

Exercise Sheet 6

Due: Sunday 07.06.2009, 23:59:59 via SVN

For help, contact aosd-staff@lists.iai.uni-bonn.de (staff only) or
aosd-course@lists.iai.uni-bonn.de (staff and participants).

Please start working on the exercises early enough so that you can contact us in time in case of problems. Don't expect us to be available during weekend!

For the first two exercises you will need [FINT](#). FINT is one of the aspect-mining tools introduced in the lecture. The name is derived from the Fan-In Analysis.

How to use FINT:

- Download the .jar file and move it into the plugin directory of your eclipse.
- To see if FINT installed correctly open "Window" -> "Show View" -> "Other..." -> "FINT"
 - If the installation does not work, try an eclipse without AspectJ
- The first step then is to "right-click" onto the project you want to analyse and to run FanInAnalysis.
 - Experiment with the filters of FINT to get good results. The developers say that a good precision for FINT is around 50%.
- To further analyse a candidate you can "right-click" onto an element and then click onto "Go Into"
- Further information on how to use FINT are given in the [HowTo](#)

In your repository you will find a version of JHotDraw that will be used as codebase for the first two exercises.

Exercise 1: "FINT True Positives" (4 Points)

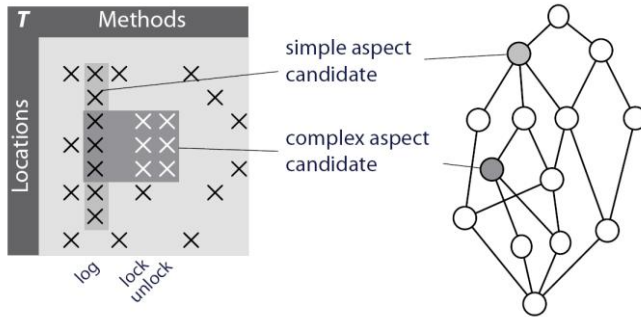
Use FINT to find two good candidates for a "refactoring to aspects". Give at least four reasons why your candidates are good candidates. The reasons should be based on your experience with aspect oriented languages, your knowledge of software engineering and the criterions given in the lecture.

Exercise 2: "FINT False Positives" (4 Points)

Use FINT to find two false positives.

While looking for them, can you find some candidates for an object-oriented refactoring?

Exercise 3: "Formal Concept Analysis" (6 Points)



One of the techniques used in aspect mining is the formal concept analysis. Among other things it is used for the grouped calls analysis. The grouped calls analysis aims to find groups of methods that invoke the same set of methods. The example gives you an expression how the binary relation "method calls method" leads to the formation of a context. The calling methods are the objects and the called methods are the attributes.

As an example you will now explore a small part of JHotDraw. The constructors of the `*Action` classes were used as objects and the called methods are their attributes.

We want you to show us that the following statements are true and where you can find them in the diagram.

- There exists one attribute that is called 13 times.
- The concept in the middle has no attributes annotated. Name the attributes that belong to this concept.
- The statement a) is represented by one concept.
- Every object that has the `ZoomAction` attribute also has another attribute.
- There are two concepts that have only two non-trivial superconcepts.
- If an object has the two attributes `MoveToFrontAction` and `CutAction` than it also has...

