CoMet: A Tool Using CUMM to Measure Unused Component Members

Msheik, Abran, Mcheick, Touloumis, Khelifi

The 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06), March 8-11, 2006, Dubai/Sharjah, UAE
Agenda

- Background on components
- Problem of component’s unused members
- Need of a Measurement Method
- Component’s Unused Member Measurement (CUMM) method
- Applying CUMM: Example
- Comet In action
- Conclusion and future directions
Background on Components

- Component
  - Simple object oriented class
  - Conforming to a component model (EJB, COM, CORBA)
  - Subsystem
  - Complete application

- Component members
  - Attributes
    - Simple
    - Nested components
  - Operations
Problem of Component’s Unused Members

- Unused attributes
- Unused operations (functionalities) [2]

Consequences
- No functional value
- Leads to waste of memory resources
- Might increase network traffic
- Might compromise the application integrity and security
Need a Quantitative Measurement Method

- How much unused **members** a component has?
- How much **memory** they consume?
Component’s Unused Member Measurement (CUMM) method

- Measures statically
  - Unused members
    - Attributes
    - Operations
    - Attributes and operations memory consumption

- Statistical formulas
  - Percentages of unused members
  - Generality degree of a component’s members
Challenge to Develop CUMM on a Sound Basis

- Traditionally measurement method are defined in terms of formulas
- Resorted to Measurement Method process defined [1]
CUMM Assignment Rules

- Number of unused attributes \( u_a = |A| \), \( A \) is the set of a component’s unused attributes
- Unit is \( ac \) (attribute per component)
- Number of unused operations \( u_a = |F| \), \( F \) is the set of a component’s unused operations
- Unit is \( fc \) (function per component)
CUMM Development process

- Total memory consumed by unused attributes where $m_{ai}$ is the memory consumed by the $i$-th unused attribute

$$t_{ma} = \sum_{i=1}^{A} m_{ai}$$

- Total memory consumed by unused operations where $m_{fi}$ is the memory consumed by the $i$-th unused operation

$$t_{mf} = \sum_{i=1}^{F} m_{fi}$$
CoMet (Component Measurement)

- CoMet automates the application of CUMM
- CoMet is a prototype developed in Java
- CoMet measures binary Java components
- Provided with a simple GUI
- Reuses apache BCEL (Byte Code Engineering Library)
CoMet in Action: Example I

```java
public class ShowWelcomeMessage {
    static String message;

    public static void main(String[] args) {
        message = new String("Hellow Dubai!");
        System.out.println(message);
    }
}
```
## Measurement Results Example I

<table>
<thead>
<tr>
<th>Element</th>
<th>Used</th>
<th>Instruction lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShowWelcomeMessage</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute members</strong></td>
<td>yes</td>
<td>1</td>
</tr>
<tr>
<td>Message</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><strong>Method members</strong></td>
<td>Yes</td>
<td>32</td>
</tr>
<tr>
<td>ShowWelcomeMessage.main</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.ShowWelcomeMessage.&lt;init&gt; ()V</td>
<td>Yes</td>
<td>5</td>
</tr>
</tbody>
</table>
Example II

```java
public class Foo {
    Bar _bar;

    public void doFoo() {
        _bar = new Bar();
        _bar.doBar();
    }

    public static void main(String[] args) {
        Foo myFoo = new Foo();
        myFoo.doFoo();
    }
} // end Foo

public class Bar {
    public int doBar () {
        int i = 1;
        int j = 2;
        int res = add(i, j);
        return res;
    }

    public int add (int v1, int v2) {
        int result = v1 + v2;
        return result;
    }

    public int mult (int v1, int v2) {
        int result = v1 * v2;
        return result;
    }
} // end Bar
```
## Measurement Results Example II

<table>
<thead>
<tr>
<th>Element</th>
<th>Used</th>
<th>Instruction lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.Bar</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method members</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.Bar.mult (II)I</td>
<td>No</td>
<td>6</td>
</tr>
<tr>
<td>testpackage.Bar.&lt;init&gt; ()V</td>
<td>Yes</td>
<td>5</td>
</tr>
<tr>
<td>testpackage.Bar.doBar ()I</td>
<td>Yes</td>
<td>13</td>
</tr>
<tr>
<td>testpackage.Bar.add (II)I</td>
<td>Yes</td>
<td>6</td>
</tr>
</tbody>
</table>
# Measurement Results Example II

<table>
<thead>
<tr>
<th>Element</th>
<th>Used</th>
<th>Instruction lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.Foo</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute members</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.Foo._bar</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td><strong>Method members</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testpackage.Foo.main ([Ljava/lang/String;)V</td>
<td>Yes</td>
<td>13</td>
</tr>
<tr>
<td>testpackage.Foo.doFoo ()V</td>
<td>Yes</td>
<td>20</td>
</tr>
<tr>
<td>testpackage.Foo.&lt;init&gt; ()V</td>
<td>Yes</td>
<td>5</td>
</tr>
</tbody>
</table>
Conclusion and Future Directions

- CoMet provides an automation tool to apply the CUMM method to Java Components
- CoMet provides an indicator of unused members
- CoMet next version
  - Measure the memory of unused members
  - Provide precise measurement result
  - Filter out library components from being measured
  - Enhanced GUI
Thank You!

hamdan.msheik.1@ens.etsmtl.ca, aabran@ele.etsmtl.ca,
References
