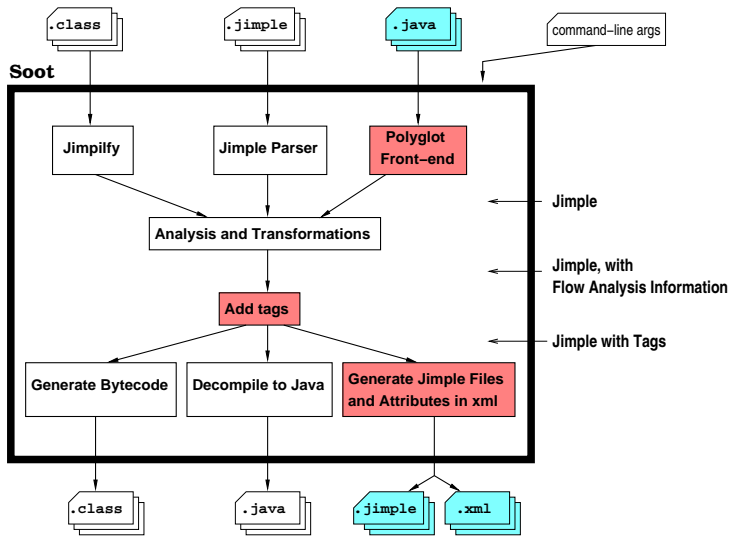


Soot



- Soot main page:
`http://www.sable.mcgill.ca/soot/`
- Soot download page:
`http://www.sable.mcgill.ca/soot/soot_download.html`
- Soot tutorials:
`http://www.sable.mcgill.ca/soot/tutorial/`

Installing Soot

Command-line Soot

Download `soot-2.4.0.jar` and add to `CLASSPATH` environment variable:

```
export CLASSPATH=soot-2.4.0.jar:$CLASSPATH
```

Soot-Eclipse plugin

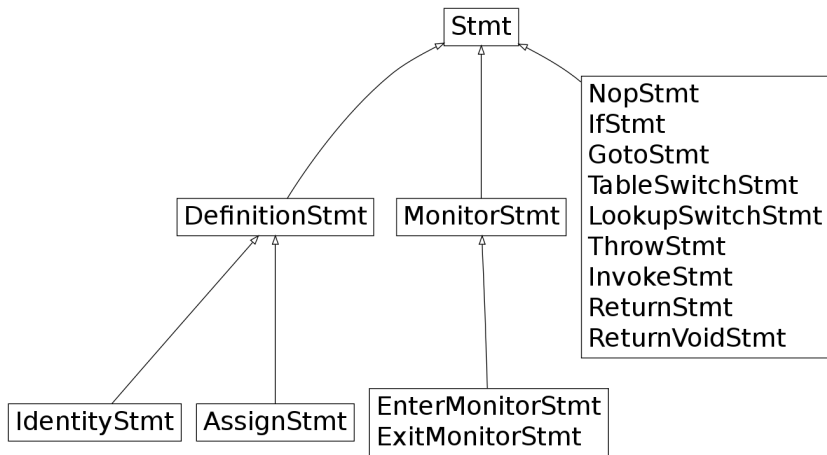
Follow installation instructions at:

<http://www.sable.mcgill.ca/soot/eclipse/updates/>

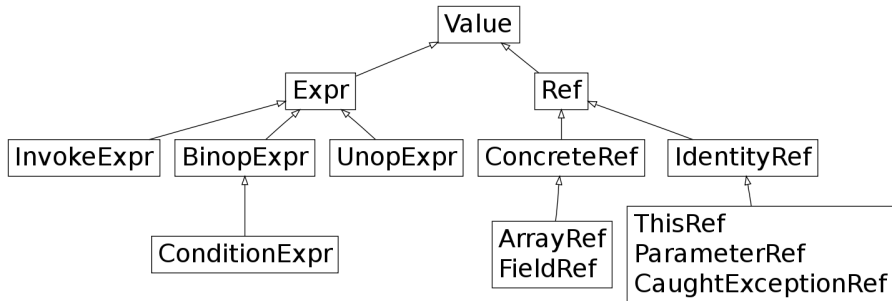
Command-line Examples

- List command line options
`java soot.Main --help`
- Process MyClass.class, output .class file
`java soot.Main MyClass`
- Process MyClass.class, output .jimple file
`java soot.Main -f jimple MyClass`
- Process MyClass.java, output .class file
`java soot.Main -src-prec java MyClass`
- Process MyClass.class, output .jimple file with tags
`java soot.Main -f jimple -print-tags MyClass`

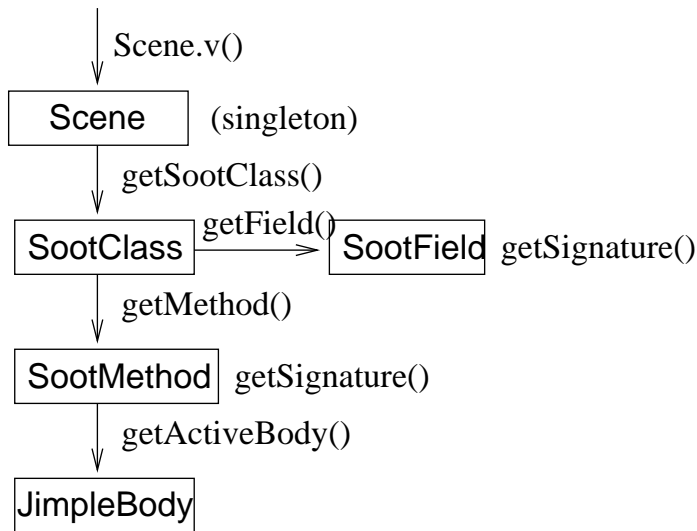
Jimple Statements



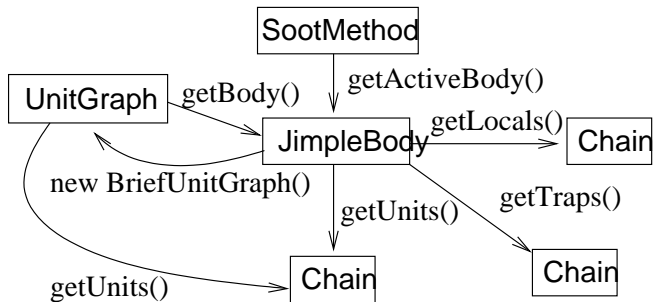
Jimple Expressions



Soot Classes

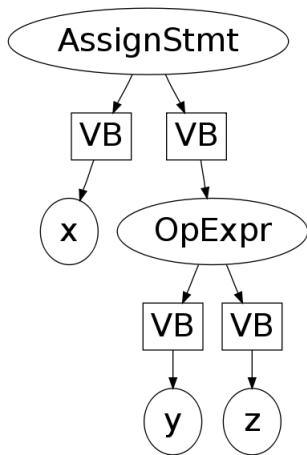


Soot Classes



“Boxes” (References)

s: $\boxed{x} = \boxed{y \text{ op } z}$



Soot Attributes

- We often want to attach annotations to code
 - to convey low-level analysis results, such as register allocation or array bounds check elimination to a VM
 - to convey analysis results to humans
 - to record profiling information

Tags are objects that can be attached to Hosts:

```
package soot.tagkit;
public interface Tag {
    public String getName ();
    public String toString();
}
```

Hosts are objects that can hold **Tags**:

```
package soot.tagkit;
public interface Host {
    public void addTag (Tag t);
    public Tag getTag (String aName);
    public List getTags ();
    public void removeTag (String name);
    public boolean hasTag (String aName);
}
```

Implementations:

SootClass, SootField, SootMethod, Body, Unit, ValueBox

Visual Representations

- Three visual representations of attribute information:
 - Text displayed in tooltips
 - Color highlighting of chunks of code
 - Pop-up links

String Tags

- **StringTags** attach a string of information to a **Host**.
`s.addTag(new StringTag(val+": NonNull"));`

- **ColorTags** attach a color to a **Host**.

```
v.addTag(new ColorTag(ColorTag.GREEN));  
v.addTag(new ColorTag(255, 0, 0));
```

- **LinkTags** attach a string of information, and a link to another part of code to a **Host**.

```
SootMethod m;  
String text = "Target:"+m.toString();  
Host h = m;  
String cName = m.getDeclaringClass().getName();  
s.addTag(new LinkTag(text, h, cName));
```