Project description - Master thesis work at UiO

Erlend Kristiansen, erlenkr

November 30, 2012

Refactoring, Software Metrics and Design Patterns

Refactorings are a common tool of the trade for programming. In general, they are assumed to be equivalent transformations of a program, to improve the quality of the source code with regard to a concrete or subjective software metric, such as coupling.

To aid in this matter, many of the common IDEs (e.g. Eclipse, IntelliJ, NetBeans, MS Visual Studio) offer tool support for automated refactorings. While the support is satisfactory in many of these environments, they generally support mostly low level refactorings, such as the rename refactoring, the extract method refactoring, the extract class refactoring and so on.

To a certain degree, literature on the subject [1, 2] suggests a relationship between refactorings and design patterns [3], in the sense that a design pattern may be the target of a series of refactorings. The aim of this project will be to pursue this relationship further. After finding a suitable tool I will try to implement one or more higher order refactorings (i.e. they are composed of lower level refactorings). If this succeeds, I will investigate the relationships between selected software metrics and design patterns, and whether this can be used to make tool support for refactoring toward patterns.

Schedule

First couple of months Some background literature must be read. In addition to the literature that has already been referred to, the literature on software metrics may for instance include one or more of [4, 5, 6, 7]. Initial work could also include looking in Open Source software repositories for examples where refactorings have been applied and classify them [13]. It would also be natural to study existing approaches to describe execution of refactorings [8, 9, 10, 11, 12, 14].

Then a suitable tool need to be chosen for the implementation of the refactoring(s). Preferably one that has accessible lower level refactorings to build on. As part of this process, the "extract-and-then-move-including-the-insertion-of-an-assertion method" refactoring from the master thesis proposal "Safer Refactorings" [15] by Volker Stolz, should be used as an excercise to understand the refactoring support of the tool investigated, starting with the Eclipse JDT [16].

Next five months (approx.) Performing implementations and experiments.

Last six months (approx.) Write research report. This work must be done partially in parallel with some of the more practical parts of the project.

References

[1] Martin Fowler et al., *Refactoring: Improving the Design of Existing Code*, 1999.

[2] Joshua Kerievsky, *Refactoring to Patterns*, 2004.

[3] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1994.

[4] Norman E. Fenton, Shari Lawrence Pfleeger, *Software Metrics: A Rigorous and Practical Approach, Revised*, 1998.

[5] Michele Lanza, Radu Marinescu, *Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems*, 2006.

[6] Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, 2002.

[7] Andreas Vogelsang, Ansgar Fehnker, Ralf Huuck, Wolfgang Reif, *Software Metrics in Static Program Analysis*, 2010.

[8] Franck Fleurey, ystein Haugen, Birger Mller-Pedersen, Gran K. Olsen, Andreas Svendsen, Xiaorui Zhang *A Generic Language and Tool for Variability Modeling*, 2009.

[9] Max Schaefer, Oege de Moor, *Specifying and implementing refactorings*, 2010.

[10] Mrcio Cornlio, Ana Cavalcanti, *Sound refactorings*, 2010.

[11] Mark O'Keeffe, Mel Cinnide, *Search-Based Refactoring: an empirical study*, 2000.

[12] Marios Fokaefs, Nikolaos Tsantalis, *Identification and application of Extract Class refactorings in object-oriented systems* 2012.

[13] Gustavo Soares, Bruno Catao, Catuxe Varjao, Solon Aguiar, Rohit Gheyi, Tiago Massoni, *Analyzing Refactorings on Software Repositories*, 2011.

[14] L. Groves, *A formal approach to program modification*, 2000.

[15] Volker Stolz, *Thesis proposal:* `http://heim.ifi.uio.no/~stolz/Volker_Stolz_-_UiO/Project_Safer_Refactorings.html` 2011.

[16] Eclipse IDE: `http://www.eclipse.org`