

FindBugs – Tool Evaluation



Group 6
Thomas Nourse
Greg Hoch
December 6, 2005

Tool Background

- ☐ A static analysis tool
- ☐ Open source under the LGPL
- ☐ Sponsored by the University of Maryland
- ☐ Differentiates itself through ease of use

The Tool

- ☐ Examines bytecode of class or JAR file
 - ☐ Matches bytecode against list of programming idioms likely to be bugs
 - ☐ Pattern checkers hard coded into the application
-

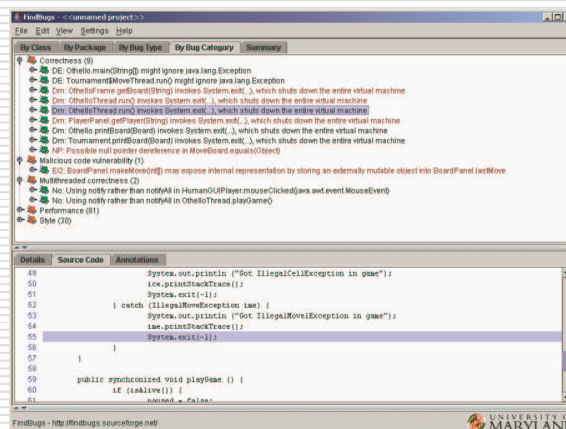
How to Use

- ☐ GUI
 - ☐ Command Line
 - ☐ Ant Task
 - ☐ Eclipse Plugin
-

Additional Features

- ☐ Aims at integration into build process
- ☐ Filter Files
 - Select bugs for special treatment
 - Can select classes to ignore
- ☐ Bug Database
 - Tracks history of bugs
 - Records who introduces bugs

Tool Demo



Error Output

IDE2: Storing reference to mutable object (1)

IDE2: BoardPanel.makeMove(int[]) may expose internal representation by storing an externally mutable object into BoardPanel.lastMove

- In class BoardPanel
- In method BoardPanel.makeMove(int[])
- Field BoardPanel.lastMove
- At BoardPanel.java [line 184]

Details | **Source Code** | **Annotations**

Method may expose internal representation by incorporating reference to mutable object

This code stores a reference to an externally mutable object into the internal representation of the object. If instances are accessed by untrusted code, and unchecked changes to the mutable object would compromise security or other important properties, you will need to do something different. Storing a copy of the object is better approach in many situations.

Details | **Source Code** | **Annotations**

```
180     public void makeMove(int[] location)
181     {
182         throws IllegalMoveException (
183         Board last = board.getClone();
184         board.makeMove(location);
185         lastMove = location;
186         lastBoard = last;
187         repaint();
188     }
```

Sample Errors

- ☐ Bad casts of object references
- ☐ Dropped or ignored exception
- ☐ Dead local store
- ☐ Dubious method used
- ☐ Null pointer dereference
- ☐ Storing reference to mutable object

Benefits

- ☐ Easy to setup and use
 - ☐ Clear explanation and indication of errors
 - ☐ Low rate of false positives
 - ☐ Integrates well into development processes
-

Drawbacks

- ☐ Checks the bytecode not the source
 - ☐ No dynamic program checking
 - ☐ Hard to add additional search patterns
 - ☐ Buggy
 - Java Web Start version does not work
 - Fails to find standard classes sporadically
 - ☐ `java.lang.Object`
 - ☐ `java.lang.Thread`
 - ☐ `java.lang.Throwable`
-

Scope

- ☐ All java code no matter the version
 - ☐ Bits and pieces of a project or the project as a whole

 - ☐ Small to medium projects or companies
 - ☐ Perfect supplement to code reviews
-

Recommendation

- ☐ Bottom Line: It's Practical
 - Limited overhead
 - Reasonable accuracy

Go for it!
