CONCEPTS EXTRACTION FROM EXECUTION TRACES

Soumaya Medini

Ph.D. Defense

Montréal, November 17th 2014
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work
Software maintenance effort is estimated to be more than 70% of the overall software cost. [Ian Sommerville, 2000]

Program comprehension require half of the effort devoted to software maintenance and evolution. [Dehaghani et Hajrahimi, 2013]
Context

- **Understand a program**: identify which concept this program implements.
- Concept location aims at identifying concepts and locating them within code regions.
- A concept represents a **functionality** of a program.
Motivation

• A typical scenario in which concept location takes part:
Motivation

• A typical scenario in which concept location takes part:

Program Failure

A problem has been detected and Windows has been shut down to prevent damage to your computer.

Fix Broken Hardware or Driver

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

1. Check to make sure your hardware or software is properly installed.
2. Check to make sure your hardware or software is properly configured.
3. Check to make sure your hardware or software is properly updated.
4. Check to make sure your hardware or software is properly installed.

If problems continue after reinstalling or updating your hardware or software, try running Windows Setup again to restore your system to its original state.

Technical Information:

*** STOP: 0x0000000A (0x00000000, 0x00000002, 0x00000000, 0x807CB57A)
Motivation

- A typical scenario in which concept location takes part:

Program Failure
Motivation

• A typical scenario in which concept location takes part:

Program Failure

Execution Trace
Motivation

• A typical scenario in which concept location takes part:

Program Failure

Analysing the one execution trace to identify the sequence of methods producing the failure.

Execution Trace
Problem Statement

- Large and noisy:
  - Execution trace corresponding to draw a rectangle in JHotDraw contains 4,000 method calls.
Problem Statement

- Several approaches address these problems:
  - Compacting execution traces (encoding the whole execution as a directed acyclic graph) [Reiss and Renieris, 2001]
  - Building high-level behavioural models (detecting and filtering utilities) [Hamou-Lhadj et al., 2005]
  - Segmenting execution traces (textual analysis or clustering algorithms) [Asadi et al., 2010] [Pirzadeh and Hamou-Lhadj, 2011]
Problem Statement

• Several approaches address these problems:
  
  • Compacting execution traces (encoding the whole execution as a directed acyclic graph) [Reiss and Renieris, 2001]
  
  • Building high-level behavioural models (detecting and filtering utilities) [Hamou-Lhadj et al., 2005]
  
  • Segmenting execution traces (textual analysis or clustering algorithms) [Asadi et al., 2010] [Pirzadeh and

None of these approaches guide developers towards segments that implements the concepts to maintain.
Identify concepts and facilitate the analysis of large execution traces for maintenance tasks.
Thesis

Execution Trace
Thesis

Execution Trace

Trace Segmentation
Thesis

Execution Trace

Trace Segmentation
Thesis

Execution Trace

Trace Segmentation

Segments Labelling
Thesis

Execution Trace

Trace Segmentation

Segments Labelling
Thesis

Execution Trace

Trace Segmentation

Segments Labelling
Thesis

Execution Trace

Trace Segmentation

Segments Labelling

Segments Relations
Outline

- Context
- Problem Statement
- **Trace Segmentation**
  - Segments Merging
  - Segments Labelling
  - Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work
Trace Segmentation

- **Asadi et al. [2010]**: identify concepts in execution trace by finding cohesive and decoupled fragments of the trace using Genetic Algorithm (GA).

- **Limitations**:
  - Not scalable (7 hours).
  - Stability problems (different segmentation).
Background

• **Steps:**

1. System instrumentation and trace collection;
2. Pruning and compressing traces;
3. Textual analysis of method source code;
4. Trace splitting using optimization techniques.
Step 1: Program instrumentation and trace collection

- We collect and tag traces.
Step 2: Pruning and compressing traces

- **Pruning:** Remove too frequent method invocations.

- **Compressing:** Remove repetitions.

```
m1 m1 m1 m1 m1  ->  m1
```
```
m1 m2 m1 m2 m1 m1 m2  ->  m1 m2
```
Step 3: Textual analysis of Method source code

- Extract identifiers from source code and comments.
- Perform stemming (waited, waiting, waits → wait).
- Remove programming language keywords and English stop words.
- Index terms and documents using the TF-IDF indexing mechanisms and apply LSI.
Step 4: Trace Splitting through optimization techniques

- Execution trace segmentation solution must be found in large search spaces.

- We must apply some optimization techniques to segment the trace.

- Approach built upon a dynamic programming algorithm to:
  - Improve scalability;
  - Compute the exact splitting.
Dynamic Programming (DP) Approach

• Solve a problem by dividing the problem into sub-problems that are recursively solved.

• The solution of the problem: combining the solutions of the sub-problems.

• The quality of the segmentation of a trace into K segments:

\[
fit(\text{segmentation}) = \frac{1}{K} \times \sum_{i=1}^{K} \frac{COH_i}{COU_i + 1}
\]
Cohesion and Coupling

• Cohesion
Cohesion and Coupling

- Cohesion
Cohesion and Coupling

- Cohesion
Cohesion and Coupling

• Cohesion

• Coupling
Dynamic Programming (DP) Approach

• Example of trace segmentation using DP.

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>m2</td>
<td>...</td>
<td>m30</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>m79</td>
<td>...</td>
<td>m90</td>
</tr>
<tr>
<td>m91</td>
<td>...</td>
<td>m133</td>
<td>m134</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>m445</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

- Create a new segment.
Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>m2</td>
<td>...</td>
<td>m30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>m79</td>
<td>m90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m133</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m134</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m445</td>
</tr>
</tbody>
</table>

- Create a new segment.

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>m2</td>
<td>...</td>
<td>m30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>m79</td>
<td>m90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m133</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m134</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m445</td>
</tr>
</tbody>
</table>

- Add the method to the last segment.
Case Study Design

• Research Questions:
  • RQ1: How do the performances of the GA and DP approaches compare?
  • RQ2: How do the GA and DP approaches perform?

• Programs:
Case Study Results

- **RQ1**: How do the performances of the GA and DP approaches compare?

<table>
<thead>
<tr>
<th>Programs</th>
<th>Scenarios</th>
<th>Number of Segments</th>
<th>Fitness</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>24</td>
<td>0.54</td>
<td>7,080</td>
</tr>
<tr>
<td></td>
<td>Create Class, Create Note</td>
<td>73</td>
<td>0.52</td>
<td>10,800</td>
</tr>
<tr>
<td></td>
<td>Draw Rectangle</td>
<td>17</td>
<td>0.39</td>
<td>2,040</td>
</tr>
<tr>
<td></td>
<td>Add Text, Draw Rectangle</td>
<td>21</td>
<td>0.38</td>
<td>1,260</td>
</tr>
<tr>
<td></td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>56</td>
<td>0.46</td>
<td>1,200</td>
</tr>
<tr>
<td></td>
<td>Spawn Window, Draw Circle</td>
<td>63</td>
<td>0.34</td>
<td>240</td>
</tr>
<tr>
<td>JHotDraw</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case Study Results

- **RQ1**: How do the performances of the GA and DP approaches compare?

<table>
<thead>
<tr>
<th>Programs</th>
<th>Scenarios</th>
<th>Number of Segments</th>
<th>Fitness</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GA</td>
<td>DP</td>
<td>GA</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>24</td>
<td>13</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>Create Class, Create Note</td>
<td>73</td>
<td>19</td>
<td>0.52</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle</td>
<td>17</td>
<td>21</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>Add Text, Draw Rectangle</td>
<td>21</td>
<td>21</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>56</td>
<td>20</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>Spawn Window, Draw Circle</td>
<td>63</td>
<td>26</td>
<td>0.34</td>
</tr>
</tbody>
</table>
Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?

<table>
<thead>
<tr>
<th>Programs</th>
<th>Scenarios</th>
<th>Number of Segments</th>
<th>Fitness</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GA</td>
<td>DP</td>
<td>GA</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>24</td>
<td>13</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>Create Class, Create Note</td>
<td>73</td>
<td>19</td>
<td>0.52</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle</td>
<td>17</td>
<td>21</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>Add Text, Draw Rectangle</td>
<td>21</td>
<td>21</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>56</td>
<td>20</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>Spawn Window, Draw Circle</td>
<td>63</td>
<td>26</td>
<td>0.34</td>
</tr>
</tbody>
</table>
Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?

<table>
<thead>
<tr>
<th>Programs</th>
<th>Scenarios</th>
<th>Number of Segments</th>
<th>Fitness</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>24</td>
<td>GA 0.54</td>
<td>DP 0.58</td>
</tr>
<tr>
<td></td>
<td>Create Class, Create Note</td>
<td>73</td>
<td>GA 0.52</td>
<td>DP 0.60</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle</td>
<td>17</td>
<td>GA 0.39</td>
<td>DP 0.67</td>
</tr>
<tr>
<td></td>
<td>Add Text, Draw Rectangle</td>
<td>21</td>
<td>GA 0.38</td>
<td>DP 0.69</td>
</tr>
<tr>
<td></td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>56</td>
<td>GA 0.46</td>
<td>DP 0.72</td>
</tr>
<tr>
<td></td>
<td>Spawn Window, Draw Circle</td>
<td>63</td>
<td>GA 0.34</td>
<td>DP 0.69</td>
</tr>
</tbody>
</table>
Case Study Results

• RQ1: How do the performances of the GA and DP approaches compare?
  • Wilcoxon test and Cliff’s delta effect size:
    ✅ Difference of the number of segments;
    ✅ Values of fitness function;
    ✅ Computation times.
Case Study Results

• RQ1: How do the performances of the GA and DP approaches compare?
  • Wilcoxon test and Cliff’s delta effect size:
    ✓ Difference of the number of segments;
    ✓ Values of fitness function;
    ✓ Computation times.
Case Study Results

• RQ1: How do the performances of the GA and DP approaches compare?
  • Wilcoxon test and Cliff’s delta effect size:
    - Difference of the number of segments;
    - Values of fitness function;
    - Computation times.
Case Study Results

• RQ1: How do the performances of the GA and DP approaches compare?
  • Wilxocon test and Cliff’s delta effect size:
    - Difference of the number of segments;
    - Values of fitness function;
    - Computation times.
Case Study Results

- **RQ2: How do the GA and DP approaches perform?**

<table>
<thead>
<tr>
<th>Program</th>
<th>Scenario</th>
<th>Concept</th>
<th>GA Jaccard</th>
<th>DP Jaccard</th>
<th>GA Precision</th>
<th>DP Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>Create Note</td>
<td>0.33</td>
<td>0.87</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Class, Create Note</td>
<td>Create Class</td>
<td>0.26</td>
<td>0.53</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Class, Create Note</td>
<td>Create Note</td>
<td>0.34</td>
<td>0.56</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle</td>
<td>Draw Rectangle</td>
<td>0.9</td>
<td>0.75</td>
<td>0.9</td>
<td>1</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Add Text, Draw Rectangle</td>
<td>Add Text</td>
<td>0.31</td>
<td>0.33</td>
<td>0.36</td>
<td>0.39</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Add Text, Draw Rectangle</td>
<td>Draw Rectangle</td>
<td>0.62</td>
<td>0.52</td>
<td>0.62</td>
<td>1</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>Draw Rectangle</td>
<td>0.74</td>
<td>0.24</td>
<td>0.79</td>
<td>0.24</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>Cut Rectangle</td>
<td>0.22</td>
<td>0.31</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Spawn Window, Draw Circle</td>
<td>Draw Circle</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>1</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Spawn Window, Draw Circle</td>
<td>Spawn Window</td>
<td>0.42</td>
<td>0.44</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Case Study Results

- RQ2: How do the GA and DP approaches perform?

<table>
<thead>
<tr>
<th>Program</th>
<th>Scenario</th>
<th>Concept</th>
<th>Jaccard</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GA</td>
<td>DP</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Note</td>
<td>Create Note</td>
<td>0.33</td>
<td>0.87</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Class, Create Note</td>
<td>Create Class</td>
<td>0.26</td>
<td>0.53</td>
</tr>
<tr>
<td>ArgoUML</td>
<td>Create Class, Create Note</td>
<td>Create Note</td>
<td>0.34</td>
<td>0.56</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle</td>
<td>Draw Rectangle</td>
<td>0.9</td>
<td>0.75</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Add Text, Draw Rectangle</td>
<td>Add Text</td>
<td>0.31</td>
<td>0.33</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Add Text, Draw Rectangle</td>
<td>Draw Rectangle</td>
<td>0.62</td>
<td>0.52</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>Draw Rectangle</td>
<td>0.74</td>
<td>0.24</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Draw Rectangle, Cut Rectangle</td>
<td>Cut Rectangle</td>
<td>0.22</td>
<td>0.31</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Spawn Window, Draw Circle</td>
<td>Draw Circle</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>JHotDraw</td>
<td>Spawn Window, Draw Circle</td>
<td>Spawn Window</td>
<td>0.42</td>
<td>0.44</td>
</tr>
</tbody>
</table>
Case Study Results

• RQ2: How do the GA and DP approaches perform?
  • Wilxocon test and Cliff’s delta effect size:
    ✓ Jaccad scores
    ✓ Precision
Case Study Results

• RQ2: How do the GA and DP approaches perform?
  • Wilxocon test and Cliff’s delta effect size:

  ![NO!]

  ✓ Jaccad scores

  ✓ Precision
Case Study Results

• RQ2: How do the GA and DP approaches perform?
  • Wilxocon test and Cliff’s delta effect size:
    - Jaccad scores
    - Precision
Outline

- Context
- Problem Statement
- Trace Segmentation
- **Segments Merging**
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario **draw rectangle:**

Original size
Compressed size
Number of segments

First Execution
Second Execution
Third Execution
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.

- Scenario draw rectangle:

  - Original size
  - Compressed size
  - Number of segments: 4850 Methods, 555 Methods, 35 Segments
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario **draw rectangle:**

<table>
<thead>
<tr>
<th>Execution</th>
<th>Original size</th>
<th>Compressed size</th>
<th>Number of segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>4850 Methods</td>
<td>555 Methods</td>
<td>35 Segments</td>
</tr>
<tr>
<td>Second</td>
<td>6668 Methods</td>
<td>240 Methods</td>
<td>21 Segments</td>
</tr>
<tr>
<td>Third</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario **draw rectangle**:

  - **First Execution**
    - Original size
    - Compressed size
    - Number of segments
    - 4850 Methods
    - 555 Methods
    - 35 Segments

  - **Second Execution**
    - 6668 Methods
    - 240 Methods
    - 21 Segments

  - **Third Execution**
    - 16 706 Methods
    - 930 Methods
    - 54 Segments
Segments Merging

• We merge segments obtained in multiple executions of the same scenario.

• Similarity:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$
Segments Merging

- We merge segments obtained in multiple executions of the same scenario.
- Similarity:

\[ J_{\text{accard}}(A, B) = \frac{|A \cap B|}{|A \cup B|} \]
Similarity Threshold

- Projects:

![Graph showing the relationship between the number of different terms and the similarity threshold. The graph includes a line for false positives and a line for false negatives.]
Segments Merging

- Example:

<table>
<thead>
<tr>
<th>Trace 1</th>
<th>Trace 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Z1</td>
</tr>
<tr>
<td>S2</td>
<td>Z2</td>
</tr>
<tr>
<td>S3</td>
<td>Z3</td>
</tr>
<tr>
<td>S4</td>
<td></td>
</tr>
</tbody>
</table>
Segments Merging

• Example:

Trace 1

Trace 2

Threshold = 70%
Segments Merging

- **Example:**

  ![Trace Diagram](image)

<table>
<thead>
<tr>
<th>Trace 1</th>
<th>Trace 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Z1</td>
</tr>
<tr>
<td>S2</td>
<td>Z2</td>
</tr>
<tr>
<td>S3</td>
<td>Z3</td>
</tr>
<tr>
<td>S4</td>
<td></td>
</tr>
</tbody>
</table>

  **Threshold = 70%**
Segments Merging

- **Example:**


```
Trace 1
<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z1</td>
<td>0.9</td>
<td>0.5</td>
<td>0.2</td>
</tr>
<tr>
<td>Z2</td>
<td>0.3</td>
<td>0.9</td>
<td>0.5</td>
</tr>
<tr>
<td>Z3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
</tr>
</tbody>
</table>
```

Threshold = 70%
Segments Merging

• Example:

Threshold = 70%

Synthetic Trace
Outline

• Context
• Problem Statement
• Trace Segmentation
• Segments Merging
  • Segments Labelling
• Segments Relations
• Usefulness Evaluation
• Conclusion and Future Work
Segments Labelling

- **Issue:** choice of the most appropriate source of information.
Segments Labelling

• **Issue:** choice of the most appropriate source of information.
  
  • **Method bodies:**
Segments Labelling

• **Issue:** choice of the most appropriate source of information.
  
  • **Method bodies:**
    
    ✔ Identifiers;
Segments Labelling

**Issue:** choice of the most appropriate source of information.

- **Method bodies:**
  - ✓ Identifiers;
  - ✓ Comments;
Segments Labelling

• **Issue:** choice of the most appropriate source of information.

  • Method bodies:
    - ✓ Identifiers;
    - ✓ Comments;

  • Method signature.
Segments Labelling

- **Issue:** choice of the most appropriate source of information.

- **Method bodies:**
  
  Method signatures provide more meaningful terms when labeling software artifacts than other sources. [De Lucia, 2012]
Segments Labelling

• **Source of information:** terms contained in the signature of methods.

• **Hypothesis:** a term appearing often in a particular segment, but not in other segments, provides important information for that segment.

• Ranks the terms of the segment by *TF-IDF* and keeps the topmost ones.
Segments Labelling

- To reduce the time and effort: segments are characterized using some unique methods (TF-IDF).

- Small version (5): result in loss of relevant information.

- Medium version (15): preserve better the relevant information.
Experiment Design

• Research Questions:
  • RQ1: How do the labels produced by the participants change when providing them different amount of information?
Experiment Design

• Research Questions:
  • RQ1: How do the labels produced by the participants change when providing them different amount of information?
  • RQ2: How do the labels produced by the participants compare to the generated labels?
Experiment Design

- **Research Questions:**
  - RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - RQ2: How do the labels produced by the participants compare to the generated labels?

- **Projects:**
Experiment Design

• Research Questions:
  • RQ1: How do the labels produced by the participants change when providing them different amount of information?
  • RQ2: How do the labels produced by the participants compare to the generated labels?

• Projects:

• Participants:
RQ1: How do the labels produced by the participants change when providing them different amount of information?
Experiment Design

• RQ1: How do the labels produced by the participants change when providing them different amount of information?
  
  • We group participants into 3 groups. Each version is assigned to a different group.
RQ1: How do the labels produced by the participants change when providing them different amount of information?

- We group participants into 3 groups. Each version is assigned to a different group.
RQ1: How do the labels produced by the participants change when providing them different amount of information?

- We group participants into 3 groups. Each version is assigned to a different group.

**Terms of small segment**

**Terms of medium segment**
RQ1: How do the labels produced by the participants change when providing them different amount of information?

- We group participants into 3 groups. Each version is assigned to a different group.

Terms of **small** segment

Terms of **medium** segment

Terms of **full** segment
• RQ1: How do the labels produced by the participants change when providing them different amount of information?

• We group participants into 3 groups. Each version is assigned to a different group.
Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
RQ1: How do the labels produced by the participants change when providing them different amount of information?

<table>
<thead>
<tr>
<th></th>
<th>2 Participants</th>
<th>5 Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Small Version</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>52</td>
<td>44</td>
</tr>
<tr>
<td>Recall</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td><strong>Medium Version</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>68</td>
<td>50</td>
</tr>
<tr>
<td>Recall</td>
<td>56</td>
<td>58</td>
</tr>
</tbody>
</table>
Experiment Results

- **RQ1:** How do the labels produced by the participants change when providing them different amount of information?

**Small Version**

<table>
<thead>
<tr>
<th></th>
<th>2 Participants</th>
<th>5 Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>52</td>
<td>44</td>
</tr>
<tr>
<td>Recall</td>
<td>44</td>
<td>46</td>
</tr>
</tbody>
</table>

**Medium Version**

<table>
<thead>
<tr>
<th></th>
<th>2 Participants</th>
<th>5 Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>68</td>
<td>56</td>
</tr>
<tr>
<td>Recall</td>
<td>50</td>
<td>58</td>
</tr>
</tbody>
</table>

92% Precision
RQ1: How do the labels produced by the participants change when providing them different amount of information?

**Experiment Results**

- **Small Version**
  - 2 Participants: Precision 52, Recall 44
  - 5 Participants: Precision 44, Recall 46
  - Precision: 92%

- **Medium Version**
  - 2 Participants: Precision 68, Recall 56
  - 5 Participants: Precision 50, Recall 58
  - Precision: 76%
Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:
    - ✔️ Number of participants;
    - ✔️ Size of the segment (full, medium, small);
    - ✔️ Their interaction;
    - ✔️ Years of programming experience.
Experiment Results

• RQ1: How do the labels produced by the participants change when providing them different amount of information?
  
  • Two-way permutation test:

  ✓ Number of participants;
  ✓ Size of the segment (full, medium, small);
  ✓ Their interaction;
  ✓ Years of programming experience.
Experiment Results

• RQ1: How do the labels produced by the participants change when providing them different amount of information?
  
• Two-way permutation test:
  
  ✓ Number of participants;
  
  ✓ Size of the segment (full, medium, small);
  
  ✓ Their interaction;
  
  ✓ Years of programming experience.
RQ1: How do the labels produced by the participants change when providing them different amount of information?

- Two-way permutation test:
  - Number of participants;
  - Size of the segment (full, medium, small);
  - Their interaction;
  - Years of programming experience.
Experiment Results

• RQ1: How do the labels produced by the participants change when providing them different amount of information?
  
  • Two-way permutation test:
    
    ✓ Number of participants;
    
    ✗ Size of the segment (full, medium, small);
    
    ✗ Their interaction;
    
    ✗ Years of programming experience.
Experiment Design

• RQ2: How do the labels produced by the participants compare to the generated labels?
Experiment Design

• RQ2: How do the labels produced by the participants compare to the generated labels?
  
  • **Oracle**: 210 segments (less than 100) manually labelled by the participants.
Experiment Design

• RQ2: How do the labels produced by the participants compare to the generated labels?
  • **Oracle**: 210 segments (less than 100) manually labelled by the participants.
  • Evaluation: 1 participant and 2 participants.
RQ2: How do the labels produced by the participants compare to the generated labels?
• RQ2: How do the labels produced by the participants compare to the generated labels?

<table>
<thead>
<tr>
<th></th>
<th>1 Participant</th>
<th>2 Participants Intersection</th>
<th>2 Participants Union</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>58</td>
<td>38</td>
<td>60</td>
</tr>
<tr>
<td>Recall</td>
<td>71</td>
<td>85</td>
<td>56</td>
</tr>
</tbody>
</table>
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work
Segments Relations

• **Formal Concept Analysis:** used to identify relations between concepts identified in different segments.

• **Groups objects that have common attributes:** objects are segments and attributes are terms.

• **An FCA concept:** maximal collection of objects that have common attributes.
Segments Relations

- FCA lattice for the execution trace of the scenario **create a class**.
Segments Relations

- FCA lattice for the execution trace of the scenario create a class.
Segments Relations

- FCA lattice for the execution trace of the scenario create a class.
• FCA lattice for the execution trace of the scenario create a class.
Segments Relations

- FCA lattice for the trace of the scenario **draw rectangle delete rectangle** (32 segments).
Segments Relations

- FCA lattice for the trace of the scenario **draw rectangle delete rectangle** (32 segments).
Experiment Design

• Research Questions:
  
  • RQ1: To what extent does our approach correctly identify relations among segments?
Experiment Design

• Research Questions:
  • RQ1: To what extent does our approach correctly identify relations among segments?

• Projects:
Research Questions:

RQ1: To what extent does our approach correctly identify relations among segments?

Projects:

Participants:
**Experiment Design**

- **RQ1**: To what extent does our approach correctly identify relations among segments?
RQ1: To what extent does our approach correctly identify relations among segments?

- 100 relations are validated by participants.
Experiment Design

- RQ1: To what extent does our approach correctly identify relations among segments?
  - 100 relations are validated by participants.

<table>
<thead>
<tr>
<th></th>
<th>Segments</th>
<th>Labels</th>
<th>Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our Approach</strong></td>
<td>9</td>
<td>listener, add, change, figure</td>
<td>sub/super concept</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>figure, listener, add, internal, multicaster, event, change</td>
<td></td>
</tr>
<tr>
<td><strong>Participant 1</strong></td>
<td>9</td>
<td>composite, figure, trigger, event</td>
<td>sub/super concept</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>manage, figure, change, event, trigger</td>
<td></td>
</tr>
<tr>
<td><strong>Participant 2</strong></td>
<td>9</td>
<td>abstract, figure, change, add, listener</td>
<td>same concept</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>figure, change, event, multicaster, add, listener</td>
<td></td>
</tr>
</tbody>
</table>
**Experiment Results**

- **RQ1**: To what extent does our approach correctly identify relations among segments?
RQ1: To what extent does our approach correctly identify relations among segments?

Experiment Results

- Agreements without distinction btw. sub/super relations
- Agreements with distinction btw. sub/super relations
**Experiment Results**

- **RQ1:** To what extent does our approach correctly identify relations among segments?

![Bar chart showing agreement results for different tools and conditions.](chart.png)

- Agreements without distinction between sub/super relations
- Agreements with distinction between sub/super relations

**Results:**
- ArgoUML: 100%
- JHotDraw: 100%
- Mars: 83%
- Maze: 82%
- Pooka: 78%

**Overall Agreement:**
- Without distinction: 96%
- With distinction: 63%
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work
Usefulness Evaluation

- During maintenance, developers are interested to understand some segments of a trace that implement some concepts of interest.
- Trace Segmentation approach groups these concepts in few segments.
- Labelling and relating segments approach guide developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.
Empirical Study

- Research Questions:
  - RQ1: Does our trace segmentation has a potential to support concept location?
  - RQ2: To what extent does our approach support concept location tasks?

- Projects:
  
  The dataset was made publicly available by Dit et al., [2013]
Empirical Results

- RQ1: Does our trace segmentation approach has a potential to support concept location?
Empirical Results

- **RQ1:** Does our trace segmentation approach have a potential to support concept location?
RQ1: Does our trace segmentation approach has a potential to support concept location?

**Empirical Results**

- Number of Segments
  - Min: 1
  - Mean: 2.2
  - Max: 5
  - Mean: 35.9
  - Max: 68

- Percentage of the size of the segments
  - Mean: 2.48
  - Max: 6.47

- Percentage of the size of the segments graph:
  - Mean: 7%
  - Max: 3.5%
Empirical Results

- RQ2: To what extent does our approach support concept location tasks if used as a standalone technique?
  
  - Title of the bug report;
  - Labels of the segments;
  - FCA lattice.
Empirical Results

- RQ2: To what extent does our approach support concept location tasks if used as a standalone technique?

![Chart showing empirical results](chart.png)
Outline

• Context
• Problem Statement
• Trace Segmentation
• Segments Merging
• Segments Labelling
• Segments Relations
• Usefulness Evaluation

• Conclusion and Future Work
Conclusion

- A typical scenario in which concept location takes part:
Execution Trace

Large, noisy, and multi-threaded
Execution Trace: Large, noisy, and multi-threaded
Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>m2</td>
<td>...</td>
<td>m30</td>
<td>...</td>
<td>m79</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m133</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m134</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m445</td>
</tr>
</tbody>
</table>

- Create a new segment.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>m2</td>
<td>...</td>
<td>m30</td>
<td>...</td>
<td>m79</td>
<td>m134</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m90</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m91</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m133</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m134</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>m445</td>
</tr>
</tbody>
</table>

- Add the method to the last segment.
Large, noisy, and multi-threaded
Segments Labelling

- **Source of information**: terms contained in the signature of methods.
- **Hypothesis**: A term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- **Ranks the terms of the segment by** *TF-IDF* and keeps the topmost ones.
Execution Trace: Large, noisy, and multi-threaded

Trace Segmentation (SSBSE’11)

Labeling Segments (WCRE’12, JSEP14)

Relating Segments (WCRE’12, JSEP14)
Segments Relations

- FCA lattice for the execution trace of the scenario “New Class”.

Execution Trace: Large, noisy, and multi-threaded
Execution Trace: Large, noisy, and multi-threaded

- Trace Segmentation (SSBSE’11)
- Labeling Segments (WCRE’12, JSEP14)
- Relating Segments (WCRE’12, JSEP14)
Usefulness Evaluation

- During maintenance, developers are interested to understand some segments of a trace that implement some concepts of interest.
- Our approach groups these concepts in few segments.
- Our approach guide developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.
Execution Trace: Large, noisy, and multi-threaded

Trace Segmentation (SSBSE’11)
Labeling Segments (WCRE’12, JSEP14)
Relating Segments (WCRE’12, JSEP14)
Usefulness Evaluation (JSEP14)

Dynamic Programming (DP) Approach
Example of trace segmentation using DP
- Create a new segment
- Add the method to the last segment

Segments Labelling
- Source of information: terms relevant in the signature of methods.
- Association of a term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- Marks the terms of the segment by TV-DRP and implies the structural ones.

Usefulness Evaluation
- During maintenance, developers are interested to understand specific segments of a trace that implement some concepts of interest.
- Our approach groups these concepts in few segments.
- Our approach guides developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.
Identify concepts and facilitate the analysis of large execution traces for maintenance tasks.
Future Work

• A tool to visualize the identified relations among segments.

• Adapting our approach to online labelling of traces while they are being generated.

• Trace segmentation of distributed systems.
Execution Trace

1. Trace Segmentation (SSBSE’11)
2. Labeling Segments (WCRE’12, JSEP14)
3. Relating Segments (WCRE’12, JSEP14)
4. Usefulness Evaluation (JSEP14)
References

