Do Crosscutting Concerns Cause Defects?


Presented by Danilo Ferreira e Silva
Motivation

Enormous effort goes into avoiding software defects

- Efforts might be better directed if we had a better understanding of their causes
- Empirical studies provide evidence that crosscutting concerns impact internal quality metrics

(But, what about external quality?)

This study considers the possibility that crosscutting concerns impact one external quality: defects
Methodology

- Formal model to measure the extent to which concerns are crosscutting
- Three case studies to gather data on scattering and defect counts
- Correlation analysis between scattering and defects
Methodology

- Formal model to measure the extent to which concerns are crosscutting
- Three case studies to gather data on scattering and defect counts
- Correlation analysis between scattering and defects

Moderate to strong correlation for all three case studies
A Model of Concerns

- $S$ is a set of concerns (items from a program’s specification)
- $T$ is a set of program elements (AST nodes)
- $R = \{(s, t) \mid s \in S, t \in T\}$
  (relationship between concerns and program elements)
Metrics

- Bug Count
- Lines of Concern Code (LOCC)
- Concern Diffusion over Components (CDC)
- Concern Diffusion over Operations (CDO)
- Degree of Scattering across Classes (DOSC)
- Degree of Scattering across Methods (DOSM)
Metrics: Degree of Scattering

\[
CONC(s, t) = \frac{\text{Source lines in element } t \text{ related to concern } s}{\text{Source lines related to concern } s}
\]

\[
\text{Variance}(s) = \frac{\sum_{t \in T} (CONC(s, t) - CONC_{\text{worst}})}{|T|}
\]

\[
DOS(s) = 1 - \frac{\text{Variance}(s)}{\text{Variance}_{\text{ideal}}(s)}
\]

\[
\text{DOSC} = 1.00 \\
\text{CDC} = 4
\]

\[
\text{DOSC} = 0.08 \\
\text{CDC} = 4
\]
Bug-concern mapping

1. Reverse engineer the concern-code mapping (manual)
   1.1 Concern selection
   1.2 Concern assignment (prune dependency rule)

2. Mine the bug-code mapping (partially automated)
   2.1 Associating bugs with bug fixes
   2.2 Associating bugs with program elements

3. Infer the bug-concern mapping
Bug-concern mapping

1. Reverse engineer the concern-code mapping (manual)
   1.1 Concern selection
   1.2 Concern assignment (prune dependency rule)

2. Mine the bug-code mapping (partially automated)
   2.1 Associating bugs with bug fixes
   2.2 Associating bugs with program elements

3. Infer the bug-concern mapping
Bug-concern mapping

1. Reverse engineer the concern-code mapping (manual)
   1.1 Concern selection
   1.2 Concern assignment (prune dependency rule)

2. Mine the bug-code mapping (partially automated)
   2.1 Associating bugs with bug fixes
   2.2 Associating bugs with program elements

3. Infer the bug-concern mapping
Bug-concern mapping

1. Reverse engineer the concern-code mapping (manual)
   1.1 Concern selection
   1.2 Concern assignment (prune dependency rule)
2. Mine the bug-code mapping (partially automated)
   2.1 Associating bugs with bug fixes
   2.2 Associating bugs with program elements
3. Infer the bug-concern mapping
Case Studies

Project characteristics:

- Open-source
- Java (tooling limitation)
- Production quality
- Identifiable concerns (at least 30)
- Accessible Issue Tracking System (ITS)
- Consistently referenced bugs (commit messages)
Case Study 1

Mylyn-Bugzilla

► Plug-in for the Eclipse IDE that enables task-focused methodology
► 28 concerns identified
  – Convert query hits to tasks
  – Support search for duplicates
► Concerns and bugs assigned manually
Case Study 2

Rhino

- Javascript interpreter
- Hierarchy of 480 concerns (357 leaves) extracted from the ECMAScript Standard
  - Regular Expression Literals
  - Scope Chain and Identifier Resolution
- Concerns assigned manually
Case Study 3

iBATIS

- Object-relational mapping tool
- Hierarchy of 183 concerns (132 leaves) extracted from the Developer’s Guide
  - Caching
    - Request Caching
    - Class Caching
- Concerns assigned manually
### Coverage Statistics

<table>
<thead>
<tr>
<th></th>
<th><strong>Mylyn-Bugzilla</strong></th>
<th></th>
<th><strong>Rhino</strong></th>
<th></th>
<th><strong>iBATIS</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>All</strong></td>
<td><strong>Mapped</strong></td>
<td><strong>%</strong></td>
<td><strong>All</strong></td>
<td><strong>Mapped</strong></td>
<td><strong>%</strong></td>
</tr>
<tr>
<td><strong>Classes</strong></td>
<td>56</td>
<td>44</td>
<td>79</td>
<td>138</td>
<td>80</td>
<td>57</td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td>427</td>
<td>253</td>
<td>59</td>
<td>1870</td>
<td>1415</td>
<td>75</td>
</tr>
<tr>
<td><strong>Fields</strong></td>
<td>457</td>
<td>230</td>
<td>50</td>
<td>1339</td>
<td>962</td>
<td>71</td>
</tr>
<tr>
<td><strong>Lines</strong></td>
<td>13649</td>
<td>5914</td>
<td>43</td>
<td>32134</td>
<td>28308</td>
<td>88</td>
</tr>
<tr>
<td><strong>Concerns</strong></td>
<td>28</td>
<td>28</td>
<td>100</td>
<td>480</td>
<td>417</td>
<td>86</td>
</tr>
<tr>
<td><strong>Bugs</strong></td>
<td>110</td>
<td>101</td>
<td>92</td>
<td>241</td>
<td>160</td>
<td>66</td>
</tr>
</tbody>
</table>
## Results

### Spearman correlation coefficients

<table>
<thead>
<tr>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.64</td>
<td>.84</td>
<td>.57</td>
<td>.38</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.77</td>
<td>.91</td>
<td>.63</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.65</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.71</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

(a) Mylyn-Bugzilla

<table>
<thead>
<tr>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.62</td>
<td>.96</td>
<td>.74</td>
<td>.60</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.63</td>
<td>.88</td>
<td>.68</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.80</td>
<td>.67</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.80</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

(b) Rhino

<table>
<thead>
<tr>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.67</td>
<td>.90</td>
<td>.73</td>
<td>.43</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.67</td>
<td>.90</td>
<td>.64</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.55</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.77</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

(c) iBATIS
## Results

### Spearman correlation coefficients

<table>
<thead>
<tr>
<th></th>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.64</td>
<td>.84</td>
<td>.57</td>
<td>.38</td>
<td>.39</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.77</td>
<td>.91</td>
<td>.63</td>
<td>.50</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.65</td>
<td>.57</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.71</td>
<td>.61</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.62</td>
<td>.96</td>
<td>.74</td>
<td>.60</td>
<td>.67</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.63</td>
<td>.88</td>
<td>.68</td>
<td>.66</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.80</td>
<td>.67</td>
<td>.73</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.80</td>
<td>.77</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.90</td>
</tr>
</tbody>
</table>

(a) Mylyn-Bugzilla

<table>
<thead>
<tr>
<th></th>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.67</td>
<td>.90</td>
<td>.73</td>
<td>.43</td>
<td>.46</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.67</td>
<td>.90</td>
<td>.64</td>
<td>.29</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.55</td>
<td>.58</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.77</td>
<td>.44</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.53</td>
</tr>
</tbody>
</table>

(b) Rhino

<table>
<thead>
<tr>
<th></th>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.67</td>
<td>.90</td>
<td>.73</td>
<td>.43</td>
<td>.46</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.67</td>
<td>.90</td>
<td>.64</td>
<td>.29</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.55</td>
<td>.58</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.77</td>
<td>.44</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.53</td>
</tr>
</tbody>
</table>

(c) iBATIS

- Moderate to strong correlation for all three case studies
### Results

#### Spearman correlation coefficients

<table>
<thead>
<tr>
<th></th>
<th>DOSM</th>
<th>CDC</th>
<th>CDO</th>
<th>LOCC</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSC</td>
<td>.64</td>
<td>.84</td>
<td>.57</td>
<td>.38</td>
<td>.39</td>
</tr>
<tr>
<td>DOSM</td>
<td>—</td>
<td>.77</td>
<td>.91</td>
<td>.63</td>
<td>.50</td>
</tr>
<tr>
<td>CDC</td>
<td>—</td>
<td>—</td>
<td>.78</td>
<td>.65</td>
<td>.57</td>
</tr>
<tr>
<td>CDO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.71</td>
<td>.61</td>
</tr>
<tr>
<td>LOCC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>.77</td>
</tr>
</tbody>
</table>

- **(a) Mylyn-Bugzilla**
- **(b) Rhino**
- **(c) iBATIS**

- Moderate to strong correlation for all three case studies
- CDC and CDO were more strongly correlated
Testing for the Confounding Effect of Size

- The size of the concern has a strong correlation with defects
- There is also a strong correlation between scattering metrics and size
- Test for a confounding effect
  - Stepwise Regression Analysis
  - Principal Component Analysis (PCA)

Conclusion
Size is not the single dominating factor, the scattering metrics contribute toward explaining the variance in bug count
Threats to Validity

Internal Validity
- Concern assignment unreliability
- Bug assignment errors
- Concern and bug assignments at the member level (not at statement level)

External Validity
- Results may not generalize
  - Programming language
  - Complexity of the problem domain
  - Tool support
  - etc
Conclusion

Main Contribution
Empirical evidence suggesting that crosscutting concerns cause defects

▶ Further studies are needed to draw general conclusions

Remaining Questions

▶ Can we reduce the likelihood of defects by reducing crosscutting?
▶ Are crosscutting concerns a byproduct of programming technology, developer aptitude, or the inherent complexity of the concern?
▶ What is the relationship between code churn and scattering?
Thanks!