Identifying Method Friendships to Remove the Feature Envy Bad Smell

Rocco Oliveto[1], Malcom Gethers[2], Gabriele Bavota[3], Denys Poshyvanyk[2], and Andrea De Lucia[3]

[1]STAT Department, University of Molise, Italy

[2]Department of Computer Science, College of William & Mary, USA

[3]Department of Mathematics and Informatics, University of Salerno, Italy

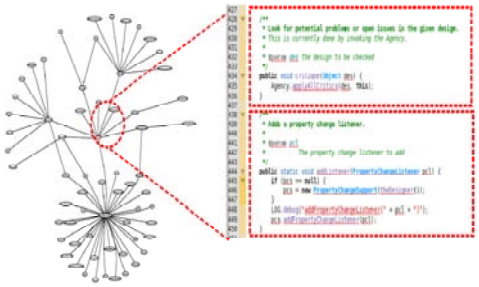
Abstract

We propose a novel approach to identify Move Method refactoring opportunities and remove the Feature Envy bad smell from source code. The proposed approach analyzes both structural and conceptual relationships between methods and uses Relational Topic Models to identify sets of methods that share several responsibilities, i.e., "friend methods". The analysis of method friendships of a given method can be used to pinpoint the target class (envied class) where the method should be moved in. The results of a preliminary empirical evaluation indicate that the proposed approach provides meaningful refactoring opportunities.

Facebook Metaphor

The social network *Facebook* is capable of suggesting friends to users based on user profiles and connections to other users.

We apply a similar concept to source code, leveraging the idea of identifying friends to address the issue of feature envy.

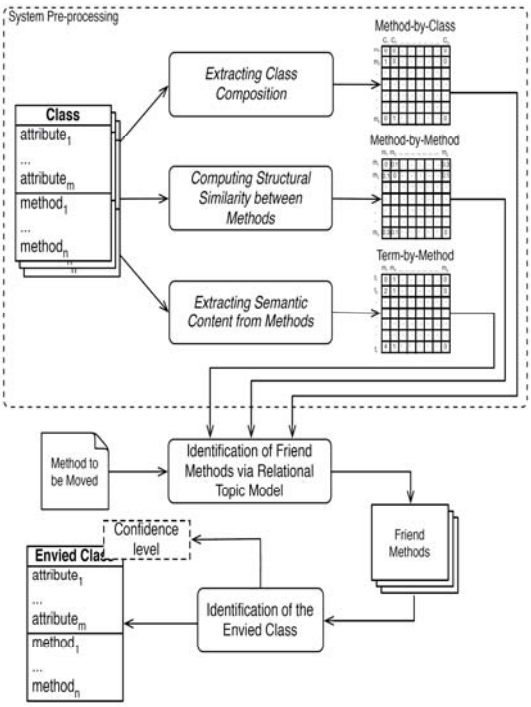


Source code as a social network where usage relationships correspond connections and textual information of methods make up profiles.

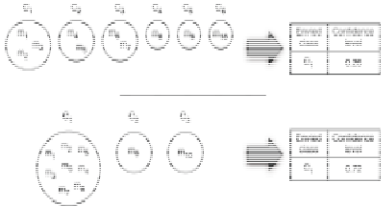
Analysis of Source Code

The method friendships are identified using Relational Topic Models through the analysis of structural and conceptual relationships among methods as well as the original structure of the classes. RTM is a probabilistic topic modeling technique designed to represent and analyze both conceptual (i.e., textual) aspects of documents within a corpus as well as known relationships among the documents (e.g., method calls in software). Once the "best" friends of a given method are identified using RTM, MethodBook analyzes the classes where these methods are implemented aiming at identifying the envied class.

MethodBook



Measuring Confidence



Confidence is determined using information entropy, considering the suggestion of the envied class as a random variable, where the probability of its states is given by the distribution of the friend methods over suggested classes.

Preliminary Evaluation

A preliminary evaluation of our approach was performed on a well-designed open-source system, namely ArgoUML version 0.16. The evaluation aimed at investigating whether MethodBook is able to identify meaningful move method refactoring operations analyzing a given input method. Moreover, we were also interested in verifying whether the proposed approach for computing the confidence level serves as a good estimation of the recommendation reliability provided by MethodBook. We randomly extracted 1,000 methods from the classes of ArgoUML. Then, we applied MethodBook to identify the envied class for each of the extracted methods. The original classes, where the extracted methods are implemented, were used as the oracle to evaluate MethodBook's recommendations for move method refactoring.

$$recall_c = \frac{|original \cap envied_c|}{|original|}\%$$

$$precision_c = \frac{|original \cap envied_c|}{|envied_c|}\%$$

Confidence level	Recall	Precision	Confidence level	Recall	Precision
0.0	0.71	0.71	0.6	0.49	0.89
0.1	0.71	0.75	0.7	0.39	0.98
0.2	0.66	0.79	0.8	0.39	0.98
0.3	0.66	0.79	0.9	0.39	0.98
0.4	0.64	0.82	1.0	0.39	0.98
0.5	0.58	0.85	-	-	-

Conclusion

We proposed to exploit method friendships to build a recommendation system supporting the software engineer in the identification of move method refactoring opportunities. The results achieved in a preliminary evaluation supported the applicability of such a metaphor and highlighted the valuable support given by RTM in the identification of refactoring opportunities.

Publications

R. Oliveto, M. Gethers, G. Bavota, D. Poshyvanyk, and A. De Lucia, "Identifying Method Friendships to Remove the Feature Envy Bad Smell (NIER Track)," in 33rd IEEE International Conference on Software Engineering, 2011.