LEVEL 0 SUMMARY

Name of student:

BURLON Carla

Name of your Level 1:

GONCALVES MELIM Maria Lolita

Source (e.g. scholars.google.com):

Paper available on Github

Paper title:

A Catalogue of Refactorings for Model-to-Model Transformations, Journal of Object Technology , Published by AITO — Association Internationale pour les Technologies Objets JOT 2011 .

Keywords specific to the paper:

Meta-model evolution, Co-evolution, OCL adaptation, and QVT-R.

Summary of the main contributions:

In the field of Model-driven Engineering, a methodology that aims to create and exploit conceptual models which are abstract representations of the data and interactions of a specific field, model to model transformations are a common subject. This paper addresses the issue of refactoring, a concept often used in traditional object oriented programing, and how it applies to model to model transformation.

This paper focuses on the maintainability aspect of M2M (Model to Model) operations. This entails mainly better organisation in the code which offers better readability, efficiency and faster iteration on changes implementations. This paper presents a set of refactoring implementations for rule based transformations. The presented refactoring while providing maintainability benefits may also offer better performances. Also their execution can be semi-automated using High Order Transformations.

The paper then presents an example of M2M transformation using ATL (specifier) and OTL (*specifier*) which clearly shows the need for a refactoring paradigm : duplicated lines of codes, useless function calls which hamper performance and all the heavy lifting done by a single rule as well as the use of deprecated constructs.

The catalogue of refactorings presented uses a format where each refactoring is described by a name, the typical situation where it should be applied, the solution, the preconditions, the parameters needed to execute it and the actual refactoring steps.

The proposed refactorings are then evaluated by a set of metrics, even though it is hard to obtain information on the readability, ease of extension and other maintainability-related aspects of the code through metrics. The metrics mainly focus on having reduced the number of elements and parameters (rules, constraints) , the number of code duplicates or similar bad practices and lastly the complexity of the dependencies. On several of these metrics the refactoring portrayed have brought significant improvements which coincidentally also led to huge performance gains.

The execution performance of several models where measured before and after refactoring and the results consist of consistent speed ups across the board which clearly showcase the relevance of these refactorings.

The main limitation of this paper is the lack of automation, as the refactoring described require the suer to manually choose which refactoring to apply (based on the description mentioned before. The transformation is however then applied automatically through the High Order Transformations.