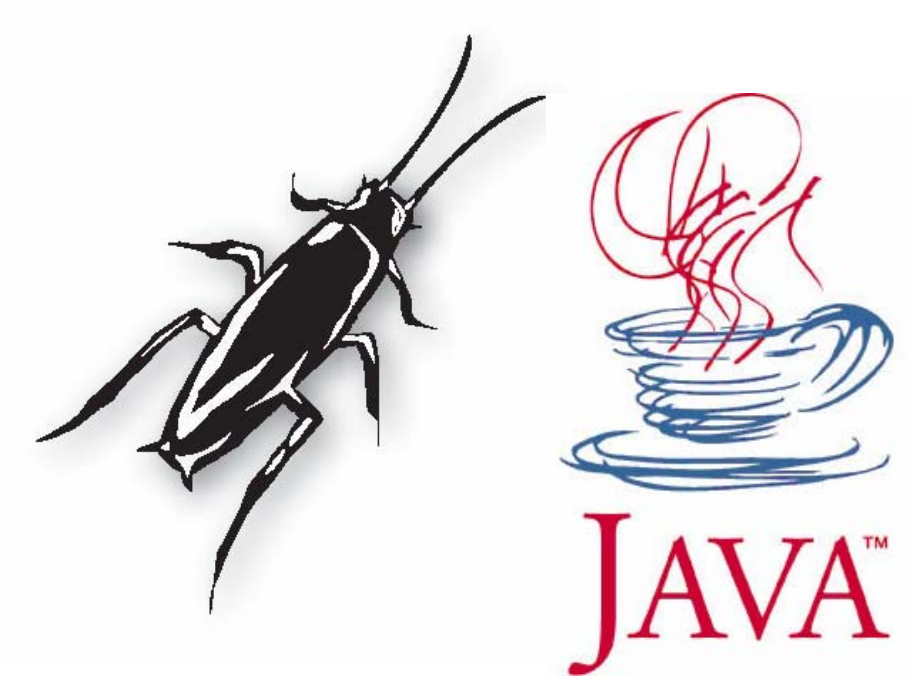
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Software Verification Tools Overview

A Comparison of Bug Finding Tools for Java

Nick Rutar, Christian B. Almazan & Jeffrey S. Foster



Clément Beffa
Vincent Pazeller
Olivier Gobet

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Introduction

- Many available tools
  - Relatively new (few years)
  - Automatically finding bugs
  - Static analysis
  - Different techniques
  - None of them strictly subsumes others
- Proposal of a meta-tool
  - Combines the best of all tools

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

5 Java tools analysed

Name	Version	Input	Interfaces	Technology
Bandera	0.3b2 (2003)	Source	CL, GUI	Model checking
ESC/Java	2.0a7 (2004)	Source[1]	CL, GUI	Theorem proving
FindBugs	0.8.2 (2004)	Bytecode	CL, GUI, IDE, Ant	Syntax, dataflow
JLint	3.0 (2004)	Bytecode	CL	Syntax, dataflow
PMD	1.9 (2004)	Source	CL, GUI, Ant, IDE	Syntax

CL - Command Line

[1] ESC/Java works primarily with source but may require bytecode or specification files for supporting types.

Small example

PMD: Avoid unused local variables

Findbugs: Method ignores results of InputStream.read()

Findbugs: Method may fail to close stream on exception

Findbugs: String comparison using ==

ESC/Java: Array index possibly too large

ESC/Java: Possible null dereference

Jlint: Compare strings as object references

```
1   import java.io.*;
2   public class Foo{
3     private byte[] b;
4     private int length;
5     Foo(){ length = 40;
6       b = new byte[length]; }
7     public void bar(){
8       int y;
9       try {
10        FileInputStream x =
11           new FileInputStream("z");
12        x.read(b,0,length);
13        x.close();}
14      catch(Exception e){
15        System.out.println("Oopsie");}
16      for(int i = 1; i <= length; i++){
17        if (Integer.toString(50) ==
18                  Byte.toString(b[i]))
19          System.out.print(b[i] + " ");
20      }
21    }
22  }
```

Findbugs: does not discover it

Jlint: does not discover it

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Categorization of bugs

Bug Category	**Example**	ESC/Java	FindBugs	JLint	PMD
General	Null dereference	√*	√*	√*	√
Concurrency	Possible deadlock	√*	√	√*	√
Exceptions	Possible unexpected exception	√*			
Array	Length may be less than zero	√		√*	
Mathematics	Division by zero	√*		√	
Conditional, loop	Unreachable code due to constant guard		√		√*
String	Checking equality using == or !=		√	√*	√
Object overriding	Equal objects must have equal hashcodes		√*	√*	√*
I/O stream	Stream not closed on all paths		√*		
Unused or duplicate statement	Unused local variable		√		√*
Design	Should be a static inner class		√*		
Unnecessary statement	Unnecessary return statement				√*

√ - tool checks for bugs in this category * - tool checks for this specific example

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Experiment

- Tested using preliminary meta-tool
  - Parses, combines and coordinates output of the various tools
  - Ranks classes, methods and lines by number of warnings
- Java 1.4
- Five mid-sized testbed
  - Apache Tomcat 5, JBoss 3, Art of Illusion 1.7, Azureus 2 & Megamek 0.29

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Performance evaluation

- Mac OS X v10.3.3
- 1.25 GHz PowerPC G4
- 512 MB RAM

Name	NCSS (Lines)	Class Files	Time (min:sec.csec)				Warning Count			
			ESC/Java	FindBugs	JLint	PMD	ESC/Java	FindBugs	JLint	PMD
Azureus 2.0.7	35,549	1053	211:09.00	01:26.14	00:06.87	19:39.00	5474	360	1584	1371
Art of Illusion 1.7	55,249	676	361:56.00	02:55.02	00:06.32	20:03.00	12813	481	1637	1992
Tomcat 5.019	34,425	290	90:25.00	01:03.62	00:08.71	14:28.00	1241	245	3247	1236
JBoss 3.2.3	8,354	274	84:01.00	00:17.56	00:03.12	09:11.00	1539	79	317	153
Megamek 0.29	37,255	270	23:39.00	02:27.21	00:06.25	11:12.00	6402	223	4353	536

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Analysis

- Too many warnings!!!
  - Impossible to study them all
- Examine a few of them:

	ESC/ Java	Find Bugs	JLint	PMD
Concurrency Warnings	126	122	8883	0
Null Dereferencing	9120	18	449	0
Null Assignment	0	0	0	594
Index out of Bounds	1810	0	264	0
Prefer Zero Length Array	0	36	0	0

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Concurrency Errors

- All tools check concurrency
- Deadlocks
- Concurrency bug pattern
  - Wait() outside while loop
- Jlint produces to much warnings to judge it
  - Duplicate warnings for the same bug

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Null Dereferences

- ESC/Java, Findbugs and JLint only
  - Not a lot of overlap
- JLint multiple report of same warnings
- ESC/Java assumes too many null because of the lack of annotation
- Findbugs uses heuristics to avoid false positive

Array Bounds Errors

- JLint and ESC/Java only
  - Not the same warnings
- Not catastrophic in Java (unlike C)
- JLint: false positive and false negative

```
public class Foo {
  static Integer[] ary = new Integer[2];

  public static void assign() {
    Object o0 = ary[ary.length];
    Object o1 = ary[ary.length-1];
  }
}
```

Jlint: does not discover it

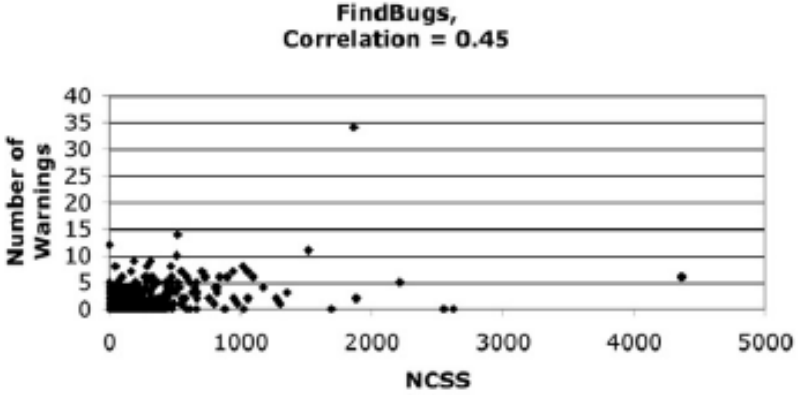
Jlint:Possible Out Of bound

Correlations

- No complete overlapping among tools

Tools	Correlation coefficient
JLint vs PMD	0.15
JLint vs FindBugs	0.33
FindBugs vs PMD	0.31

- No general correlation among lines of code and number of warning



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Issues of the tools

- Not compatible with latest java
- Should offer GUI for ease of use (hyperlink and group bugs) and CL for extracting result (like for a meta-tool)
- Should avoid cascading errors
- Should allow annotation to suppress warnings

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Limitation of the work

- Only Java
- Limited test suite size
- Limited selection of tools
- No deep analysis of found warnings
  - Huge amount of warnings makes it hard to analyze
- No distinction of bugs severity
  - Though severity is relative and subjective

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Conclusion

- Meta-tools seem a good solution to highlight real bugs from warnings
- No tools is sound
- Understanding the right tradeoffs of bugs finding tools is still a wide area of open research

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Project Status

- Listing of tools and characteristics
  - Only open source and downloadable
- Installation and test of tools for java
- Construction of a test case
- Remaining work
  - Enlarging test case scope
  - Analyzing results of the tool
  - Imagining a theoretical meta-tool

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Questions?