Case Study: Supplementing Program Analysis with Natural Language Analysis to Improve a Reverse Engineering Task

David Shepherd, Lori Pollock, and K. Vijay-Shanker
University of Delaware
Consider the Reverse Eng Task…

Aspect-Oriented Programming, language support for:

Aspect Mining

Locates refactoring candidates
Classic Aspect Candidates

**Before/After Aspects** [Mastering AspectJ, Aspectj.com]

- Circle.setRadius(x){
  
  \[ r = x; \]

  Display.update()

  
}

- Square.setSize(x){

  \[ size = x; \]

  Display.update()

  
}

- Aspect Updater

  After Shape.set*()

  Call Display.update()

**Certain Design Patterns** [Hanneman 02]

- public MultiFigure()

  numFigs = n; ....

  addListener(Drawing)

}

- public Square(y){

  side = y;

  addListener(MultiFigure)

  
}

- Aspect Listener

  After Figure.construct()

  Add parent as listener
Current Aspect Mining Analyses

Traditional Program Analyses (TPA)

Fan-In Analysis
[Marin et al. WCRE 2005]

Code Clone Analysis
[Shepherd SERP 04, Bruntink ICSM 04]

Traditional Program Analyses (TPA) can be effective
Consider Natural Language Analyses (NLA)

Scenario

✓ Class: DrawApplication
✓ Methods
  ✓ addViewChangeListener
  ✓ removeViewChangeListener

Opposite Verbs

Natural Language Analyses can be effective
Quiz: Is TPA Enough?

Which method(s) are good candidates?

Traditional Program Analyses (TPA) can be misleading
Quiz: Is NLA Enough?

Scenario A
✓ Method uses verb “add”
✓ Another method in the same class uses verb “remove”

Scenario B
✓ Method uses verb “add”
✓ Another method in the same class uses verb “remove”

Natural Language Analyses can be misleading
Summary: TPA & NLA Complementary

Using NLA and TPA together to build a case for refactoring.

Supplementing TPA with NLA makes a stronger case for refactoring.

Method
Auctions.startBlocking()

Fan-in is high
Opposite verb is used
Called in a clone
Talk Progression

✓ Aspect Mining Framework: Timna

✓ Integration of Natural Language in iTimna

✓ Evaluation
Timna: An Aspect Mining Framework

**Phase 1: Learning (occurs once)**

- Tagged Program
- Mining Analyses
- Analyses Results
- Learner
- Mining Rules

**Method name | Fan-in | Code Clone | Is Void**

- `openWindow`: 3 | T | T
- `addListener`: 7 | F | F

If (Fan-in > 3 and Is Void) then the method is a refactoring candidate

**Phase 2: Classifying (occurs many times)**

- Program
- Mining Analyses
- Analyses Results
- Classifier
- Marked Program
iTimna: Adding NLA to Timna

TPA Analyses
✓ Code clone
✓ Signature analysis
✓ Callsite pairings

NLA Analyses
✓ Opposite verbs
✓ Observer words
✓ Constraint verbs

Add the NL Analyses here
An Example NLA

✓ NLA: Past-Tense Verb
✓ Intuition: Past-Tense indicates reactionary
✓ Process:
  1. Extract verb [AOSD 06]
  2. Use morphological analysis to detect verb tense
✓ Examples:
  ✓ figureSelectionChanged()  $\Rightarrow$ changed $\Rightarrow$ past-tense
  ✓ setSelection  $\Rightarrow$ set $\Rightarrow$ not past-tense
  ✓ drawingINVALIDated()  $\Rightarrow$ invalidated $\Rightarrow$ past-tense
Evaluation

✔ Question: What effect does integrating NLA with TPA have on AM?

✔ Methodology:
Manually tag a program, Use AM tools, Check results

✔ Measures: Precision (quality) and Recall (completeness)

✔ Subject Program: JHotDraw (2,739 methods)

✔ AM Tools: Fan-in, Timna, iTimna
### Quantitative Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan-in</td>
<td>37</td>
<td>2</td>
</tr>
<tr>
<td>Timna</td>
<td>62</td>
<td>60</td>
</tr>
<tr>
<td>iTimna</td>
<td>81</td>
<td>73</td>
</tr>
</tbody>
</table>
Qualitative Results

NLA and Simple Program Information

```java
private void endEdit() {
    if (fURLTarget != null) {
        setURL(fURLTarget, fTextField.getText());
        fURLTarget = null;
        fTextField.endOverlay();
    }
}
```

If (Void Return Type & Opposite Verb & No Parameter)

Then the method is a candidate
Qualitative Results

NLA and Simple Program Information

If (Not Void Return Type && Constraint Verb)
Then the method is a candidate
Related Work

- **Mining Dynamic Information**
  [Breu ASE 04, Ceccato WCRE 04]

- **Mining Static Information**
  [Marin WCRE 04, Shepherd SERP 04, Bruntink ICSM 04]

- **Combining/Comparing Approaches**
  - **Sorts** [Marin WCRE 2006]
  - **Qualitative Comparison** [Tourwe IWPC 05]
Conclusions

✓ Created NLAs for Aspect Mining
✓ Showed that TPA and NLA can work together
✓ Evaluated prototype, shows strong promise

Supplementing TPA with NLA improves the effectiveness of our aspect mining framework and warrants further evaluation