Code Coverage Analysis for Eclipse

EclipseCon 2008, Santa Clara

Marc R. Hoffmann, hoffmann@mountainminds.com
Gilles Iachelini, giachelini@csc.com
17.03.2008
Focus

Technical
Java
Eclipse
Topics

- Principles and Techniques
- Code Coverage Tools in Eclipse
- Coverage Analysis of Eclipse Apps
Tutorial Prerequisites

- Eclipse 3.3.x
- Tutorial Data

http://www.eclemma.org/research

EclipseCon 2008

Tutorial: Code Coverage Analysis for Eclipse by Marc R. Hoffmann and Gilles Iachellini at EclipseCon, 2008/03/17
- Presentation
- Tutorial Data ← Download for tutorial attendees
Usage Scenario

Unit Tests

INDIRECT METRIC

White Box Testing

System Test

Automated UI Test

Manual Execution

Black Box Testing
Coverage Units

- Control Flow Coverage
  - Classes
  - Methods
  - Lines
  - Statements
  - Branches
  - Paths

\[
\text{Coverage Ratio} = \frac{\text{Covered Units}}{\text{Total Units}}
\]
public int clip(int lower, int upper, int x) {
    if (x < lower) {
        x = lower;
    }
    if (x > upper) {
        x = upper;
    }
    return x;
}

Test Set for Full Statement Coverage:
clip(1, 9, 0)
clip(1, 9, 10)
public int clip(int lower, int upper, int x) {
    if (x < lower) {
        x = lower;
    }
    if (x > upper) {
        x = upper;
    }
    return x;
}

Test Set for Full Branch Coverage:
clip(1, 9, 0)
clip(1, 9, 10)
public int clip(int lower, int upper, int x) {
    if (x < lower) {
        x = lower;
    }
    if (x > upper) {
        x = upper;
    }
    return x;
}

Test Set for Full Path Coverage:
clip(1, 9, 0)
clip(1, 9, 10)
clip(1, 9, 5)
clip(9, 1, 5)
Granularity

OVERALL COVERAGE SUMMARY

<table>
<thead>
<tr>
<th>name</th>
<th>method, %</th>
<th>line, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>all classes</td>
<td>85% (5733/6738)</td>
<td>79% (27538/34699)</td>
</tr>
</tbody>
</table>

COVERAGE BREAKDOWN BY PACKAGE

<table>
<thead>
<tr>
<th>name</th>
<th>method, %</th>
<th>line, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>org.apache.commons.collections</td>
<td>81% (1632/2017)</td>
<td>75% (9500.0/12251)</td>
</tr>
<tr>
<td>org.apache.commons.collections.bag</td>
<td>69% (157/228)</td>
<td>77% (726.2/945)</td>
</tr>
<tr>
<td>org.apache.commons.collections.list</td>
<td>91% (534/585)</td>
<td>79% (2457.6/2720)</td>
</tr>
<tr>
<td>org.apache.commons.collections.set</td>
<td>89% (287/221)</td>
<td>78% (688.8/1041)</td>
</tr>
<tr>
<td>org.apache.commons.collections.comparators</td>
<td>89% (125/141)</td>
<td>80% (525.6/517)</td>
</tr>
<tr>
<td>org.apache.commons.collections.map</td>
<td>98% (1255/1255)</td>
<td>81% (4203.6/4449)</td>
</tr>
<tr>
<td>org.apache.commons.collections.weakMap</td>
<td>91% (549/668)</td>
<td>79% (2402.1/3006)</td>
</tr>
<tr>
<td>org.apache.commons.collections.collection</td>
<td>90% (201/224)</td>
<td>83% (945.4/1141)</td>
</tr>
<tr>
<td>org.apache.commons.collections.buffer</td>
<td>91% (206/215)</td>
<td>86% (1455.9/1532)</td>
</tr>
<tr>
<td>org.apache.commons.collections.functional</td>
<td>74% (160/221)</td>
<td>86% (659.8/709)</td>
</tr>
<tr>
<td>org.apache.commons.collections.hashmap</td>
<td>92% (428/467)</td>
<td>89% (1851.4/2070)</td>
</tr>
<tr>
<td>org.apache.commons.collections.keyvalue</td>
<td>92% (101/110)</td>
<td>88% (417.6/473)</td>
</tr>
</tbody>
</table>

```java
152  public boolean addAll(int index, Collection c) {
153      if(c.isEmpty()) {
154          return false;
155      } else if(size == index || size == 0) {
156          return addAll(c);
157      } else {
158          Listable succ = getListItem(index);
159          Listable pred = (coll <= succ) ? succ.getPrev();
160          Iterator it = c.iterator();
161          while(it.hasNext()) {
162              pred = insertListItem(pred, succ, it.next());
163          }
164          return true;
165      }
166  }
```
## Tool Matrix

<table>
<thead>
<tr>
<th></th>
<th>Cmd</th>
<th>Ant</th>
<th>Eclipse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Coverclipse</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Cobertura</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EMMA/EclEmma</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TPTP</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Coverage While You Work

Install EclEmma
02_EclEmma_1.3.1/ecleemma-1.3.1.zip

Import sample application
03_Exercise_1/...simplemath.zip

Run FormulasTest as JUnit test

Identify lines in class Formulas not covered by the test cases.

Add test cases to get full coverage
Coverage Data Analysis

- Import Apache Collections Library
  04_Exercise_2/...collections.zip

- Run `TestAllPackes` as JUnit test

- Explore Results in Coverage view
Feedback Loop

Write Tests

Fix It

© 2008 Marc R. Hoffmann, made available under EPL v1.0
Implementation Strategies

- **Compiler**
- **Class Loader**
- **JVM**

**Instrumentation**
- **Java Source**
- **Byte Code**

**Runtime Profiling**
- Offline
- On-the-Fly
Why Byte Code Instrumentation?

- Performance Issues with JVM Profilers
- No Source Required
- Works on 3rd party JAR‘s
- Platform independent
Byte Code Instrumentation

JVM

Coverage Runtime

Structure Information

Instrumentation
Byte Code Probes

// access flags 9
public static clip(III)I

L0
LINENUMBER 6 L0
ILOAD 2
ILOAD 0

IF_ICMPGE L1
L2
LINENUMBER 7 L2
ILOAD 0
ISTORE 2

[...]

// access flags 9
public static clip(III)I

L0
GETSTATIC Formulas.$VRc : [[Z
L1
ICONST_1
AALOAD
ASTORE 3
L2
LINENUMBER 6 L2
ILOAD 2
ILOAD 0
L3
ALOAD 3
ICONST_0
ICONST_1
BASTORE
IF_ICMPGE L4
L5
LINENUMBER 7 L5
ILOAD 0
ISTORE 2
ALOAD 3
ICONST_1
ICONST_1
BASTORE
[...]
Code Coverage for Eclipse/OSGi Bundles
Code Coverage Analysis for Eclipse

Code Coverage for JUnit Plug-In Tests

Import EclEmma Plug-ins
05_Exercise_3/eclelemma_core.zip

Run AllEclEmmaCoreTests JUnit plug-in test
Headless?

IDE ➔ Target

Eclipse/OSGi
Class Loading via OSGi

Compiler → Class Loader → OSGi Framework Implementation → JVM
Equinox Adapter Hooks
Copy to your plugins folder
06_Exercise_4/... osgihook

Insert into config.ini

```ini
osgi.framework.extensions=
    com.mountainminds.eclemma.osgihook
```

Start/Stop Eclipse
TDD/Test First ↔ Code Coverage

TDD Test First

superseeds

verifies

Code Coverage
Thank You!

- Use Code Coverage
- It’s Easy and Helpful
- OSGi/Equinox is Code Coverage Ready
- Makes Writing Unit Tests Fun
References

- Code Coverage Tools
  - Cobertura, http://cobertura.sourceforge.net/
  - CodeCover, http://codecove.org/

- Byte Code Library and Outline Plug-in:

- Code Coverage for Equinox
• This presentation is contributed by Marc R. Hoffmann, Mountainminds GmbH & Co. KG, made available under EPL 1.0
• Some example code taken from the Apache Jakarta Commons project, provided under Apache License Version 2.0.
• All pictures in this presentation taken from stock.xching, http://sxc.hu/