Ranking reusability of software components using coupling metrics

Gui Gui, Paul D. Scott

*JSS 2007*

2008. 12. 9
AhRim Han
Contents

- Introduction
- Established coupling metrics
- Newly proposed coupling metrics
  - WTCoup and WICoup
- Experimentation
- Conclusion and Future Work
- Discussion
This study is started as part of a project of developing a search engine for retrieving Java components from the internet.

Roles of a search engine

- Identify components that offer required functionality
- Assess each retrieved component
  - How fault-free
  - How readily may be adapted for inclusion as part of a larger software system

Reusability
Established coupling metrics to assess the reusability of Java components

- Several empirical studies demonstrated that lower coupling is correlated with fewer faults
- However, established coupling metrics achieved only limited success in predicting modification effort

In this paper,

- Develop new coupling measures to predict the modification effort in the context of
  - Time required
  - Amount of code modification needed to extend the functionality of Java components
## Established coupling metrics (1/3)

### Representative set

- **NLM** = number of local methods in a class
- **NRM** = number of remote methods call from a class

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CBO</strong> (Coupling Between Object classes)</td>
<td>$\text{CBO}(c) = # {\text{class } d \mid \text{a method of class } c \text{ calls a method or references a field of class } d}$</td>
</tr>
<tr>
<td><strong>RFC</strong> (Response set For Classes)</td>
<td>$\text{RFC}(c) = # \text{ methods of } c \ (\text{NLM}) + # \text{ methods of other classes invoked by } c \ (\text{NRM})$</td>
</tr>
<tr>
<td>**RFC}_∞</td>
<td>$\text{RFC}_∞(c) = # \text{ methods of } c \ (\text{NLM}) + # \text{ methods of other classes invoked by } c \ (\text{NRM, recursively})$</td>
</tr>
<tr>
<td><strong>DAC</strong> (Data Abstraction Coupling)</td>
<td>$\text{DAC}(c) = # \text{ attributes in a class that have other class as their type}$</td>
</tr>
<tr>
<td><strong>MPC</strong> (Message Passing Coupling)</td>
<td>$\text{MPC}(c) = # \text{ of messages (method invocations) a class sends to other class}$</td>
</tr>
<tr>
<td><strong>CF (= COF)</strong> (Coupling Factor)</td>
<td>$\text{CF} = \frac{# \text{ non-inheritance coupling}}{# \text{ both inheritance and non-inheritance coupling}}$</td>
</tr>
</tbody>
</table>
Modification effort prediction

- Using six static coupling metrics
- Results
  - (+) Most of the metrics exhibited some significant correlations with reuse effort
  - (-) Metrics fell far short of perfect ranking and their performance was not consistent across different types of components
    - CBO: worst performer
    - RFC\(_\infty\): best performer
Modification effort prediction (Cont’d)

Three limitations of established coupling metrics

A. Only consider direct coupling except RFC∞
   - Intransitive relation

B. Not consider of the extent of the dependency in CBO and CF
   - All-or-nothing quantity

C. Not consider functional complexity of the class being assessed
Newly proposed coupling metrics (1/5)

- Two kinds
  - WICoup (Weighted Intransitive Coupling)
  - WTCoup (Weighted Transitive Coupling)

- Same basis of established coupling metrics
  - Interactions between classes by invoking methods and accessing variables
Background for calculating WICoup and WTcoup

- $C = \{C_1, C_2, \ldots, C_n\}$
  - a system comprising of $n$ classes
- $M_j$
  - set of members of class $C_j$
- $X_{i,j}$
  - set of members of class $C_j$ invoked by class $C_i$
- $X_i = \bigcup_{1 \leq j \leq n} X_{i,j}$
  - set of all members of other classes invoked by class $C_i$
Calculating WICoup

- Establish an intransitive (i.e., direct) coupling measure between classes of \( C_i \) to \( C_j \)
  - Estimation of the proportion of \( C_i \)’s functionality that is obtained from \( C_j \)

\[
WCD(i, j) = \frac{|X_{i,j}|}{|X_i| + |M_i|}
\]

- Construct a measure for an entire system of \( n \) classes

\[
WICoup = \frac{\sum_{i,j=1}^{n} WCD(i, j)}{n^2 - n}
\]

- Total functionality of class \( C_i \) = RFC

| \( M_i \) | : measure of the functionality achieved through its own members
| \( X_i \) | : measure of the functionality achieved through members of other classes
Calculating WTCoup

- Establish a transitive coupling measure between classes of $C_i$ and $C_j$ along path $\pi$
  - Estimation of the proportion of the functionality that $C_i$ obtains from $C_j$ (via other classes lying on the path $\pi$)
  - WCD is a special case of WCT corresponding to a path of length 1

$$WCT(i, j, \pi) = \prod_{\text{edge}(p,q) \in \pi} WCD(p, q) = \prod_{\text{edge}(p,q) \in \pi} \frac{|X_{p,q}|}{|X_p| + |M_p|}$$
Calculating WTCoup (Cont’d)

- Combine the values associated with the various paths between classes of $C_i$ and $C_j$
  - Exclude all paths containing cycles
  - Select the path which has the largest WCT values as a measure

\[
\text{MaxWCT}(i, j) = \text{WCT}(i, j, \pi_{\text{max}}(i, j))
\]

, where \( \pi_{\text{max}}(i, j) = \arg \max_{\pi \in \Pi} \text{WCT}(i, j, \pi) \)

- Construct a measure for an entire system of $n$ classes

\[
\text{WT}Coup = \frac{\sum_{i,j=1}^{n} \text{MaxWCT}(i, j)}{n^2 - n}
\]
Goal

- Evaluate two proposed measures (WTCoupl and WICoup) as predictors of the required modification effort

Study setting

- Total 60 components (3924 classes) retrieved
  - HTML parser (15 components)
  - Lexical Tokenizer (20 components)
  - Barcode (25 components)
- The single experienced Java programmer was required to make extensions to retrieved JAVA components
Study setting (Cont’d)

- Modification effort
  - Number of changes - NLOC that were added, deleted and modified
  - Time required - minutes taken to determine and carry out the requisite changes

- Six established coupling metrics was used for comparison with two proposed measures
  - $CF$, $CBO$, $RFC$, $RFC_{\infty}$, $DAC$, $MPC$
Results

Rank correlation

Spearman rank correlations of coupling measures against NLOC (upper line) and time (lower line)

<table>
<thead>
<tr>
<th></th>
<th>HTML</th>
<th>Lexical T</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTCoup</td>
<td>0.975**</td>
<td>0.951**</td>
<td>0.974**</td>
</tr>
<tr>
<td></td>
<td>0.973**</td>
<td>0.948**</td>
<td>0.973**</td>
</tr>
<tr>
<td>WICoup</td>
<td>0.924**</td>
<td>0.889**</td>
<td>0.910**</td>
</tr>
<tr>
<td></td>
<td>0.916**</td>
<td>0.891**</td>
<td>0.911**</td>
</tr>
<tr>
<td>RFC∞</td>
<td>0.911**</td>
<td>0.846**</td>
<td>0.713**</td>
</tr>
<tr>
<td></td>
<td>0.916**</td>
<td>0.846**</td>
<td>0.712**</td>
</tr>
<tr>
<td>RFC</td>
<td>0.893**</td>
<td>0.823**</td>
<td>0.656**</td>
</tr>
<tr>
<td></td>
<td>0.894**</td>
<td>0.823**</td>
<td>0.659**</td>
</tr>
<tr>
<td>MPC</td>
<td>0.842**</td>
<td>0.770**</td>
<td>0.718**</td>
</tr>
<tr>
<td></td>
<td>0.851**</td>
<td>0.766**</td>
<td>0.715**</td>
</tr>
<tr>
<td>DAC</td>
<td>0.507</td>
<td>0.817**</td>
<td>0.800**</td>
</tr>
<tr>
<td></td>
<td>0.533*</td>
<td>0.820**</td>
<td>0.785**</td>
</tr>
<tr>
<td>CF</td>
<td>0.882**</td>
<td>0.291</td>
<td>0.758**</td>
</tr>
<tr>
<td></td>
<td>0.878**</td>
<td>0.301</td>
<td>0.772**</td>
</tr>
<tr>
<td>CBO</td>
<td>0.465</td>
<td>0.117</td>
<td>0.485*</td>
</tr>
<tr>
<td></td>
<td>0.478</td>
<td>0.135</td>
<td>0.485*</td>
</tr>
</tbody>
</table>

Values of the coefficient of determination ($R^2$) obtained when NLOC is regressed on the coupling metrics

<table>
<thead>
<tr>
<th></th>
<th>HTML</th>
<th>Lexical T</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTCoup</td>
<td>0.850**</td>
<td>0.837**</td>
<td>0.958**</td>
</tr>
<tr>
<td>WICoup</td>
<td>0.822**</td>
<td>0.788**</td>
<td>0.854**</td>
</tr>
<tr>
<td>RFCInf</td>
<td>0.812**</td>
<td>0.756**</td>
<td>0.590**</td>
</tr>
<tr>
<td>RFC</td>
<td>0.786**</td>
<td>0.729**</td>
<td>0.534**</td>
</tr>
<tr>
<td>MPC</td>
<td>0.771**</td>
<td>0.509**</td>
<td>0.507**</td>
</tr>
<tr>
<td>DAC</td>
<td>0.254</td>
<td>0.739**</td>
<td>0.507**</td>
</tr>
<tr>
<td>CF</td>
<td>0.621**</td>
<td>0.097</td>
<td>0.695**</td>
</tr>
<tr>
<td>CBO</td>
<td>0.260</td>
<td>0.010</td>
<td>0.121</td>
</tr>
</tbody>
</table>

** denotes significant at 1% level.

* denotes significant at 5% level; ** denotes significant at 1% level.
Results (Cont’d)

Partial correlation and multiple regression

Result of stepwise multiple regression for the three component groups

<table>
<thead>
<tr>
<th>Measure</th>
<th>First variable</th>
<th>Measure</th>
<th>Second variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML P.</td>
<td>WT Coup</td>
<td>MPC</td>
<td>HTML P.</td>
</tr>
<tr>
<td></td>
<td>0.850</td>
<td>0.045</td>
<td>0.958</td>
</tr>
<tr>
<td></td>
<td>&lt;0.001</td>
<td>0.042</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Lexical T.</td>
<td>WT Coup</td>
<td>DAC</td>
<td>Lexical T.</td>
</tr>
<tr>
<td></td>
<td>0.837</td>
<td>0.077</td>
<td>0.958</td>
</tr>
<tr>
<td></td>
<td>&lt;0.001</td>
<td>0.001</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Barcode</td>
<td>WT Coup</td>
<td>-</td>
<td>Barcode</td>
</tr>
<tr>
<td></td>
<td>0.958</td>
<td>-</td>
<td>0.958</td>
</tr>
<tr>
<td></td>
<td>&lt;0.001</td>
<td>-</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>
Results (Cont’d)

Discussion

- WTcoup and $R_\infty$ performed good
  - Advantageous of taking account of transitive metrics
- CBO and CF was consistently poor
  - Treated coupling as a simple binary quantity
- WICoup and WTcoup significantly outperformed than other metrics
  - Considered total functionality of a class as a normalizing factor in assessing the extent
Conclusion and Future Work

- Developed two new coupling measures
- Proved these to be successful in predicting modification effort
  - Time required
  - Amount of code modification needed to extend the functionality of Java components
Discussion – Need to be improved (1/3)

- Complex dependencies
  - Transition relationship
Discussion – Need to be improved (2/3)

- Complex dependencies (Con’d)
  - Inheritance relationship

![Diagram of complex dependencies involving classes A, B, C, and D with inheritance arrows and possible alternatives.]

or
Measure of the overall coupling between two classes

- Ignored couplings via alternative paths and this may definitely affect the accuracy of prediction
- Need to develop a method for combining the values associated with various paths