1. Metrics and Visualization

(Last Update: 2024-02-20)

The purpose of the laboratory exercises in general is to stimulate discussion about the properties of a well-structured object-oriented design. The tasks and assignments are designed to stimulate your comprehension of the topic and the tools. In most cases, you can use alternative projects and/or tools to accomplish the same or similar tasks, or even more!

This session is to provide you the first contact with the sample projects and tools. Consider the tools used in this session as the providers of data required to build up an argument for a system report or action plan.

Sample Projects and Tools

Projects

- pacman-python (https://github.com/kilincceker/pacman-python-ua-sre)
- django CMS (https://github.com/django-cms/django-cms)

IDE

- PyCharm CE (https://www.jetbrains.com/pycharm) – You can use other IDEs.

Tools

- CodeScene (https://codescene.io) - No installation necessary, but it requires a GitHub account. This tool's integration with GitHub allows it to visualize your repositories.
- JsCity (https://github.com/ASERG-UFMG/JSCity/wiki/JSCITY) – This is an implementation of CodeCity (http://wettel.github.io/codecity.html) to analyze JavaScript code.

Book

- Object-Oriented Reengineering Patterns (OORP - http://scg.unibe.ch/download/oorp)

Setup / Preparation

First, for a project that you want to analyze into your GitHub account. This is necessary because the free version of CodeScene can only see projects in your own account.

Second, go to CodeScene website, and log in with your GitHub account. Once logged in, you can choose to create a new project, where CodeScene will show all the GitHub projects in your account. Select the project you want to analyze, click next and wait until it finishes (it might take a while, depending on the project).

Download/Clone the same project and open it on the IDE of your choice (for example, PyCharm). Build and run the project. Refer to the additional documentation if necessary.

Also, if you have not already, download the book for this course *"Object-Oriented Reengineering Patterns"* *(Note: OORP, p.xx refers to a page in the pdf version of this book.)*

Task 1: Introduction -- React

This first task has two goals: (i) to help familiarize yourself with the CodeScene interface; and (ii) to observe that not every visualization is useful for refactoring.

Visit CodeScene website and click at the Showcases. These are examples of projects analyzed by CodeScene (from a full version of CodeScene and not only the free version we are using). Select "React" project.

Visit the JsCity website and look at the examples. You may want to select a simple example first to get acquainted. After that, also select "React" project to visualize (and be patient because it may take a while).

What do you think of these visualization tools? Can you extract important information about the visualized project from them? Which visualization would be more useful to plan refactoring activities?

Task 2: Hands-on -- pacman-python

For the second task, the goal is to start getting acquainted with pacman-python source code. Download/clone the repository and run the application. Now look at the source code and other files in the project. Try to understand its internal structure.

As stated in the book (OORP, p.36), this is your "First Contact" with the software that needs reengineering activities. Often, we ask ourselves: "Where do I start?" (OORP, p.37).

Well, pacman-python is a simplified Python implementation of Pacman game (https://en.wikipedia.org/wiki/Pac-Man). What features can be added to pacman-python?

Is it possible to implement those features right now; or should we reengineer the project to make it easier to add the new features? The last question is rhetorical for this task, but you should think about it when doing a reengineering project.

Task 3: Visualization tool -- pacman-python

The third task is to use our visualization tool of choice (for example, CodeScene) to identify possible reengineering targets. Since we already had the "First Contact", now we should move to the "Initial Understanding" (OORP, p.83) of the system. One important reengineering pattern is to "Study the Exception Entities" (OORP, p.107).

Using CodeScene, can you identify artifacts that appear to be out of place? Could they benefit from refactoring activities? How is the complexity measurements for such artifacts compared to others?

Task 4: Another Project -- django CMS

Repeat the previous tasks using Django CMS project.

Discussions and Conclusion

We have used metrics and visualization for the Initial Understanding, thereby demonstrating their potential support to the reengineering process.

Did the metrics and visualization tools used in this lab session provide you with the information you needed to fulfill the goals of the different Reengineering Patterns? Set up an argument about the issues concerning the use of such tools: when would you use them, when wouldn't you? Can you envision other usages of these tools in other phases in the Reengineering Lifecycle?