The Conceptual Cohesion of Classes

Andrian Marcus and Denys Poshyvanyk

Department of Computer Science
Wayne State University
Detroit, MI, USA

Motivation

- Cohesion is the degree to which the elements in a design unit (class, package) are logically related or "belong together" [Briand'00]
- A cohesive class represents a crisp abstraction from a problem domain
- Different views of cohesion
- No accepted standard in the community
- Class cohesion can significantly affect the design, understandability, maintainability

Related Work – Class Cohesion

- Structural metrics:
  - LCOM1, LCOM2 [Chidamber 94][1]; LCOM3, LCOM4 [Hitz 94]
  - LCOM5 [Henderson 96]
  - Connectivity [Hitz 94]; Coh [Briand 97, 98]
  - ICH[2] [Lee 95]; TCC[3], LCC[4] [Bieman 95, 98]
- Semantic metrics
  - LORM[5] [Etzkorn 00]; SCDE[6] [Etzkorn 02]; SCF[7] [Maletic 01]
- Information entropy-based metrics; Metrics based on data mining; Slice-based metrics; etc.

1. *Lack of cohesion in methods*
2. *Information-flow based cohesion*
3. *Tight class cohesion*
4. *Loose class cohesion*
5. *Logical relatedness of methods*
6. *Semantic class definition entropy*
7. *Semantic cohesion of files*

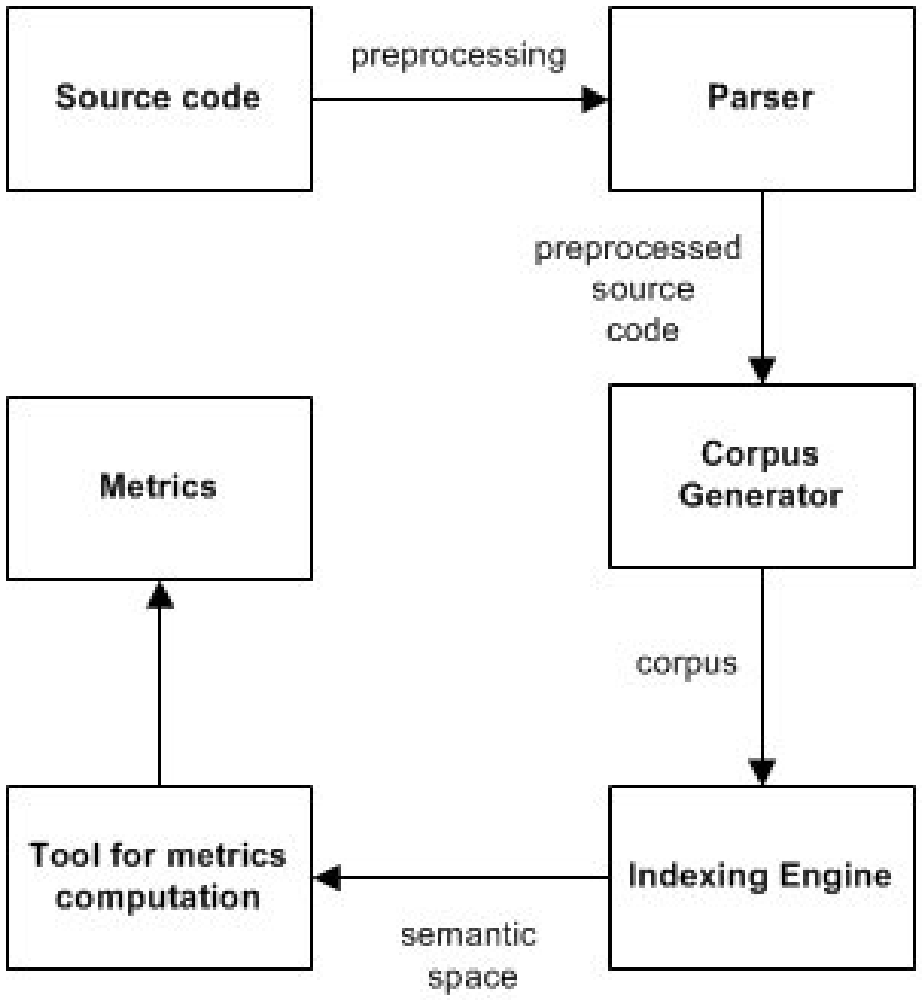
Types of Cohesion

- Functional
- Informational
- Communicational
- Procedural
- Temporal
- Logical
- Coincidental

Information Retrieval Approach for Cohesion Measurement

- Using semantic information (i.e., comments, identifiers, etc.) to measure cohesion (i.e., how related are the elements of a class)
- Use information retrieval (IR) methods to extract and analyze the semantic information
- We are using Latent Semantic Indexing (LSI)

Measuring Methodology



System Representation

- Set of classes C = $\{c_1, c_2 \ldots c_n\}$
- For each class $c \in C$, $M(c) = \{m_1, \ldots, m_k\}$
- v_{mk} and v_{mj} are the vectors corresponding to the $m_k, m_j \in M(c_i)$
- Conceptual similarity between two methods

$$CSM(m_k, m_j) = \frac{vm_k^T vm_j}{|vm_k|_2 \times |vm_j|_2}$$

The Conceptual Cohesion of Classes

- Average conceptual similarity of the methods in a class (ACSM) c ∈ C

$$\text{ACSM(c)} = \frac{1}{N} \times \sum_{i=1}^{N} CSM(m_i, m_j)$$

- Conceptual cohesion of a class (C3) c ∈ C

$$\text{C3(c)} = \begin{cases} ACSM(c) & if \quad ACSM(c) > 0 \\ else & 0 \end{cases}$$

Example of measuring C3

	m1	m2	m3	m4	m5
m1	**1**	**0.21**	**0.72**	**0.33**	**0.42**
m2		**1**	**0.28**	**0.91**	**0.66**
m3			**1**	**0.37**	**0.27**
m4				**1**	**0.89**
m5					**1**

Conceptual similarities between the methods in class c.

ACSM(c)=0.5 -> C3(c) = 0.5

Shortcomings of C3

- Are two classes with the same C3 value equally cohesive? (SD of the CSM values)
- Measure the influence of highly related methods in a class with a low C3 cohesion
- Define a new measure based on the counting mechanism utilized in LCOM2
  - Do not take into account intersections of methods based on common attribute usage
  - Count intersections of method pairs based on the CSM value between them and the ACSM

Lack of Conceptual Similarity between Methods (LCSM)

- Let $M_i = \{m_j \mid (m_i, m_j) \in E, m_i \neq m_j\}$ be the set of neighbor methods of mi (with which mi has a higher CSM value than the average)
- Let $P = \{(M_i, M_j) \mid M_i \cap M_j = \emptyset\}$
- Let $Q = \{(M_i, M_j) \mid M_i \cap M_j \neq \emptyset\}$
- Lack of conceptual similarity is

$$\text{LCSM(c)} = \begin{cases} |P| - |Q| & if \quad |P| > |Q| \\ else \quad 0 \end{cases}$$

Example of Measuring LCSM

	m1	m2	m3	m4	m5
m1	1	0.21	**0.72**	0.33	0.42
m2		1	0.28	**0.91**	**0.66**
m3			1	0.37	0.27
m4				1	**0.89**
m5					1

	M1	M2	M3	M4	M5
M1		Ø	Ø	Ø	Ø
M2			Ø	m5	m4
M3				Ø	Ø
M4					m4
M5					

ACSM(c) = 0.5
M1 = {m3},
M2 = {m4, m5},
M3 = {m1},
M4 = {m2, m5},
M5 = {m2, m4}

|P| = 7; |Q| = 3;

LCSM(c) = 7-3 =4

Limitations

- C3 and LCSM do not take into account polymorphism and inheritance
- Method invocation, parameters, attribute references, and types are of interest only at identifier level
- C3 and LCSM do not make distinction between constructors, accessors, and other method stereotypes. Some of these methods can artificially increase or decrease cohesion

C3' and LCSM'

- Same measurement as C3 and LCSM
- Eliminate comments from the analysis
- Keep identifiers only
- Assess the influence of comments quality over C3 and LCSM

Case Study

- Compare C3, C3', LCSM, and LCSM' with [$LCOM_1$-$LCOM_5$], Coh, C, ICH, TCC, and LCC

- WinMerge with 51KLOC and 11K comments
- Metrics computed for 34 classes with 522 methods
- Structural metrics computed with Columbus [Ferenc'04], C3 and LCSM – our tool

- Analysis of correlations between metrics

Results

- C3 and C3' very close values (WinMerge has 20% of code as comments)
- LCSM and LCSM' are less conclusive in this respect, but the differences are still not major
- C3 and LCSM do not correlate – interesting!
- Significant correlations between C3 and ICH, and C3 and LCOM5 – not major surprise
- No significant correlation between any structural metric and LCSM – somewhat surprising! – expected LSOM2 to correlate

Interesting Cases

- *"It is after all possible to have a class with high internal, syntactic cohesion but little semantic cohesion"* – Henderson-Sellers

Class name	C3	LCOM2
IVSSItem	0.64	528
IVSSDatabase	0.635	136
IVSSItemOld	0.632	465
BCMenuData	0.434	0
CDirDoc	0.294	0
RescanSuppress	0.392	1

Interesting Cases

- IVSSItem, IVSSDatabase, IVSSItemOld – wrapper classes with few or no data members. Wrappers tend to group together methods that are conceptually similar or have similar usage
- BCMenuData is a class that implements a “property container”. Small class with many accessor methods. Many unique identifiers. Cohesive – C3 did not capture it.
- RescanSuppress only three methods: constr., destr., clear – C3 limited in this case
- On the top and bottom values, C3 and LCOM2 agreed.

Metrics are Complementary

- Structural metrics tell us if a class is built cohesively
- Semantic/conceptual metrics tell us if a class is written cohesively
- We desire both -> increase maintainability

Future Work

- More case studies
- Investigate new measuring mechanism using the semantic information (e.g. like $LCOM_3$)
- Combine with structural metrics