

Software Product Line Evolution Tool

Tutorial

1-Importing the Plugin Project

Open the Git Repositories View in the eclipse. Use the option *“Clone a Git Repository and add the clone to this view”*.

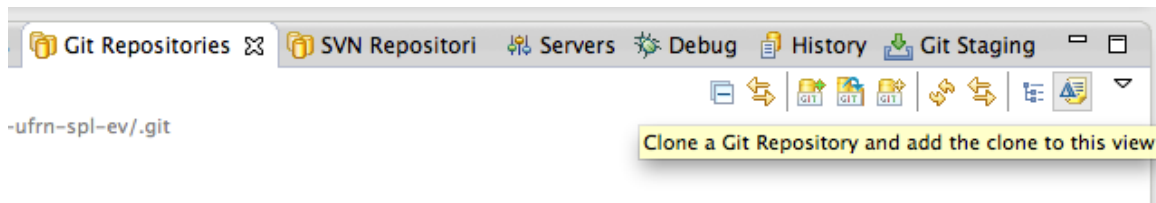


Figure 1: Clone the tool git repository

Enter the repository address (<https://jadsonjs@bitbucket.org/jadsonjs/br-ufrn-spl-ev.git>) and your user and password

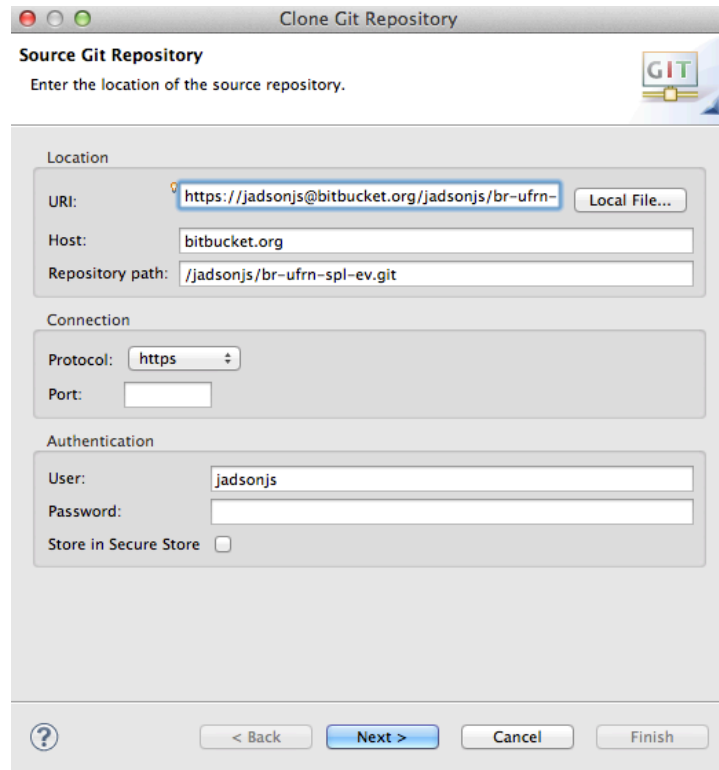


Figure 2: Clone the tool Git Repository – step 2

Import the plugin project to your eclipse workspace:

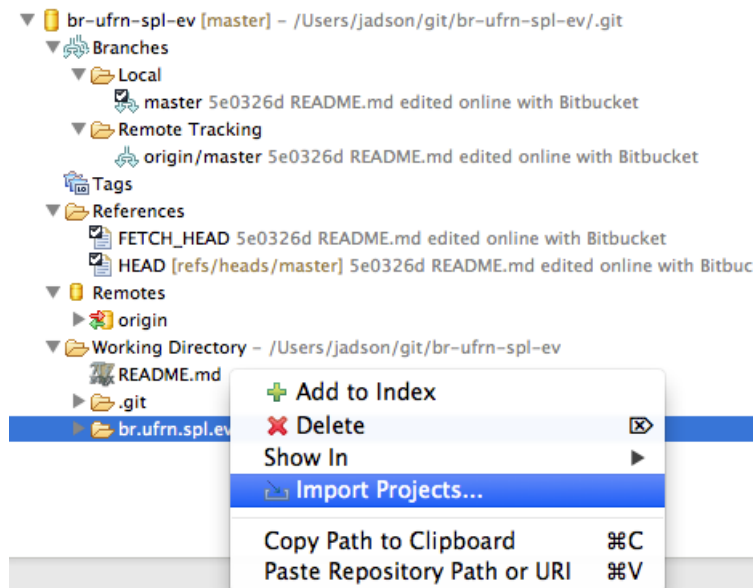


Figure 3: Import the Project to the eclipse workspace

If you find some conflicts problems when checkout the tool, please make sure that your eclipse have all the plugin dependences declared in the plugin.xml file.

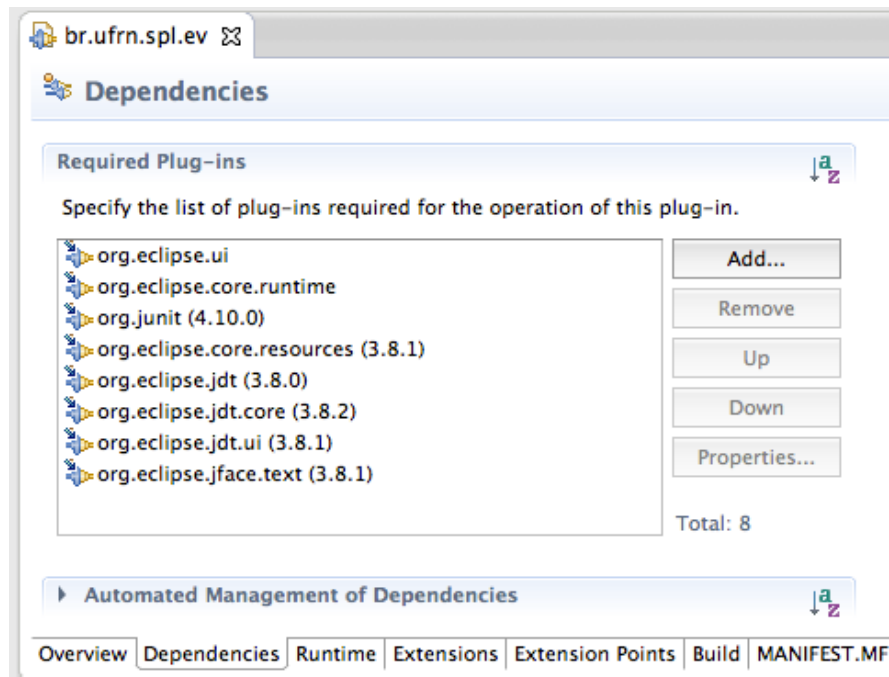


Figure 4: Checking the plugin dependence

2 – Configuring the Tool

After import the project to the eclipse workspace, it is necessary to configure the tool parameters in the **config.properties** file, show in the Listing 1.

a - config.properties file:

Listing 1. config.properties file

```
#####  
#      file of general tool configurations      #  
#####  
  
# where save the things, ends with "/"  
DEFAULT_WORK_DIRECTORY=/Users/your_user/temp_directory/  
  
# the indirect analysis strategy that will be used: allow values: "JDT" or "WALA"  
INDIRECT_ANALYSIS_TYPE=JDT  
  
# the level of the dependence analysis  
DEPTH_ANALYSIS_LEVEL=2  
  
#The miner the will be used on the source side  
#Have to be declared in the miners.properties file  
#Actual values are: IPROJECT, REDMINE, SIGPROJECT an  GITHUB  
SOURCE_MINER=IPROJECT
```

The configuration options of this file are describe bellow:

DEFAUL_WORK_DIRECTORY = The tool work directory. It is the place where the tool will read all information necessary and save the result of the analysis. It must to and with “/”

ANALYSIS_TYPE = The type of dependence analyzer used, if JDT or ~~WALA~~ (deprecated).

DEPTH_ANALYSIS_LEVEL = The depth indirect conflict analysis level.

SOURCE_MINER = The source miner used. Supported (IPROJECT, SIGPROJECT, REDMINE, GITHUB). If now informed, the miner on the source side will not be executed.

TARGET_MINER = The target miner used. Supported (IPROJECT, SIGPROJECT, REDMINE, GITHUB). If now informed, the miner on the source side will not be executed.

REPOSITORY_CONNECTOR_SOURCE = The connector used in the source side, have to be declared in the connectors.properties file.

Actual values are: SVN, GIT

REPOSITORY_CONNECTOR_TARGET = The connector used in the source side, have to be declared in the connectors.properties file. Actual values are: SVN, GIT

EXTRACT_EVOLUTION_MODEL = The way of the miner will be executed, if accessing the configuration manager system (ONLINE) or reading theses information of the file (OFFLINE)

INDIRECT_CONFLICT_MODEL= The way of the indirect conflict will be execute, if will try to checkout the source code of the cloned of a control version system (ONLINE) or if will consider that the source code already has been checkouted (OFFLINE)

CHANGELOG_TREE_VIEW = The way that the evolution information will be organized on the conflict tree (MODULE or FILE).

STATISTIC_ANALYSIS_TYPE = The type of statistic analysis that will be executed (SIMPLE, COMPLETE). Used to collect data for our researches.

STATISTIC_ANALYSIS_MODEL= The way how the tool will get the project path. Two values supported: (DIRECT) or (INDIRECT)

SOURCE_PROJECT_TO_ANALYSIS= Define the source project name where the statistics analysis will executed. Fill just if you use the value "DIRECT" in STATISTIC_ANALYSIS_MODEL property

TARGET_PROJECT_TO_ANALYSIS= Define the target project name where the statistics analysis will executed. Fill just if you use the value "DIRECT" in STATISTIC_ANALYSIS_MODEL property

b - connections.properties file:

It is necessary also to configure the connections information of Configuration Manager Systems and Control Version Systems in the file **connections.properties**. Figure 1 shows the location of this file inside of the tool project.

Listing 2 shows an example of a configuration for analyze SIGAA evolutions from version 3.11.3 to version 3.11.18 in the source side.

Listing 2. connections.properties file

```
# SOURCE SYSTEM
SOURCE_SYSTEM_URL=http://localhost:8080/iproject/public/rest/build evolution information
SOURCE_SYSTEM_USER=
SOURCE_SYSTEM_PASSWORD=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
SOURCE_SYSTEM_NAME=SIGAA
SOURCE_MODULE_NAME=LIBRARY
SOURCE_SYSTEM_START_VERSION=3.11.3
SOURCE_SYSTEM_END_VERSION=3.11.18
SOURCE_SYSTEM_OFF_LINE_EVOLUTION_FILE=.xml

# SOURCE REPOSITORY
SOURCE_REPOSITORY_URL=http://desenvolvimento.info.ufrn.br/projetos
SOURCE_REPOSITORY_PATH=/trunk/SIGAA
SOURCE_REPOSITORY_USER=user
SOURCE_REPOSITORY_PASSWORD=123456
```

3 – Running the Tool

After finishing all configurations, you can run the tool as an Eclipse Plugin as shown on Figure 2.

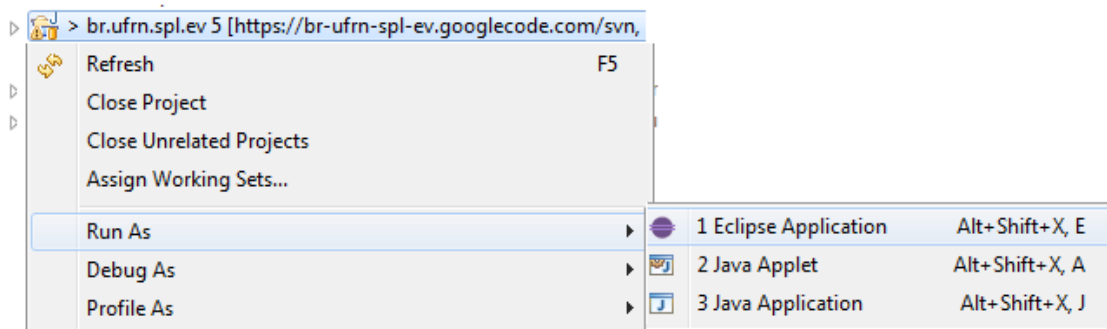


Figure 5: Run as an Eclipse Plugin

Ps.: The plugin will be executed in the **runtime-EclipseApplication** workspace, created automatically when you execute a project as an eclipse plugin.

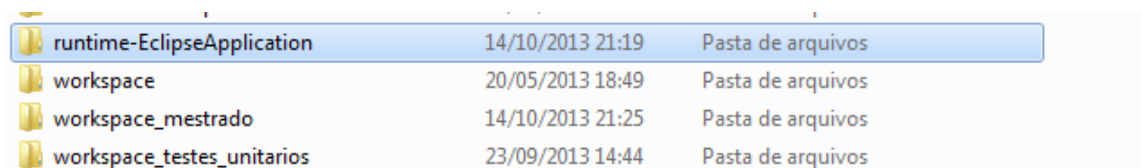


Figure 6: Workspace of plugin execution

4 – Starting the Analysis

As shown at Figure 4, a new menu called “Evolution Analysis” will be present in the eclipse main menu. This menu opens the main screen of the tool.

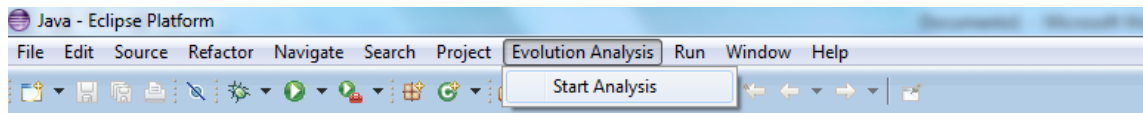


Figure 7 Starting the Tool

In the tool main screen, Figure 5, click in the “Execute new Evolution Analysis” button. The analysis will start according with the tool configuration done in the Section 2.

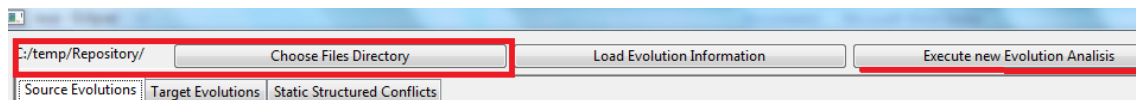


Figure 8: Execute new Evolution Analysis

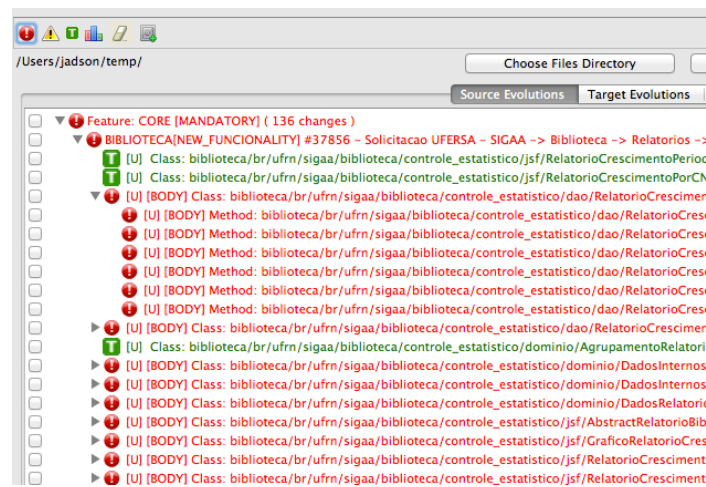
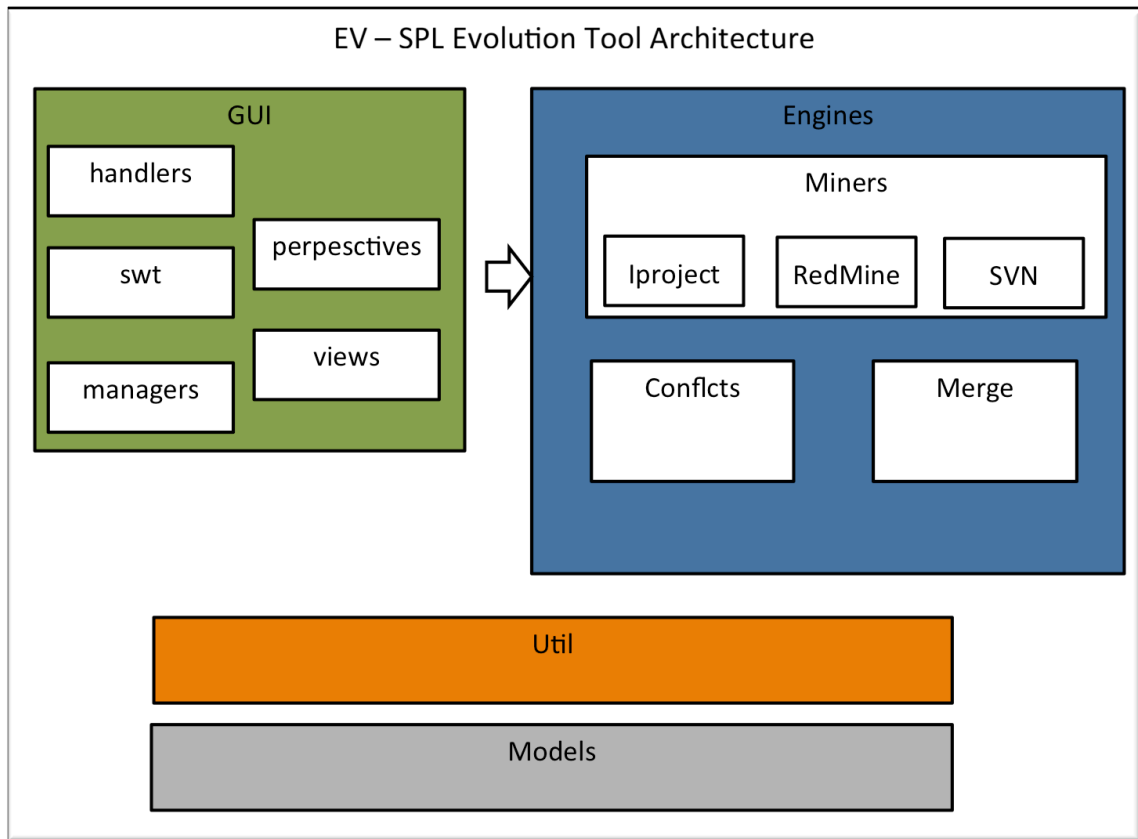


Figure 9: EV – software product line evolution tool

5 – Tool Architecture



The tool is divided if four main parts. GUI classes have the graphic user interface. Engines execute the miner and conflict analysis. The package Models contains the information models generated by the tool. Finally, a package with utility classes.

To understand to tool source code, start with the classes of the package **Handlers**. These classes are classes that capture the events of eclipse graphic interface.

Usually at this point, a class of the **SWT** package will be executed. The tool main screen, where the most of operations start, is the **ShowEvolutionsUI.java**.

The SWT classes call classes from the **Managers** package. Managers are classes the control the tool execution flow.

Usually the next classes that will be executed are the classes in the **Engine** package.

Inside of the Engine package we have the **Miners** package. IProjectMiner, SIGProjectMiner and RedmineMiner are the miners implemented until the moment.

Inside of Engines package we have another package called **Conflicts**. This package contains the classes responsible by the conflict analysis.

The tool execution flow is showing in the Figure 10 (in Portuguese).

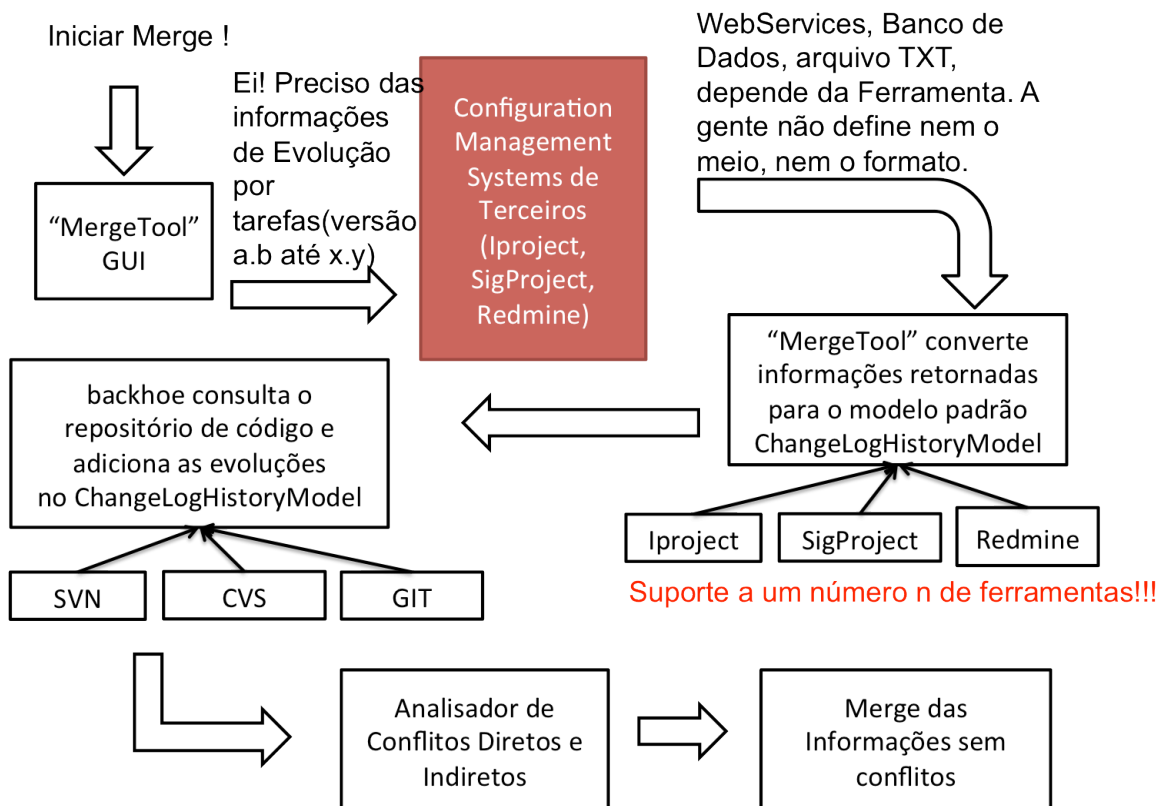


Figure 10: Tool Execution Flow