

Reengineering the Modularity of OO Systems

Fernando Brito e Abreu¹, Gonçalo Pereira, Pedro Sousa²

¹ISEG/UTL ²IST/UTL
INESC, R. Alves Redol, n^o9, Lisboa, Portugal
{fba, gpereira, pms}@inesc.pt

Introduction

The MOOD project within the Software Engineering Group at INESC aims at delivering a broad spectrum quantitative framework to support the design of object oriented systems. A central role in this effort is played by GOODLY (a **Generic Object Oriented Design Language? Yes!**) [Abreu97] which was first conceived for facilitating the extraction of MOOD design metrics [Abreu94]. This language allows to express the most relevant object oriented design information and is used in MOODKIT G2, a set of tools to support the quantitative assessment of object oriented design. One of them allows the extraction of categorized information about couplings between classes, the starting point for the described experiment.

The Methodological Approach

Cluster analysis [Kaufman90, Romesburg90] designates a group of techniques concerned with the classification of similar items into groups. To use it we need some measure of the dissimilarity between items and an algorithm to drive the clustering process, a clustering method. The dissimilarities are summarized in a square symmetric **dissimilarity matrix**. We used a **hierarchical agglomerative clustering** approach and applied several methods: average linkage within and between groups, single and complete linkage, centroid, median and ward methods.

Thirteen different **coupling categories** could be identified in GOODLY: direct inheritance, class parameter, attribute type, employed attribute, employed type, parameter in operation, parameter in message, parameter in call, return in operation, return in message, return in call, local attribute in operation and message recipient. The interconnection strength between two classes, which we called **affinity**, was determined through six distinct schemes of combination (**ratings**) of the available coupling information. The dissimilarity D between classes i and j , depends on the affinity, to which we applied a standardization transformation, so that $D(i,j) \in [0,1]$.

A **sample** of systems was selected according to a set of criteria such as: diversity in level of abstraction, language and application domain, considerable size and maturity. A **dissimilarity matrix** was then calculated for each tuple (*system, affinity rating*) and fed to SPSS, a statistical tool that allows the application of the mentioned hierarchical clustering methods. The execution of the clustering algorithm was stopped when the

number of groups was the same as the original number of modules in the system (as defined by their producers). For each modularization proposal generated by SPSS (one for each clustering method), we produced an **inclusion matrix**, a symmetric square matrix where both columns and rows are the classes in the system under consideration. A “one” in position (i, j) represents that class i and j are in the same module and a “zero” otherwise. A **matching algorithm** was then performed between the original inclusion matrix (based on the proposal of the system producer) and each of those based on the cluster analysis driven proposals. Only the elements below (or above) the diagonal of both matrixes were considered. Matchings in the range of 70 to 95% were obtained for most systems.

Conclusions

We could not observe a significant impact on the resulting matching of either the affinity ratings, or the number of couplings. However, *the adopted clustering method has a strong influence on the matching level*. The two best results were obtained with the average linkage within groups and ward methods and the two worst with the median and centroid methods. For a chosen clustering method, *the matching level increases as the average number of classes per module decreases*.

As a corollary we may say that for systems whose average module membership (classes per module) is not too high, the proposed cluster analysis approach leads to modularization solutions that match very closely those proposed by human experts. This approach can then be used in the reengineering of OO legacy systems, allowing to identify ill-defined modularization situations and to propose alternative ones.

References

- [Abreu94] F. B. Abreu, R. Carapuça, "Object-Oriented Software Engineering: Measuring and Controlling the Development Process", *Proceedings of the 4th International Conference on Software Quality*, McLean, Virginia, USA, October 1994.
- [Abreu97] F. B. Abreu, L. Ochoa, M. Goulão, "The GOODLY Design Language for MOOD Metrics Collection", (<http://www.esw.inesc.pt/ftp/pub/esw/mood>), March 1997.
- [Kaufman90] L. Kaufman, P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Mathematical Statistics, 1990.
- [Romesburg90] H. C. Romesburg, *Cluster Analysis for Researchers*, Robert E. Krieger Publishing Company, Malabar, Florida, USA, ISBN 0-89464-426-2, 1990.