@tComment:
Testing Javadoc Comments to Detect Comment-Code Inconsistencies

Shin Hwei Tan
University of Illinois

Darko Marinov
University of Illinois

Lin Tan
University of Waterloo

Gary T. Leavens
University of Central Florida
/*
 * Returns a synchronized map backed by the given map.
 * ...
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */

static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)

• @param - Parameter name, Description
• @return - Description
• @throws - Exception name, Condition under which the exception is thrown
Popularity of *Javadoc* Comments

- Writing *Javadoc* comments is a common practice
- Lots of *Javadoc* comments exist in Java libraries.

<table>
<thead>
<tr>
<th>Project</th>
<th>Number of methods</th>
<th>Number of <em>Javadoc</em> comments for methods</th>
<th>Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections</td>
<td>3,874</td>
<td>2,434</td>
<td>63</td>
</tr>
<tr>
<td>GlazedLists</td>
<td>2,753</td>
<td>1,741</td>
<td>63</td>
</tr>
<tr>
<td>JFreeChart</td>
<td>6,205</td>
<td>6,186</td>
<td>100</td>
</tr>
<tr>
<td>JodaTime</td>
<td>3,887</td>
<td>2,917</td>
<td>75</td>
</tr>
<tr>
<td>Log4j</td>
<td>2,115</td>
<td>958</td>
<td>45</td>
</tr>
<tr>
<td>Lucene</td>
<td>5,222</td>
<td>2,205</td>
<td>42</td>
</tr>
<tr>
<td>Xalan</td>
<td>5,404</td>
<td>3,229</td>
<td>60</td>
</tr>
</tbody>
</table>
Javadoc Comments can be Inconsistent with Code

```java
/*…
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */
static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)
```

Expected behavior for synchronizedMap (null):
- Throws IllegalArgumentException

Actual behavior:
- Throws NullPointerException
Checking Comments

Important to find bugs in comments
- Comments are not executed
- Comments are read by developers to understand code
- Incorrect comments could cause developers to write wrong code [SOSP’07]

Challenging to automatically analyze comments
- Ambiguities in understanding general text
- NLP made progress but struggles with general text
Build domain-specific analyses

- **iComment** (locking protocols & function calls) [SOSP’07]
- **aComment** (interrupts) [ICSE’11]
  - System code (C/C++)
  - Extract machine-checkable rules from comments
  - Use static analysis to check consistency between code and comments/rules
New Domain
• Method properties for null values and related exceptions

Dynamic Analysis (Random Testing)
• Fewer false alarms compared to static analysis

Improved Testing
• Reduce false alarms in test generation tool

Evaluation on 7 Libraries
• Found 28 inconsistencies, 12 were fixed
Example Inconsistency of Type 1: Correct Code, Incorrect Comment

```java
/* …
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */
static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)
```
Example Inconsistency of Type 1: Correct Code, Incorrect Comment

```java
/* ...
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */
static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)
```

@tComment
Example Inconsistency of Type 1: **Correct Code, Incorrect Comment**

```java
/*…
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */
static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)
```

```java
public void test1() throws Throwable {
    java.util.Map var0 = null;
    try {
        java.util.Map var1= ...synchronizedMap(var0);
    } catch (IllegalArgumentException expected) {return;}
    fail("Expected exception of type IllegalArgumentException but got NullPointerException");
}  
```
Example Inconsistency of Type 1: **Correct Code, Incorrect Comment**

```java
/* ...
 * @param map the map to synchronize, must not be null
 * @return a synchronized map backed by the given map
 * @throws IllegalArgumentException if the map is null
 */
static <K,V> Map<K,V> synchronizedMap(Map<K,V> map)
```

```java
public void test1() throws Throwable {
    java.util.Map var0 = null;
    try {
        java.util.Map var1= ...synchronizedMap(var0);
    } catch (IllegalArgumentException expected) {return;}
    fail("Expected exception of type IllegalArgumentException but got NullPointerException");
}
```

✓ Confirmed & fixed by Collections developers
Example Inconsistency of Type 2: Fault in Code, Correct Comment

```java
/*...
* @param anchor the anchor (<code>null</code> not permitted).
*/
void setRotationAnchor(TextAnchor anchor)
```
Example Inconsistency of Type 2: Fault in Code, Correct Comment

```java
/*...
 * @param anchor the anchor (<code>null</code> not permitted).
 */
void setRotationAnchor(TextAnchor anchor)
```
Example Inconsistency of Type 2: Fault in Code, Correct Comment

```java
public void test2() throws Throwable {
    ...CategoryPointerAnnotation var0 = new ...CategoryPointerAnnotation("$0.00", (java.lang.Comparable),'#', 10.0d, 10.0d);
    org.jfree.ui.TextAnchor var1 = null;
    try {
        var0.setRotationAnchor(var1);
        fail("Expected exception but got Normal Execution");
    } catch (Exception expected) {} 
}
```
Example Inconsistency of Type 2: Fault in Code, Correct Comment

```java
public void test2() throws Throwable {
    ...CategoryPointerAnnotation var0 = new ...CategoryPointerAnnotation("$0.00", (java.lang.Comparable)'#', 10.0d, 10.0d);
    org.jfree.ui.TextAnchor var1 = null;
    try {
        var0.setRotationAnchor(var1);
        fail("Expected exception but got Normal Execution");
    } catch (Exception expected) {}
}
```

/*...
 * @param anchor the anchor (<code>null</code> not permitted).
 */

```java
void setRotationAnchor(TextAnchor anchor)
```
@tComment Design

- Javadoc Comments
- Java Methods
- @tComment
- Infer properties
- Properties
- Modified Randoop
- Tests with Inconsistency
@tComment Design

1. Infer properties
2. @tComment
3. Modified Randoop
4. Tests with Inconsistency

Javadoc Comments
Java Methods
Properties for Method Parameters: If the Parameter is Null...

- **Null Normal**
  - …, the method should execute normally (no exception)
  - Eg: `@param` predicate the predicate to use, may be null

- **Null Any Exception**
  - …, the method should throw some exception

- **Null Specific Exception**
  - `id==null => IllegalArgumentException` if the id is null

- **Null Unknown**
  - …, the method behavior is unknown
  - Eg: `@param` array the array over which to iterate
Properties for Method Parameters:
If the Parameter is Null...

- **Null Normal**
  - …, the method should execute normally (no exception)
  - Eg: `@param` predicate the predicate to use, may be null
- **Null Any Exception**
  - …, the method should throw some exception
  - Eg: `@param` collection the collection to add to, must not be null
Properties for Method Parameters: If the Parameter is Null...

- **Null Normal**
  - …, the method should execute normally (no exception)
  - Eg: `@param` *predicate* the predicate to use, may be `null`

- **Null Any Exception**
  - …, the method should throw some exception
  - Eg: `@param` *collection* the collection to add to, must not be `null`

- **Null Specific Exception** (`id==null => IllegalArgumentExceptionExc`)
  - …, the method should throw a specific type of exception
  - Eg: `@throws` `IllegalArgumentException` if the *id* is `null`
Properties for Method Parameters: If the Parameter is Null…

- **Null Normal**
  - …, the method should execute normally (no exception)
  - Eg: `@param` predicate the predicate to use, may be null

- **Null Any Exception**
  - …, the method should throw some exception
  - Eg: `@param` collection the collection to add to, must not be null

- **Null Specific Exception (id==null => IllegalArgumentExc)**
  - …, the method should throw a specific type of exception
  - Eg: `@throws` IllegalArgumentException if the id is null

- **Null Unknown**
  - …, the method behavior is unknown
  - Eg: `@param` array the array over which to iterate
Inferring Properties from Comments

Parse *Javadoc*
(Standard Doclet)

Extract `@param`
and `@throws`

Analyze text
Inferring Properties from Comments

Parse **Javadoc** (Standard Doclet)

Extract `@param` and `@throws`

Analyze text

- Negation words (±3 distance within `null`) → Null Any Exception
- No negation words → Null Normal
- `null` in `@throws` tag, then searches the list of parameter names. If found → Null Specific Exception
  
  \[
  \text{param} == \text{null} \Rightarrow \text{SpecificException}
  \]

- Generates multiple properties for “or” and “either” in the `@throws` tag
while timeLimit not reached:

a. Create a new sequence
   1. Randomly pick a method call \( m(T_1...T_k) \)
   2. For each parameter, randomly pick a sequence \( S_i \) that can construct the object for that parameter
      • Select null with some probability (nullRatio)
   3. Create new sequence \( S_{\text{new}} \leftarrow S_1;...;S_k; m(\text{var}_1...\text{var}_k) \)

b. Classify the new sequence \( S_{\text{new}} \)
   • Execute & check contract violations?
     • Yes - Output as failure-revealing test case
     • No - Add to sequences if not redundant, else discard
@tComment Modified Randooop

while timeLimit not reached:

a. Create a new sequence
   1. Randomly pick a method call m(T₁...Tₖ)
   2. For each parameter, randomly pick a sequence Sᵢ
      that can construct the object for that parameter
      • Select null with some probability (nullRatio)
   3. Create new sequence S_{new} ← S₁,..,Sₖ, m(var₁..varₖ)

b. Classify the new sequence S_{new}
   • Execute & check contract violations & @tComment properties?
     • Violate contract - Output as failure-revealing tests
     • Violate properties - Output as comment-code inconsistency
     • No - Add to sequences if not redundant, else discard
From Properties to Inconsistency

/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)
/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)
From Properties to Inconsistency

/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)
/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)

Execute

exiting(null, "", null)
From Properties to Inconsistency

/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)

Execute

Actual Exception Thrown

exiting(null, "", null)
/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */
void exiting(Logger logger, String sourceClass, String sourceMethod)

Execute

Matches

Actual Exception Thrown

Null Normal

Null Any Exception

exiting(null, "", null)
```java
/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */

void exiting(Logger logger, String sourceClass, String sourceMethod)
```

From Properties to Inconsistency

Matches

Execute

exiting(null, "", null)
/**
 * @param logger logger, may not be null.
 * @param sourceClass source class, may be null.
 * @param sourceMethod method, may be null.
 */
void exiting(Logger logger, String sourceClass, String sourceMethod)

exiting(null, "", null)

isCommentCodeInconsistency?
Different kinds of Matches

Throws Exception?

Execute
Different kinds of Matches

- Execute
- Throws Exception?
  - No
    - Null Any / Null Specific Exception?
Different kinds of Matches

- Execute
  - Throws Exception?
    - No
      - Null Any / Null Specific Exception?
        - Yes: Missing Exception
        - No
Different kinds of Matches

Execute

Throws Exception?

No

Null Any / Null Specific Exception?

Yes

≥1 Null Specific Exception?

Missing Exception
Different kinds of Matches

1. Execute

2. Throws Exception?
   - Yes
     - ≥1 Null Specific Exception?
       - Yes
         - Missing Exception
       - No
         - No
   - No

3. Null Any / Null Specific Exception?
   - Yes
     - Missing Exception
   - No

4. ≥1 Null Normal?
Different kinds of Matches

- **Throws Exception?**
  - Yes
    - ≥1 Null Specific Exception?
      - Yes: **Expected Exception**
      - No: ≥1 Null Normal?
        - Yes: **Unexpected Exception**
        - No: Null Any / Null Specific Exception?
          - Yes: **Missing Exception**
          - No: Execute

- No
  - Execute
Different kinds of Matches

Throw Exception?

Yes

≥1 Null Specific Exception?

No

Null Any / Null Specific Exception?

Yes

Missing Exception

No

≥1 Null Normal?

Yes

Unexpected Exception

No

Unknown Status

Execute
Different kinds of Matches

Execute

Throws Exception?

No

Null Any / Null Specific Exception?

Yes

Missing Exception

Yes

≥1 Null Specific Exception?

No

≥1 Null Normal?

Yes

Unexpected Exception

No

Yes

Unknown Status

Yes

Exception in the set of expected?

No

No
Different kinds of Matches

- Execute
  - Throws Exception?
    - Yes
      - ≥1 Null Specific Exception?
        - No
          - Exception in the set of expected?
            - No
              - Different Exception
            - Yes
              - ≥1 Null Normal?
                - Yes
                  - Unexpected Exception
                - No
                  - Unknown Status
          - Yes
            - Null Any / Null Specific Exception?
              - Yes
                - Missing Exception
              - No
                - Different Exception
          - No
            - Different Exception

- ≥1 Null Specific Exception?
  - No
    - Exception in the set of expected?
      - Yes
        - Different Exception
      - No
        - Different Exception
Different kinds of Matches

1. Execute
   - Throws Exception? (Yes -> ≥1 Null Specific Exception? (Yes -> Exception in the set of expected? (Yes -> Expected Exception)
   - Throws Exception? (No -> Null Any / Null Specific Exception? (Yes -> Missing Exception
2. Execute
   - ≥1 Null Specific Exception? (Yes -> Exception in the set of expected? (Yes -> Expected Exception
   - ≥1 Null Specific Exception? (No -> ≥1 Null Normal? (Yes -> Unexpected Exception
   - ≥1 Null Normal? (No -> Different Exception
3. Execute
   - Different Exception
Different kinds of Matches

- Execute
  - Throws Exception?
    - Yes
      - ≥1 Null Specific Exception?
        - Yes
          - Missing Exception
        - No
          - ≥1 Null Normal?
            - Yes
              - Unexpected Exception
            - No
              - Exception in the set of expected?
                - Yes
                  - Expected Exception
                - No
                  - Different Exception

- Is Comment-code Inconsistency?
7 open source projects
Ran @tComment to infer properties
Ran Modified Randoop to check properties
  Varying 2 options on Modified Randoop
    nullRatio (How often null is chosen as input for a method?)
    timeLimit (How long should Modified Randoop take for generating tests before stopping?)
Best configuration for all projects
  nullRatio $= 0.6$
  timeLimit $= 3600s$ (reached plateau effect after that)
## Subject Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th># LOC</th>
<th># Classes</th>
<th># Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections</td>
<td>Collection library and utilities</td>
<td>19,417</td>
<td>274</td>
<td>3,874</td>
</tr>
<tr>
<td>GlazedLists</td>
<td>List transformations in Java</td>
<td>19,203</td>
<td>239</td>
<td>2,753</td>
</tr>
<tr>
<td>JFreeChart</td>
<td>Chart creator</td>
<td>51,376</td>
<td>396</td>
<td>6,205</td>
</tr>
<tr>
<td>JodaTime</td>
<td>Date and time library</td>
<td>18,428</td>
<td>154</td>
<td>3,887</td>
</tr>
<tr>
<td>Log4j</td>
<td>Logging service</td>
<td>14,452</td>
<td>221</td>
<td>2,115</td>
</tr>
<tr>
<td>Lucene</td>
<td>Text search engine</td>
<td>38,051</td>
<td>422</td>
<td>5,222</td>
</tr>
<tr>
<td>Xalan</td>
<td>XML transformations</td>
<td>53,642</td>
<td>510</td>
<td>5,404</td>
</tr>
</tbody>
</table>
# Number of Matches

<table>
<thead>
<tr>
<th>Project</th>
<th>Missing Exception</th>
<th>Different Exception</th>
<th>Unexpected Exception</th>
<th>Unknown Status</th>
<th>Expected Exception</th>
<th>Tested Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections</td>
<td>12</td>
<td>4</td>
<td>6</td>
<td>94</td>
<td>36</td>
<td>115</td>
</tr>
<tr>
<td>GlazedLists</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>151</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>JFreeChart</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>127</td>
<td>6</td>
<td>42</td>
</tr>
<tr>
<td>JodaTime</td>
<td>3</td>
<td>0</td>
<td>13</td>
<td>37</td>
<td>3</td>
<td>31</td>
</tr>
<tr>
<td>Log4j</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>186</td>
<td>152</td>
<td>179</td>
</tr>
<tr>
<td>Lucene</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>368</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Xalan</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>544</td>
<td>32</td>
<td>43</td>
</tr>
<tr>
<td>Total</td>
<td>30</td>
<td>4</td>
<td>34</td>
<td>1507</td>
<td>232</td>
<td>433</td>
</tr>
</tbody>
</table>
## Number of Matches

<table>
<thead>
<tr>
<th>Project</th>
<th>Missing Exception</th>
<th>Different Exception</th>
<th>Unexpected Exception</th>
<th>Unknown Status</th>
<th>Expected Exception</th>
<th>Tested Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections</td>
<td>12</td>
<td>4</td>
<td>6</td>
<td>94</td>
<td>36</td>
<td>115</td>
</tr>
<tr>
<td>GlazedLists</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>151</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>JFreeChart</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>127</td>
<td>6</td>
<td>42</td>
</tr>
<tr>
<td>JodaTime</td>
<td>3</td>
<td>0</td>
<td>13</td>
<td>37</td>
<td>3</td>
<td>31</td>
</tr>
<tr>
<td>Log4j</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>186</td>
<td>152</td>
<td>179</td>
</tr>
<tr>
<td>Lucene</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>368</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Xalan</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>544</td>
<td>32</td>
<td>43</td>
</tr>
<tr>
<td>Total</td>
<td>30</td>
<td>4</td>
<td>34</td>
<td>1507</td>
<td>232</td>
<td>433</td>
</tr>
</tbody>
</table>

Tested only a fraction of properties inferred
## True Inconsistencies & False Alarms

<table>
<thead>
<tr>
<th>Project</th>
<th>Missing Exception = TI + FA</th>
<th>Different Exception = TI + FA</th>
<th>Unexpected Exception = TI + FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections</td>
<td>12 + 0</td>
<td>3 + 1</td>
<td>0 + 6</td>
</tr>
<tr>
<td>GlazedLists</td>
<td>0 + 0</td>
<td>0 + 0</td>
<td>1 + 5</td>
</tr>
<tr>
<td>JFreeChart</td>
<td>1 + 0</td>
<td>0 + 0</td>
<td>2 + 0</td>
</tr>
<tr>
<td>JodaTime</td>
<td>3 + 0</td>
<td>0 + 0</td>
<td>0 + 13</td>
</tr>
<tr>
<td>Log4j</td>
<td>1 + 0</td>
<td>0 + 0</td>
<td>0 + 3</td>
</tr>
<tr>
<td>Lucene</td>
<td>0 + 4</td>
<td>0 + 0</td>
<td>1 + 1</td>
</tr>
<tr>
<td>Xalan</td>
<td>4 + 5</td>
<td>0 + 0</td>
<td>0 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>21 + 9</strong></td>
<td><strong>3 + 1</strong></td>
<td><strong>4 + 30</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>Total</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>True Inconsistencies (TI)</td>
<td>21</td>
</tr>
<tr>
<td>False Alarms (FA)</td>
<td>9</td>
</tr>
</tbody>
</table>
Sources of False Alarm

Incorrect inference
  • Wrong type of inferred properties

Missing properties
  • Missing property (*Null Unknown*) causes unexpected exception

Incorrect/missing properties for another method
  • Method under test depend on other method, other method causes unexpected exception
## Comment Analysis Results

<table>
<thead>
<tr>
<th>Project</th>
<th>Precision [%]</th>
<th>Recall [%]</th>
<th>Accuracy [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Norm</td>
<td>Any</td>
<td>Spec</td>
</tr>
<tr>
<td><strong>Collections</strong></td>
<td>75</td>
<td>92</td>
<td>100</td>
</tr>
<tr>
<td><strong>GlazedLists</strong></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>JFreeChart</strong></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>JodaTime</strong></td>
<td>100</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td><strong>Log4j</strong></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>Lucene</strong></td>
<td>100</td>
<td>67</td>
<td>100</td>
</tr>
<tr>
<td><strong>Xalan</strong></td>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>Total/Overall</strong></td>
<td>98</td>
<td>98</td>
<td>100</td>
</tr>
</tbody>
</table>

2 properties are inferred, 1 incorrect

High accuracy (97-100%)
Summary of Results

Comment-Code Inconsistency Detection

• Detected 28 comment-code inconsistencies
• 40 false alarms

Comment Analysis Result

• High accuracy of 97–100% without using NLP techniques
  • Javadoc comments are well-structured
  • Not much variance in paraphrases
Conclusion

• An inconsistency between comment and code is highly indicative of program faults
• @tComment checks consistency of Java method bodies and Javadoc comments properties related to null values and exceptions
• Evaluated on 7 open-source projects
• Discovered 28 inconsistencies and 12 were already fixed

Acknowledgement: Travel expenses sponsored by NSF CCF 07-46856 grants and ACM-W Scholarships for Attendance at Research Conferences.