quanto a sua utilidade em relação ao QQM.

### Ameaças à validade

- A escolha da variável dependente usada nos Li e Henry (1993) para medir a capacidade de manutenção (ou esforço de manutenção) é questionável: "O número de linhas alteradas em uma classe" pode dizer pouco sobre o esforço real (em termos de pessoa horas etc)

### Validade Interna

\*Medidas podem ser aplicáveis em estágios iniciais do processo de desenvolvimento, poupando assim tempo e esforço de retrabalho, no entanto, o estudo nesta medição foi realizado somente após a implementação e a descrição do sistema antes da implementação é menos completo do que após a implementação.

### *Validade externa*

Basili lista os seguintes fatos que podem restringir a generalização de seus resultados

- Os sistemas desenvolvidos são pequenos: eles ficam entre 5.000 e 14.000 linhas de código de código C++.
- Os sistemas desenvolvidos têm uma complexidade conceitual limitado.
- Os desenvolvedores podem não ser tão bem treinados e experientes como média programadores profissionais.

### Pontos importantes destacados

- Uma razão para a pequena quantidade de trabalho de validação empírica é que precisamos definir e medir um atributo interno e um atributo externo, e medir um atributo de qualidade externa pode ser difícil na prática.
- Trocas na definição do atributo externo para contornar estes problemas limitam a utilidade do estudo. Por exemplo, em Li e Henry (1993) o esforço de manutenção foi definido como o número de linhas modificadas em uma classe. Uma vantagem é que o número de linhas modificadas em uma classe pode ser calculado. No entanto, como já realçado, o número de linhas modificadas não é um indicador direto do esforço real gasto na manutenção da classe.

### Autores

- LIONEL C. BRIAND
- JOHN W. DALY
- JURGEN WUST
- 1998