abc : Extending Java to AspectJ

Ganesh Sittampalam
Oxford University
What is AspectJ?

• An extension of Java designed to support *cross-cutting concerns*

• “Static” features
  – *Intertype declarations* (Open classes)
  – *declare parents* (Hierarchy changes)

• “Dynamic” features
  – Aspect code observes a base program
  – Advice runs at certain *join points* as specified by a *pointcut*
ABC

- A new compiler for AspectJ
- Full, robust implementation
- Research-oriented
  - Extensible
  - Optimising
- Frontend based on Polyglot
- Backend based on Soot
Architecture
Jimple IR

- Three address code
- Originally designed for analysis and optimisations of bytecode
- Framework for propagating information through the process with “tags”
- Great for a compiler backend too
Local io=localgen.generateLocal(BooleanType.v(),"checkType");
Stmt instancetest=Jimple.v().newAssignStmt(io, 
                               Jimple.v().newInstanceOfExpr(v,type));
Expr test=Jimple.v().newEqExpr(io,IntConstant.v(0));
Stmt abort=Jimple.v().newIfStmt(test,fail);
units.insertAfter(instancetest,begiin);
units.insertAfter(abort,instancetest);

checkType = t instanceof Foo;
if checkType == 0 goto labell;
Conclusions

• Soot has performed well as a code generation system

• But needs:
  – Quoting features
  – Code validation