



## Trial Report of Sensor

发件人: [REDACTED]@neusoft.com<[REDACTED]@neusoft.com>

时 间: 2020年2月25日(星期二) 下午2:13

收件人: [REDACTED]

Dear [REDACTED]

I am happy to provide my trial results of Sensor. Generally speaking, Sensor is an useful tool for detecting the semantic inconsistency problems. The detailed trial report of Sensor is described as follows:

## Trial Report of Sensor

**Requirement.** In general, large-scale industrial projects could introduce hundreds of third-party libraries. These direct or transitive libraries evolve asynchronously, which can easily cause dependency conflicts (also known as “Jar hell” in companies). If more than one versions of a library are present on the classpath, the behaviors will be unpredictable. First, because of the class loading mechanism of JVM, Maven or Gradle, only one version of them can be packaged into the final Jar file, and the other versions will be shadowed. As such, we always get `NoClassDefFoundError` or `NosuchmethodError` crashes. Even worse, unexpected semantic program behaviors could also be introduced in this scenario, and they are very difficult to detect. When a library is upgraded, we should perform time-consuming tests focus on these problems, before releasing our projects. This task is challenging for our developers. Sensor is an effective tool to diagnose semantic conflict issues caused by “Jar hell” problem. We applied this tool in our projects and obtained satisfying diagnosis results.

**Installation.** Sensor has been tested on Windows 10 operating system (64 bit) with JDK 1.8, Maven 3.5.4, soot 3.2.0, Evosuite (a secondary developed version), and JD-core 1.0 installed.

Installing the third party dependencies of Sensor, needs to perform the following three steps:

1. Launch the windows console as an administrator and execute the commands:

```
mvn install:install-file -Dfile=Soot.jar -Dpom=pom.xml -DgroupId=org.sensor -DartifactId=soot -Dversion=3.2.0 -Dpackaging=jar
```

```
mvn install:install-file -Dfile=evosuite.jar -Dpom=pom.xml -DgroupId=org.sensor -DartifactId=evosuite-client-sensor -Dversion=1.0 -Dpackaging=jar
```

```
mvn install:install-file -Dfile=jdcore.jar -Dpom=pom.xml -DgroupId=org.sensor -DartifactId=jd-core -Dversion=1.0 -Dpackaging=jar
```

2. Change the working directory to the one of a Java project being unzipped.
3. Run the following command to diagnose semantic conflict problems:

```
mvn -f=pom.xml -DresultPath=./ -DignoreTestScope=true -Dmaven.test.skip=true neu.lab.X:1.0:semanticsConflict -e
```

-DresultPath= is a path of the output results' directory, which can be customized.

**Effectiveness.** We used Sensor to analyze 10 Java projects and found 56 issues. We confirmed 46 issues are bugs need to be fixed. Overall, it achieves a high precision of 0.821. The detailed results are listed in the following table:

**Table Results of analyzing ten projects (we anonymized the project names)**

Projects	LOC	The number of API pairs with the same signature in the conflicting libraries	The number of API pairs with the same signature having different implementations in the conflicting libraries (detected by static analysis)	The number of API pairs causing inconsistent semantic behaviors across different library versions detected by test cases (True positive results)	False positive results
Project 1	>100K	103	44	7	2
Project 2	>100K	78	9	3	0
Project 3	>100K	213	36	10	2
Project 4	>100K	52	20	5	2
		89	25	6	1
Project 5	>50K	31	7	2	1
Project 6	>50K	64	12	3	0
		45	11	3	1
Project 7	>20K	23	7	2	0
Project 8	>20K	42	8	3	0
Project 9	>20K	15	4	1	1
Project 10	>20K	7	2	1	0
Precision	Precision = 46/56 = 0.821			46	10

**User Experience.** This tool can automatically output the following key information that can help us debug the semantic inconsistent/ Jar hell issues:

### 1) *Root cause:*

It generates the warnings of the conflicting library versions in our project's dependency tree and points out the versions that cannot be loaded.

### 2) *APIs with changed implementations in the conflicting libraries:*

It can also output all the subtle differences between the above conflicting API pairs, including the added code lines, deleted code lines or changed code lines.

### 3) *Test cases:*

The generated tests can effectively help us reproduce such semantic inconsistencies caused by dependency conflicts. The arguments used in the test cases are meaningful for our program, which gives us a hint that what inputs can trigger these bugs. The above information is indeed useful and time-consuming for us to manually obtained.

### 4) *Differences of test outcomes:*

This tool specifies the variables in the methods of our projects that have different states when running on different library versions. This is the most important symptoms of the semantic conflict issues.

The diagnosis results generated by Sensor, also solved two bugs that have already reported in our issue tracking systems and puzzled us for three weeks. These are the surprising results to us!

**Performance.** On average, it took about 20.5 hours to obtain the diagnosis report for a large-scale Java project, and the run time depends on the number of conflicting APIs in the project. Although the testing task is time-consuming, Sensor did a great job in automatically detecting such semantic conflict issues. The generated diagnosis reports indeed helped us identify many issues that could hardly be found using our existing test suites and developers' observations.



资深软件工程师  
医学影像产品部

医疗解决方案事业本部 临床医疗事业部  
东软集团股份有限公司  
沈阳市浑南区新秀街2号东软软件园  
邮编: 110179  
手机: [REDACTED]  
邮箱: [REDACTED]@neusoft.com  
[www.neusoft.com](http://www.neusoft.com)

**Neusoft**

-----  
Confidentiality Notice: The information contained in this e-mail and any accompanying attachment(s) is intended only for the use of the intended recipient and may be confidential and/or privileged of Neusoft Corporation, its subsidiaries and/or its affiliates. If any reader of this communication is not the intended recipient, unauthorized use, forwarding, printing, storing, disclosure or copying is strictly prohibited, and may be unlawful. If you have received this communication in error, please immediately notify the sender by return e-mail, and delete the original message and all copies from your system. Thank you.  
-----