Combining Conceptual and Domain-Based Couplings to Detect Database and Code Dependencies

Malcom Gethers, Amir Aryani, Denys Poshvyvanyk

RMIT University, Australia

2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation
Common problems in software maintenance

- Systems with legacy code, e.g., COBOL
- Hybrid systems, e.g., Python and Java
- Multi-tier systems
- Inaccessible maintenance history
Despite the issues...

- Perform impact analysis
- Information suitable for domain experts
- User Interface Components (UIC)
How?

- Domain-based coupling
- Conceptual coupling
- Combination
Motivations

- Domain-based approach works without access to source code or design documents.
- Conceptual coupling approach is language independent.
- The approaches complement each other.
Example of UICs

Domain variables
Case Study

- 120,111 times downloaded in 2011
- 3,531 Java Classes
- 2,569,854 lines of code
- Four distinct interfaces
- 347 screens
Dependencies

- **Architectural dependencies**: 17,605
- **Source code dependencies**: 14,898
- **Database dependencies**: 20,310

M. Lungu and M. Lanza, Softwareaut, CSMR 2006
Case Study - Orthogonality

Do conceptual and domain-based coupling identify orthogonal dependencies?
## Case Study - Orthogonality

<table>
<thead>
<tr>
<th>Architectural Dependencies (UICs)</th>
<th>CP 10</th>
<th>CP 20</th>
<th>CP 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>C (int) D</td>
<td>26%</td>
<td>26%</td>
<td>26%</td>
</tr>
<tr>
<td>C (diff) D</td>
<td>35%</td>
<td>38%</td>
<td>39%</td>
</tr>
<tr>
<td>D (diff) C</td>
<td>25%</td>
<td>25%</td>
<td>25%</td>
</tr>
</tbody>
</table>

- **C (int) D**: Set intersection of correct dependencies identified by both conceptual and domain-based coupling
- **C (diff) D**: Set difference of correct dependencies identified by conceptual and domain-based coupling
- **D (diff) C**: Set difference of correct dependencies identified by conceptual and domain-based coupling
# Case Study - Orthogonality

<table>
<thead>
<tr>
<th>Architectural Dependencies (UICs)</th>
<th>CP 20</th>
<th>CP 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>C (int) D</td>
<td>26%</td>
<td>26%</td>
</tr>
<tr>
<td>C (diff) D</td>
<td>35%</td>
<td>38%</td>
</tr>
<tr>
<td>D (diff) C</td>
<td>25%</td>
<td>25%</td>
</tr>
</tbody>
</table>

**Metrics are orthogonal!**

- **C (int) D**: Set intersection of correct dependencies identified by both conceptual and domain-based coupling.
- **C (diff) D**: Set difference of correct dependencies identified by conceptual and domain-based coupling.
- **D (diff) C**: Set difference of correct dependencies identified by conceptual and domain-based coupling.
Case Study - Accuracy

Does combing conceptual and domain-based coupling improve the accuracy our ability to identify dependencies?
Case Study - Accuracy

Is it possible to improve the accuracy?

- Conceptual
- Domain
- Conceptual + Domain
Case Study - Accuracy

The combination of conceptual and domain dependencies yields an improvement for identifying dependencies.

Wilcoxon sign-ranked test indicates our findings are typically statistically significant.
Case Study - Accuracy

The combination of conceptual and domain dependencies yields an improvement for identifying dependencies. The combination outperforms either individual technique. Wilcoxon sign-ranked tests indicate our findings are typically statistically significant.
Conclusion

Conceptual and domain-based coupling identify orthogonal sets of dependencies

Combining the metrics improves our ability to predict dependencies

Recall improvements of up to 7% over the baseline approach

Precision improvement up to 24% over the baseline approach
Thank You

SEMERU @ William and Mary

http://www.cs.wm.edu/semeru