Abstract

Refactoring has become a well-known technique for improving the code in a way that preserves behavior. The application of refactorings during development process of an object oriented or procedure oriented software improves the design and therefore the quality of software. During the evolution of software it is a requirement to refactor them in order to make it more compatible and flexible with the new environment. Much work is being done in refactoring object oriented code with aspect oriented programming. But this paper describes the various types of refactoring being done on procedural codes for eg: C language and the utility of refactoring the procedural codes with the help of aspect oriented programming. The paper also proposes certain refactorings that could be achieved in a better way using AOP.

Reference

- Garrido, A, Ralph Johnson "Challenges of Refactoring C Programs", 2002
- M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts. Refactoring: Improving the
- Jan Hannemann, "Aspect-Oriented Refactoring: Classification and Challenges", 2005
- Wolfgang De Meuter, Bram Adams, "Can we Refactor Conditional Compilation into
  Aspects, 2008
- Magiel Bruntink, Arie van Deursen, "Discovering Faults in Idiom Based Exception
  Handling", 2005
- S.Schulze, M.Kuhlemann, M Rosenmuller "Towards a Refactoring guideline Using Code
  Clone Classification", 2008.

Index Terms
Computer Science Programming

Languages

Key words
Legacy Systems
Aspect Oriented Programming
Refactoring techniques
Object Oriented Refactorings
Procedural languages challenges