DependEnCy Conflicts Assessment

Released raw data

We release the following datasets and tool under the MIT License:

- DECCA, our dependency issue assessment tool, which is implemented as a Maven plugin to warn developers whether the issues are benign or harmful. You can find it in package **plugin-decca.zip**, which contains the following files and directories:

  (1) File **decca-1.0.jar** is a copy of DECCA's implementation for running analysis tasks.

  (2) File **decca-1.0.pom** is the pom file (**p**roject **o**bject **m**odel) of DECCA project.

  (3) File **soot-1.0.jar** is a program static analysis tool referenced by DECCA.

  (4) The directory **apache-maven-3.2.5** is the Maven project. As DECCA is a Maven plugin, its implementation depends on Maven project.

  (5) File **script.jar** is the script for evaluating the experimental results shown in Section 5.1. With the help of this script, analyzing the ground truth dataset can be run in batch mode.

- Raw data we used in our experiments(Section 5). You can find raw datasets in package **RawData.zip** which is available at:

  https://we.tl/PiShDzvpaF

  The package **RawData.zip** contains the following two parts of data:

(a) The ground truth dataset used to verify the effectiveness of DECCA (Section 5.1). You can find it in directory **RawData\Ground_truth_dataset**, which contains the following files and directories:

  (1) File **Ground truth dataset.xlsx** provides the explanation for the dataset.

  (2) The directory **RawData\Release_conflict** contains the collected host projects with 43 high-severity and 176 low-severity dependency issues.

  (3) The directory **RawData\Release_fixed** contains the above host projects whose high-severity dependency issues have been fixed.

  Running DECCA on **RawData\Release_conflict** and **RawData\Release_fixed** can reproduce our experimental results described in Section 5.1.

(b) 24 Java projects we analyzed to verify the usefulness of DECCA (Section 5.2). In directory **RawData\Issue_report_dataset**, you can find the following files and directories:

  (1) File **Issue_report_dataset\Reports.docx** lists the links of dependency conflict issue reports we submitted to their corresponding bug tracking systems(in Section 5.2).

  (2) The directory **Issue_report_dataset\Projects** contains the 24 subjects with dependency conflict issues.

  (3) The directory **Issue_report_dataset\Generated issue reports** contains the issue reports generated by DECCA.

Running DECCA on them can reproduce our experimental results described in Table 3 of Section 5.2.

How to use DECCA

DECCA can take a Maven based project (it should contain the complete Maven built project directory and file pom.xml) as input for analysis. The expected running environment is 64-bit Window operating system with JDK 1.7 or 1.8. **As Maven built projects need to download dependencies from Maven Central Repository, DECCA cannot work offline**.

You can run DECCA on our experimental subjects based on the following steps:

Step 1: Unzip the **plugin-decca.zip** to local directory. Recommended directory structure is:

D:\plugin-decca
  ├──decca-1.0.jar :
  ├──decca-1.0.pom
  ├──soot-1.0.jar
  ├──apache-maven-3.2.5
  ├──script.jar

*Note: To facilitate testing, please keep the unzip directory to be consistent with the above example.

Step 2: Install DECCA.

(a) Execute the following Windows CMD command to install soot:

```
D:\plugin-decca\apache-maven-3.2.5\bin\mvn.bat   install:install-file   -Dfile=D:\plugin-decca\soot-1.0.jar   -DgroupId=neu.lab   -DartifactId=soot -Dversion=1.0 -Dpackaging=jar
```

(b) Execute the following Windows CMD command to install DECCA:

```
D:\plugin-decca\apache-maven-3.2.5\bin\mvn.bat   install:install-file   -Dfile=D:\plugin-decca\decca-1.0.jar   -DgroupId=neu.lab   -DartifactId=decca -Dversion=1.0 -Dpackaging=maven-plugin -DpomFile=D:\plugin-decca\decca-1.0.pom
```

Step 3: Detect and assess the dependency conflict issues. (Please unzip the package **RawData.zip** under directory **D:\.**)

Execute the following Windows CMD command to analyze the project:

```
D:\plugin-decca\apache-maven-3.2.5\bin\mvn.bat   -f=D:\RawData\Issue_report_dataset\Projects\hadoop-rel-release-3.0.0\hadoop-common-project\hadoop-minikdc\pom.xml   -Dmaven.test.skip=true   neu.lab:decca:1.0:detect   -DresultFilePath=D:\Report\resultFile.xml -DdetectClass=true -e –Dappend=false –e
```

> Command explanation:
> (1) -f=pom file : Specify the project under analysis;
> (2) -DresultFilePath=output issue report directory : Output the issue report to the specified file;
> (3) -DdetectClass=Boolean : Specify the tool whether reports the class level conflicts or not;
> (4) -Dappend=Boolean : Specify the result output mode (whether in append mode or not).

Then you can get the dependency issue report in your specified directory (e.g., **D:\Report\resultFile.xml**).

Reproducing the experimental results in Section 5.1

We provide the script to run DECCA on ground truth dataset in batch mode. Please unzip the package **RawData.zip** under directory **D:**. After completing the installation **steps 1** and **2**, please execute the following Windows CMD command:

```
java -jar D:\plugin-decca\script.jar D:\RawData\Ground_truth_dataset
```

Then you can automatically get the *Precision*, *Recall* and *F*-measure results. The overall process would cost you about 2~3 hours.

*Note: We improved the source code of DECCA after paper submission deadline. The current release can identify dependency conflict issue with higher *Precision*, *Recall* and *F*-measure values. We will update the results in the camera ready version.

Reproducing the experimental results in Section 5.2

According to installation **steps 1~3**, you can run DECCA on any one of the 24 java projects (under the directory **RawData\Issue_report_dataset\Projects**) described in Section 5.2. Then the experimental results shown in Table 3 can be reproduced.

To facilitate evaluation, we also provide the issue reports of the 24 projects generated by DECCA (described in **RawData\Issue_report_dataset\Generated issue reports**) and the issue report records we submitted to their corresponding bug tracking systems (**RawData\Issue_report_dataset\Reports.docx**).