641 COMS: Soot Tutorial

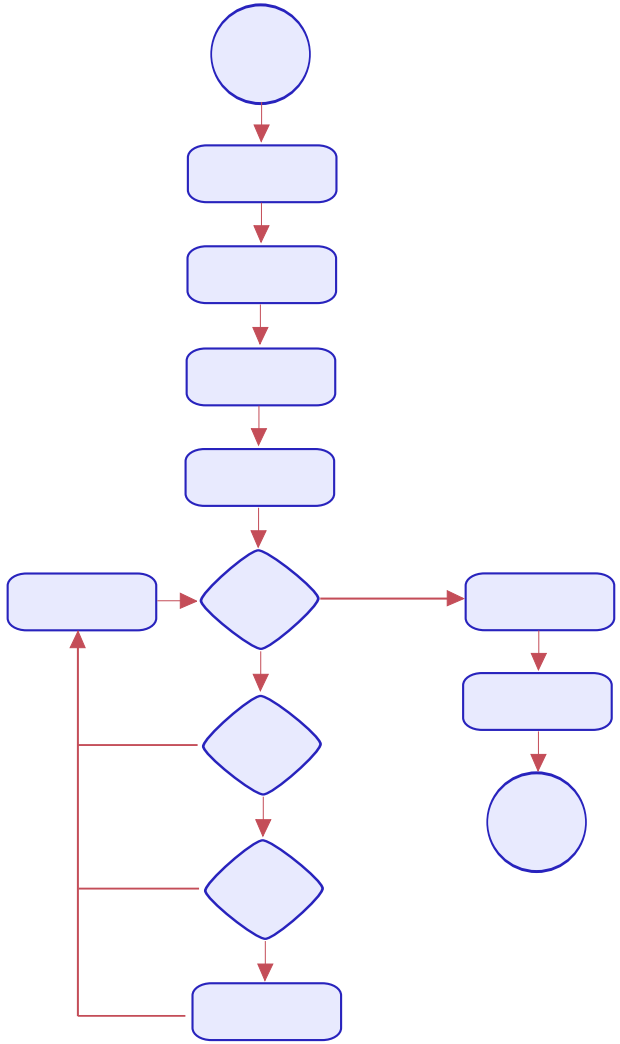
Danilo Dominguez Perez
danilo0@iastate.edu

Departament of Computer Science
Iowa State University

September 26, 2014

What is Soot

- Compiler infrastructure for Java applications
- Provide four Intermediate Representations (IR) for Java: Baf, Jimple, Shimple and Grimp
- Provide different algorithms for points-to analysis, callgraph generation and optimizations



Jimple

- Typed 3-address IR
- Normally just two operands which must be locals or constants (just method invocations have more than two operands)
- All locals explicitly declared
- Only 15 kinds of statements versus over 200 in JVM bytecode

Jimple Formats

Jimple

Jimple API

Jimple ASCII

Java Code

```
public int stepPoly(int x) {
 if(x < 0) {
  System.out.println("error");
  return -1;
 }
 else if(x <= 5)
  return x * x;
 else
  return x * 5 + 16;
}
```

Jimple ASCII Code

```
public int stepPoly(int) {
java.io.PrintStream r1;
Example r0;
int i0;
r0 := @this;
i0 := @parameter0;
if i0 >= 0 goto label0;
r1 = java.lang.System.out;
r1.println("error");
return -1;
label0:
if i0 > 5 goto label1;
i0 = i0 * i0;
return i0;
label1:
i0 = i0 * 5;
i0 = i0 + 16;
return i0;
}
```

Locals

Typed locals are declared at the top of the method

```
 java.io.PrintStream r1;
Example r0;
int i0;
```

3-address IR

Java Code

```
return x * 5 + 16;
```

Jimple ASCII Code

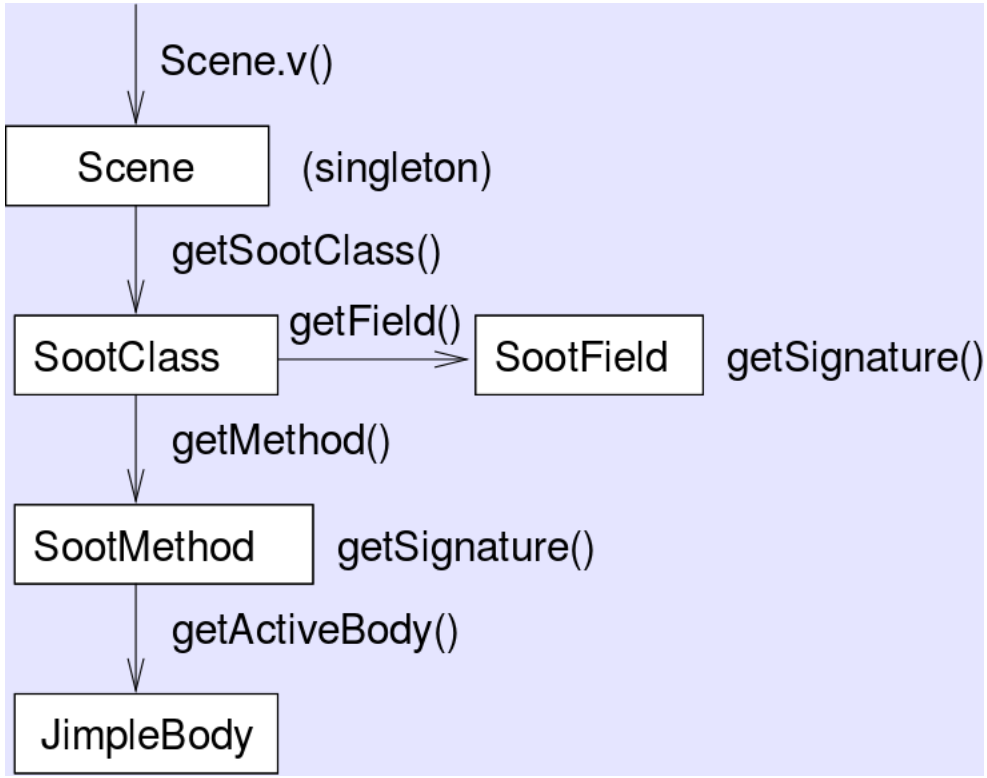
```
i0 = i0 * 5;
i0 = i0 + 16;
return i0;
```

Scene

- The Scene class represents the complete environment the analysis takes place in
- Contains points-to information, call graphs, application classes and more

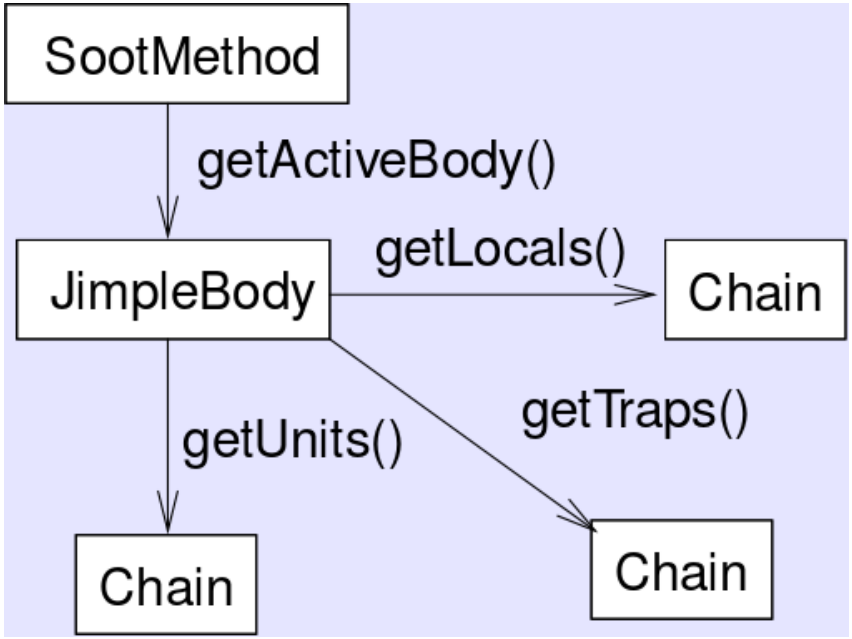
Jimple API

Scene Hierarchy



Jimple API

Method Hierarchy

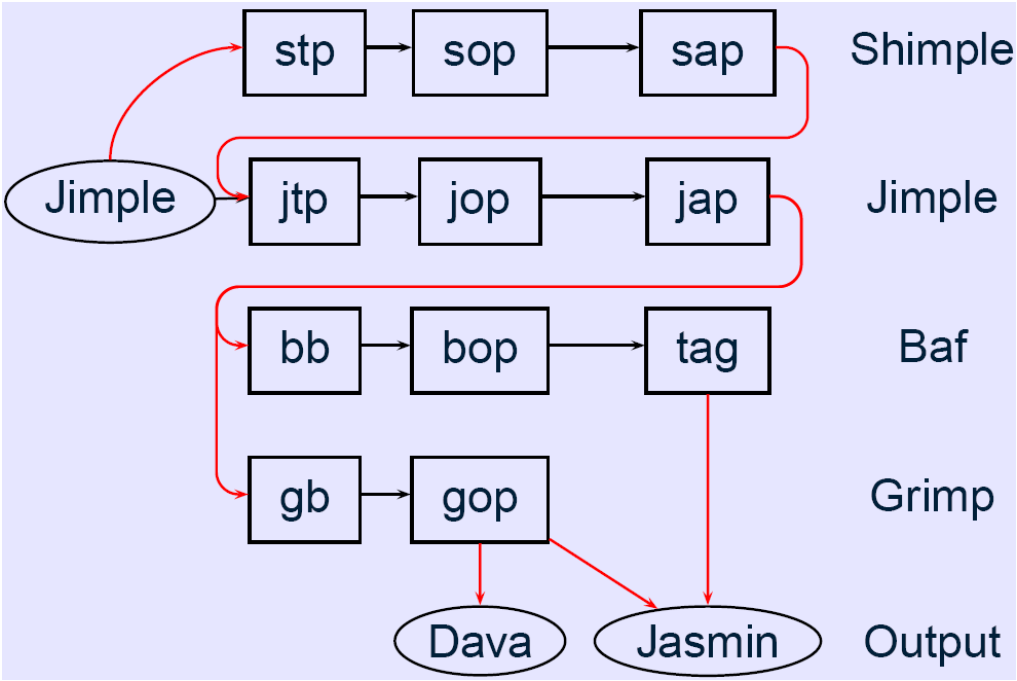


Soot Pashes

- In Soot phases are called *packs*
- There are phases for intra-procedural and inter-procedural analyses

Soot Phases

Intra-procedural Analysis



Code Instrumentation

DEMO 1

Running Demo 1

Parameters

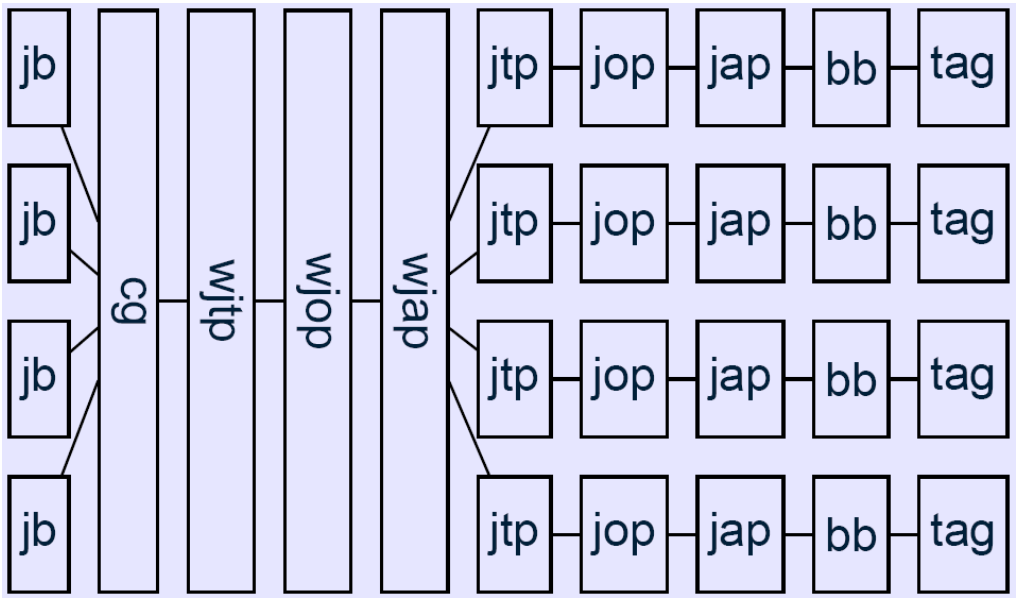
-soot-class-path .:benchmarks/scimark2lib.jar -pp
-allow-phantom-refs -app jnt.scimark2.commandline

Parameters

- -soot-class-path: use *path* as the list of directories in which Soot should search for classes
- -pp: instead of replacing the default soot classpath with the classpath given on the command line, prepent it with that classpath
- -allow-phantom-refs: allow Soot to process a class even if it cannot find all classes referenced by that class
- -app: run in application mode, processing all classes referenced by argument classes

Soot Phases

Inter-procedural Analysis



Static Profiler

DEMO 2

Running Demo 2

Parameters

-soot-class-path .:benchmarks/scimark2lib.jar -pp
-allow-phantom-refs -w -app jnt.scimark2.commandline

Parameters

- -w: perform whole program optimizations on the application classes. This enables the Whole-Jimple Optimization pack as well as whole program mode and intraprocedural optimizations