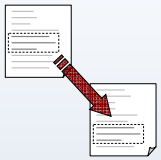
Visualization of Clone Detection Results

http://www.cis.uab.edu/tairasr/visual

Robert Tairas (tairasr@cis.uab.edu)
Jeff Gray (gray@cis.uab.edu)
Ira Baxter (idbaxter@semanticdesigns.com)

Software Composition and Modeling Laboratory
Department of Computer and Information Sciences
University of Alabama at Birmingham

Introduction



- The goal of a clone detection tool is to identify sections of code that are duplicated in a program. The result of the detection is presented in some manner for the user to view, which is usually in the form of a list of clones that are grouped together. Previous research has shown how scatter plots can be used to render a graphical representation of the results.
- This poster describes the integration of a stand-alone clone detection tool into Eclipse and a corresponding alternative visualization of clone detection results.
- An Eclipse plugin is described that displays the results of a clone detection tool called CloneDR™. The visualization of the results is implemented as an extension to the AspectJ Development Tool (AJDT) Visualiser plugin, which is primarily used to view crosscutting concerns in aspect-oriented programs.

Tool Integration & Extension

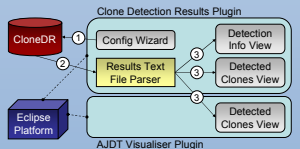
CloneDR

- A clone detection tool from Semantic Designs, that examines the abstract syntax tree representation of a program where subtrees are matched according to a calculated similarity value to detect code clones in the program.
- Can evaluate large software applications and is applicable to multiple languages.
- Java version is freely available at: http://www.semanticdesigns.com/Products/Clone/register-download.html

The AJDT Visualiser Plugin

- Part of the AspectJ Development Tools (AJDT) project.
- Recently opened for adaptation to allow visualization of entities other than aspects.
- Examples of other tools that use the Visualiser include applications toward Google search results and CVS file histories.
- AJDT Visualiser website: http://www.eclipse.org/ajdt/visualiser

Plugin Architecture and Process

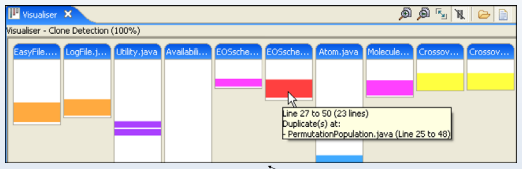


The plugin plays the role of an interface between CloneDR and the user. It performs the duties of setting up the configuration file, executing CloneDR as an external task, and displaying the clone detection results to the user.

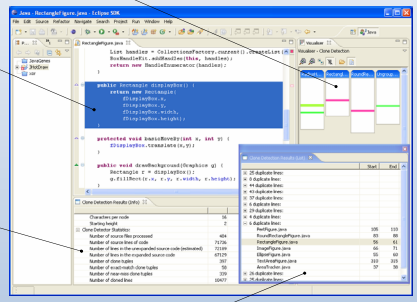
- When the process is started, the plugin will display a configuration wizard that assists the user to determine the settings for the clone detection procedure. The configuration settings are stored in a file.
- CloneDR is then called and given the location of the configuration file (step 1). CloneDR will execute and produce a text file containing its clone detection results.
- The results are parsed by the plugin (step 2) and sent to three Eclipse views (step 3). One of these views is the Visualiser extension that produces a graphical representation of the results of the clone detection.

Inside the Plugin

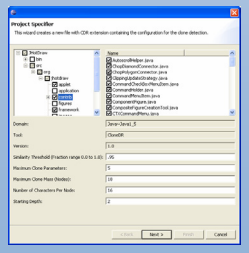
- In the visualization, bars represent source files and stripes represent clones.
- Stripes with the same color are clones of each other.

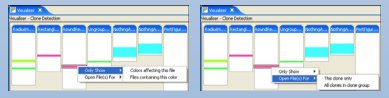


- The source file (with the clone lines highlighted) will be displayed if a clone representation is double-clicked in either the graphical or textual views.
- Statistical information about the clone detection procedure itself in addition to the configuration settings are displayed in the clone detection results information view.



- The clone detection results textual view displays clones in groups, where each clone is represented by the file name that it is in and its starting and ending lines.
- Displaying clones in a text list is similar to that used in the Eclipse plugins for Simian and SimScan, which are also clone detection tools. However, these plugins don't provide any visualizations.

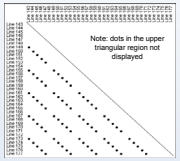




- The context menu when right-clicking on a clone displays options that include:
  - Display all bars containing the same clones as in the selected bar
  - Display all bars containing the selected clone
  - Open the file containing the selected clone
  - Open all files containing clones in the same clone group as the selected clone
- All but one of the options above required editing the internal code of the Visualiser, because no related extension points were available.
- The configuration wizard allows users to select the configuration settings for the clone detection procedure, including the selection of files and specific detection values.

Experimental Results

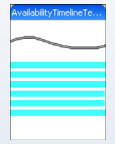
- The plugin was applied on two open source programs, JavaGenes (a scientific NASA program) and JHotDraw (a GUI framework).



(a) Scatter plot

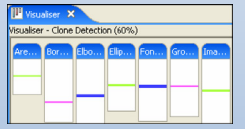
5 duplicate lines:		
AvailabilityTimelineTest.java	143	147
AvailabilityTimelineTest.java	149	153
AvailabilityTimelineTest.java	155	159
AvailabilityTimelineTest.java	161	165
AvailabilityTimelineTest.java	167	171
AvailabilityTimelineTest.java	173	177

(a) Text listing



(c) Visualiser view

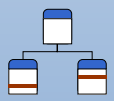
- Both JavaGenes and JHotDraw contain an example where one group of clones is located in multiple locations in the same source file.
- The figures above provide a sample of how the clones are displayed in the three different types of representations.
- The scatter plot consists of dots where each dot represents two statements that are duplicates of each other. When multiple dots generate a diagonal line, the section of code that is represented by the dots are clones.
- Compared to the diagonal lines, the Visualiser view provides a straightforward representation of the sections of code that are duplicated in a single source file.



- In the figure on the left, several clones can be seen to be ubiquitous in that they are present throughout multiple source files in JHotDraw.
- The Visualiser view can filter a display to show only bars containing a specific clone.
- Non-connected sections that contain the same clone can not be grouped together in scatter plots.

Future Work

- *A more structured view of the source code files*: If the language is object-oriented, the classes could be displayed with respect to their relationships in a manner similar to a UML class diagram. This can provide more understanding of clones compared to the current flat display.



- *Demarcation of methods in bars representing source files*: Clones can also be duplicate methods. Displaying the borders of methods in the view can assist in determining which clones are method-level clones.

- *Tighter integration of CloneDR in Eclipse*: A mechanism that can allow CloneDR to be integrated further into Eclipse could simplify and speed up the detection and visualization process by providing the Visualiser direct access to the CloneDR internal representation.

Eclipse
CloneDR