Abstract

Software quality based applications development is the main concern is user satisfaction. It increases the reliability and efficiency of information retrieval and management. As the bundle of code created day by day the repository storing such code is regularly migrates the older code into legacy systems. To develop and facilitate new object-oriented model based application with improved problem-solving capabilities such code has to be re-factored and reused effectively. The legacy systems have the collection of both the types of the code: procedural and object-oriented. The procedural code is converted into object-oriented code by using the phenomenon of re-engineering and the object-oriented code database is searched for reusable code components. Thus to make the effective and timely detection of such reusable components tools is required. All the existing tools for such detection use various metrics for measuring and analysis of compatibility, price and development effort required to re-engineer those components. Also the current system will only focuses on using cohesion and coupling based metrics. But accuracy is the problematic issues in all of them because of their few metrics usage conditions. This work proposes a novel RUCM (Reusability Utility Count Model) for analyzing the reusability value. It takes various key features of code for calculating the above. The work focuses on satisfying the quality attributes by applying all the modularity principles in metrics design and measurement. To do that effectively this work had developed
six composite metrics: LOC, LMD, MD, DOC UOS, and IC. In its primary work level the proposed approach seems to provide effective results in near future.

References

- Andreas S, Evigoni D Reiner D, Erik F and Micheal W, "Conception and Experience of metrics based software reuse in practice", Published in International Workshop of Software and Maintenance (WSM99), Sep 1999. pp 178-189
- Benjamin Van Ryseghem, Stephane Ducasse and Johan Fabry, "A Framework for the Specification and Reuse of UIs and their Models", Published in International Workshop on Smalltalk Technologies (IWST 12), ACM 2012.
- Shamsher Singh, Pushpinder Singh and Neeraj Mohan, "Identification of Object Oriented Reusable Components Using Multilayer Perceptron Based Approach", in International Conference on Computer Engineering and Multimedia Technologies (ICCEMT), Sep 2012.
- Nasib S. Gill, "Reusability Issues in Component-Based Development", in Department of Computer Science & Applications, M. D. University, Rohtak, Haryana (India).
- Hani Abdeen, Houari Sahraoui, Osama Shata, Nicolas Anquetilz and Stephane Ducasse, "Towards Automatically Improving Package Structure While Respecting Original Design Decisions", in Research Grant NPRP grant #09-1205-2-470 from the Qatar National Research Fund, Qatar University, Qatar.
- Fernando Britoe Abreu and Rogério Carapuça, "Candidate Metrics for
Object-Oriented Software within a Taxonomy Framework,
- Danail Hristov, Oliver Hummel, Mahmudul Huq and Werner Janjic, "Structuring Software Reusability Metrics for Component-Based Software Development,
- Philip Newcomb, "Reengineering Procedural Into Object-Oriented Systems",
   in IEEE, ISSN: 0-8186-7111-4, 1995. pp 237-249
- K. K. Aggarwal, Yogesh Singh, Arvinder Kaur and Ruchika Malhotra, "Software Reuse Metrics for Object-Oriented Systems,
- Amit Sharma, Sanjay Kumar Dubey, "Comparison of Software Quality Metrics for Object-Oriented System",
- Johny Antony P, "Predicting Reliability of Software Using Thresholds of CK Metrics",
- Anupama Kaur, Himanshu Monga, Manupreet Kaur , "Performance Evaluation of Reusable Software Components",
- Parul Gandhi and Pradeep Kumar Bhatia, "Reusability Metrics for Object-Oriented System: An Alternative Approach",
- Jannik Laval, Jean-Rémy Falleri, Philippe Vismara, Stephane Ducasse, "Efficient Retrieval and Ranking of Undesired Package Cycles in Large Software Systems",

Index Terms

Computer Science
Software Engineering

Keywords
RUPM (Reusability Utility Count Model)  Object Oriented  UOS (Understandability of
Software)  Complexity)  IC (Interface
DOC (Degree of Cardinality)

LMD (Low Modification Degree)

ALOC

MD (Modularity Degree)

Cohesion

Coupling;