Cohesion and Coupling Metrics on Systems of Systems

To-ju Huang
Computer Science Department
University of California, San Diego
October 9, 2009
Outline

- Background
  - Separation of concerns and modularity
  - Cohesion and coupling
- Survey of C&C metrics
  - Coupling metrics
  - Cohesion metrics
- C&C metrics revisited regarding SoS
  - Revisiting coupling metrics
  - Revisiting cohesion metrics
- Summary
Decomposing software into smaller pieces (modules) is an important application of separation of concerns [GHEZZI].

Separation of concerns (SoC) is commonly seen in daily life [GHEZZI]
- E.g. software lifecycle, network layering

Good modularity helps achieve
- Productivity
- Comprehensibility
- Maintainability
Cohesion and Coupling

- Two important module measures
  - Cohesion: how a module is focused
    - The more a module is cohesive [YOURDON]
      - Comprehensibility increases
      - Maintainability increases
  - Coupling: how modules interrelate [YOURDON]
    - if we keep it low as much as possible
      - Productivity increases
      - Comprehensibility increase
      - Maintainability increases
Outline

- Background
  - Separation of concerns and modularity
  - Cohesion and coupling
- Survey of C&C metrics
  - Coupling metrics
  - Cohesion metrics
- C&C metrics revisited regarding SoS
  - Revisiting coupling metrics
  - Revisiting cohesion metrics
- Summary
Coupling metrics

- Ordinal (not metrics)
- Static metrics
- Slicing-based metrics
- Object-oriented metrics
Ordinal levels of coupling [LAWRENCE]

- Content coupling
- Common coupling
- Control coupling
- Stamp coupling
- Data coupling

- Tight:
  - Modifying internal data of the other
  - Sharing global data
  - Passing control flag
  - Passing composite data
  - Passing pure data

- Loose
Static coupling metrics

- Dhama’s definition of coupling [Dhama]

\[
\frac{1}{(i_d + u_d + g_d) + 2(i_c + u_c + g_c) + (w + r)}
\]

favors loose coupling

data coupling

control coupling

environment coupling
Slice-based coupling metrics \[\text{[HARMAN]}\]

- Information flow between modules \(f\) and \(g\) percentage of \(f\)'s elements which are included in \(g\)'s slice

\[
FF_{(f,g)}^P = \frac{FF_{(f,g)}^P \times \text{length}(f) + FF_{(g,f)}^P \times \text{length}(g)}{\text{length}(f) \times \text{length}(g)}
\]

- Coupling metric between modules \(f\) and \(g\)

- Coupling metric of a module \(f\)

\[
\frac{\sum_i \text{coupling}(f,g_i) \times |g_i|}{\sum_i |g_i|}
\]
Object-oriented coupling metrics

- CBO (Coupling Between Objects) [CHIDAMBER]
  - Number of other classes that a class is coupled to
Cohesion metrics

- Ordinal (not metrics)
- Static metrics
- Slicing-based metrics
- Object-oriented metrics
Ordinal levels of cohesion [YOURDON]

- Coincidental
- Logical
- Temporal
- Procedural
- Communicational
- Sequential
- Functional

- Not related at all
- In the same logical category
- Temporally close
- Close In the a process
- Around data
- Sequentially chained
- Implement a single function
Static cohesion metrics

Dhama’s cohesion metrics is the summation of four cohesion measurements [DHAMA]

- Functional cohesion
  - Inverse of weight variables counts
  - The more variables used in a module, the more functionality it convey, the less focused it is

- Data flow cohesion
  - Counts of pairs of statements linked by data flow, divided by the statements distance

- Action-bundling cohesion
  - Counts of pairs of statements performed on the same

- Logical bundling cohesion
  - The deeper a block of statements are nested, more cohesive
  - The bigger a block is, the more cohesive it is
# Slicing-based cohesion metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
<th>Key component</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tightness</strong> [WEISER]</td>
<td>$\frac{</td>
<td>SL_{\text{int}}</td>
</tr>
<tr>
<td><strong>MinCoverage</strong> [OTT]</td>
<td>$\min_i</td>
<td>SL_i</td>
</tr>
<tr>
<td><strong>Coverage</strong> [WEISER]</td>
<td>$\frac{1}{</td>
<td>V_o</td>
</tr>
<tr>
<td><strong>MaxCoverage</strong> [OTT]</td>
<td>$\max_i</td>
<td>SL_i</td>
</tr>
<tr>
<td><strong>Overlap</strong> [WEISER]</td>
<td>$\frac{1}{</td>
<td>V_o</td>
</tr>
</tbody>
</table>
Object-oriented cohesion metrics

- LCOM (Lack of Cohesion on Methods) [CHIDAMBER]
  - The number of methods that access one or more of the same attributes
  - The more methods have to share attributes, the less they fulfill features individually
- LCOM has been refined a lot since invented; LCOM5 is the latest refinement
Outline

- Background
  - Separation of concerns and modularity
  - Cohesion and coupling (C&C)
- Survey of C&C metrics
  - Coupling metrics
  - Cohesion metrics
- C&C metrics revisited regarding SoS
  - Revisiting coupling metrics
  - Revisiting cohesion metrics
- Summary
Revisit “coupling” metrics

In SoS realm, some cases of tight coupling are allowed and utilized

- Content coupling e.g. template engine, VM
- Common coupling e.g. Shared libraries
- Control coupling
- Stamp coupling
- Data coupling (No coupling)

Not problematic because they do not change frequently
Revisit “coupling” metrics

- Dhama’s definition of coupling

\[
\frac{1}{(i_d + u_d + g_d) + 2(i_c + u_c + g_c) + (w + r)}
\]

Low level details may not be available due to lack of source code

Far more other than “variables” can cause coupling, e.g. specification, messages, events
Revisit “coupling” metrics

- Slice-based coupling of a module $f$

\[
\frac{\sum_i \text{coupling}(f, g_i) \times |g_i|}{\sum_i |g_i|}
\]

where

\[
\text{coupling}(f, g) = \frac{FF^P_{(f, g)} \times \text{length}(f) + FF^P_{(g, f)} \times \text{length}(g)}{\text{length}(f) \times \text{length}(g)}
\]

Boundary of authority domains and lack of source code and may thwart the calculation of slicing
Revisit “coupling” metrics

- CBO (Coupling Between Objects)
  - Number of other classes that a class is coupled to

Can perfectly be applied to systems of systems
Revisit “coupling” metrics

- Whether coupling is suitably tight depends on types of changes and frequencies.
- When defining coupling on SoS, a broader spectrum of “modules” should be considered, e.g. specification, virtual machines, middleware.
- The “interdependence” between “modules” should be extended to include connections other than variables, method calls. For example, message passing, statements in a specification, event, interception which are performed by middleware.
Revisiting “cohesion” metrics

- Dhama’s cohesion metrics is the summation of four cohesion measurements
  - Functional cohesion
    - According to the definition, a large system is of low cohesion
  - Data flow cohesion
  - Action-bundling cohesion
  - Logical bundling cohesion

  Good for represent coordination-level coherence

  Not clear
## Revisiting “cohesion” metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
<th>Key component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tightness</td>
<td>( \frac{</td>
<td>SL_{int}</td>
</tr>
<tr>
<td>MinCoverage</td>
<td>( \min_i</td>
<td>SL_i</td>
</tr>
<tr>
<td>Coverage</td>
<td>( \frac{1}{</td>
<td>V_o</td>
</tr>
<tr>
<td>MaxCoverage</td>
<td>( \max_i</td>
<td>SL_i</td>
</tr>
<tr>
<td>Overlap</td>
<td>( \frac{1}{</td>
<td>V_o</td>
</tr>
</tbody>
</table>

Slices may be hard to get. Probably dynamic slices may help?
Revisiting “cohesion” metrics

- Systems of systems are usually complex, providing variety of services
- It is still not clear what it means by a system which focused on single function
- Defining cohesion on coordination could probably the a suitable direction
  - E.g. a coordination embed two parallel, independent set of message patterns
References


Summary

- Concepts of cohesion of coupling
- Survey of cohesion and coupling metrics
  - Ordinal levels
  - Static metrics
  - Slice-based metrics
  - Object-oriented metrics
- Revisit C&C metrics regarding SoS
  - Coupling makes more sense in the light of change types and frequency
  - Service-level cohesion is still a vague idea
Thank you!