

# Software Architectures

## Assignment 1: Design Patterns

Assistants: Kennedy Kambona, Janwillem Swalens

Mail: {kkambona, jswalens}@vub.ac.be

Office: {10F730, 10F719}

Deadline: 5<sup>th</sup> March 2015, 23:59

In this assignment, you will explore the source code of jEdit to examine its use of design patterns and to study its coupling and cohesion. jEdit<sup>1</sup> is an open-source text editor, written in Java, with support for amongst others syntax highlighting, code folding, find and replace, and plug-ins.

The assignment consists of three exercises, in which you will need to examine the jEdit source code to find instances of a number of given design patterns, recognize design patterns in the code, and lastly study the coupling and cohesion of some components of the code.

## Assignment

For this assignment you will write a report, answering the three exercises described in this document.

**Deadline 5<sup>th</sup> March 2015 at 23:59.** The deadline is fixed and will not be extended.

**Deliverables** Write a report (in English) answering the questions given in this document. The report should be handed in as a single PDF file.

The file should follow the naming schema (Firstname-Lastname\_)\*1.pdf, for example: Kennedy-Kambona-Janwillem-Swalens\_1.pdf.

Submit it on the Software Architectures course page<sup>2</sup> in PointCarré, by clicking on *Assignments (Opdrachten) > Assignment 1*.

**Team work** You are allowed to work alone or in a team of two. Only one of you should submit the report on PointCarré, but be sure to mention both names in the report! Note that copying – whether from previous years, from other teams, or from the

---

<sup>1</sup>See <http://jedit.org/> and <https://en.wikipedia.org/wiki/JEdit>

<sup>2</sup>Use the English variant “Software Architectures”, rather than the Dutch one “Software Architecturen”.

internet – will not be tolerated, and can lead to a zero for the complete course. If you use any other resources besides those provided in the lectures and in this document, remember to cite them in your report.

**Grading** The exercises will be graded and can become subject of an additional defense.

## Exercise 1: Find Instances of Design Patterns

First, set up Eclipse to view the jEdit source code, using the instructions in the last section of this document: *Preparing Eclipse*.

In the first exercise, you should find instances of a number of design patterns used in the jEdit source code. In the folder `org/gjt/sp/jedit`, identify one instance of each the following design patterns:

1. Singleton
2. Abstract Factory
3. Observer
4. Adapter
5. Visitor

For each instance:

- Say whether this is a creational, structural, or behavioral pattern.
- Describe the participants: give their class or method name.
- Create a class diagram showing the involved participants. Only include elements necessary for the design pattern.
- Describe the motivation and application of the pattern in its concrete situation in 50-100 words. Do not give a general description of the design pattern.

In practice, instances of these patterns might slightly deviate from the situation described in the Design Patterns book. In that case, describe where the instance deviates from the original pattern, and (if possible) why.

Be aware that real instances of these patterns do not need to follow the exact naming conventions of the Design Patterns book.

## Exercise 2: Recognize Design Patterns

jEdit supports undo and redo. The file `org/gjt/sp/jedit/buffer/UndoManager.java` contains the component that manages the undo and redo functionality. It also contains a number of inner classes, which represent the actions that can be undone, such as `Insert`, `Edit`, `Replace`, and `CompoundEdit`. These classes implement two design patterns, one structural and one behavioral. For each of these two patterns:

- Say which design pattern you found, and whether this is a creational, structural, or behavioral pattern. A list of patterns can be found in the lecture slides.
- Describe the participants.
- Describe the motivation and application of the pattern in its concrete situation in 50-100 words. Do not give a general description of the design pattern.

## Exercise 3: Coupling and Cohesion

(a) First, answer the following questions:

- Which is preferable: high or low cohesion? Why?
- Which is preferable: tight or loose coupling? Why?

(b) **Cohesion** For each of the following classes:

- `MiscUtilities`
- `GUIUtilities`
- `io/VFSFile.java`

answer the following questions:

- Which type of cohesion do you find in this class? The types of cohesion (coincidental, logical, etc.) are described in the lecture slides.
- Is this high or low cohesion, and why? If it is a bad type of cohesion, how could it be improved in this case?

(c) **Coupling** Examine the relation between the classes `jEdit` and `GUIUtilities`:

- Which type of coupling do you find between these classes? The types of coupling (content, common, etc.) are listed in the lecture slides.
- Is this tight or loose coupling? If it is a bad type of coupling, how could it be improved in this case?

## Preparing Eclipse

The assignment uses **Eclipse for Java EE Developers** and **jEdit 5.2.0**.

What follows is a short step by step description on how to setup Eclipse for your platform and open the given source code of jEdit. You are not required to use Eclipse, you can use any editor to find the design patterns in the given code.

1. Download “Eclipse for Java EE Developers” from <http://www.eclipse.org/downloads/>.
2. Extract the downloaded file to a path of your choice and start Eclipse. We recommend creating an empty workspace for these assignments.
3. Download `jedit.zip` from PointCarré.
4. In Eclipse, choose File > Import...
5. Next, choose General > Existing Projects into Workspace (**not** Archive File).
6. Choose “Select archive file” and select the downloaded `jedit.zip`. Press Finish.
7. The `jedit` project contains the source code of jEdit 5.2.0. You will be working in the folder `org/gjt/sp/jedit`.