Development of Quality Software Using Object Oriented Framework: Reuse Reengineering Concept

Shikha Singh, Prof. (Dr.) Rajendra Kumar Isaac

Abstract— Software Reuse and re-engineering is a complex subject that is highly knowledge-intensive. It is an important way to reduce software cost and to make programmers and designer more efficient.An Object Oriented framework is a set of cooperating classes,both abstract and concrete classes and may be the solution for a family of related problem..This paper proposes the framework technique , Highly Cohesive and Low Coupled Object Oriented (HCLCOO) Framework . This framework provides a generic structure to develop application software .HCLCOO is responsible for performing a specific task.This framework contains abstract class and concrete class ,abstract class contain general structure by simple change it may be used for different application.

Index Terms—Reuse,Re-engineering,Software Quality,OOA,Framework,Cohesion, Coupling.

---------------------- ◆ ---------------------------

1. Introduction

The idea of Reuse Re-engineering approach , first introduced by Mcllroy , was proposed at the NATO Software engineering Conference in 1968.[Smolarva97].To reduce the cost and improve the quality of the software. The development and maintenance of software in an organization are costly and time consuming .Several legacy systems exist in the word as the critical assets for most of the organizations .This system is too difficult to modify and too important to be discarded. Therefore these systems need to be re-engineered onto a modern platform for better operating efficiency and maintenance. Reuse and Reengineering enhance the reusability, maintainability, testability and enhance the quality of software.

The framework allows for specific design and quality requirements (performance, and maintainability) of the target system to be considered during the re-engineering process. Frameworks are just one of many reuse techniques [johnson97].They become popular in the software industry during the 1990s .They are object oriented re-engineering techniques .

An object-oriented framework, is the reusable design and implementation of a system or subsystem [Beck and Johnson 1994], it is typically implemented as a set of Abstract classes which define the core functionality of the framework along with concrete classes for specific application included for completeness.

Designing a Framework differs from designing single application in two manners:-

- Frameworks are used to provide a general solution for a set of similar or related problem or entire application , while application provides a concrete solution for a particular problem.
- .A framework does not cover all of the functionality required by particular domain ,it abstracts the common functionality to be filled in by the framework user, whereas an application design has all the components it needs to execute and perform its task.

Due to these differences ,framework design focus on providing flexible abstraction that cover the functionality required by applications within a domain and making the framework easy to use. These abstractions in turn provide ways in which application developers can customize the framework. Object oriented technology is suitable technology for framework. An Object –oriented framework is a set of cooperating classes , both abstract and concrete , that make up a reusable design for a specific class of software.[EST95,Gam94].The framework determines the

architecture of the application .Built using it by partitioning the design into abstract classes and defining their responsibility and collaborations and the thread of control. The design decisions that are common to its application domain are captured, so that the application designer can concentrate on the specifics of application. Framework development thus emphasizes design reuse over code reuse.[Gam94].
An object –oriented framework, is the reusable design and implementation of a system or subsystem [Johnson,1997].It is implemented as a set of abstract classes which define the core functionality of the framework along with concrete classes for specific applications included for completeness.

2. Existing Techniques

In this section we attempt to put existing software re-engineering term in some perspective.

2.1 Reuse and Re-engineering by Inheritance:

In an Object Oriented System Inheritance allows the derivation of new classes by modification of existing classes. Specialization (adds new functionality to sub-classes) and overriding (refine, redefine or even hide the parent's functionality) are two ways of modifying an existing class. Inheritance makes it easier to include existing code by extending the original class and adding new ones. But it allows also redefinition and hiding that may have harmful effects on reuse. Such diverse uses of the same inheritance mechanism cause difficulties in class reuse.

2.2 Reuse and Re-engineering by Repositories

Reuse can be beneficial only if reusable items are easily available. For this reuse repository is created in such a manner that developers can take benefit of pre –existing reusable items. An effective reuse repository often contains numerous components ,which makes it impossible for software developers to anticipate the existence of all the components[Ye02]. Traditional reuse repository system employs information access mechanisms (browsing and querying) and require software developers to initiate the reuse process. On the other hand ,active reuse repository system infer queries from syntactic and semantic cues present in partially constructed programs in development environments, and proactively deliver components that match the inferred reuse queries.

2.3 Component Reuse and Reengineering

Software component reuse is the technique of creating new software applications from existing components. Reusable components can be requirements specifications, design

documents, source code, user interfaces, user documentation, or any other items associated with software.

2.4 Quality Driven Object –Oriented Re-engineering

[Ladan] proposes quality Driven Object –Oriented Reengineering.Laden propses in her paper that software re-engineering processes can be applied at different levels.At the lowest levels migration takes the form of transforming the code from one language into another.At the higher level ,the structure of the system may be changes as well to make it for instance more object –oriented.At even higher levels of abstraction the global architecture of the system may be changed as part of the migration process.The re-engineering process include the following steps

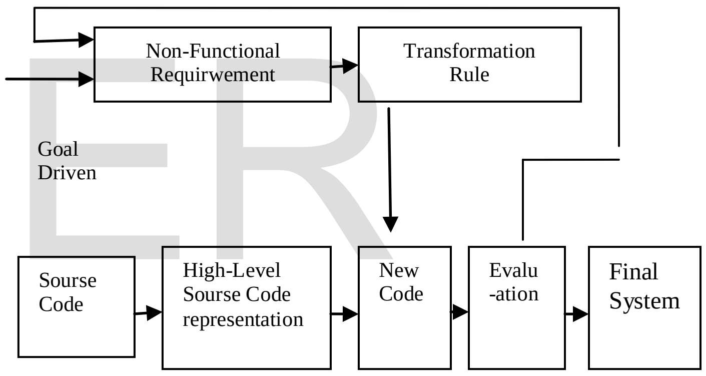


Figure 1: The Block Diagram of the Quality –Based Re-engineering Process.

3. PROPOSED SYSTEM METHODOLOGY

Highly Cohesive And Low Coupled Object Oriented Framework

To the improvement of existing software and creating new software this paper proposes a methodology Highly

Cohesive And Low Coupled Object Oriented (HCLCOO) Framework.By the use of this framework designer can update the existing system or create new system as shown in following figure.

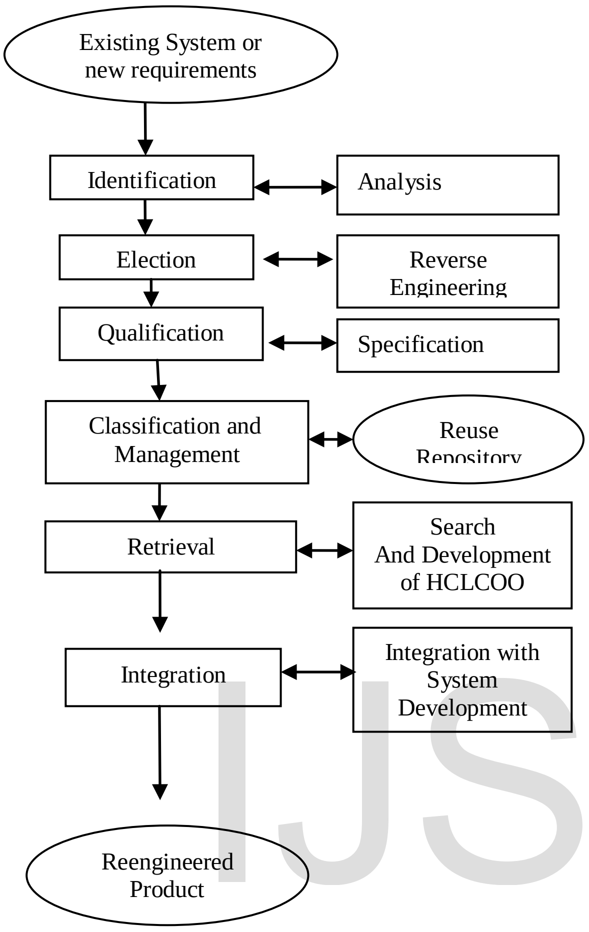

Existing System or new requirements
Identification
Analysis
Election
Reverse Engineering
Qualification
Specification
Classification and Management
Reuse Repository
Retrieval
Search And Development of HCLCOO
Integration
Integration with System Development
Reengineered Product

Figure 2: Process of Re-engineer the new system by existing code

This framework is responsible for performing a specific logical task(that may be performed differently in similar application), Highly Cohesive and Low coupled Object Oriented framework does one thing well,Highly cohesive and low coupled classes have clearly defined relationship with other classes and collaborate in clear ways with a clear interface encapsulating behavior. This type of framework is specially designed for specific tasks that can be performed in different application. A similar application that required the same type of job can use Highly Cohesive and Low Coupled Object Oriented Framework. Highly Cohesive and Low Coupled Object Oriented Framework is desirable because it means the Framework does one job well. Highly Cohesive Object Oriented Framework fixes the way of performing most of activities in the framework itself. Thus these frameworks are highly useful for those applications that require performing such activity in a particular way as defined and implemented in the framework. Highly Cohesive and Low Coupled Object Oriented Framework increases the reusability of class and is easy to understand and implement in other application program.

4. RESULT

4.1 Highly Cohesive Low Coupled Object Oriented Framework for Unit testing

The purpose of this thesis to develop a framework which is useful for many applications, here I developed a framework for the unit test environment.
Thus this framework supports the development of a number of applications that would test a module based on condition coverage criteria but accept the stub and drivers manually (from human being) not automatic.

4.2 Object Oriented Descriptions:

After studying the problem ,we specify the data flow diagram to specify how users will interact with the framework. A module to be tested, drivers and the stub can either be provided to the system manually or automatically using software systems.

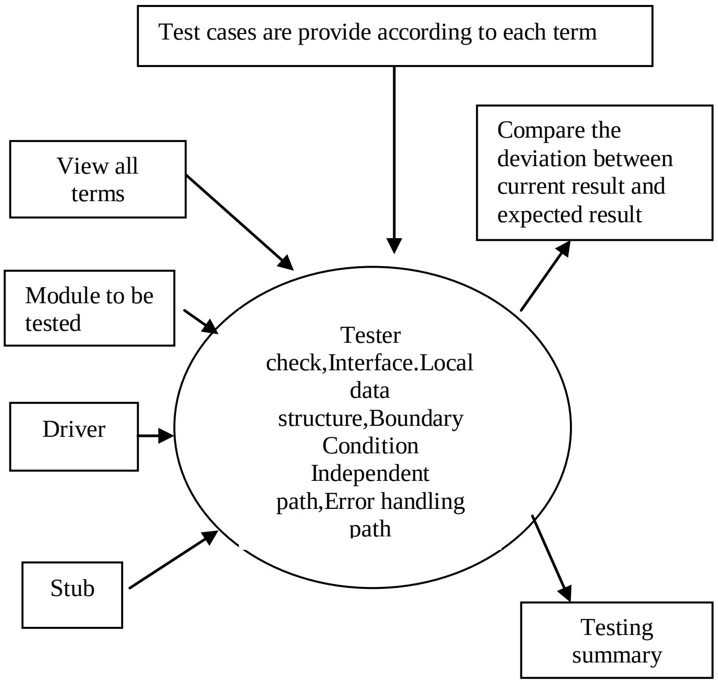
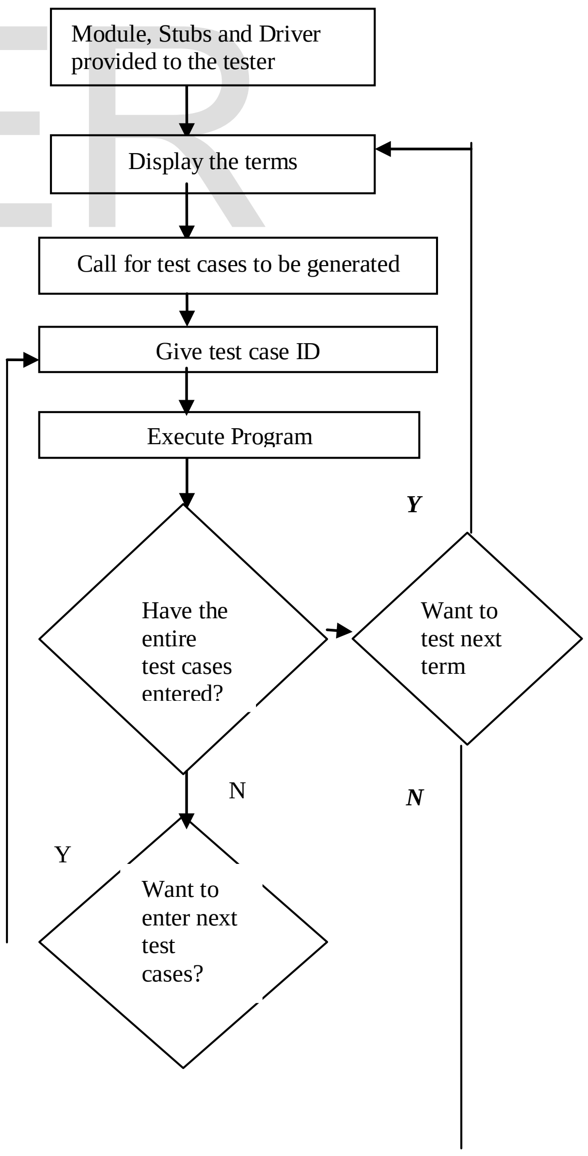

Test cases are provide according to each term
View all terms
Compare the deviation between current result and expected result
Module to be tested
Tester check,Interface.Local data structure,Boundary Condition Independent path,Error handling path
Driver
Stub
Testing summary

Figure 3:Data Flow Diagram for Unit Test Environment

4.3 Design Concept for Highly Cohesive and Low Coupling Object Oriented Framework in Unit Test Environment

- A module to be tested is provided to the tester
- The driver is provided to the tester.
- Stubs are provided to the tester.
- By all the Condition is tested
    - Next condition is identified
    - The system displays this Condition to the tester and asks the number of test Cases (N) that need to be generated to test this condition.
    - For this number (N) of times
        - Tester asks the next test case.
        - Tester input the test cases
        - Tester executes this test case and redirects the output values of the execution to a file.
        - The actual output value is compared with the specified output value and is shown to the tester.
        - Tester according to the test result update a testing summary report.
        - Tester/Software ask whether to continue or quit.
        - If tester wants to quit (in case of getting wrong result to correct the unit) , the Path and name of the Test Summary report are displayed to the tester and system setups.
    - Tester asks whether to proceed further or quit.
    - If tester wants to quit (in case of getting wrong result to correct the unit) , the path name of the Test Summary Report is displayed to the tester and tester stops.

4.4 Following Flow Charts shows the above steps


Module, Stubs and Driver provided to the tester
Display the terms
Call for test cases to be generated
Give test case ID
Execute Program
Have the entire test cases entered?
Want to test next term
Y
N
N
Y
Want to enter next test cases?

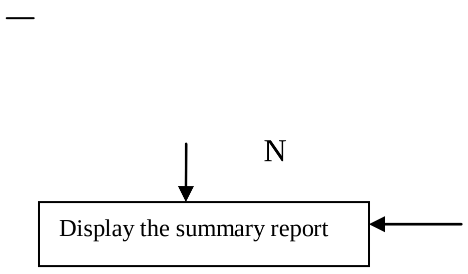


Figure 4: Steps for generating test case

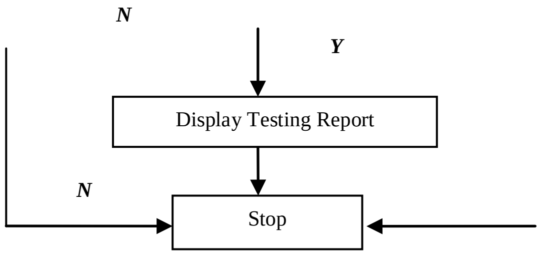
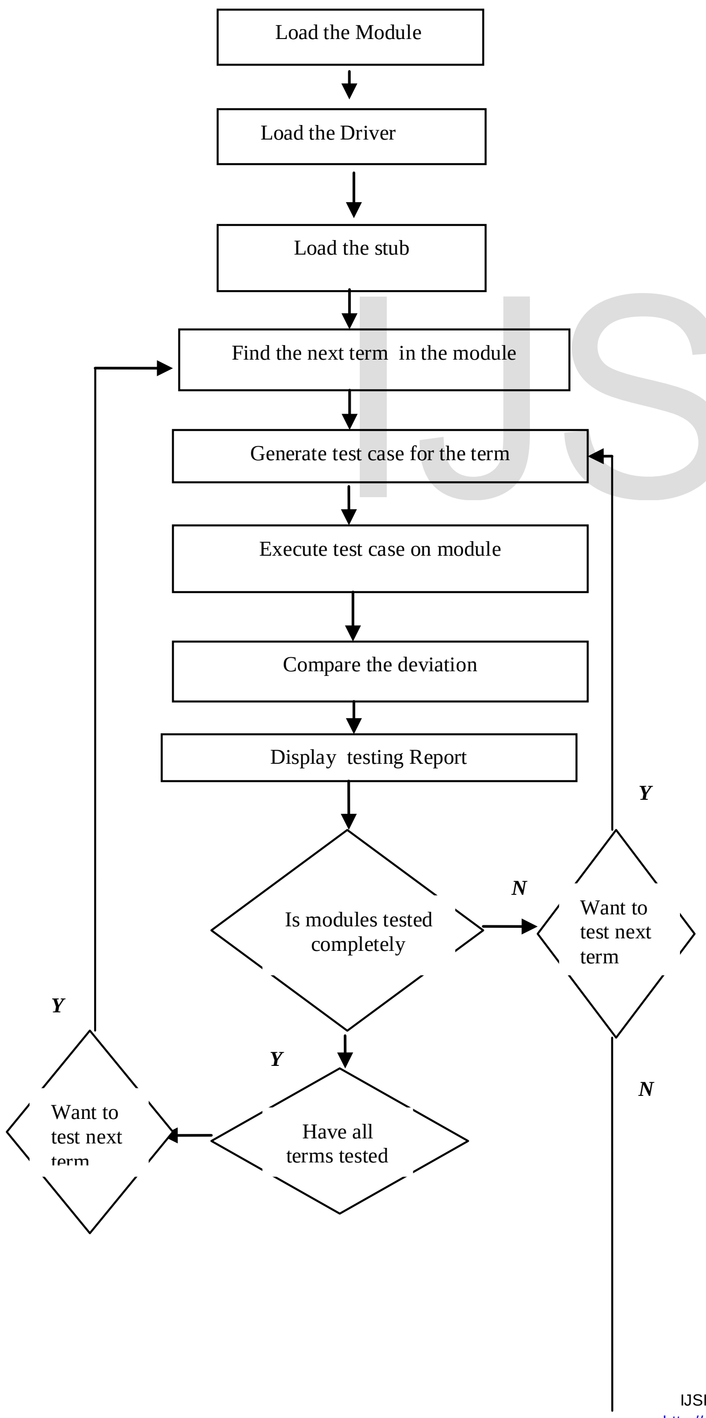


Figure 5: Steps of HCLCOO Framework for unit test environment

4.5 Following flow Chart shows the step for Highly Cohesive low Coupled Object Oriented Framework for Unit test Environment



5 CONCLUSIONS

In this paper ,we have presented a framework for reenginering the existing system and improving the quality of exesting system also we may create new system by reusing this framework .We propose a Highly Cohesive Low Couple Object Oriented Framework (HCLCOO) technique to solve the problem of reuse and reengineering.In this technique framework provide generic structure by which many similar application problem may be solve.HCLCOO framework is responsible for performing a specific logical tasks(that may be performed differently in similar application), Highly Cohesive and Low coupled Object Oriented framework does one thing well,Highly cohesive and low coupled classes have clearly defined relationship with other classes and collaborate in clear ways with a clear interface encapsulating behaviour. This type of framework is specially designed for specific tasks that can be performed in different application. A similar application that required the same type of job can use Highly Cohesive and Low Coupled Object Oriented Framework.

REFERENCES

[1] Basili,V.R.,"Viewing maintenance as reuse –Oriented software development" Software,IEEE,Jan 1990.
[2] Bosch J." Frameworks Problems and Experiences ",Book ,Building Application Frameworks : Object –Oriented Foundations Of Framework Design,1999.
[3] Froehlich G., Hoover H. J. , Ling Liu Ling , Sorenson P.,"Designing Object Oriented Framework",CiteSeer,1998.
[4] Fayad M.E.,"Object Oriented Application framework",Proceeding in communication of the ACM ,page(s) : 32-38,/Vol. 40, October 1997.

[5] Johnson R. E.," Frameworks=(Components+Patterns)" ,communication of the ACM, Vol. 40, October 1997.
[6] Johnson R. E.," Components, Frameworks, Patterns",Proceeding of 1997 on software reusability , page(s):10-17,1997.
[7] Johnson R. E.,Foot B., "Reusing Object Oriented Designs,Technical Report UIUCDCS 91-1696, University of Illinois ,May 1991.
[8] Jose A.," An Object-Oriented Framework for Building Software Agents", Journal of Object Technology, vol. 2, Page(s): 85-97. January-February2003.
[9] Landin Niklas," Development of Object-Oriented Frameworks", thesis report,1995 .
[10] Mili A.," A survey of software reuse libraries",Journal Annals of Software Engineering,Page(s):349-414,Volume 5,1998.
[11] Mishra A., "Software quality assurance models in small and medium organizations: a comparison", International journal of Information Technology and Management, Vol. 5, No. 1, 2006.
[12] Marcus A. Rothenberger, "A Cost Benefit Model for Systematic Software Reuse ", ECIS 2002 Gdańsk Poland , June 6–8 2002.

AUTHOR

- Shikha Singh, Research Scholar , Sam Higginbottom Institute of Agriculture, Technology and Sciences (Formerly Allahabad Agricultural Institute) Deemed-to-be University, Allahabad(U.P.) shikhasinghjyoti@gmail.com.

- Prof. (Dr.) Rajendra Kumar Isaac,Professor,Faculty of Engineering and Technology,Sam Higginbottom Institute of Agriculture, Technology and Sciences,(Formerly Allahabad Agricultural Institute) Deemed-to-be University, Allahabad(U.P.)